UNED

Análisis estadístico de datos espaciales con QGIS y R

Yolanda Cabrero Ortega
Alfonso García Pérez

https://www.facebook.com/groups/stats.ebooksandpapers/

Análisis estadístico de datos espaciales con QGIS y R

YOLANDA CABRERO ORTEGA
ALFONSO GARCÍA PÉREZ

UNIVERSIDAD NACIONAL DE EDUCACIÓN A DISTANCIA

ANÁLISIS ESTADÍSTICO DE DATOS (ESPACIALES CON QGIS Y R

www.uned.es/publicaciones

Fotografía de la portada: Hoces del Duratón. Segovia

ISBN electrónico: 978-84-362-7091-4

Edición digital: noviembre de 2015

El mundo es un lugar que va más allá
de nuestro entendimiento

Paul Auster

Índice

Prólogo

El presente texto es una introducción al Análisis de Datos Espaciales, entendidos éstos como datos en los que, además de las variables que se estén considerando en el estudio, aparece su localización geográfica. Ésta no tiene porqué ser siempre su latitud y longitud; en ocasiones, la distancia a la costa de un banco de peces es más informativa que sus coordenadas geográficas.

Una peculiaridad de este libro es que el Análisis de Datos Espaciales se hace, tanto con los Sistemas de Información Geográfica SIG, o mejor GIS (*Geographical Information System*) si utilizamos el acrónimo inglés, como mediante la Modelización de los datos espaciales.

En este análisis global se hace uso del software Quantum GIS, QGIS, y del paquete estadístico R, ambos gratuitos y que interactúan perfectamente.

Los Sistemas de Información Geográfica son un *visor de datos*. Aprenderemos el manejo de QGIS en la primera parte del texto. El Análisis Estadístico de esos datos se realizará en la segunda parte del texto con R.

El Análisis de Datos Espaciales es de gran interés en muchos campos en donde los objetivos pueden ser distintos. En Ecología, por ejemplo, suele ser de interés estimar una *distribución espacial* que explique las localizaciones acaecidas en un área de estudio o que permita comparar las localizaciones de varias especies.

En Epidemiología el interés suele ser el de poder concluir si las causas de una cierta enfermedad están concentradas en una determinada región (piense el lector en los recientes casos de ébola). Esto puede conseguirse comparando la distribución espacial de los casos observados con las localizaciones de un conjunto de controles elegidos al azar de la población en estudio.

En Arqueología la localización geográfica es de sumo interés. Por ejemplo, en la parte occidental de las Islas Británicas se encuentran yacimientos con monumentos megalíticos puesto que estas zonas absorbieron influencias del Atlántico. Por contra, la parte oriental de dichas islas recibieron influencias de sus vecinos europeos, las cuales dieron lugar a vasos campaniformes.

En Economía, la localización de una nueva empresa es de vital importancia para el incremento de sus beneficios ya que si debe enviar sus productos

deberá reducir costes de transporte para lo que deberá conocer en dónde se localizan sus principales clientes y esto, no sólo a nivel empresarial sino a nivel nacional. Hoy en día se comenta que el siglo XIX fue europeo, el siglo XX americano y que el XXI será oriental.

El ejército necesita conocer la localización geográfica de objetivos propios y ajenos para una mejor defensa.

En Ciencias Ambientales, la localización geográfica lleva asociado un clima específico con unas implicaciones que deben ser analizadas.

Y, por supuesto, los Sistemas de Información Geográfica son imprescindibles en Geografía la cual depende de ellos como la Estadística depende de los paquetes estadísticos.

Hoy en día no puede mantenerse al margen la localización geográfica al analizar unos datos. Además, esta implicación interactúa entre los diversos campos, de manera que el clima en un momento determinado lleva a unos asentamientos en los que se obtuvieron unos determinados yacimientos arqueológicos, o las rutas comerciales las cuales dependen de la topografía del terreno. De ahí que en estos días es obligado un estudio interdisciplinario de los datos, surgiendo campos como la paleoclimatología, paleobiogeografía, paleoecología, por citar sólo unos campos relacionados con la Arqueología, lo que obliga a la formación de equipos de investigación también multidisciplinarios.

El lector debe de tener unos conocimientos básicos de Estadística y de R. Si no los tiene, para la primera, le recomendamos el libro de García Pérez (2008a), a veces denominado CB, y para R el texto de García Pérez (2008c). La unión de ambos libros es el de García Pérez (2010).

En algunos momentos se citará el texto TA que corresponde a García Pérez (2005a) y TAEA que es el libro base del Máster *Técnicas Actuales de Estadística Aplicada*.

Los ficheros de datos que utilizaremos en el libro, así como información y ejemplos complementarios, están disponibles en la dirección

`http://www.uned.es/pfacs-estadistica-aplicada/gis.htm`

Yolanda Cabrero Ortega
Tutora de Geografía e Historia
Centros Asociados de Madrid y Madrid Sur. UNED
(ycabrero@madrid.uned.es) (ycabrero@madridsur.uned.es)

Alfonso García Pérez
Catedrático de Estadística e I.O.
Depart. de Estadística. Fac. de Ciencias. UNED
(agar-per@ccia.uned.es)

Capítulo 1

Introducción al QGIS

1.1. Introducción

En este capítulo introduciremos qué son y para qué sirven los denominados Sistemas de Información Geográfica SIG (o GIS como preferimos denominarlos aquí utilizando el acrónimo anglosajón), analizaremos brevemente los más conocidos, para pasar después a describir con detalle el que utilizaremos en este libro, el Quantum GIS o, más brevemente, QGIS, software que hemos elegido por actuar con el paquete estadístico R.

1.2. Sistemas de Información Geográfica

La información que nos llega diariamente a través de los medios de comunicación, o la que podemos obtener en una base de datos de unos grandes almacenes que buscan la existencia de un producto, o la disponibilidad de hoteles en una zona determinada, o el seguimiento que podemos hacer de un envío a través de una web, etc., será una información más útil si está *georreferenciada*, es decir, si esta información incluye las coordenadas geográficas de dónde se produce.

Los *Sistemas de Información Geográfica* son herramientas desarrolladas para gestionar esa información que se obtiene en un territorio y, dado que tienen una gran potencia, poder trabajar con un volumen de datos muy elevado como los que habitualmente proceden del mundo real.

Aunque podríamos adoptar la definición sobre lo que es un GIS dada por Burrough y McDonnell (1998, pp. 11), "... un potente conjunto de herramientas para recopilar, almacenar, recuperar a voluntad, transformar y mostrar datos espaciales del mundo real para un conjunto particular de propósitos", precisaremos un poco más diciendo que es una técnica informática que permite:

- Trabajar con una gran cantidad de datos es decir, permite aplicaciones en lo que hoy en día se denomina *Big Data*.
- Capturar datos espaciales (o utilizar los nuestros), editarlos, almacenarlos, gestionarlos y consultarlos de forma rápida.
- Analizar dichos datos de forma espacial, es decir, utilizando la información proporcionada por sus coordenadas.
- Obtener conclusiones y resultados, tanto desde un punto de vista *descriptivo* como, lo que es más importante, desde un punto de vista *inferencial* lo que permitirá modelizarlos y hacer predicciones.
- Generar resultados y exportarlos: visualizarlos, crear informes, gráficos, mapas, etc., pero no es sólo una herramienta de diseño cartográfico sino que analiza la realidad, la modeliza y la gestiona.

El enlace de los GIS y los datos es a través de la *georreferenciación*, es decir, la localización de datos en un territorio. Los datos utilizados por los GIS tienen 2 propiedades:

- Geométricas: los datos se localizan en un lugar determinado, es decir, están georreferenciados con unas coordenadas que permiten localizar *puntos*, *líneas* y *polígonos*.
- Información Estadística (o descriptiva): cada dato o "geodato" tiene asociada una *matriz de datos* con información de variables, información que aparece recogida en tablas de atributos, tales como por ejemplo el número de empresas, número de empleados, productividad, población de municipios, tipos de cultivos, extensión de usos del suelo, etc., asociadas a unas coordenadas.

La mayoría de los GIS (como por ejemplo ArcGIS) están construidos en lo que podríamos denominar *arquitectura de caja de herramientas* (*toolbox*) con toda la información (como por ejemplo los datos) contenida en dichos programas y dependiendo de cada ordenador. No obstante, hoy en día, con la gran información digital disponible en medios móviles, por ejemplo a través de Google Earth o Google Maps, se tiende a utilizar los GIS denominados de *arquitectura de servicios* en donde la información utilizada (datos) puede provenir de Internet, incluso del espacio ya que la *Shuttle Radar Topography Mission* (SRTM) de Febrero 2000 permite manejar mapas del tiempo por ejemplo.

Aunque las imágenes de satélite requieren meta-datos, las del terreno que forman los mapas se obtienen fácilmente mediante los *Global Positioning System* (GPS) proporcionando coordenadas (bidimensionales) precisas, en un sistema de referencia conocido.

Además, este último tipo de GIS (como el que estudiaremos en este libro), tiene la ventaja de que, aunque los datos no sean gratis, sus visualizaciones sí lo son, además de no tener que estar pendientes de sus actualizaciones.

Un aspecto muy importante que queremos destacar de los datos utilizados en los GIS es que se almacenan en lo que denominaremos *capas*, las cuales pueden combinarse, es decir, ponerse unas capas sobre otras, para crear mapas diferentes los cuales podrán ser objeto de consultas, tanto por su geometría como por su información descriptiva. Si analizamos un paisaje, nuestras capas (datos) serían por ejemplo, relieve, litología, hidrología, núcleos urbanos, usos del suelo, red viaria, etc. Con un GIS podemos individualizar cada una de estas capas, o seleccionar aquellas con las que queramos trabajar para destacar algunos aspectos, incorporar información actualizada y crear nuevos mapas.

Cada una de las capas con las que trabajamos tienen información geográfica (mapa digital) sobre el que se añaden datos alfanuméricos generando en nuestro trabajo dos tipos de ficheros: cartográfico (mapas digitales) y datos (tablas de atributos). Con los GIS podremos localizar los objetos en el espacio (georreferenciar, que nos permite calcular áreas, distancias, ...) y relacionarlos (topografía, que nos permite conocer zonas conectadas).

Uno de los aspectos más interesantes de un GIS, aparte de capturar y almacenar información, es la posibilidad de hacer consultas de forma rápida teniendo en cuenta que podemos estar manejando miles de datos. De esta forma,

- Podremos establecer ubicaciones óptimas teniendo en cuenta las características de una zona: su topografía, comunicaciones, usos del suelo, población, calidad medioambiental de la zona, nivel socioeconómico, etc.

- Podremos establecer las zonas con menor o mayor impacto medioambiental para ubicar actividades de mayor o menor riesgo o necesidad: vertederos, carreteras, parques eólicos, etc.

- Podremos localizar las zonas con necesidades, o no, de infraestructuras: hospitales, colegios, paradas de autobuses, etc.

- Podremos analizar superficies mezclando fenómenos como topografía, clima, ..., para obtener modelos digitales del terreno de forma tridimensional.

- Podremos hacer un seguimiento y monitorización de un territorio que nos permita ver su evolución, su cambio a lo largo del tiempo (cambio de usos del suelo, de población, degradación de un espacio natural, ...).

1.2.1. Utilidad de los Sistemas de Información Geográfica

Rhind (1990, pp. 218-223) establece seis tipos de preguntas que pueden ser respondidas con un GIS:

1. Localización (¿Qué hay en ...?): posicionándonos sobre un mapa, podemos saber lo que hay en un lugar determinado al tener acceso a la información contenida en la tabla de atributos.
2. Condición (¿Dónde ...?): podemos hacer consultas determinando el cumplimiento de ciertas condiciones impuestas según nuestro criterio.
3. Evolución temporal o Tendencias (¿Qué ha cambiado ...?): podemos determinar la variación de un aspecto concreto viendo su evolución a lo largo del tiempo (para ello necesitamos tener mapas de la zona que cubran espacios temporales diferentes).
4. Rutas (¿Cuál es el camino óptimo hacia ...?): podemos calcular las mejores rutas entre dos puntos.
5. Pautas (¿Qué pautas existen para ...?): podemos determinar la regularidad en la aparición de un fenómeno.
6. Modelos (¿Qué ocurriría si ...?): podemos generar modelos como veremos en los capítulos siguientes, que nos ayuden a prever lo que sucedería en una zona ante un hecho determinado. Este aspecto es sin duda el más importante y a los que se dedica la segunda parte del libro.

1.2.2. Aplicaciones de los Sistemas de Información Geográfica

En sus inicios, las aplicaciones de los GIS se centraban en la geografía cuantitativa y espacial pero, en la actualidad, y teniendo en cuenta que un GIS permite trabajar con datos y hacerlos útiles, se ha convertido en una herramienta de trabajo fundamental para numerosas disciplinas: prevención de catástrofes naturales, enfermedades, elección de la mejor ubicación para un negocio, ... Por lo tanto, su campo de aplicación es tan variado como lo puedan ser las actividades que se puedan desarrollar. Por poner algunos ejemplos:

- Ordenación del territorio, como la planificación y gestión de infraestructuras: redes hidrológicas, carreteras, redes ferroviarias. También la elaboración de mapas de usos del suelo, la localización de servicios como industrias, servicios sanitarios o educativos, así como el establecer planes catastrales.
- Empresas: investigación de mercados, determinación de localizaciones óptimas, geomarketing, logística como el diseño de repartos, seguimiento de mercancías, etc.

- Medioambiente, Ecología, Geología, Oceanografía: estudio y localización de especies naturales, ver el estado de conservación del medio natural, su evolución, estudio del clima, planificación parcelaria, uso de fertilizantes, etc.
- Sanidad: evolución de enfermedades, determinación de focos de enfermedad.
- Arqueología, Paleontología: localización geográfica de yacimientos.
- Estudios sociodemográficos: determinar estructuras de población, necesidades por barrios (hospitales, colegios, ...).
- Establecimiento de planes de emergencia: mapas de actuación en casos de incendios, inundaciones, ...

1.2.3. Sistemas de Información Geográfica más utilizados

En el mercado contamos con una gran variedad de software que nos permite trabajar con información geográfica. Hay empresas comerciales (ESRI, Intergraph, MapInfo, Bentley Systems, Autodesk o Smallworld) que ofrecen un completo conjunto de aplicaciones. Desde los gobiernos también se ha trabajado para crear programas de GIS de código abierto de acuerdo a sus necesidades.

La gran mayoría de los GIS están adaptados a todos los sistemas operativos (Windows, Mac, Linux) pero se recomienda hacer una consulta previa antes de elegir un GIS u otro. Podemos clasificarlos en tres grandes grupos:

No libres, creados por empresas: ArcGis, ESRI, Intergraph, MapInfo Autodesk Map, IDRISI, ABACO DbMAP, Bentley Map, Caris, CartaLinx, GE Smallworld, Geomedia; GeoStratum, GestorPojet-PDAProject, LatinoGis, Manifold, Maptitude, MiraMon, ortoSky, SITAL, SuperGIS, TatukGIS, TNTMips, TransCAD.

De acceso libre: gvSIG, QGIS, Capawere, MapServer, SAGA GIS, SEXTANTE, LocalGIS, Kosmo, JUMP, ILWIS, GRASS, GeoServer, GeoPista, Generic Mapping Tools, El Suri, uDIG, MapGuide Open Source, MapWindow GIS.

Y, por último, un grupo de GIS *creados por organismos gubernamentales* como: SIGNA (Sistema de Información Geográfico Nacional), SITGA (Xunta de Galicia), SITNA (Gobierno de Navarra), SIGPAC (Ministerio de Agricultura sobre parcelas agrícolas), etc.

En este libro nos centraremos en el QGIS no sólo porque es gratuito sino porque es el que tiene una mayor interconexión con el paquete estadístico R (también gratuito) que utilizaremos en la segunda parte del libro para realizar el análisis estadístico de los datos espaciales.

1.3. Instalación de QGIS

Para instalarlo vaya a

`http://www.qgis.org/es/site`

y descargue el ejecutable (o instálelo directamente) desde el botón

Descargar ahora

que allí aparece.

Al descargar e instalar Quatum GIS también descargará e instalará el programa GRASS GIS; ambos trabajan juntos. También instalará otros programas con SAGA GIS o Python.

Cuando haya instalado QGIS, de todos los iconos que aparecerán en su Escritorio, lo podrá abrir con el icono QGIS Desktop.

1.3.1. Descripción del área de trabajo

Al abrir QGIS encontramos un área de trabajo interactiva a base de menús y botones de acceso rápido (Figura 1.1) en la que vemos arriba unos *Menús* que denominamos zona 1, que dan acceso a diversas utilidades.

Vemos también unas *Herramientas*, zona 2, que son botones de acceso rápido a las funciones del menú y que podemos configurar según nuestras necesidades. En la zona de visualización de las *Capas*, que en la Figura se ha denominado zona 3, es donde aparecen las que vamos incorporando y que podemos activar-desactivar clicando sobre la casilla con su nombre. También podremos cambiarlas de posición arrastrándolas, o agruparlas de tal manera que abriendo un grupo de ellas se incorporen automáticamente todas las capas que componen el grupo.

Las capas siempre se sobre-impresionan en el mapa según el orden en el que vamos incorporándolas a QGIS.

Está también la zona de *Visualización gráfica* (zona 4) en donde se irá formando nuestro mapa y, por último, la *Barra de estado* (zona 5) en donde aparecen la *Escala*, las *Coordenadas* y el *Sistema de Referencia de Coordenadas* SRC. Existen varios SRC dependiendo del país, de la proyección, etc.

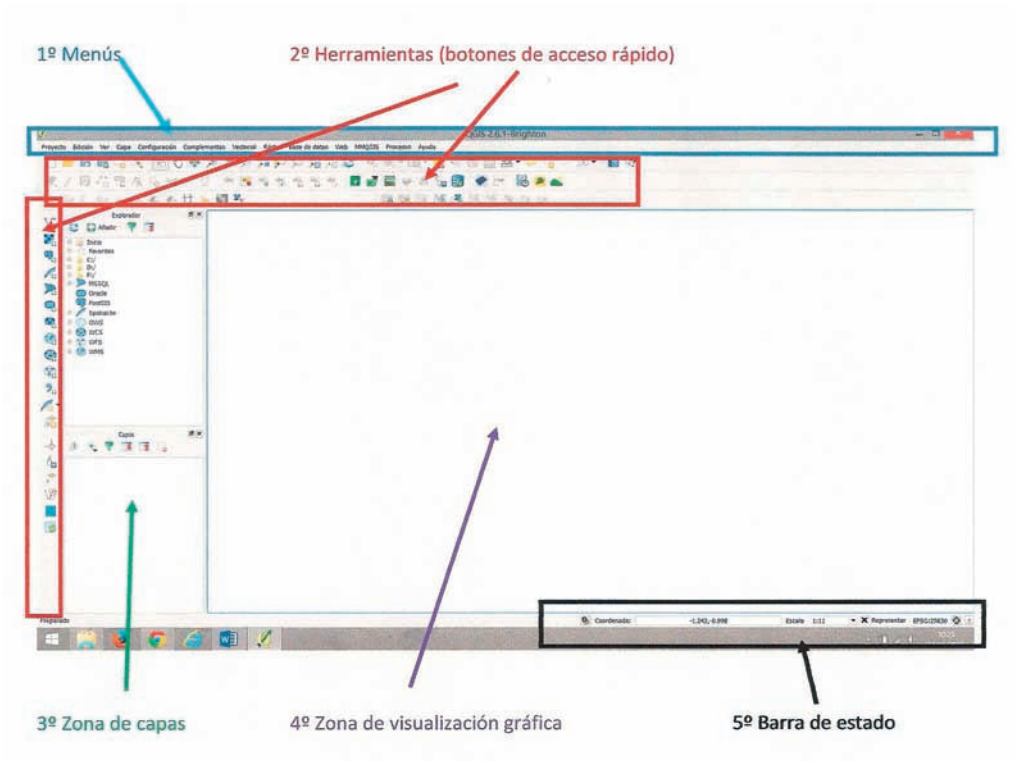

Figura 1.1

1.4. Tipos de datos GIS

Con un GIS trabajamos con una parte de la realidad elegida de acuerdo a nuestras necesidades. Esto nos lleva a modelizar dicha realidad de dos formas principalmente (aparte de los modelos tridimensionales que no trataremos en el libro): *Vectorial* y *Raster*. De hecho, se habla de un *GIS vectorial* o de un *GIS raster* cuando predomina información de una u otra forma.

Un Gis vectorial está formado por datos geográficos, es decir, datos espaciales, representados por medio de coordenadas. Se distinguen tres tipos de datos: Puntos, Líneas (segmentos que unen dos puntos) y Polígonos (unión de varias líneas).

Los GIS raster son lo que en matemáticas se ha denominado Redes o Mallas (*grids*), que en resumen van a ser matrices de unos y ceros que se asocian con píxeles. Por esta razón suele asociarse a las fotos con GIS raster. Si además de unos y ceros aparecen otros dígitos, tendremos raster con colores en lugar de blanco y negro.

1.4.1. GIS vectorial

El uso de datos vectoriales permite trabajar con datos dispersos por el espacio geográfico, como por ejemplo la localización de hospitales en una comunidad autónoma, localización de las principales carreteras de acceso a una ciudad, localización de áreas protegidas por su valor ambiental, etc. Los datos vectoriales se usan para representar elementos que son el resultado de la acción del hombre (divisiones administrativas, usos del suelo, propiedad del suelo, redes viarias, etc.)

Se tienen en cuenta las propiedades de las *entidades* (elementos que no pueden subdividirse en otros menores, tales como escuelas, lagos, carreteras, etc.) y éstas se representan digitalmente por medio de *objetos*. En el GIS vectorial se consideran tres tipos de objetos:

- *Puntos*. Representan objetos espaciales que sólo están localizados, no tienen dimensiones, es decir, ni largo (i.e., longitud) ni ancho (anchura). La posición de cada objeto queda fijada a través de las coordenadas de los sistemas de referencia, están lo que se dice, georreferenciados, es decir, dotados de coordenadas (x, y). De esta forma representamos escuelas, hospitales, catedrales, yacimientos, etc.

- *Líneas*. Son una sucesión de puntos y representan objetos espaciales con una dimensión, longitud. Su posición se fija con dos pares de coordenadas. Con este objeto representamos carreteras, ríos, líneas de ferrocarril, etc.

- *Polígonos*. Son una sucesión de líneas cerradas y representan objetos espaciales con dos dimensiones, longitud y anchura. La posición de cada objeto se fija con dos o más líneas cuyas coordenadas inicial y final coinciden. Con ellos representamos lagos, pueblos, etc.

Dependiendo de la escala utilizada, una misma entidad puede ser representada por diferentes objetos, es decir, una ciudad por ejemplo puede aparecer como un punto, pero también como un polígono.

Un GIS vectorial tiene una componente cartográfica (mapa geográfico sobre el que trabajamos) y una base de datos que compone lo que se denomina la *tabla de atributos*. En estas tablas, que básicamente son lo que en Estadística se denominan Matrices de Datos, se registran las características de cada dato GIS vectorial y nos servirán para realizar consultas, para unir datos de otras tablas, para hacer cálculos de forma automática, etc.

Los formatos de datos vectoriales son muy variados y dependen del software GIS utilizado. Los más habituales son ficheros con extensiones shp, svg, mif, mid, E00, mdb, dgn, dwg, dxf y dxn.

Los ficheros shp (*shapefile*) son los más utilizados por QGIS. Este tipo de ficheros está relacionado con otros ficheros que tienen extensiones dbf, shx, prj

y xml. Por comentar algunas de sus características, digamos que los ficheros de extensión shp, contienen la *geometría*. Esto es, los puntos o vértices que definen la forma de los elementos geográficos. Los de extensión dbf son ficheros en el conocido programa Dbase y son ficheros que contienen la tabla de atributos o descripciones de cada uno de los elementos. Los ficheros shx contienen un índice para el manejo de tablas entre archivos y permiten facilitar las búsquedas. Los de extensión prj contienen la definición del sistema de coordenadas, proyección cartográfica, datum y unidades que usa el shapefile para registrar los elementos geográficos. Por último, los ficheros de extensión xml contienen metadatos (es decir, descripción de los geodatos) en un formato estandarizado.

Cada fichero shp sólo puede contener un dato geométrico: línea, o punto, o polígono. Los polígonos están formados por líneas pero no se consideran del tipo línea.

Hay también otros ficheros vinculados con shp que opcionalmente se pueden utilizar para mejorar el funcionamiento en las operaciones de consulta de la base de datos, información sobre la proyección cartográfica, o almacenamiento de metadatos. Estos archivos son de extensiones sbn y sbx, que almacenan el índice espacial de las entidades; con extensiones fbn y fbx, que almacenan el índice espacial de las entidades para los shapefiles que son inalterables (solo lectura), y los de extensiones ain y aih, que almacenan el índice de atributo de los campos activos en una tabla o el tema de la tabla de atributos.

Los GIS vectoriales permiten representar información tanto de datos cualitativos como cuantitativos de formas muy variadas: eligiendo símbolos, tamaños o colores diferentes en mapas de puntos; eligiendo grosores o colores diferentes en mapas de líneas; creando mapas de isolíneas a partir de mapas de puntos por interpolación (TIN); usando tramas o colores diferentes en mapas de polígonos, etc.

Según nuestras necesidades también podremos hacer cambios en los mapas para lograr una mejor representación de los datos, simplificando o suavizando trazados lineales; reduciendo el número de categorías realizando reclasificaciones (unión de campos con valores comunes); eliminando arcos que separan polígonos con variables similares (disolución) para crear uno mayor en polígonos contiguos y de iguales características; redefiniendo polígonos (fusión); podremos unir hojas si nuestro mapa digitalizado está dividido en varias de ellas, ajustando bordes para lograr una mejor representación de entidades.

Un GIS como herramienta de análisis permite hacer cálculos de forma automática: medir longitudes, áreas, perímetros, calcular centroides y estadísticas básicas, los datos pueden exportarse y tratarse con paquetes estadísticos más potentes como R. Pueden hacerse consultas por atributos y hacernos preguntas del tipo ¿Dónde,...?, las cuales responderemos utilizando operadores como =, >, <, AND, OR, NOT, ... y que pueden combinarse para hacer búsquedas más selectivas.

Con los resultados obtenidos pueden crearse nuevos mapas. Podremos hacer consultas espaciales, haciéndonos preguntas del tipo ¿Qué, ...? por medio de la selección de registros en el mapa o en la base de datos, combinando varias capas, buscar coincidencias, intersecciones, ...

Podremos medir distancias, por ejemplo buscando la distancia más corta, ...; se pueden hacer análisis de proximidad creando áreas de influencia (buffer) alrededor de un elemento espacial; superponer mapas mezclando datos de capas diferentes, ...

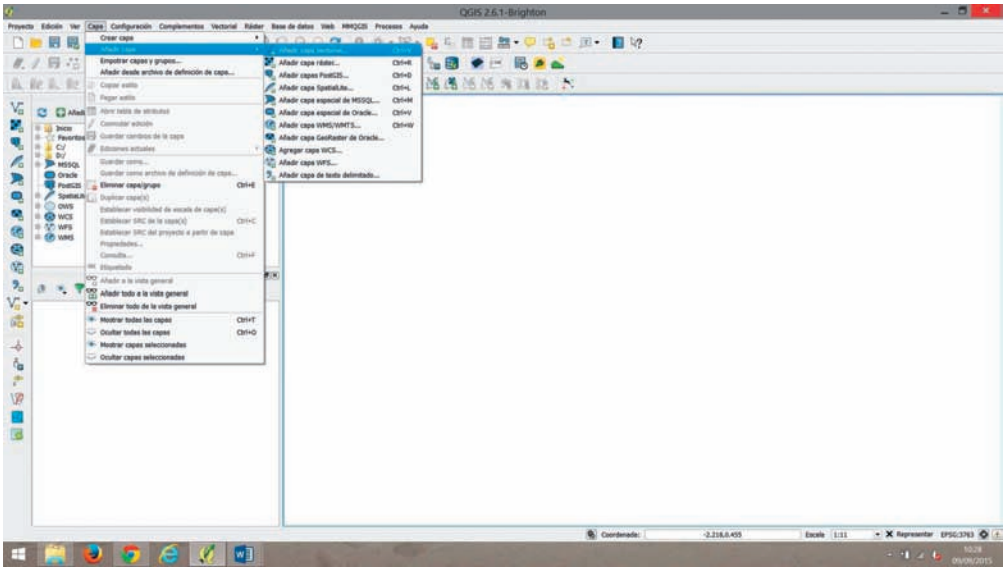

Figura 1.2

También podremos hacer un análisis de *redes* (arcos conectados como carreteras, red de ferrocarril, tendidos eléctricos, ...), pudiendo calcular caminos óptimos entre dos puntos teniendo en cuenta aspectos como distancia, tiempo empleado, coste, tráfico, ... o podremos calcular áreas de influencia de zonas de servicio.

Si trabajamos con superficies en donde se observan variables de tipo continuo (modelos TIN) podremos hacer cálculos de pendientes, calcular cuencas de drenaje, etc.

Figura 1.3

1.4.2. Ejemplo de QGIS vectorial

A continuación veremos un ejemplo de utilización de QGIS vectorial que permitirá ilustrar conceptos antes considerados. Estos datos han sido obtenidos del Instituto Geográfico Nacional y del Ministerio de Agricultura, Alimentación y Medio Ambiente. En el Prólogo de este libro damos una dirección de Internet de dónde se pueden descargar los datos de este ejemplo y todos los utilizados en el libro.

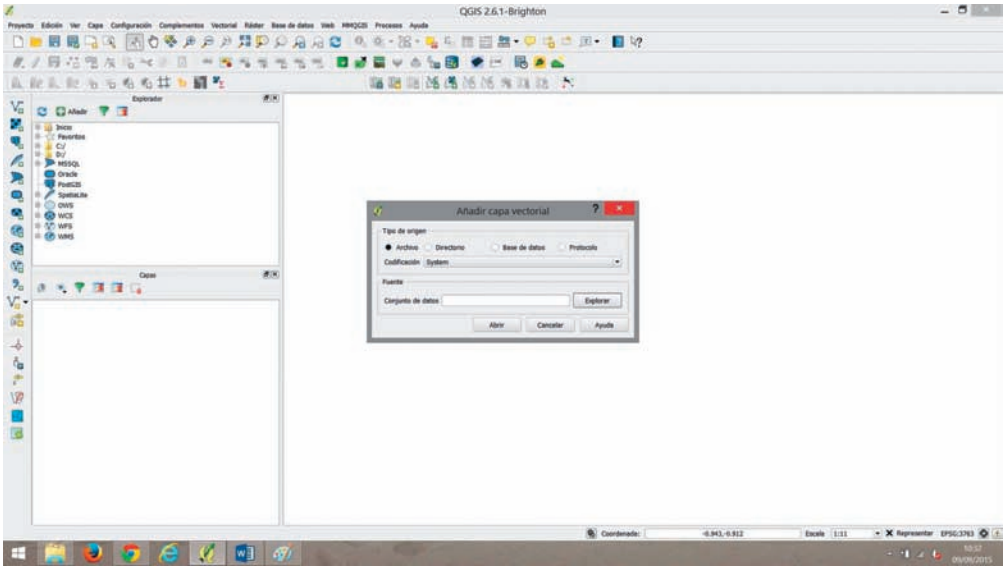

Figura 1.4

Ejemplo 1.1

Se está realizando un estudio de los humedales españoles para comprobar su situación y comprobar su viabilidad, conservación, etc. Para analizarlo con QGIS primero incorporamos la capa geográfica de la Península Ibérica que será el marco geográfico necesario para situar espacialmente los humedales. Esta capa geográfica es el fichero

```
recintos_autonomicas_inspire_peninbal_etrs89.shp
```

Cuando ejecute QGIS deberá incluir en el subdirectorio en el que esté este fichero Shapefile otros ficheros con información complementaria que utiliza QGIS; se trata de los ficheros con el mismo nombre y extensiones dbf, shx y prj.
Podemos incluir este mapa base de la Península (el fichero shp anterior) de dos formas: bien con la secuencia

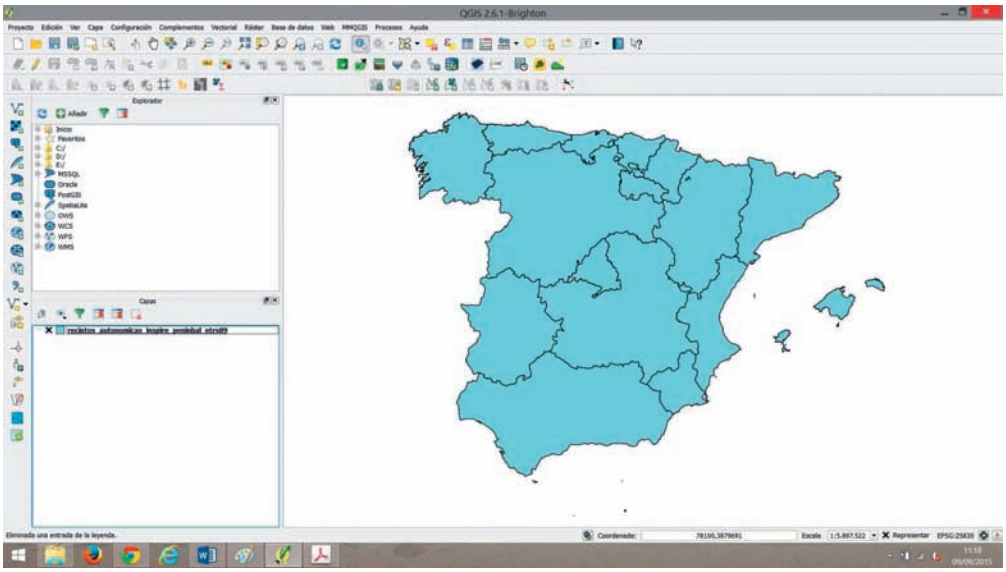

Figura 1.5

`CAPA -> Añadir Capa -> Añadir Capa Vectorial`

en donde `CAPA` está en el zona 1 de Menús (véase la Figura 1.1). Este proceso aparece en la Figura 1.2. Alternativamente se puede incorporar este fichero utilizando el botón de acceso rápido, Figura 1.3, que se encuentra en el lateral izquierdo, zona 2 de Herramientas (Figura 1.1).

Se abrirá a continuación un cuadro de diálogo donde seleccionaremos la opción de abrir un archivo y lo buscaremos en nuestro directorio, Figura 1.4.

Después de estas operaciones obtendremos la Figura 1.5 con el mapa deseado.

Al terminar este proceso aparecerá un nuevo elemento en la zona 3 de Capas (véase la Figura 1.1).

QGIS elige los colores por lo que es posible que en su ordenador le aparezca otro color distinto al de la Figura 1.5 o uno diferente cada vez que importe el mapa. Algunas de las muchas posibilidades que ofrece QGIs, como el cambio de colores, se analizarán en el capítulo siguiente.

Ahora añadiremos al mapa de fondo antes seleccionado, la capa de los humedales de la Península Ibérica siguiendo los mismos pasos anteriores y abriendo el fichero `IEZH.shp`. De esta forma habremos obtenido la Figura 1.6. Observemos que ha aparecido una nueva capa en la zona 3 de capas de QGIS. De hecho, como dijimos más arriba, cada vez que incorporemos capas, éstas irán apareciendo como elementos nuevos en esta zona.

Al igual que antes, indicamos al lector que, en el mismo subdirectorio que tenga este fichero, deberá tener otros 5 ficheros con el mismo nombre y extensiones dbf, sbn, sbx, shx, prj y xml.

Podemos observar en nuestro mapa de la Figura 1.6 de la Península Ibérica, los humadales añadidos. En algunos casos sólo se aprecian como puntos al existir otros humedales de mayor

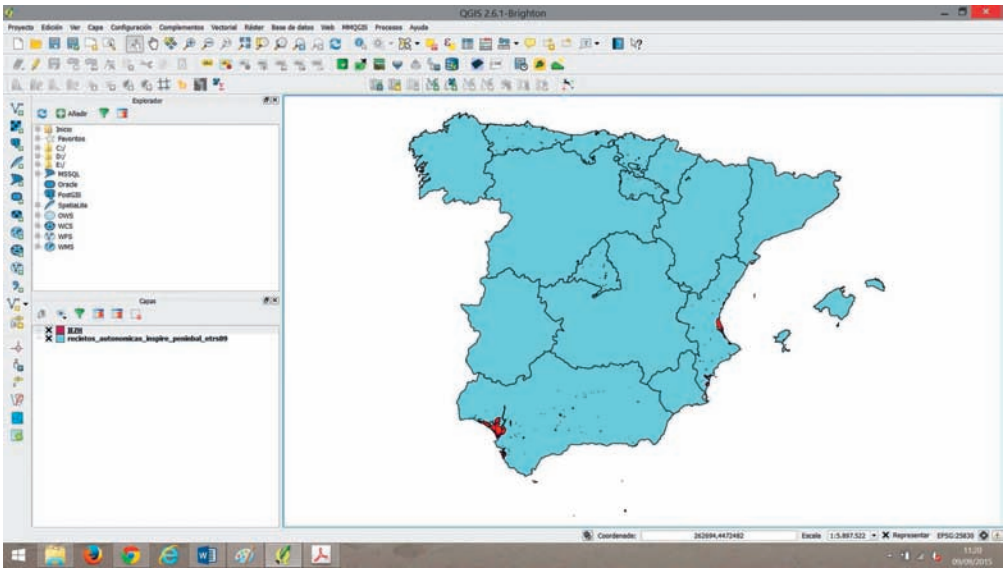

Figura 1.6

tamaño. En otros lugares, los humedales aparecen como manchas al tener éstos una mayor extensión, pero todos ellos se corresponden con humedales localizados geográficamente.
Si abrimos la tabla de atributos de la capa, podemos observar que ésta contiene los datos incorporados: nombre del humedal, longitud, anchura, etc. Para abrir la tabla de atributos, clicaremos con el botón derecho del ratón sobre el nombre de la capa de la zona 3 de capas, seleccionado la opción `Abrir Tabla de Atributos`. Alternativamente, podemos abrirla mediante el botón de acceso directo situado en la zona 2, debajo de la línea de menús, Figura 1.7, obteniendo así el mapa dado por la Figura 1.8 en donde aparece sobre-impresionada la tabla de atributos.
Podemos obtener información de cada humedal mediante el botón que aparece en la Figura 1.9 que, al presionarlo, convierte el cursor del ratón en un puntero. Ahora, posicionando éste sobre un humedal determinado y clicando, aparece información sobre este humedal en un cuadro, información que se corresponde con la que aparece en la tabla de atributos, Figura 1.10.
Nuestro mapa de la Península Ibérica final con los humedales, podemos guardarlo como un *proyecto QGIS* mediante la opción Guardar como del menu Proyecto de la zona 1. De esta forma, podremos abrir este fichero recién creado en otra sesión posterior para reanudar o modificar el mapa antes guardado.

Figura 1.7

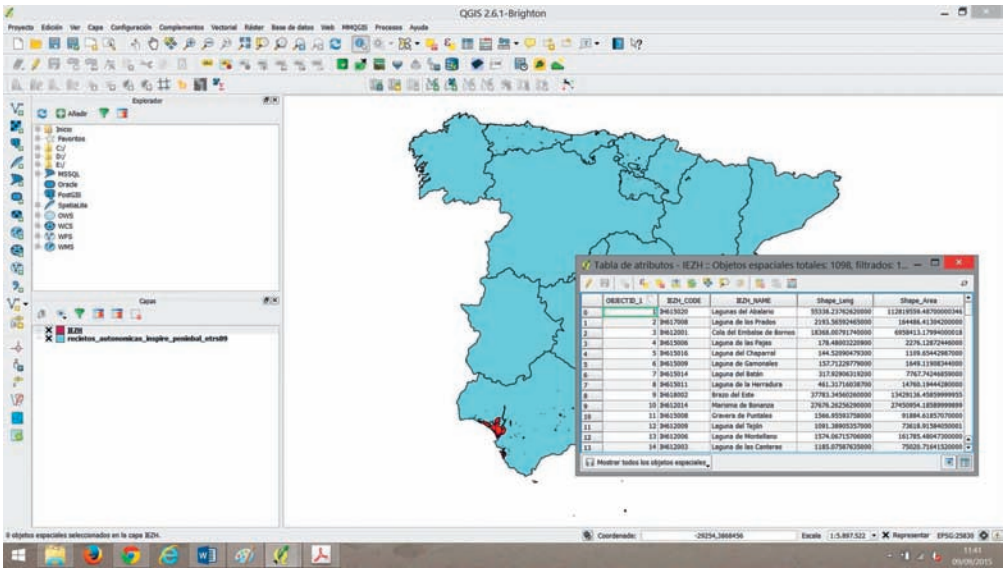

Figura 1.8

Figura 1.9

1.4.3. GIS raster

Los GIS raster se basan en análisis y representación de localizaciones espaciales. Se utilizan para representar datos continuos, sin límites marcados, zonas de transición y datos que cambian (espacios naturales, altitudes, precipitaciones, ...) aunque, de forma más imprecisa, también permiten representar entidades.

En un GIS raster trabajamos con mapas, fotografías de satélites, ortofotografías, ..., en donde la realidad se representa como una retícula rectangular dividida en cuadrículas, celdillas o píxeles de igual tamaño donde las celdas no se solapan: los datos se definen por la posición en una fila y columna, localización relativa, (coordenadas (x, y)).

Dependiendo del tamaño del píxel, la resolución será mejor o peor (a mayor tamaño del píxel, menor resolución). Para determinar el tamaño correcto, puede seguirse la norma de utilizar como tamaño del píxel la mitad de la longitud más pequeña a representar.

En el caso en que dos valores queden representados en la misma celda, se

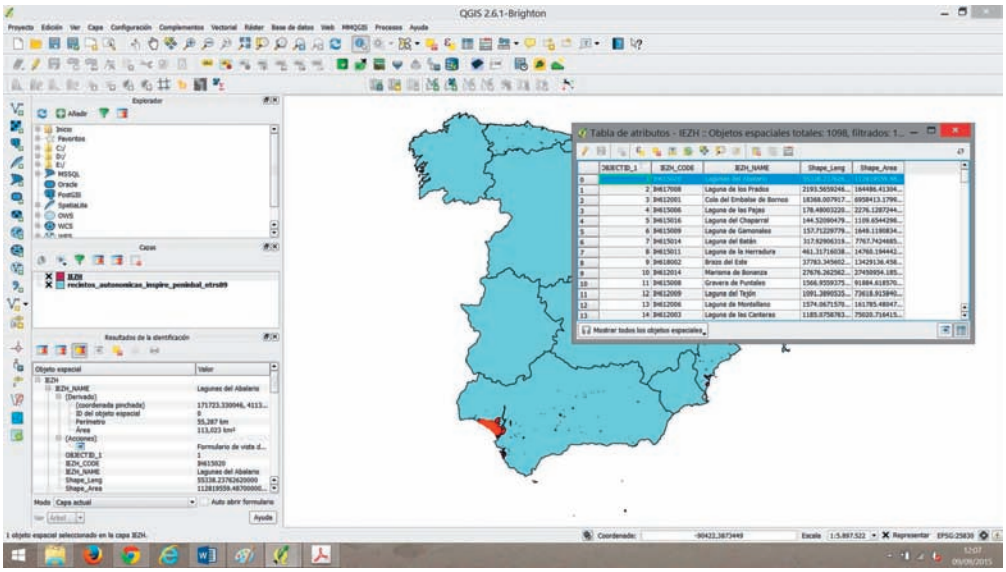

Figura 1.10

asignará como valor de la celda el valor que tenga una mayor presencia o el que quede en el centro.

Sobre un mapa podemos crear una retícula para incorporar los valores en formato raster.

Cada celda o píxel representa una parte del espacio geográfico y tiene unas características propias, valores relacionados con la variable que representa.

Agruparemos las celdas formando los objetos que existen en la realidad.

Para representar entidades utilizamos una celda para un punto, celdas alineadas para representar una línea y celdas contiguas para un polígono. Este sencillo sistema plantea problemas por una falta de exactitud: un punto o línea no ocupa toda una celda, sino una parte de ella y esto presenta dificultades a la hora de saber cuál es su tamaño o forma, y sólo aumentando la resolución podremos concretarlo.

Representar altitudes, precipitaciones, presiones atmosféricas, etc., variables que presentan variaciones continuas en el espacio es sencillo pues cada celda incluye el valor de dicha variable.

Al igual que en los GIS vectoriales, también se trabaja creando capas con cada grupo de datos y, empleando la misma retícula, podremos hacer estudios más complejos al relacionar unas capas con otras.

Los formatos de ficheros de datos raster son ASCII y ficheros con extensiones bil, bin, bsq y grid. Los ficheros de dibujos, mapas y fotos que utilizaremos

en el GIS raster tendrán habitualmente extensiones tiff, jpeg, gif, png y eps.

Cada capa raster genera dos tipos de archivos: en uno se guardan los valores de las celdas (formato ASCII) y en el otro se guarda la información general de la retícula, de la leyenda, orientación, resolución, número de filas y columnas, tipo de variable, etc.

Combinando diversos mapas y analizando sus variables generaremos mapas nuevos con nueva información y nuevas variables.

Como hemos indicado anteriormente, los GIS raster trabajan con mapas que podemos obtener de forma variada como por ejemplo, *escaneándolos* aunque, en este caso, tendríamos que tener cuidado con la resolución, para no incrementar notablemente en número de celdas, así como con los mapas que ofrecen datos de más de una variable, como un mapa topográfico donde tenemos variables de altitudes, hidrografía, usos del suelo, etc.

Otra forma de obtener mapas es por medio de los *satélites* que aportan información actualizada pero que puede presentar inexactitudes. Su uso es muy recomendable en estudios sobre inundaciones, incendios, ect. Podríamos importar ficheros en distintos formatos y convertirlos en alguno de los formatos raster más utilizados como los anteriormente mencionados.

También se puede rasterizar información vectorial (pasarla a formato raster) creando un malla de celdas y volcando la información en ellas. Las celdas tendrán valores si se corresponden con zonas donde hay puntos, líneas o polígonos; si no, aparecen en blanco.

Cuando abrimos una capa raster es fundamental que la información sea clara para poder interpretarla. Lo más habitual es asignar colores a los valores si se representan variables cualitativas adjuntando una leyenda explicativa. Si trabajamos con variables cuantitativas que tomen numerosos valores, lo mejor es agruparlos en intervalos y asignar colores a cada intervalo siguiendo una gradación progresiva para facilitar el análisis. También podríamos utilizar los propios valores dentro del mapa lo que nos permitiría validar los resultados.

Trabajar con mapas tridimensionales también es una opción en donde podríamos elegir la perspectiva, la escala, etc. Son mapas como los topográficos o geológicos que hoy en día pueden utilizarse para representar otro tipo de datos: la forma de representar las altitudes podrían usarse para representar datos de población, concentración de algún elemento, etc.

Dependiendo de nuestras necesidades podemos alterar un mapa raster para hacerlo más funcional cambiando la orientación, la resolución uniendo celdas por suma o por suavizado (media aritmética). Si la información está dividida en varias hojas podemos unirlas, dividirlas o extraer una zona concreta.

También podemos hacer operaciones de análisis basadas únicamente en el valor de la celda, considerándola de forma aislada. Podemos hacer dos tipos de análisis: El primero que denominamos *reclasificaciones* o también *recodificaciones* que consisten en, partiendo de un mapa con celdas obtener otro mapa

nuevo con valores diferentes en dichas celdas. Aquí tenemos dos posibilidades dependiendo del tipo de variables que se estén considerando:

Si trabajamos con variables cualitativas podemos hacer, o bien una recodificación de clases dando nuevos valores a las celdas con las que vamos a trabajar, o bien podremos hacer una agregación de clases agrupando los valores del mapa original creando grupos y recodificando de nuevo.

Por ejemplo, si contamos con una distribución de 50 enfermedades, cada una con un valor asignado, y en nuestro mapa sólo aparecen 5 enfermedades, podemos recodificarlas del 1 al 5 y podremos hacer una agregación de clases si las dividimos en contagiosas o no (valores 1 y 2).

Por otro lado, si trabajamos con variables cuantitativas podemos, o bien agrupar los valores en intervalos que previamente hemos definido o calculado de forma automática, o bien podremos expresar los valores finales de forma diferente realizando operaciones matemáticas (suma, resta, división, multiplicación, etc.).

Por ejemplo podemos expresar los valores en otro sistema de medidas pasando de metros a kilómetros o eliminar decimales por redondeo hacia el número entero más próximo o por truncado eliminando los decimales directamente, etc.

El segundo tipo de análisis se basa en realizar *superposiciones de mapas* en donde los valores de las celdas de nuestro mapa final se obtendrán combinando los valores de las celdas de los mapas fuente. La combinación de valores podemos obtenerla, o bien realizando operaciones aritméticas, usando ecuaciones matemáticas con sumas, restas, etc., que nos permitan unificar valores, calcular porcentajes, ..., o bien podemos crear premisas que deben cumplirse (superposición lógica) y crear un mapa con las zonas donde se cumplan dichas premisas. Para ello utilizamos las operaciones de lógica booleana como OR, donde se cumple una de las dos premisas planteadas y AND, en donde se cumplen las dos premisas.

Otra opción de análisis de valores de celdas que el GIS raster ofrece es calcular los nuevos valores de las celdas teniendo en cuenta las celdas que la rodean, es decir, las celdas contiguas. Se pueden realizar filtrados calculando el nuevo valor de la celda que será el resultado de calcular la media, o moda, o mediana en relación con las celdas que la rodean y realizar suavizados o realces de los datos obtenidos según las necesidades.

Trabajando con celdas contiguas, si los valores contienen datos de altitud, podemos generar otro tipo de mapas como mapas de pendientes, calcular su orientación y, utilizando estos nuevos mapas, podremos hacer análisis más complejos como determinar cuencas de drenaje, etc.

Si las celdas no están contiguas también podemos generar nuevos mapas basados en el cálculo de las distancias que hay entre las celdas. La mayoría de los GIS permiten hacer estos análisis de proximidad, buffer, de forma rápida

y sencilla generando mapas donde se cumpla la condición lógica que determinemos (qué zona se encuentra a una distancia determinada de un punto de referencia, etc.).

Si tenemos puntos repartidos por nuestro mapa, estos pueden representarse como una celda y se pueden crear polígonos (*polígonos Thiessen*) con las celdas cercanas que tengan igual valor (*teselación de Voronoi*). Este sistema es utilizado para crear áreas de influencia y para trabajar con datos cualitativos.

Sabemos que el espacio geográfico real no es uniforme, que tiene desniveles, ríos, etc., que determinan los usos de dicho espacio, la distribución de infraestructuras, etc. Esto tendrá que ser tenido en cuenta para trabajar modelizando el terreno o calculando distancias.

1.4.4. Ejemplo de QGIS raster

En esta sección analizaremos unos datos que se utilizarán en otras partes del libro.

Figura 1.11

Ejemplo 1.2

Los datos `meuse10.txt` corresponden a localizaciones y concentraciones (en un área de aproximadamente 15×15 metros) de metales pesados en la capa superior del suelo, recogidos en una llanura de inundación del río Mosa, cerca de la localidad holandesa de Stein datos tomados de Rikken y van Rijn (1993). La matriz de datos es de la forma

```
     x      y cadmium copper lead zinc   elev       dist   om ffreq soil lime landuse dist.m
181072 333611    11.7     85  299 1022  7.909 0.00135803 13.6     1    1    1      Ah     50
181025 333558     8.6     81  277 1141  6.983 0.01222430 14.0     1    1    1      Ah     30

.....................................................................................

179466 330381     0.8     21   51  162  9.406 0.35860600  5.7     3    1    0       W    460
180627 330190     2.7     27  124  375  8.261 0.01222430  5.5     3    3    0       W     40
```

en donde las dos primeras columnas son las localizaciones en coordenadas RDM (un sistema de coordenadas topográficas holandés); las cuatro siguientes, concentraciones en partes por millón de metales pesados; `elev` la elevación relativa sobre la llanura; `dist` la distancia GIS al Mosa; `om` materia orgánica del suelo; las cuatro siguientes, variables de tipo cualitativo y, finalmente, `dist.m` la distancia en metros al Mosa.

Para analizar estos datos con QGIS, primero vamos a incorporar la tabla txt de datos para trabajar con datos raster. Nuestra tabla tiene las coordenadas planas (x, y) pero también incorpora datos de en lenguaje QGIS se denomina *elevación*, que no necesariamente se refiere a altura sino que, en este caso, son las cantidades encontradas de minerales en las capas superficiales de la tierra. Estos valores nos servirán, mediante la creación de un modelo de

elevaciones del terreno (MDT), para medir la mayor o menor presencia de estos minerales y marcar la zona en donde se encuentran.
Para ello creamos la capa raster a partir del fichero de texto, `meuse10.txt`. Podemos hacerlo de dos formas: utilizando el botón de acceso rápido situado en el margen izquierdo, Figura 1.11, o utilizando el botón `Capa` de la zona 1, eligiendo después la opción `Añadir Capa` y, dentro de ésta, `Añadir capa de texto delimitado`, Figura 1.12.

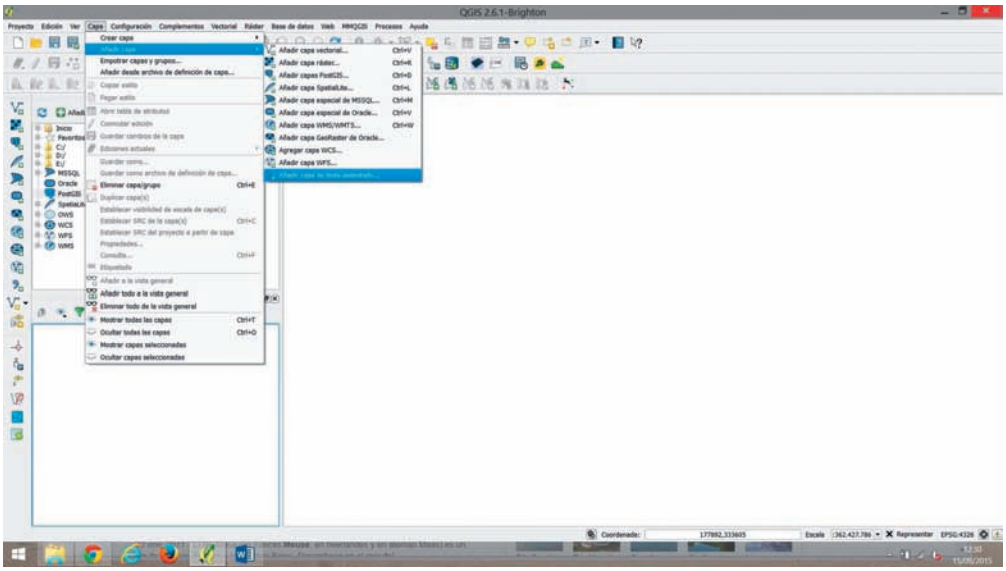

Figura 1.12

En el cuadro de diálogo que se abre, buscaremos nuestro fichero txt con la opción `Explorar`. Daremos un nombre a nuestra capa: Mosa-raster y marcaremos las opciones de formato: "delimitadores personalizados" y "espacio". Como opciones de registro marcaremos "El primer registro tiene los nombres de los campos" y como Definición de geometría marcaremos "Coordenadas del punto" y "Aceptaremos", Figura 1.13.
En algunas ocasiones cuando incorporamos capas o datos, QGIS nos pide especificar el SRC de la capa para que su representación sea correcta. Para ello, al aceptar las opciones anteriores, puede abrirse un cuadro de diálogo para establecer el SRC de la capa y eligiendo el sistema WGS84 EPSG 4326. Si no aparece en las opciones, podemos ponerlo en el filtro y QGIS lo buscará. Ya sólo tendremos que marcarlo y aceptar, Figura 1.14.
Al aceptar, se cargarán los datos pero, como no es una capa editable, la tendremos que salvar como shp. Para ello, con la capa marcada en la zona de capas y clicando con el botón derecho, guardaremos la capa como shp manteniendo el SRC de la capa y, en codificación, marcaremos la opción "Añadir archivo guardado al mapa", Figura 1.15.
En nuestra zona de capas aparecerá la capa shp con el nombre que hemos dado. La visualización de los datos es la misma que tiene el fichero txt (clicando en la cruz al lado de las capas podemos ocultarlas y mostrarlas) pero la capa shp en sus opciones (clicar con el botón derecho sobre el nombre de la capa) tiene habilitada la opción "Conmutar edición" lo que nos permite trabajar con los datos, cambiarlos, añadir columnas, ..., mientras que la capa txt no presenta esta opción.

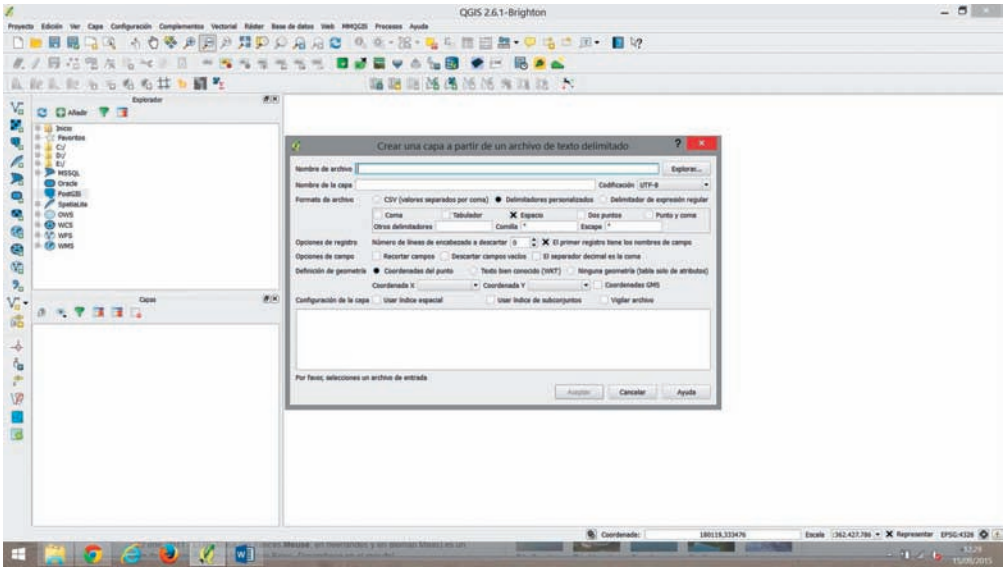

Figura 1.13

A partir de ahora trabajaremos con la capa shp. Lo primero que haremos será una interpolación utilizando la columna "elev" de nuestra tabla de atributos. Para ello iremos al menú `Raster` eligiendo la opción `Interpolación`, Figura 1.16.

En el cuadro de diálogo que se abrirá marcaremos las siguiente opciones: En las opciones de entrada buscaremos nuestra capa vectorial (`meuse10.shp`) y como atributo para realizar la interpolación elegiremos la columna "elev". A continuación clicaremos en `Añadir` para cargar nuestra selección. En las opciones de salida elegiremos como método de interpolación, la interpolación triangular (TIN) y daremos un nombre de salida a nuestra nueva capa, Figura 1.17.

Al aceptar la selección, aparecerá la capa cargada en la gama de colores grises. Para verla correctamente, clicaremos dos veces sobre el nombre de la capa y abriremos la opción `estilo`, en donde elegiremos como tipo de randerización: `unibanda pseudocolor`. En `Generar nuevo mapa de color`, elegiremos los colores según nuestro criterio y clicaremos sobre `clasificar` para incorporar los nuevos colores a nuestra capa, Figura 1.18.

Lo que obtenemos es una gama de colores adaptada a la elevación del terreno. Para hacer la lectura de nuestra capa más útil podemos elegir que la gama de colores sea degradada con lo cual podremos ver mayores elevaciones en las zonas con los colores más intensos y menores elevaciones donde los colores sean más suaves, Figura 1.19.

Ahora podemos incorporar a nuestra capa *curvas de nivel* que unen todas las zonas que tienen la misma elevación. Para ello, en el menú `Raster`, elegimos la opción `Extracción` y en ella, `Curvas de Nivel`, Figura 1.20.

Se abrirá de esta forma un cuadro de diálogo en donde marcaremos el archivo de entrada donde se incorporarán las curvas de nivel, en este caso el archivo es la capa creada con la interpolación. Daremos un nombre de salida a la capa que creamos y seleccionaremos el intervalo de separación de cada curva. En este ejemplo, al no tener unas altitudes muy diferenciadas, bajamos a 1 metro la separación de dichas curvas. Aceptamos la selección y

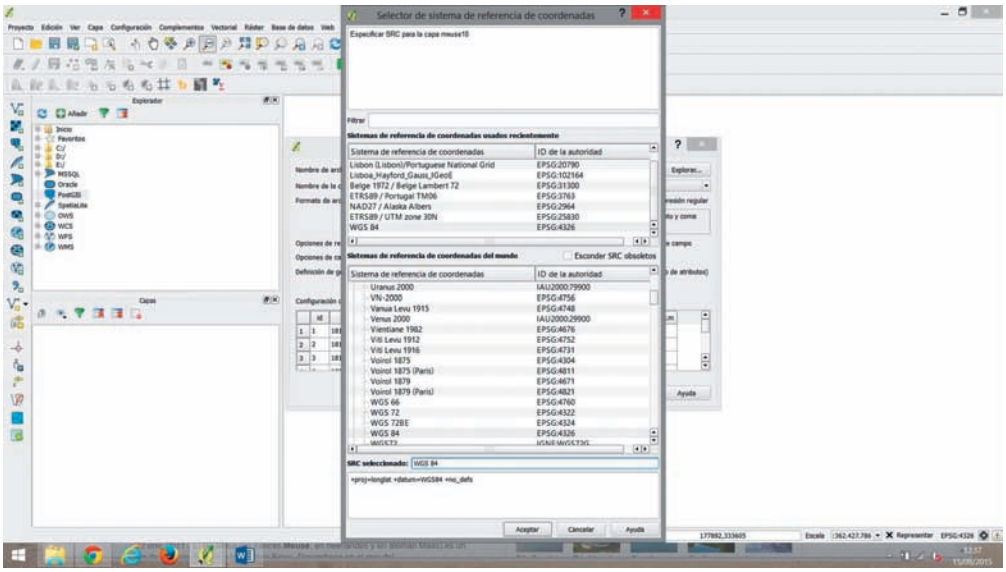

Figura 1.14

esperamos a que QGIS genere las curvas y las incorpore a nuestra capa, Figura 1.21.

De esta forma se añadirá una nueva capa tal en nuestra zona de capas y, dependiendo del color en el que aparezcan, podremos cambiarlas e incrementar su grosor para lograr una mejor visualización accediendo a la opción estilo, tal y como vimos anteriormente, Figura 1.22.

A simple vista vemos que la menor o mayor intensidad del color nos hace pensar donde el terreno es más o menos elevado. Podemos realizar de forma muy sencilla perfiles topográficos que reafirmen lo que visualmente intuimos. Para ello es necesario tener incorporado en QGIS un complemento denominado "Profile Tool". Es un complemento de Python que descargaremos, si no lo tenemos, desde el menú `Complementos`, para después elegir `Administrar e instalar complementos`. De entre todos los complementos que se nos muestran, elegiremos `Profile Tool` e instalaremos.

Posicionados sobre la capa de interpolación creada y activando el icono de `Profile Tool` (Figura 1.23) o accediendo al complemento desde el menú `Complementos`, se abre un área de trabajo en donde aparecerán los perfiles que nosotros dibujemos.

Posicionándonos sobre la zona de trabajo y clicando, al arrastrar con el ratón trazando líneas, podremos ver el perfil, Figura 1.24, en la ventana que se ha abierto, e incluso, a través de la pestaña `Table` podremos ver todos los registros de elevaciones de nuestro trazado y podremos copiarlos para pegarlo en una hoja de cálculo si lo necesitamos, Figura 1.25.

Una opción interesante que ofrece QGIS es realizar análisis del terreno, donde podemos ver sus características físicas como por ejemplo, generar una capa de pendientes de esta zona. Para ello, en el menú `Raster` elegiremos la opción `Análisis del terreno` y, después, `Pendiente`, Figura 1.26.

En el cuadro de diálogo que se abre a continuación marcaremos la capa de altitud, la creada con la interpolación, daremos un nombre de salida con formato tiff y con `factor` Z, "1". Con la opción marcada de `Añadir resultados al proyecto`, aceptaremos la selección realizada,

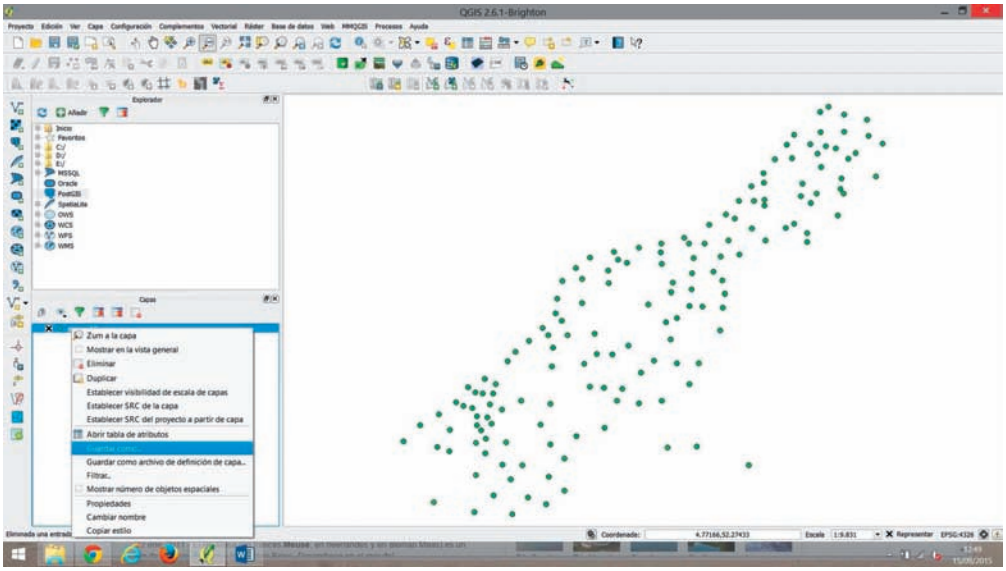

Figura 1.15

Figura 1.27.
El resultado nos ofrece una capa raster en colores grises, los cuales cambiaremos igual que hicimos anteriormente accediendo con doble clic a los ajustes de estilo de la capa creada para obtener una mejor imagen, Figura 1.28.

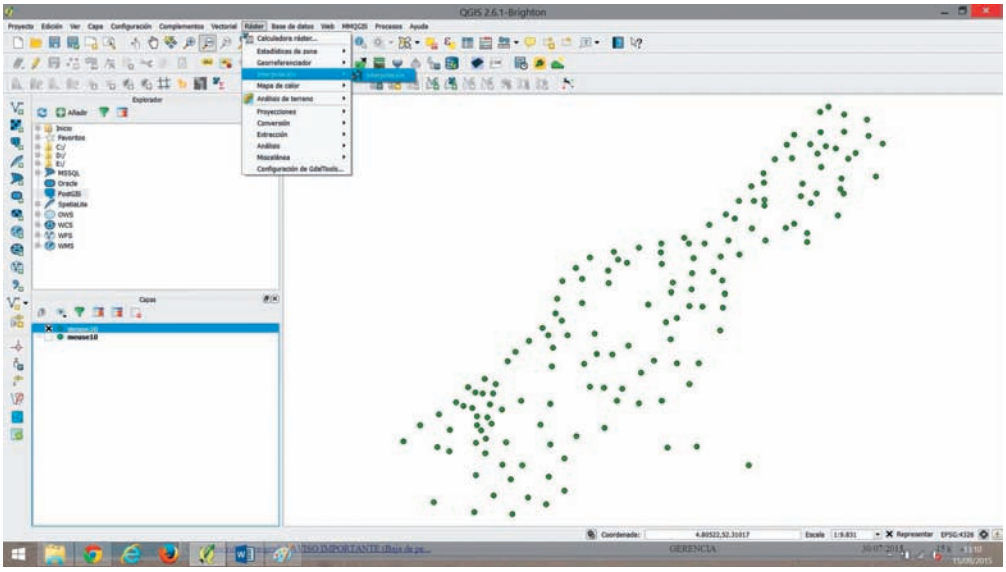

Figura 1.16

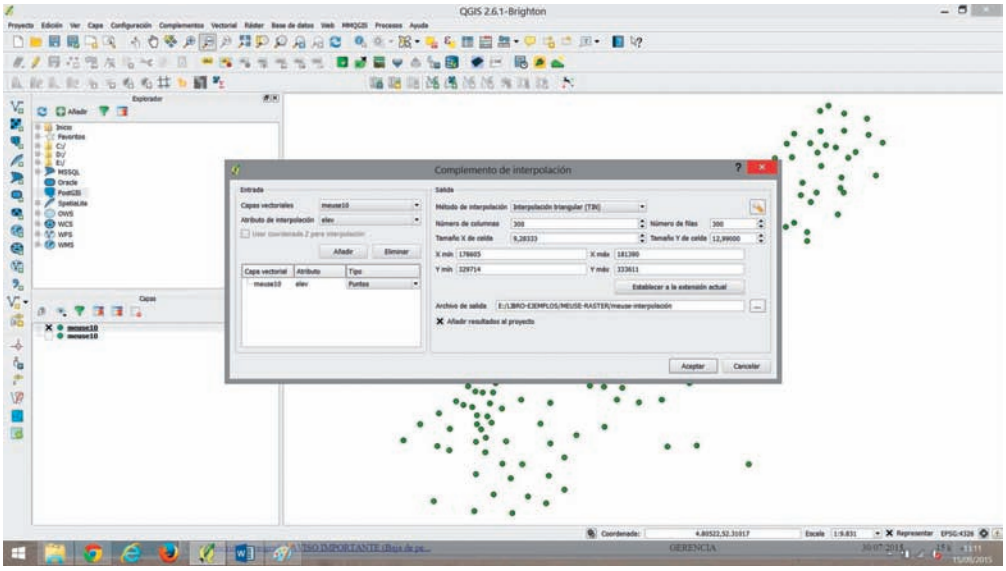

Figura 1.17

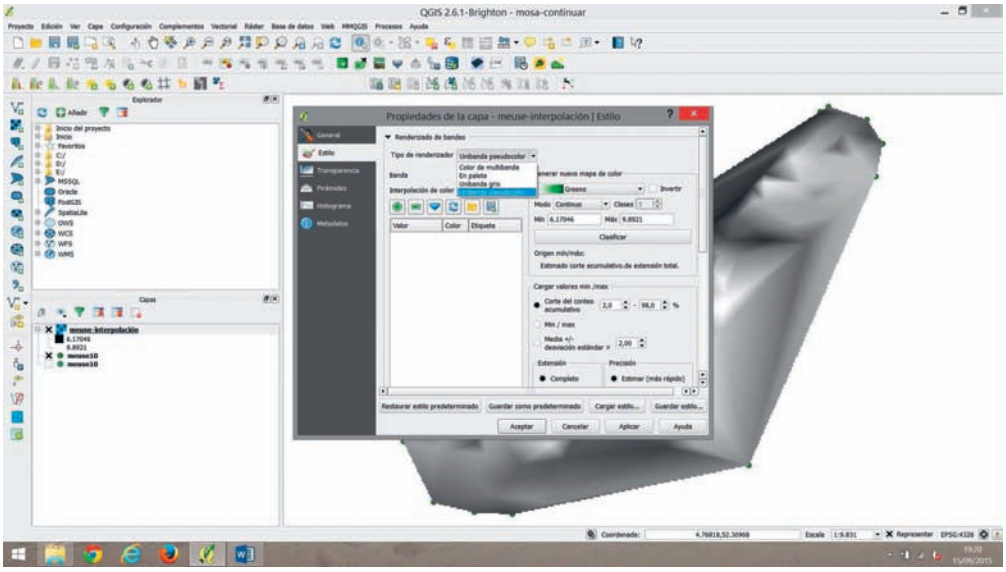

Figura 1.18

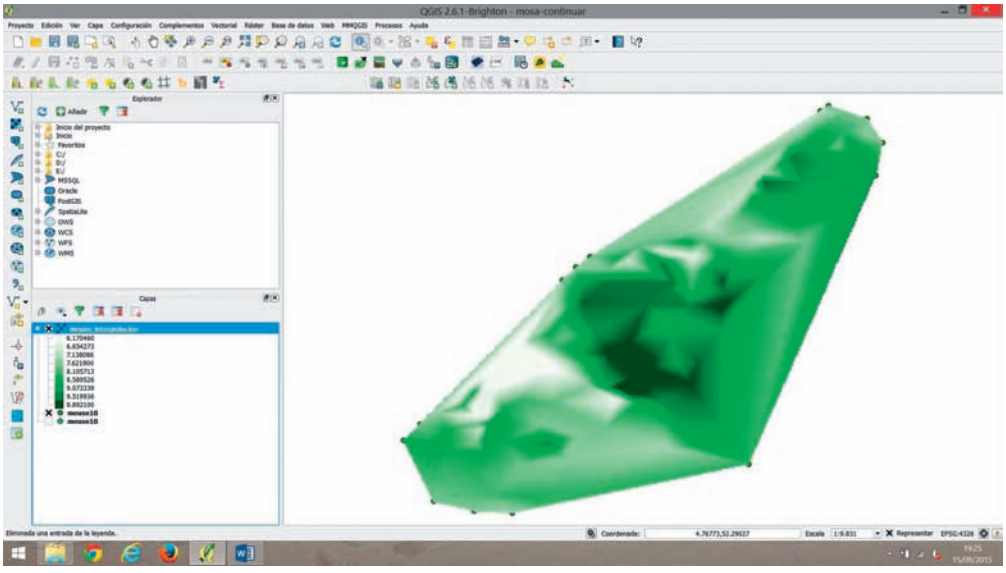

Figura 1.19

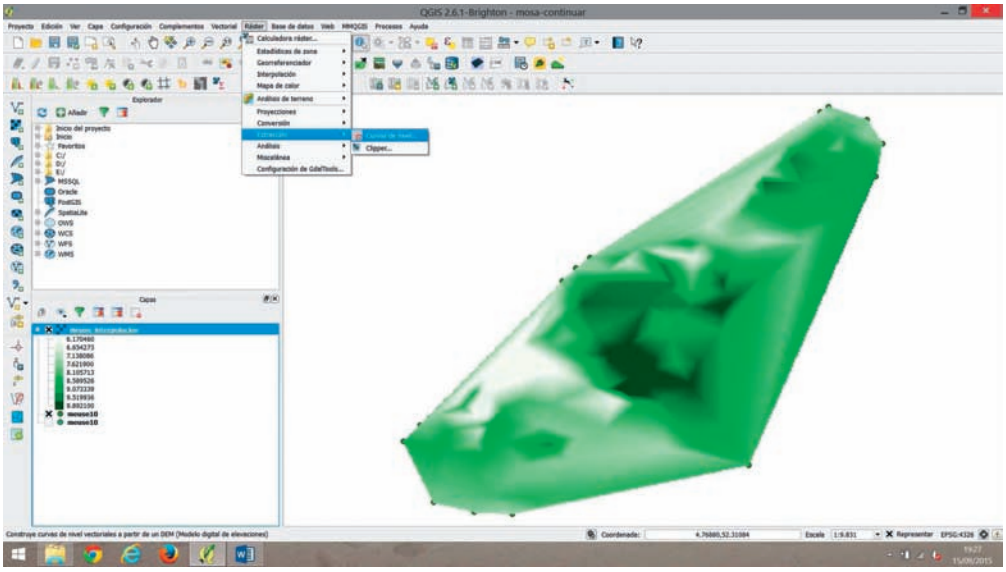

Figura 1.20

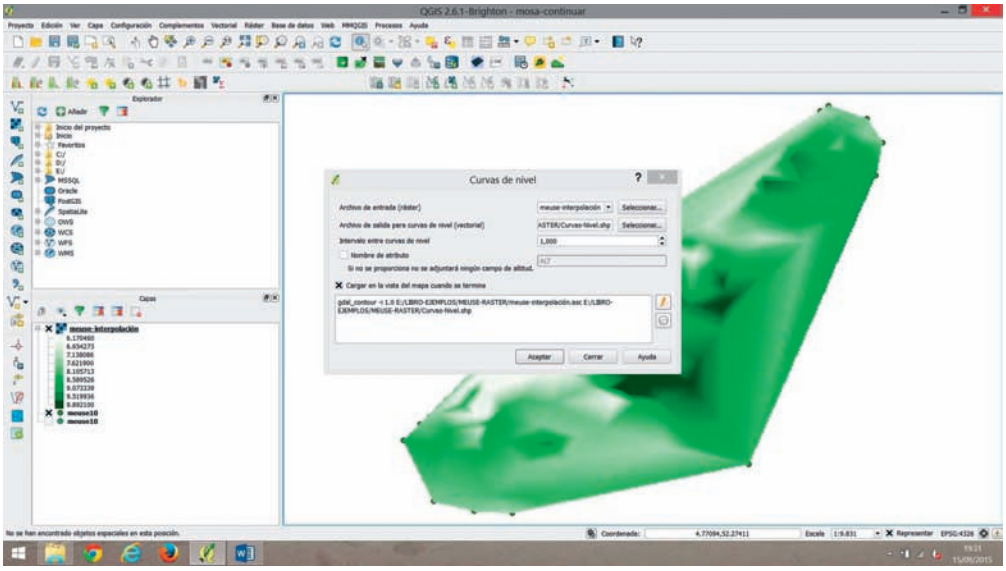

Figura 1.21

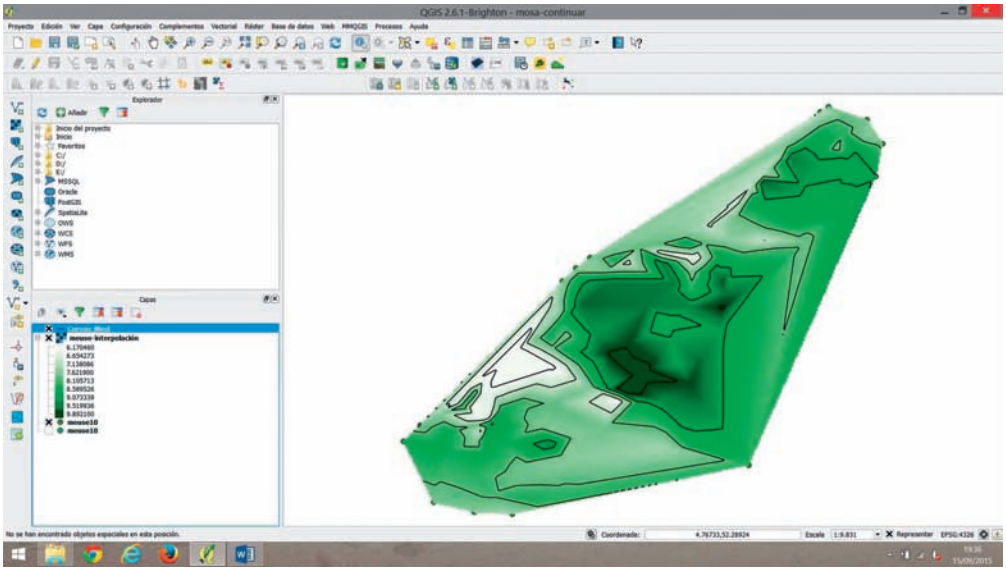

Figura 1.22

Figura 1.23

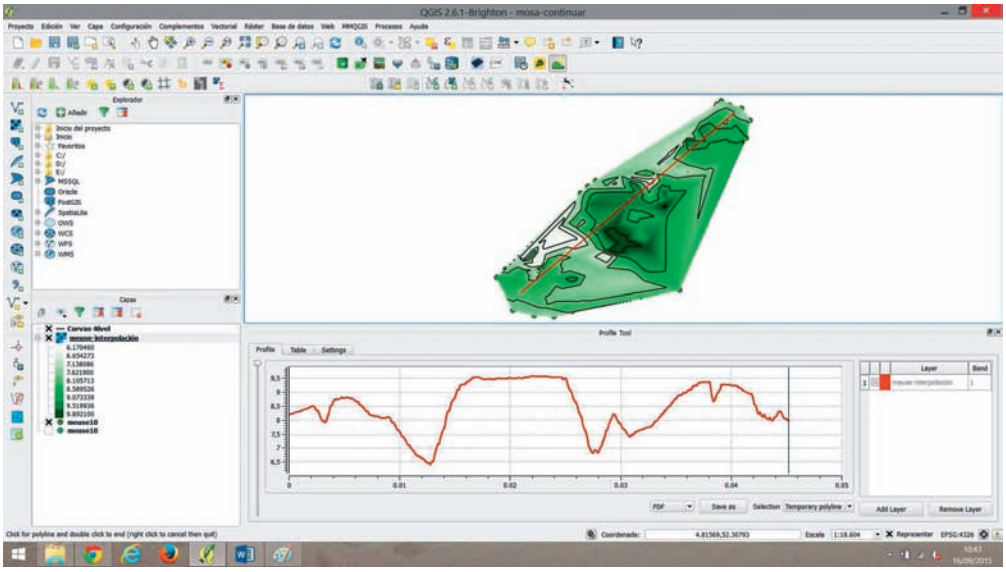

Figura 1.24

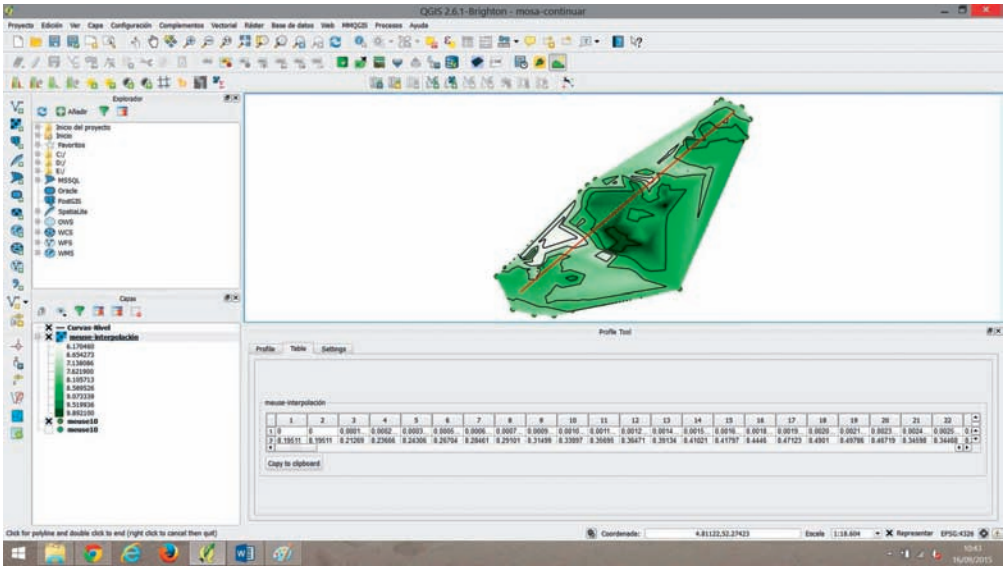

Figura 1.25

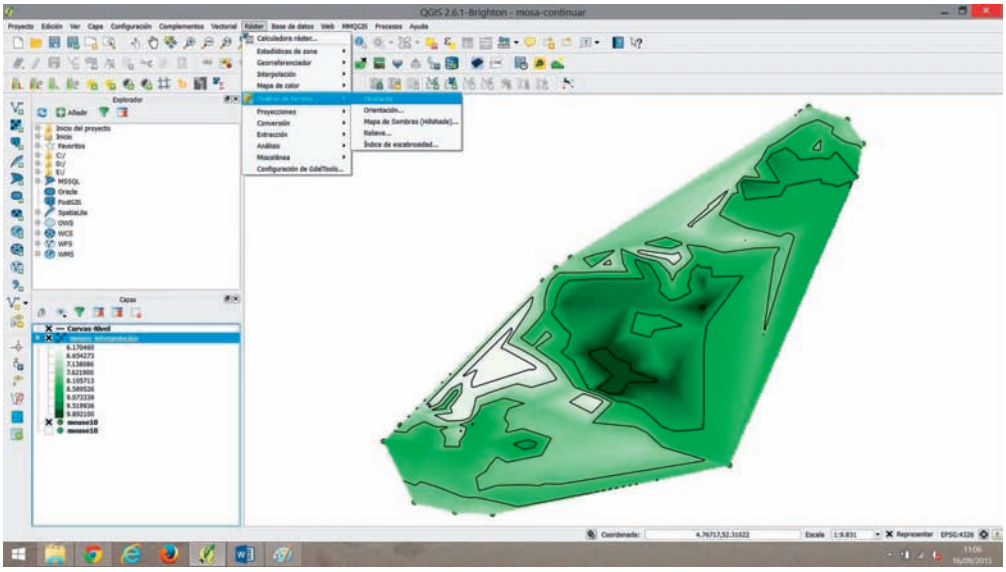

Figura 1.26

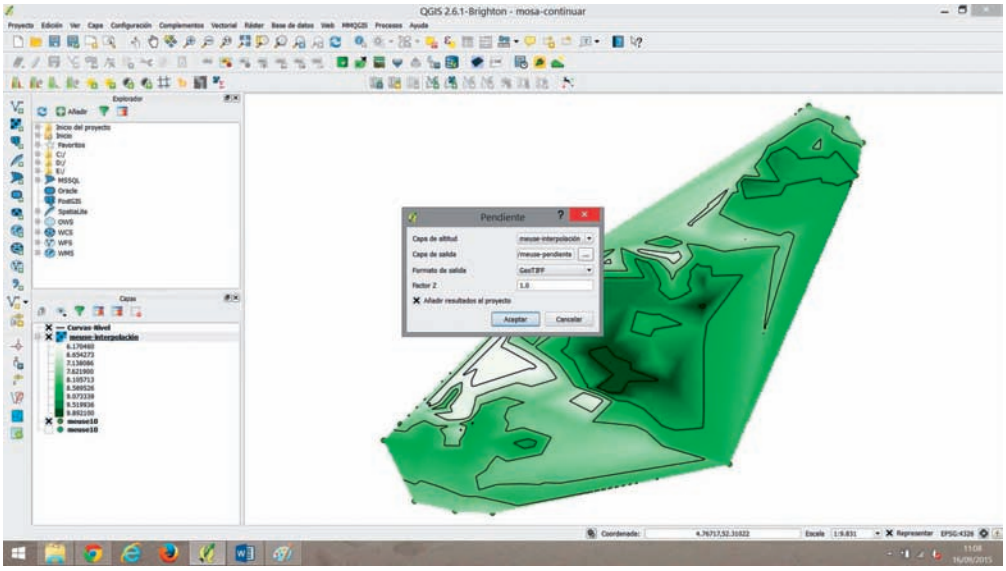

Figura 1.27

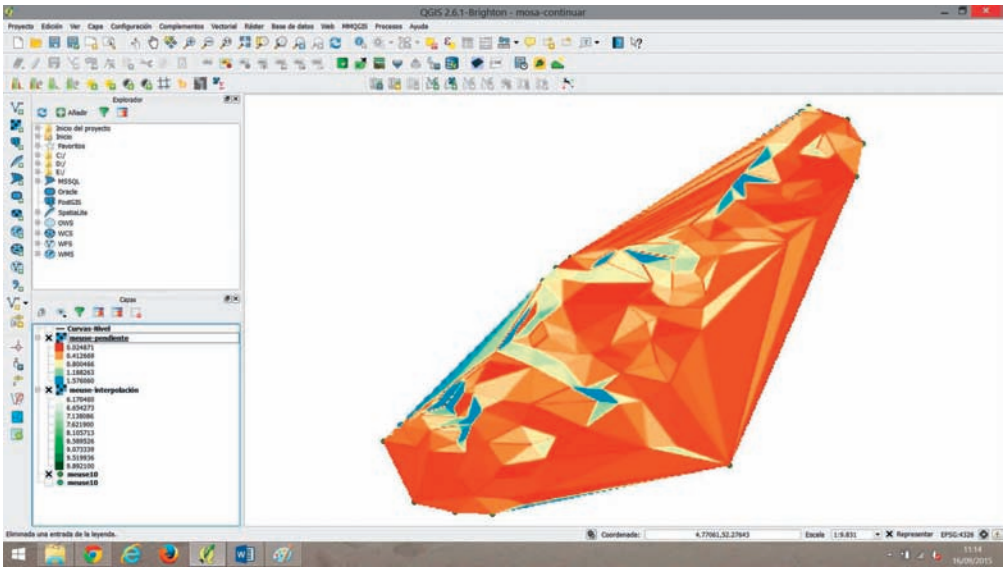
QGIS 2.6.1-Brighton - mosa-continuar
Proyecto Edición Ver Capa Configuración Complementos Vectorial Ráster Base de datos Web MMQGIS Procesos Ayuda
Explorador
Añadir
Inicio del proyecto
Inicio
Favoritos
C:/
D:/
E:/
MSSQL
Oracle
PostGIS
SpatiaLite
OWS
WCS
WFS
WMS
Capas
Curvas-Nivel
mosae-pendiente
mosae-interpolación
mosae10
mosae10
Eliminada una entrada de la leyenda.
Coordenada:
4,77061,52,27643
Escala
1:9.831
Representar
EPSG:4326
11:14
16/09/2015

Figura 1.28

Capítulo 2

Utilización y Manejo de QGIS

2.1. Introducción

En este capítulo estudiaremos algunas de las numerosas posibilidades que ofrece QGIS en cuanto al análisis descriptivo de datos espaciales. No pretende ser un manual de QGIS sino que enseña las utilidades más frecuentes de este programa aunque, dada la brevedad del capítulo, muchas de ellas quedarán fuera.

En los dos primeros ejemplos utilizaremos como *mapa de fondo*, es decir como base geográfica, mapas obtenidos del Instituto Geográfico Nacional. En el tercero utilizaremos parte de unos datos similares a los que aparecen en algún manual de QGIS. (Ver página web del libro para mayor detalle de obtención de dichos mapas.)

2.2. Incorporación de Tablas de Datos

Comenzaremos la sección con un ejemplo de representación de datos que viene recogidos en una tabla, la cual se ha transformado en un fichero dbf. Estos datos se incorporarán al mapa mediante un *identificador* común, es decir, una columna común tanto a la tabla como al mapa. En éste y en los sucesivos ejemplos utilizaremos determinadas propiedades de QGIS. Aquí la que utilizaremos será Uniones.

Ejemplo 2.1

Los datos de la Tabla `Cáncer-Pulmón` corresponden a la incidencia del cáncer de pulmón en las distintas comunidades autónomas españolas de la península. Para representar dicha incidencia, lo primero que haremos será incorporar la capa geográfica vectorial

`recintos_autonomicas_inspire_peninbal_etrs89.shp`

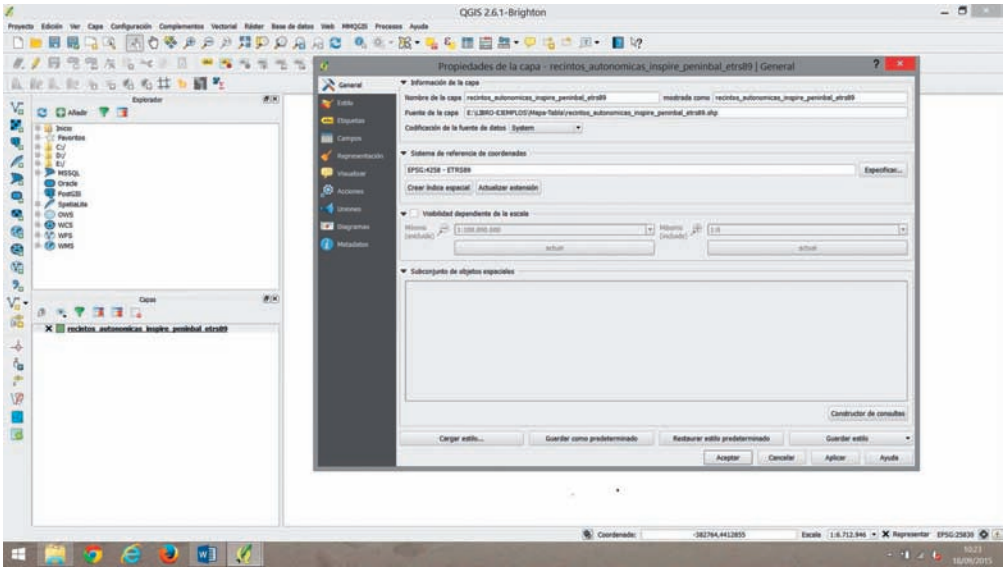

Figura 2.1

Figura 2.2

como hicimos en el Ejemplo 1.1.

Los datos de incidencia del cáncer están en una hoja de cálculo y no en un fichero shp aunque para incluirlos en QGIS debemos hacerlo como si fuera una capa vectorial shp eligiendo el fichero de la hoja de cálculo.

Este fichero aparecerá como una capa en la relación de capas, pero no se visualiza en la zona de trabajo puesto que no es una capa gráfica.

Para poder incorporar datos de la tabla a la capa geográfica del mapa de España, uniendo ambas informaciones, tenemos que asegurarnos que en ambas tablas de atributos, hay al menos una columna con identificadores comunes, puesto que será ésta la que emplearemos para realizar la unión.

En este caso al abrir las tablas de atributos de ambas capas observamos la existencia de varias columnas comunes con igual extensión e información.

Posicionándonos sobre la capa geográfica y clicando dos veces sobre ella (en la zona de capas) se abre un cuadro que nos va mostrando la información de dicha capa y distintas opciones para adaptarla a nuestras necesidades, Figura 2.1.

En la pestaña `General` aparece información genérica sobre nuestra capa: nombre, donde está el fichero, el SRC, etc. Pero vamos a centrarnos ahora en la pestaña `Uniones` pues es aquí desde donde podemos incorporar nuestros datos. Más adelante profundizaremos en otras pestañas que nos ayudaran a mejorar la presentación de los datos.

Al activar esta pestaña se abren una serie de opciones para incorporar tablas de datos a la

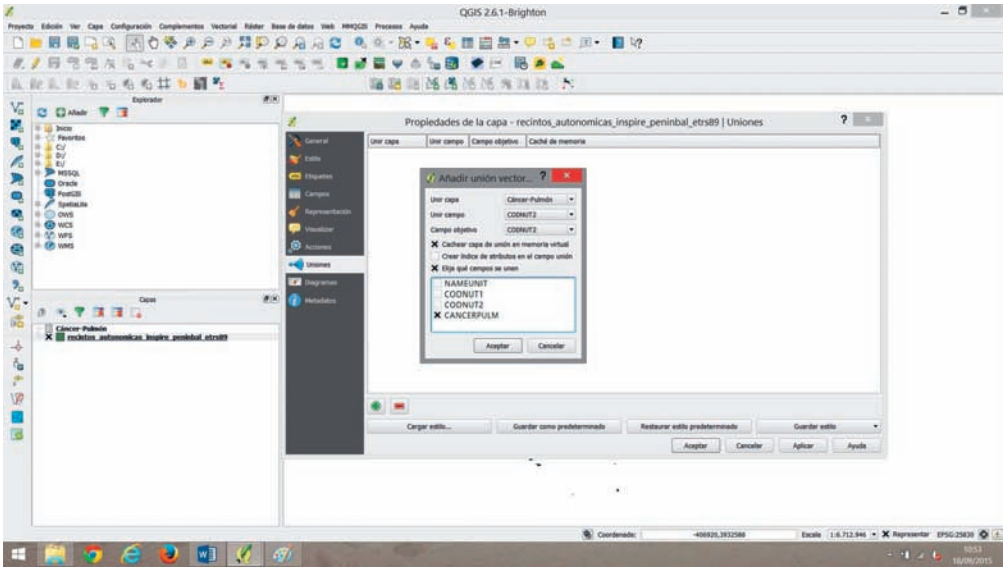

Figura 2.3

capa geográfica. Lo primero será clicar sobre el botón `mas` para que se abra un cuadro de diálogo. Este botón está situado en la parte inferior izquierda, Figura 2.2.

Rellenaremos a continuación los datos que nos pide. En `Unir capa` le indicamos, Cáncer-Pulmón; en `Unir campo`, elegimos la columna CODNUT2; en `Campo objetivo`, CODNUT2; y, finalmente, activaremos la casilla `Elegir qué campos se unen`, seleccionado aquí CANCERPULM. Por último, aceptamos nuestras selecciones con `Aceptar` para incorporar los datos, Figura 2.3.

Ahora trabajaremos con la pestaña `Estilo`. Es interesante pues nos permite cambiar la apariencia de la capa. Observemos que al incorporar la capa, el color en el que aparece lo elige QGIS al azar, pero podemos cambiar su apariencia a través de las opciones que se nos muestran.

Seleccionaremos aquí la opción `Categorizado` y elegiremos la columna que hemos incorporado con los datos.

En cuanto al color, para facilitar la lectura, elegiremos un color graduado dentro de la `Rampa de Color`; en este caso azules y, finalmente, clicaremos en `Clasificar` para que nos aparezcan los datos con su color asociado. Observemos que la intensidad del color se incrementa a medida que los porcentajes son mayores. Figura 2.4.

Al `Aplicar` y `Aceptar`, los cambios se incorporan a la capa geográfica. Podremos cambiar la apariencia de los datos tantas veces como queramos.

Ahora a través de la pestaña `Etiquetas` incorporaremos los datos numéricos en porcentaje. En el cuadro de diálogo, elegiremos `Etiquetar la capa con` y activaremos el botón de la Figura 2.5 para acceder a las siguientes opciones que se nos muestras con QGIS, Figura 2.6

Vamos a elegir que en nuestra Etiqueta aparezca el nombre de las comunidades autónomas y, en línea aparte, el porcentaje de cáncer de pulmón observado en cada una de ellas.

Para ello podemos teclear en el campo `Expresión` toda la sentencia o ir buscándola en las opciones que se nos ofrecen. Cada campo incorporado aparecerá con dobles comillas, mientras

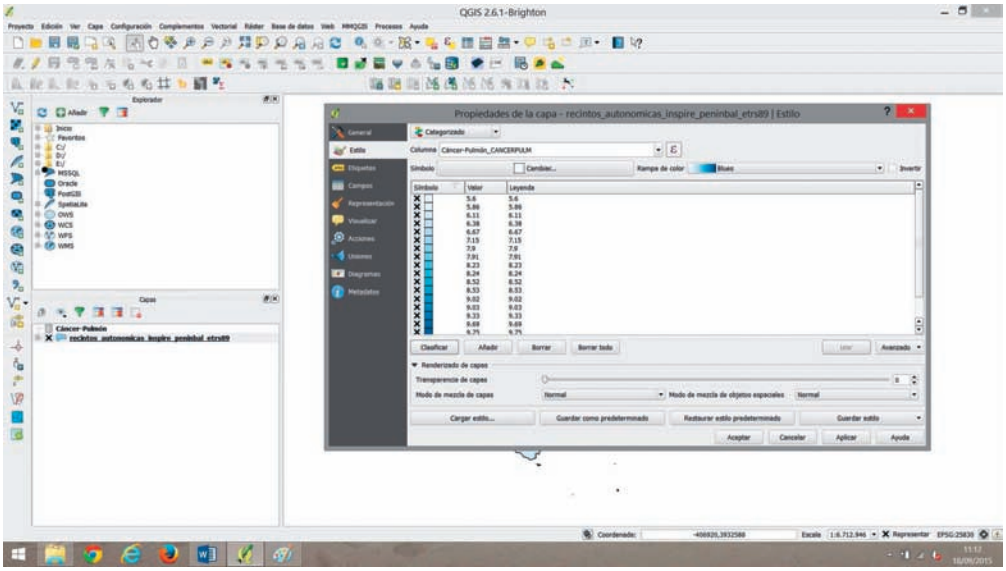

Figura 2.4

Figura 2.5

que la condición que queremos se cumpla, deberá ponerse entre comillas simples, utilizando el operador || para separar sentencias.
De esta forma, en el desplegable `Campos y valores` marcamos la opción `nameunit`, que es el nombre de la columna donde aparecen, en nuestra tabla, los nombres de las comunidades autónomas. Con doble clic lo incorporamos al cuadro de expresión, apareciendo en éste con dobles comillas.
Como queremos que el porcentaje de incidencia de cáncer aparezca en línea aparte, escribiremos otra sentencia para indicar el cambio de línea, separada de la anterior con el operador || e indicando entre comillas simples

```
\n
```

Volveremos a poner el operador || y añadiremos con doble clic el campo de datos del cáncer cuya información está en la columna

```
Cáncer-Pulmón_CANCERPULM
```

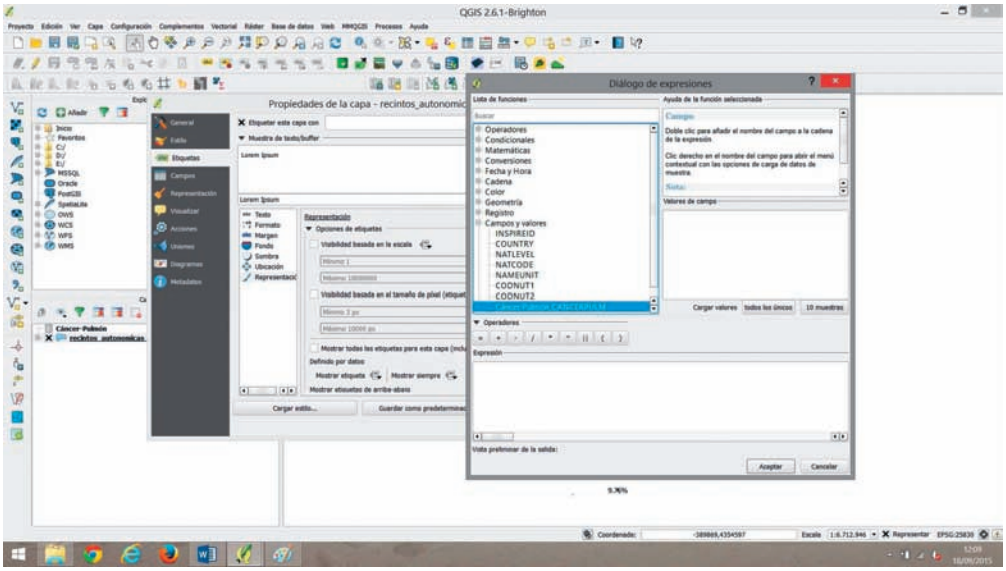

Figura 2.6

Volveremos a utilizar ahora el operador || para indicar que los datos aparezcan como porcentajes.

Podríamos escribir entre comillas simples % o elegir este símbolo en el desplegable `Operadores` añadiendo nosotros las comillas simples.

Si la sentencia que hemos escrito:

```
"NAMEUNIT"  || '\n' ||  "Cáncer-Pulmón_CANCERPULM"  ||  '%'
```

es correcta, en la vista preliminar de la salida, aparecerá, pero si nos hemos equivocado, aparecerá un mensaje de error indicando que la expresión no es válida.

Todo esto viene recogido en la Figura 2.7.

Aceptaremos las opciones elegidas y seguiremos trabajando con la pestaña `Etiquetas` para indicar en la opción `Margén` que dibuje `Buffer de texto`. En `Ubicación` marcaremos `Forzar puntos dentro del polígono` y en `Representación`, marcaremos `Todos`, para que QGIS incorpore todos los datos. Aplicaremos y Aceptaremos para ver nuestros datos incorporados al mapa, Figura 2.8.

Ahora guardaremos nuestro trabajo como Proyecto QGIS para poderlo utilizar de nuevo si fuera necesario.

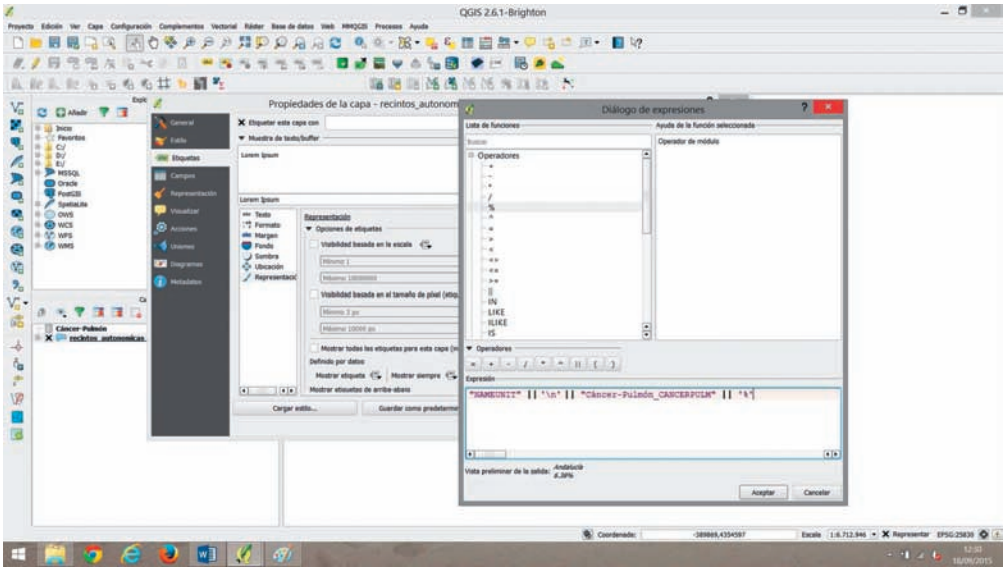

Figura 2.7

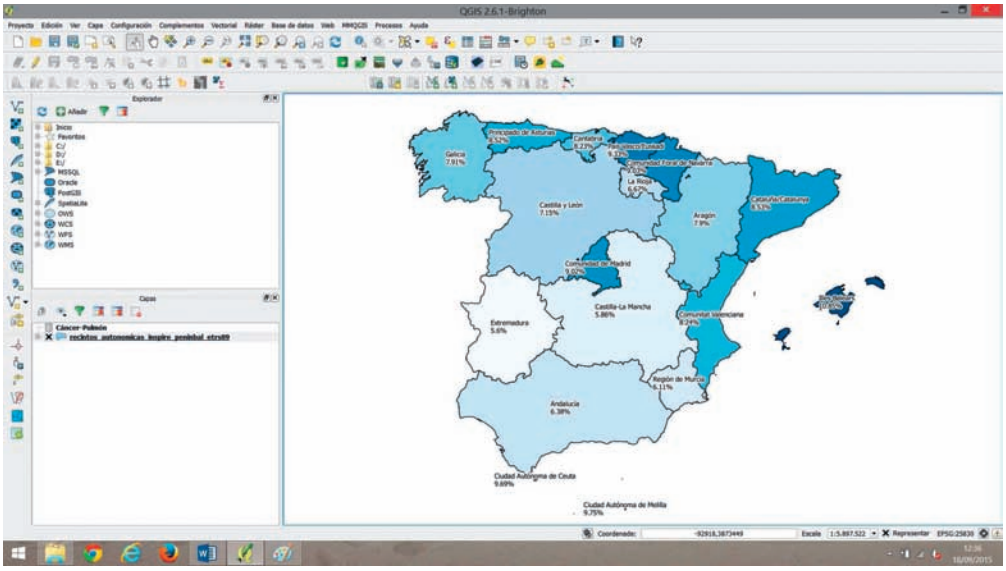

Figura 2.8

2.3. Selección Espacial

En este apartado explicaremos cómo seleccionar una parte del mapa geográfico y sus característica asociadas. De hecho, éstas las incorporaremos mediante capas sucesivas.

Ejemplo 2.2

A continuación analizaremos un municipio, en este caso de la comunidad de Madrid, y estudiaremos su entramado viario.

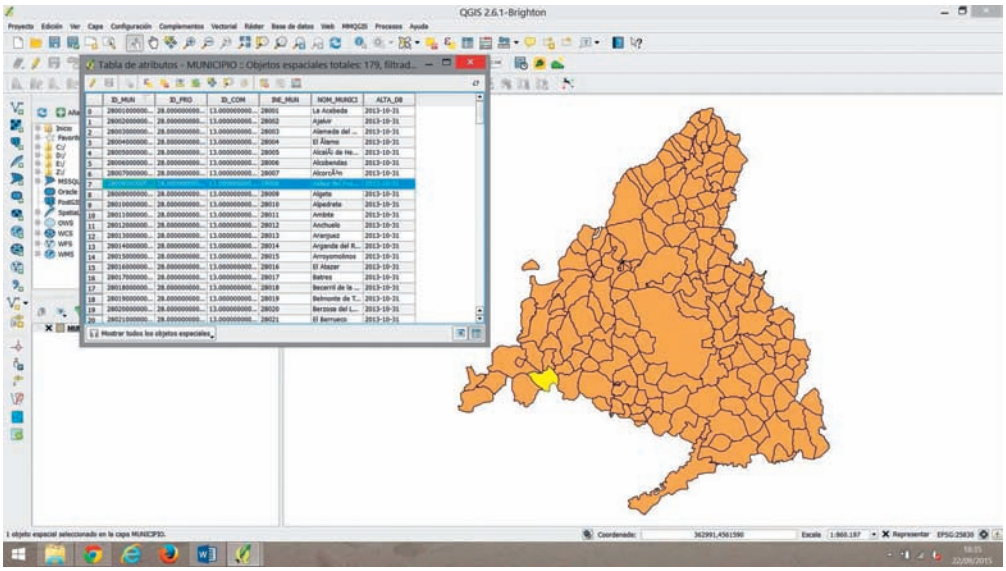

Figura 2.9

Como en el ejemplo anterior, primero incorporaremos el *mapa de fondo*, es decir, la capa geográfica vectorial. En este caso será de la Comunidad de Madrid, capa obtenida también del Instituto Geográfico Nacional al igual que el resto de capas y datos. Ver la página web mencionada en el Prólogo del libro.
La primera capa con la que trabajaremos es

```
MUNICIPIO.shp
```

la cual se incorpora de la misma forma que estudiamos en el Capítulo 1, en concreto en el Ejemplo 1.1.
Tras la incorporación de esta capa aparecerán los municipios pertenecientes a la Comunidad de Madrid. Si abrimos la tabla de atributos, veremos que están todos ellos. Si marcamos ahora una fila de dicha tabla de atributos, el municipio aparecerá resaltado en el mapa como se aprecia en la Figura 2.9.
En este ejemplo vamos a incorporar también las carreteras que pasan por la Comunidad de Madrid. Para ello incorporamos otra capa vectorial, la capa

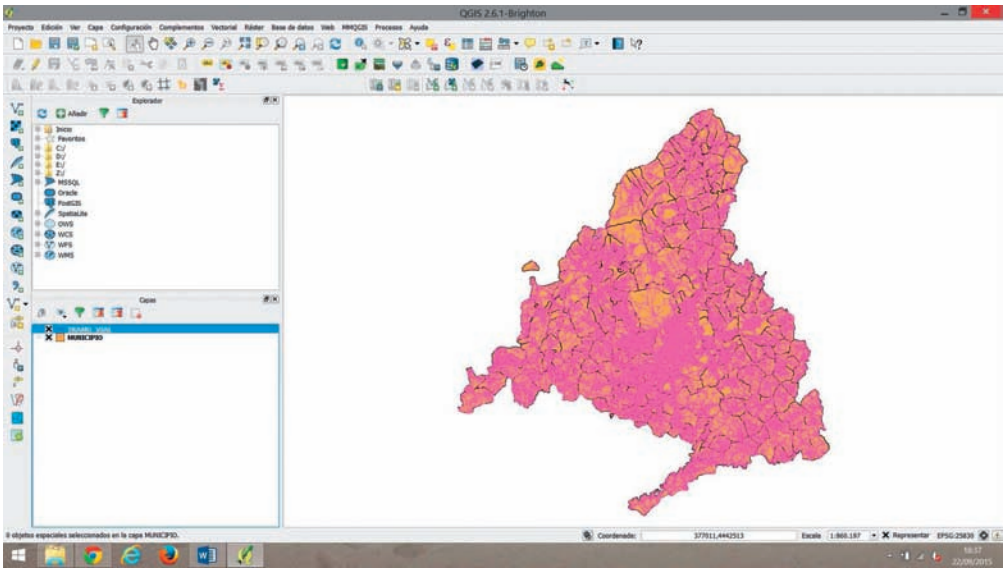

Figura 2.10

Figura 2.11

`TRAMO_VIAL.shp`

Después de incorporarla tendremos un gráfico como el de la Figura 2.10, con la salvedad que hemos hecho en anteriores ocasiones de una posible diferencia en los colores que puede obtener el lector, los cuales se pueden modificar como de hecho, haremos más adelante.
Si queremos trabajar con unos municipios en concreto o con un tipo de carreteras, podemos hacer una selección y guardarla como una capa independiente con formato shp. Esta selección puede hacerse de varias formas: Una, abriendo la tabla de atributos, buscando el municipio y marcando la fila. En este caso, al igual que antes, éste se marcará en nuestro mapa y podremos guardar la selección así efectuada, siguiendo las indicaciones que daremos más abajo.
La otra posibilidad se recomienda utilizar cuando tengamos muchos individuos; es decir, muchas filas en nuestra matriz de atributos. En este caso, es mejor que QGIS realice la selección de forma automática.
Para ello activaremos el botón `Seleccionar objetos espaciales usando una expresión`, Figura 2.11, el cual está en la zona 2 de botones de acceso rápido. Este botón también está disponible en la tabla de atributos.
En ambos casos, el cuadro de diálogo es el mismo. Seleccionaremos en él `Campos y valores` y la columna que contiene los nombres de los municipios,

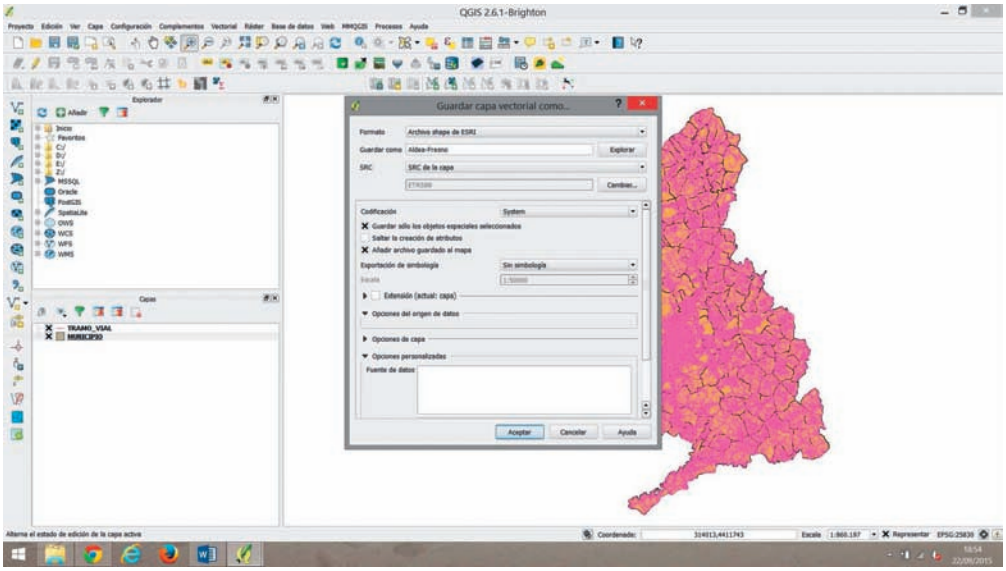

Figura 2.12

Figura 2.13

`NOM_MUNICI`

Con un doble clic la incorporamos a nuestra área de expresión (donde ponemos la premisa que se debe cumplir), añadimos a continuación el símbolo = y, activando la pestaña `Todos los únicos`, nos aparecerán los nombres de todos los municipios para poder elegir el que queremos.
También podemos escribir el nombre del municipio directamente entre comillas simples en nuestra área de expresión. Clicamos sobre `Seleccionar` y nos aparecerá marcado en el mapa (y en la tabla de atributos).
Para crear una capa vectorial con esta selección, posicionados sobre el nombre de la capa, clicaremos con el botón derecho del ratón y elegiremos la opción `Guardar como`.
Ponemos el nombre del municipio, en este caso Aldea-Fresno, y marcamos la opción de que guarde la selección. Si no lo hacemos de esta forma, lo que crearemos es una copia de la capa de municipios. Ver Figura 2.12.
En nuestra zona de capas aparece como capa independiente la que acabamos de crear y con el `Botón de zoom a la selección` (Figura 2.13), la ampliamos para verla mejor.
Observemos que deberemos ocultar las otras capas y ampliar esta acabada de crear con objeto de verla mejor y obtener una gráfica como la Figura 2.14.
A continuación vamos a activar la `Capa del viario` y, de esta forma, podremos hacer una

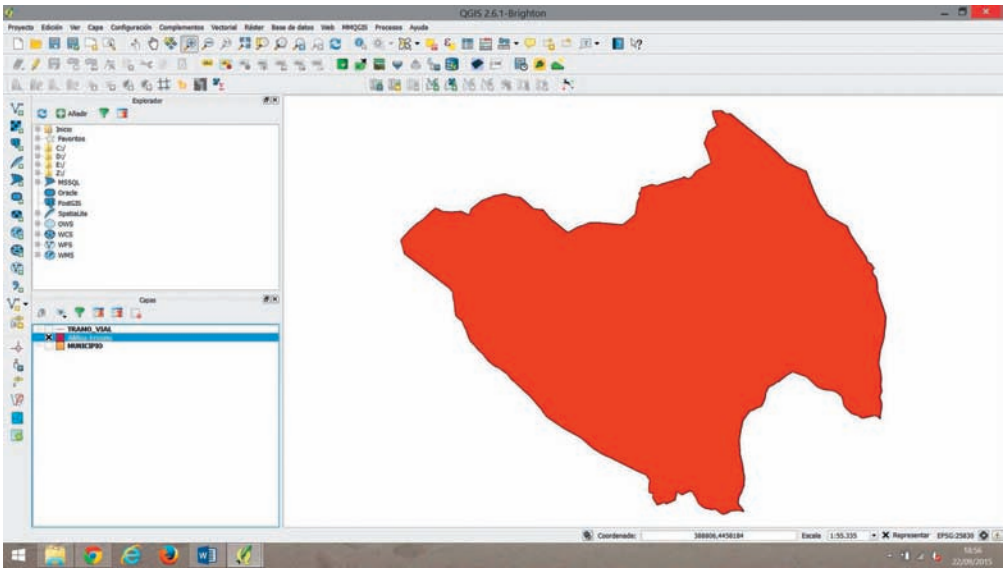

Figura 2.14

Figura 2.15

selección espacial extrayendo los tramos del viario que están dentro de este municipio. Para ello, a través de la secuencia

```
Vectorial -> Consulta Espacial -> Consulta Espacial
```

o clicando sobre el botón de acceso rápido, Figura 2.15, seleccionaremos la capa

```
TRAMO_VIAL
```

en el cuadro de diálogo que se abre y dentro del apartado `Objeto espacial de origen`. En el apartado `Donde el objeto espacial`, podemos elegir diversas opciones del tipo que esté dentro, inconexo, intersecta o toca. Nosotros hemos elegido en esta ejemplo la opción `Dentro`, y `Como objeto espacial de referencia` elegimos la capa del municipio que hemos creado, Aldea-Fresno. A continuación aplicamos. Figura 2.16.
QGIS procesa los datos de acuerdo a nuestras indicaciones y nos abre una ventana en donde nos informa que de 256550 tramos de vial, ha seleccionado 897, que son los que están dentro del municipio elegido, Figura 2.17.

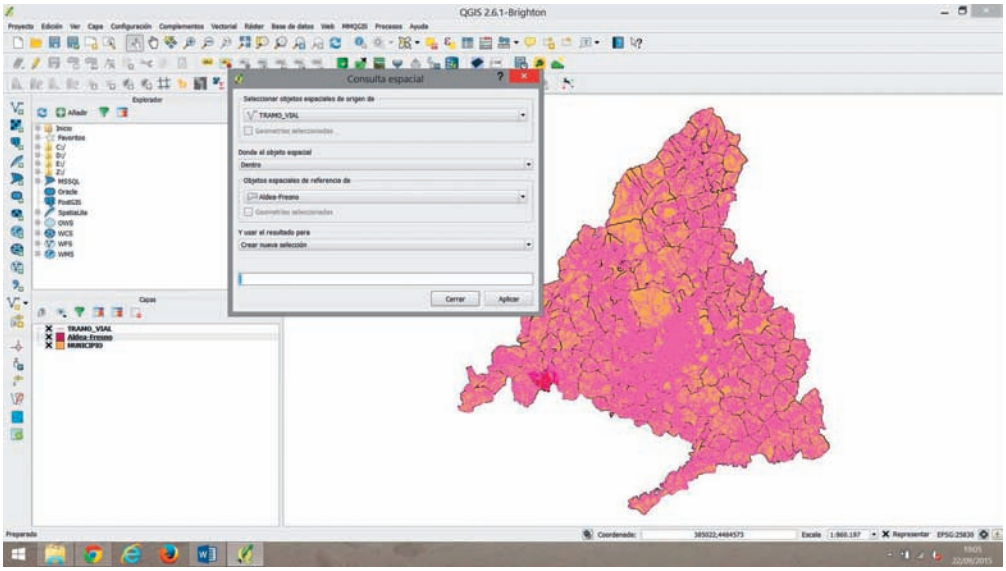

Figura 2.16

Observe el lector en la parte inferior izquierda de la pantalla que QGIS le informa de la selección que ha realizado. De hecho, ésta es una forma rápida de verificar que la selección realizada se ha creado correctamente.
A continuación debemos activar el botón de la Figura 2.18 que aparece dentro del cuadro anterior para incorporar la selección como una capa a QGIS, aplicando a continuación.
De esta forma, QGIS habrá cargado como una capa más en nuestra zona de capas la selección realizada. Figura 2.19.
La tabla de atributos de esta capa nos muestra los tramos en esta zona, y podríamos seleccionar unos en concreto en los que estuviéramos interesados, marcándose sobre el mapa de la misma forma a como lo hicimos más arriba.
En concreto si queremos elegir un tipo particular de viario como las sendas, procederemos de la siguiente forma: en `Campos y Valores` deberemos seleccionar el nombre de la columna que contiene los datos que es

```
TIPO_V_DES
```

añadiendo a continuación el símbolo =, pues queremos seleccionar sólo un tipo de viario, el que sea *igual a* sendas. La opción de abrir `Todos los únicos` es muy útil pues permite incorporar la selección con la grafía correcta. Figura 2.20.
QGIS ha seleccionado 20 sendas dentro de nuestro municipio, las que aparecen marcadas en amarillo y en la parte inferior izquierda de la pantalla.
También aparecen marcadas las filas de la tabla de atributos que hacen referencia a las 20 sendas. Podemos ver todas las sendas juntas seleccionando en la pestaña inferior izquierda `Mostrar objetos seleccionados`. Figura 2.21 y Figura 2.22.

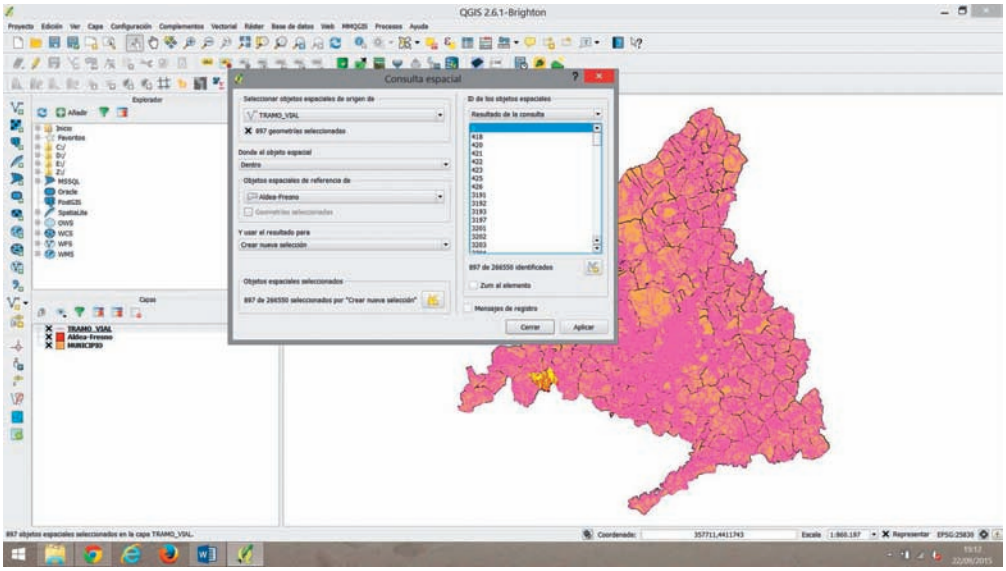

Figura 2.17

Figura 2.18

Observe como en la parte superior de la tabla de atributos (Figura 2.22) QGIS nos indica que, de un total de 897 objetos espaciales (el total del viario que está dentro de nuestro municipio) se han filtrado 20 tramos, que es nuestra selección de sendas.
Ahora podemos guardar esta selección como una nueva capa con el botón derecho sobre el nombre de la capa

```
Tramo_Vial_Dentro_Aldea-Fresno
```

y guardarla como lo hicimos anteriormente. En nuestra zona de capas se incorporará una nueva que hemos llamado `Sendas-Aldea-Fresno`.
Siguiendo este mismo procedimiento, podemos seleccionar todos los tipos de vías que aparecen en este municipio tal y como hemos hecho con Sendas. Una vez que tengamos las capas podemos hacer un grupo con todas ellas y darlo un nombre, como por ejemplo, Viario, el cual incluirá todos los tipos de vías seleccionadas.
Esta opción de hacer grupos con capas que guarden alguna relación nos ayuda a agruparlas en nuestra zona de capas, a visualizarlas o apagarlas todas a la vez con un clic sobre la casilla que antecede al nombre del grupo.
Para hacer grupos, primero clicamos sobre el icono de la Figura 2.23 y, después, arrastramos

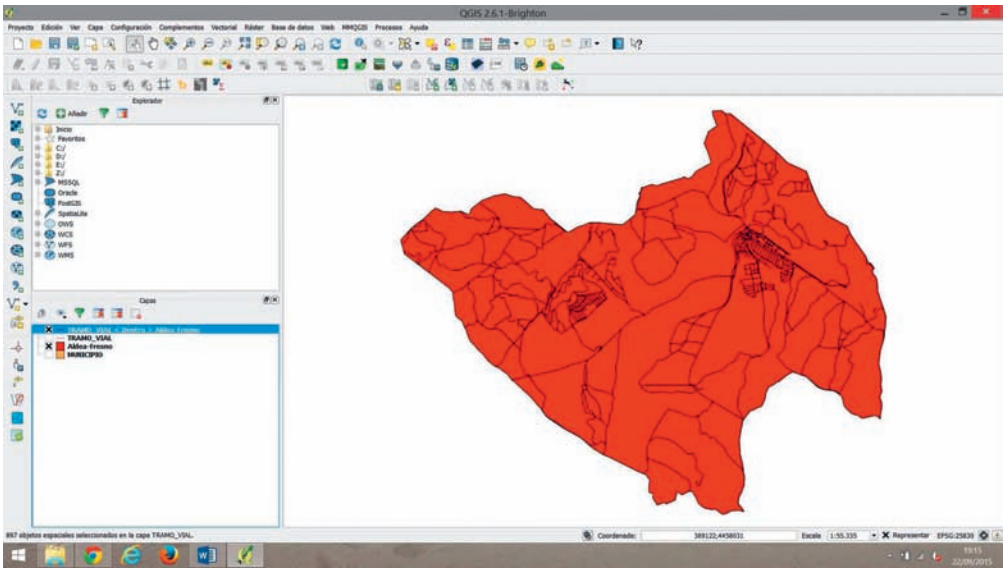

Figura 2.19

una a una las capas que queremos agrupar al nuevo icono con la figura anterior que se habrá creado.

Cuando tenemos una capa que está formada por categorías de datos variados más o menos amplia, como en nuestro caso el Viario que incluye sendas, caminos, carreteras convencionales, pistas y vías urbanas, podemos, desde la capa, visualizar de forma diferente cada uno de estos grupos.

Para ello clicaremos dos veces sobre la capa para abrir el menú `Propiedades de la capa` (o lo abriremos clicando con el botón derecho del ratón sobre el nombre de la capa y abriendo la opción `Propiedades`) y trabajaremos con la pestaña `Estilo`, en donde elegiremos la opción `Categorizado`. En el cuadro de diálogo que se abre marcaremos la columna que contiene nuestros datos

```
TIPO_V_DES
```

y elegiremos la opción `Clasificar`. Nos aparecerán los cinco tipos de vías, con su color que podremos cambiar, incrementar el tamaño de la línea, elegir otro formato, etc., utilizando las opciones `Cambiar` y `Rampa de Color`.

Cuando realizamos este tipo de clasificaciones suele aparecer una casilla que no tiene asociado ningún dato y que podemos eliminar marcándola y clicando sobre `Borrar`. De acuerdo a las opciones que hemos elegido, tendríamos un mapa como el de la Figura 2.24 en el que hemos cambiado el color de la capa Aldea-Fresno para mejorar la visualización.

Por último, guarde este ejemplo como un proyecto QGIS para posteriores estudios.

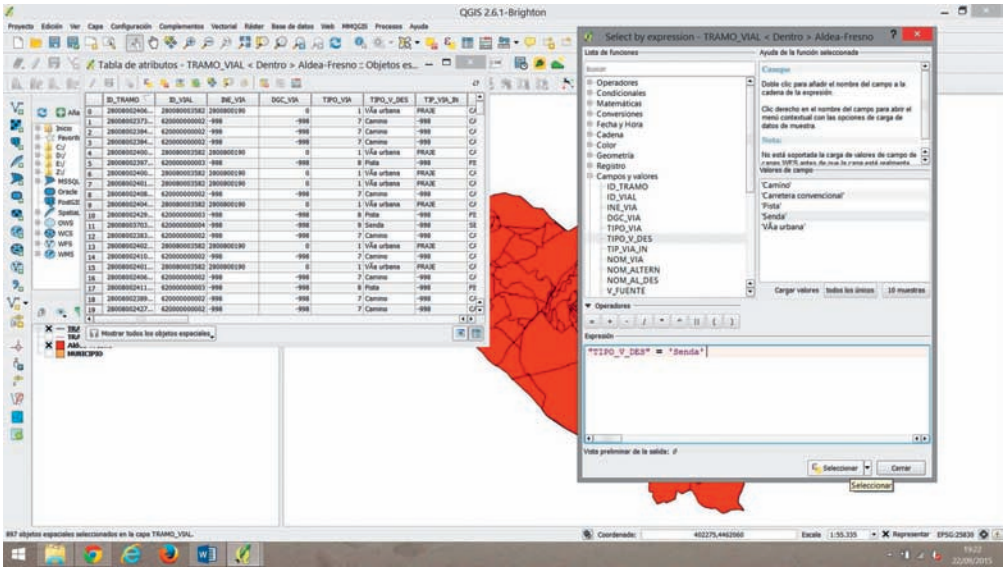

Figura 2.20

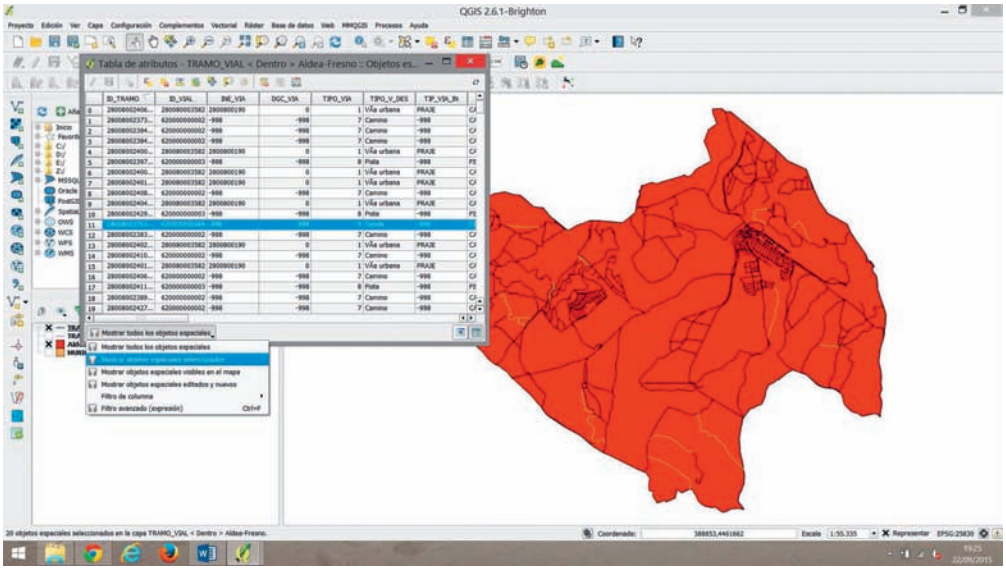

Figura 2.21

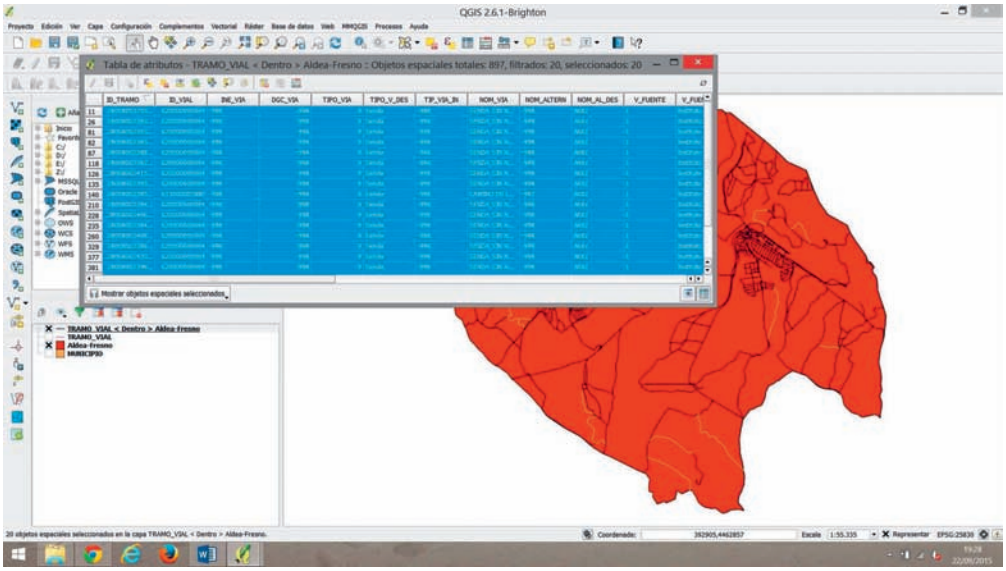

Figura 2.22

Figura 2.23

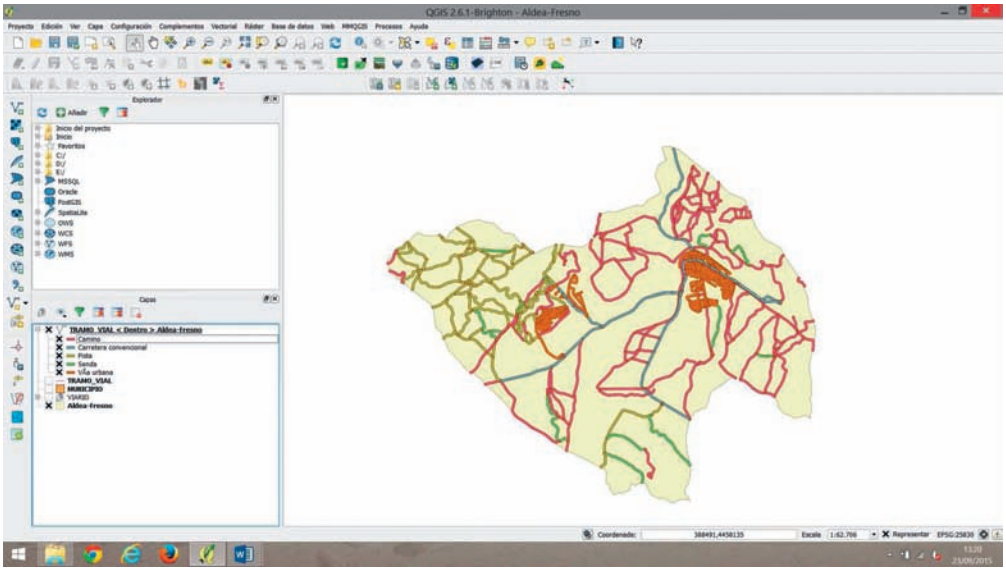

Figura 2.24

2.4. Análisis Espacial de Proximidad

Entre las muchas aplicaciones de QGIS está la de prevención: riesgos naturales, contaminación excesiva, incendios, etc. Son numerosas las ocasiones en las que determinar el número de viviendas que se verían afectadas en una catástrofe puede salvar numerosas vidas.

A continuación estudiaremos un ejemplo en el que vamos a delimitar el área de influencia de un fenómeno.

Ejemplo 2.3

En este ejemplo partimos de la existencia de una zona rural en donde se localizan diversas granjas y por donde circula un río. Vamos a determinar cuáles son las granjas que se verían afectadas ante un posible desbordamiento del río por fuertes lluvias.
Para ello vamos a cargar los ficheros shp que necesitamos para este estudio, pero esta vez con la particularidad de que añadiremos a nuestra área de trabajo todos los ficheros al mismo tiempo.

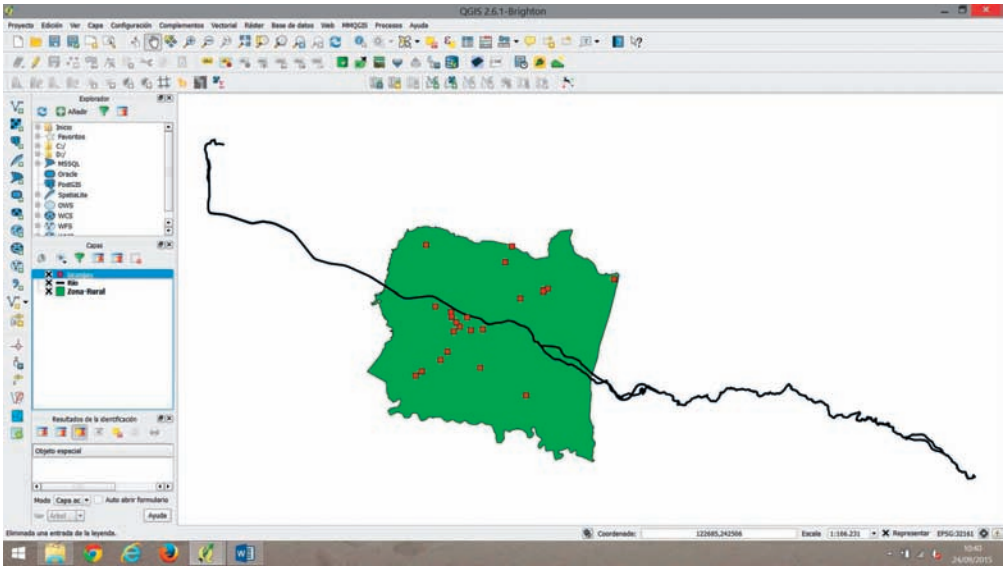

Figura 2.25

Como siempre, primero activaremos el botón de acceso rápido (Figura 1.3) con el que añadiremos capas vectoriales y, tras seleccionar la primera capa, mantendremos apretada la tecla CTRL del ordenador, seleccionando a continuación las restantes capas shp que necesitamos.
En este ejemplo utilizaremos las capas Zona-Rural, Granjas y Río, las cuales, como en anteriores ocasiones, están en la página web del libro mencionada en el Prólogo.
Con esta forma mencionada de inclusión simultánea de varias capas obtendríamos la Figura 2.25. Nótese el cambio del SRC para obtener la correcta visualización.

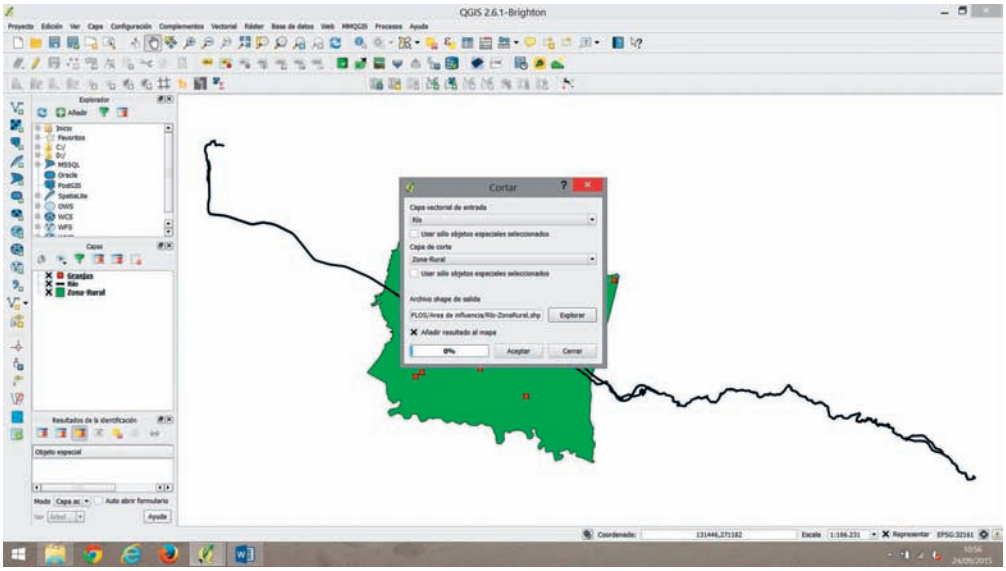

Figura 2.26

En general, QGIS puede pedirnos antes de incorporar las capas que seleccionemos el SRC. Si no lo hace y no las vemos correctamente, deberemos seleccionarlo desde la Barra de Estado, clicando sobre el icono del EPSG.
En este ejemplo hemos elegido el EPSG 32161. Para hacerlo, podemos utilizar el filtro, escribiendo en nuevo EPSG, para posteriormente, Aplicar y Aceptar.
Lo primero que vamos a hacer es seleccionar la parte de río que afecta a nuestra área rural. Para ello, abriendo el menú `Vectorial` y eligiendo la opción de `Herramientas de Geoproceso`, elegimos `Cortar`.
En el cuadro de diálogo que se abre a continuación, elegimos la capa `Río` como capa vectorial de entrada y la capa `Zona Rural` como capa de corte. Daremos un nombre a nuestro fichero de salida, en este ejemplo `Río-ZonaRural`, y Aceptamos a continuación. Figura 2.26.
Después QGIS procesa la orden dada, obteniendo la nueva capa.
Ahora, al ocultar la capa Río (desactivando su casilla) podemos ver la parte de río que pasa por la zona rural. Cambiamos su aspecto para una mejor visualización mediante la secuencia

```
Propiedades -> Estilo
```

A continuación, para verlo mejor, hacemos un zoom a la capa con el botón del mismo nombre, Figura 2.13, obteniendo la Figura 2.27.
Vamos a realizar ya el *Análisis Espacial de Proximidad*, estableciendo un área de influencia alrededor del río.
Esta área de influencia la determinaremos de acuerdo con nuestros criterios y previsiones, como por ejemplo si en esta zona ya se han producido desbordamientos podemos tener en cuenta los límites alcanzados.

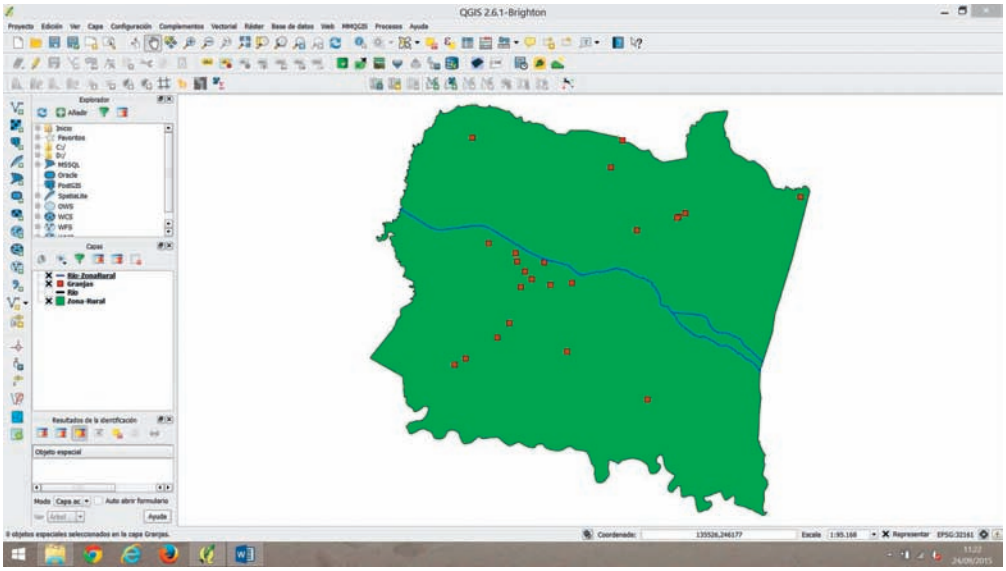

Figura 2.27

También podemos valorar el volumen de precipitaciones caídas en los últimos días o semanas, o quizás se puede tener en cuenta si estamos en época de deshielos, etc.

A modo de ejemplo, estableceremos un área de influencia o Buffer de 1000 metros. Para ello accedemos a esta opción a través del menú `Vectorial`, eligiendo la opción `Herramientas de Geoproceso` y, dentro de ellas, `Buffer`.

En el cuadro de diálogo que se abre, marcamos la capa donde aplicaremos el área de influencia, en este caso `Río-ZonaRural`.

Después, señalaremos la distancia: 1000 y daremos un nombre a nuestra selección que aparecerá como una nueva capa, la cual hemos denominado aquí, `Área-inf-1000`.

Aceptando se marcará alrededor del río la nueva área que hemos establecido, Figura 2.28.

Como queremos saber cuántas granjas se encuentran dentro del área de influencia así creada, vamos a cambiar la presentación de esta capa accediendo a las `Propiedades de la capa` (botón derecho del ratón una vez posicionados sobre el de la capa de la zona de capas o clicando dos veces sobre la capa) y eligiendo la pestaña `Estilo`, opción utilizada para cambiar el color y marcar una transparencia a la capa. De esta forma podremos ver las granjas situadas en esta zona de influencia, Figura 2.29.

Para facilitar la correcta localización de estas granjas, que se verían afectadas por la crecida del río, vamos a realizar una *consulta por localización*.

Para ello, desde el menú `Vectorial` accedemos a la opción de `Herramientas de Investigación` y, dentro de ella, elegimos la opción `Seleccionar por Localización`, Figura 2.30.

En el cuadro de diálogo que se abre a continuación, señalaremos como `Objetos Espaciales a Seleccionar` la capa `Granjas` que intersecta (utilizando la terminología QGIS y no el habitual interseca) con `objetos espaciales de`. Aquí elegiremos la capa que hemos creado anteriormente con el Buffer (`Área-Inf-1000`).

En nuestra zona de trabajo aparecerán en color amarillo las granjas que intersecan con el área de influencia del río. Recordamos que también aparece el número de granjas seleccionadas

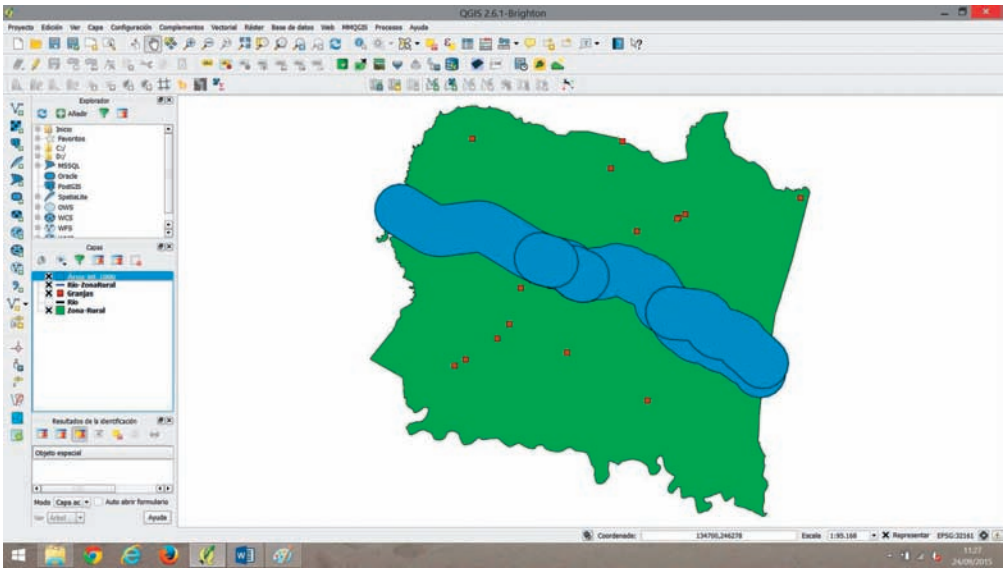

Figura 2.28

en la parte inferior izquierda de QGIS, Figura 2.31.
Si abrimos ahora la tabla de atributos de la capa `Granjas`, veremos marcadas las filas que se corresponden con las granjas afectadas.
Podremos activar la opción de QGIS que nos permite mostrar sólo la selección realizada y podremos guardar esta selección como una capa shp eligiendo `Guardar como` y marcando la opción de `Guardar la selección`, sobre la capa `Granja`.
Desactivando la capa `Granjas` y la capa creada con el área de influencia, podremos ver las granjas afectadas y sus datos figurarán en la tabla de atributos que QGIS crea de forma automática, Figura 2.32.
Por último, guardamos el trabajo como proyecto QGIS.

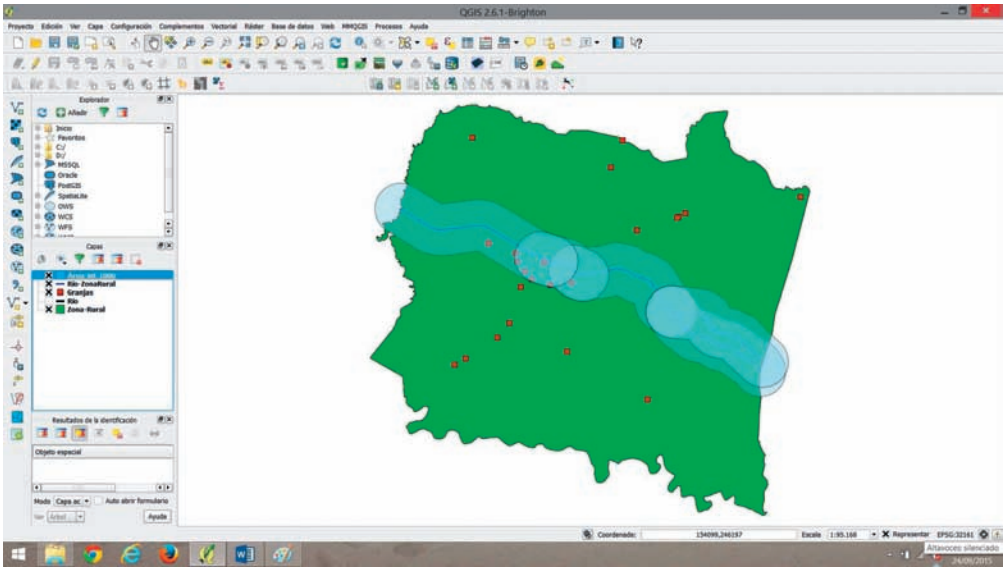

Figura 2.29

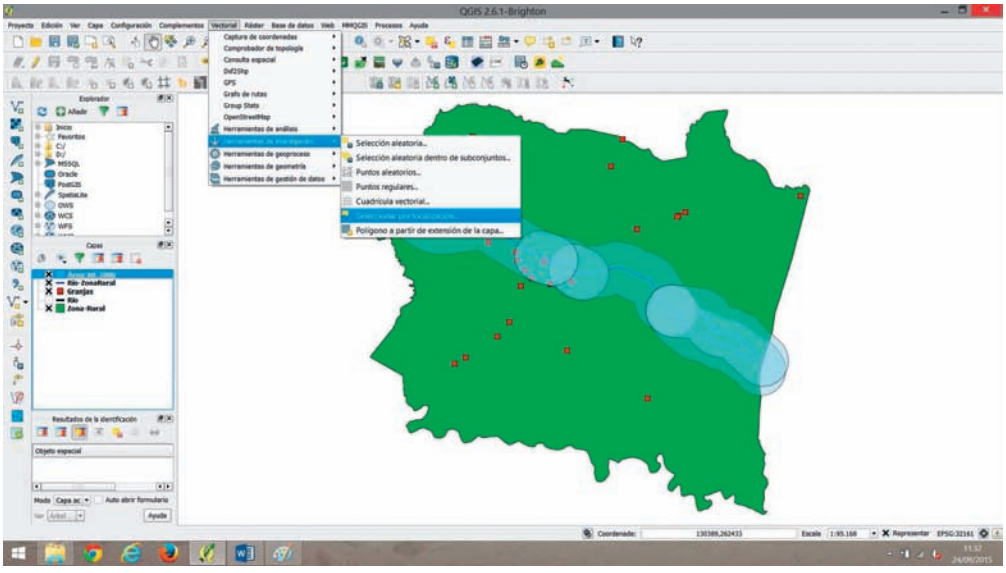

Figura 2.30

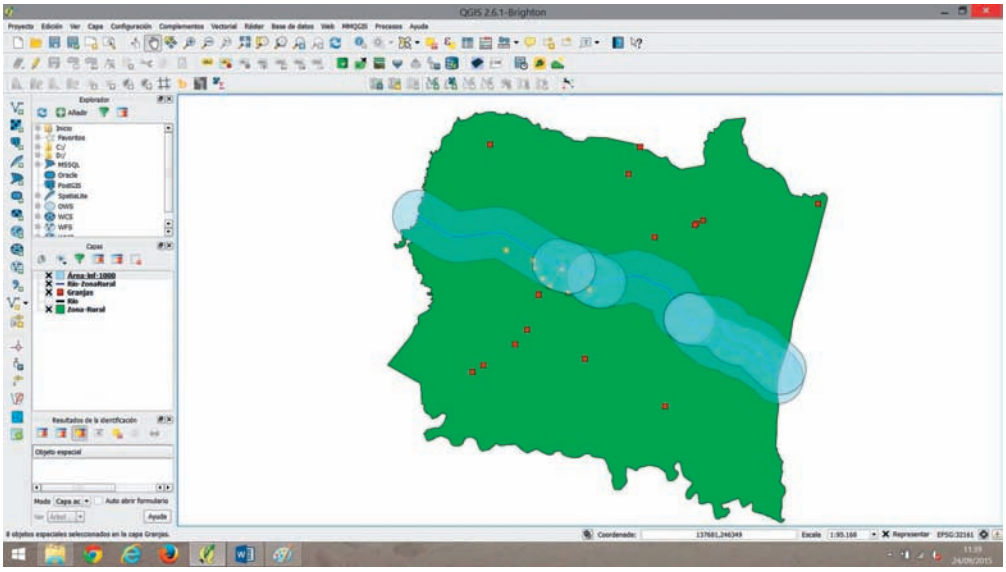

Figura 2.31

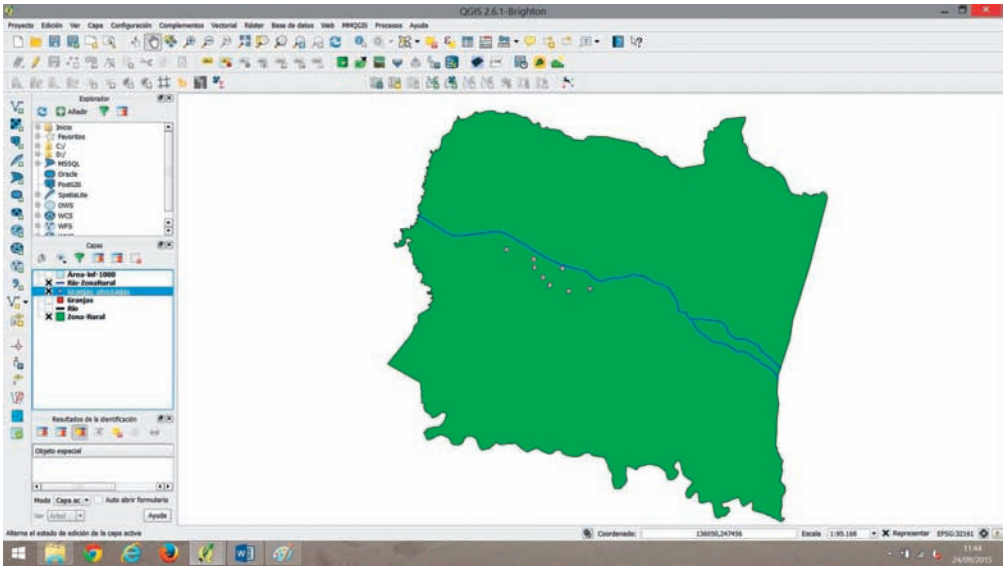

Figura 2.32

2.5. Presentación e Impresión

En la última sección de este capítulo mostraremos cómo preparar un Proyecto QGIS para exportarlo en pdf y poder incorporarlo a un documento.

Ejemplo 2.4

Utilizaremos en este ejemplo uno de los proyectos QGIS que hemos creado anteriormente y que habíamos guardado; en concreto el proyecto Aldea-Fresno.qgs.
Para ello, previamente deberemos abrirlo. Desde el menú `Proyecto` seleccionamos `Abrir` y recuperaremos, de esta forma, el mapa y sus capas tal y como lo habíamos dejado cuando lo salvamos, es decir, obtendremos la Figura 2.24.
Ahora, otra vez desde `Proyecto` accedemos a `Nuevo diseñador de impresión` y elegimos un nombre para éste.
Se abrirá entonces una zona de trabajo totalmente diferente, en donde tendremos que incorporar nuestro proyecto QGIS, Figura 2.33.

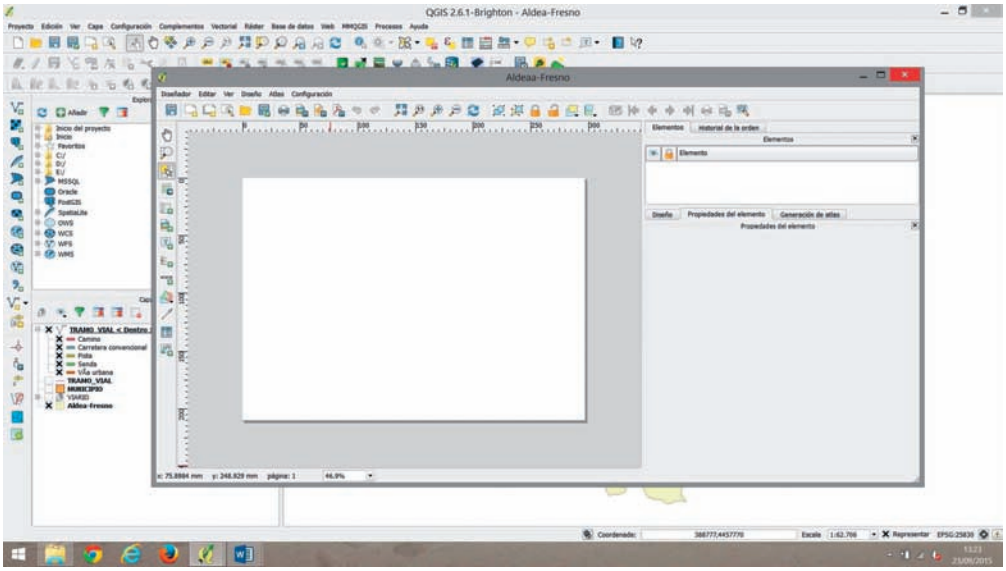

Figura 2.33

La forma de incorporar nuestro mapa es, eligiendo dentro del menú `Diseño` la opción `Añadir Mapa` o activando el botón de acceso rápido `Añadir mapa nuevo`, situado a la izquierda del área de trabajo, Figura 2.34.

Figura 2.34

Ahora, sobre el lienzo, haremos un rectángulo con el ratón y al soltarlo, el mapa quedará incorporado, Figura 2.35.

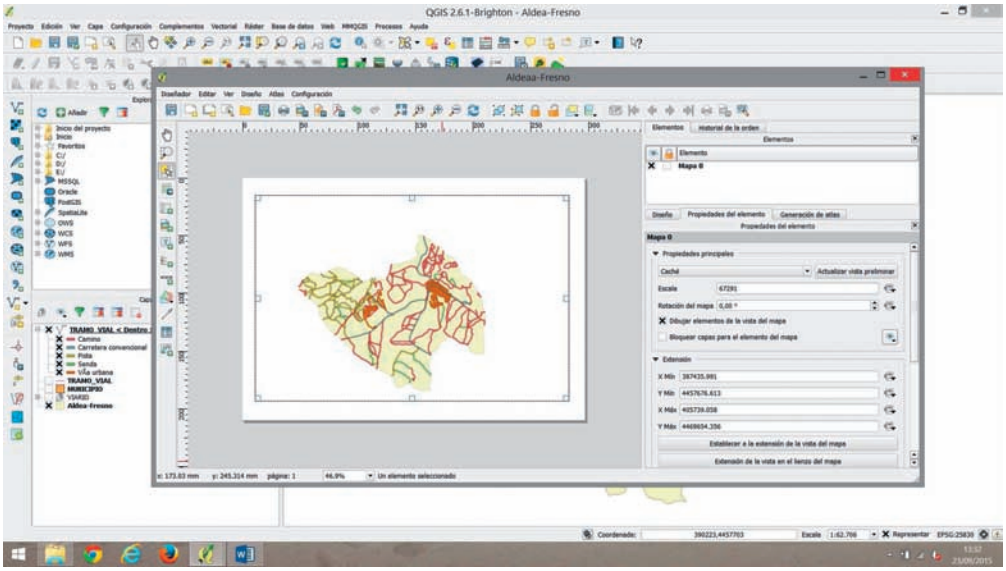

Figura 2.35

En este punto podemos elegir diversas opciones, tales como añadir un título, la leyenda, la escala, la orientación, etc. Las opciones aparecen dentro del menú `Diseño`. También podemos utilizar los botones de acceso rápido situados en el margen izquierdo.
Estas opciones abren cuadros de diálogo en la pantalla. Cada uno de ellos permite hacer cambios según nuestro criterio, es decir, elegir colores, posición, tipo y tamaño de letra, etc. Algunas de estas opciones se abren desplegándolas o clicando sobre ellas. En el caso de que no aparezcan, se puede acceder a ellas a través del menú `Ver`, eligiendo `Paneles` y `Propiedades del elemento`.
Arrastrando con el ratón, elegimos la zona en donde queremos situarlas.
Como ejemplo, vamos a poner un título seleccionando la opción `Añadir etiqueta nueva` a nuestro mapa. Para ello, con el ratón crearemos una caja en el lugar elegido para poner el título, Figura 2.36.
A continuación vamos a incorporar la escala y la leyenda. En el caso de la escala podremos elegir los segmentos que la forman.
En el caso de la leyenda podremos quitar el título que se ha creado de forma automática y cambiar algunos de los elementos que aparecen, accediendo a las opciones de `Elementos de la Leyenda`. Incluso podremos cambiar el nombre de algunas capas, como hacemos en este ejemplo con objeto de corregir la grafía. El resultado final es la Figura 2.37.
Ahora podremos guardar el resultado eligiendo la opción que más nos convenga dentro del menú `Diseñador`. En nuestro ejemplo hemos elegido el formato pdf, Figura 2.38.
Una vez guardado el resultado final, podremos imprimirlo o exportarlo fácilmente.

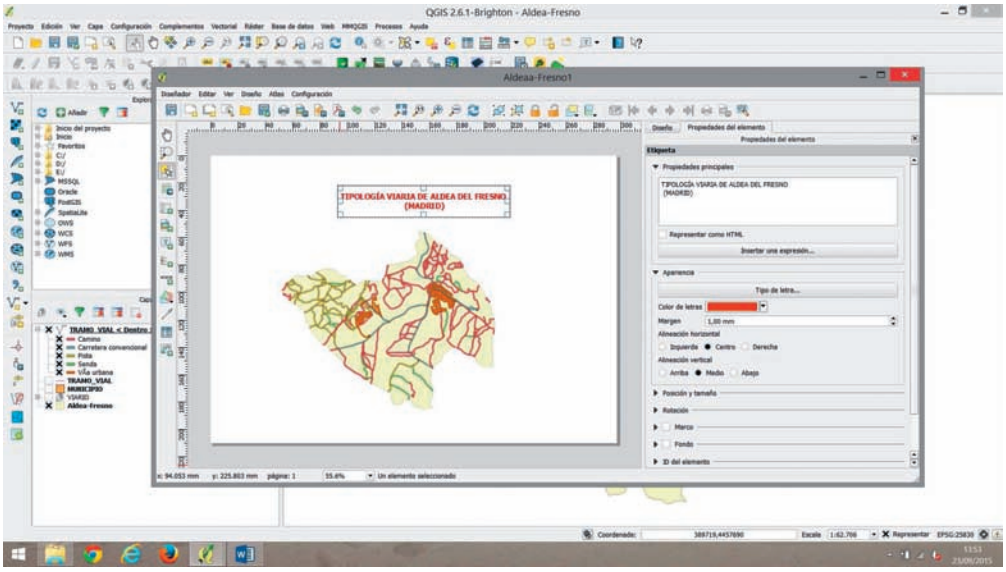

Figura 2.36

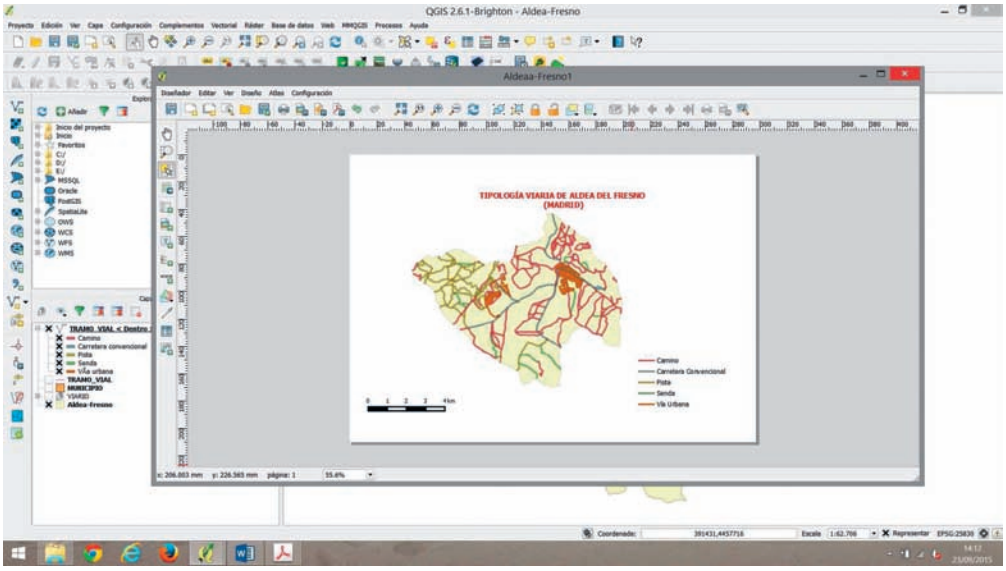

Figura 2.37

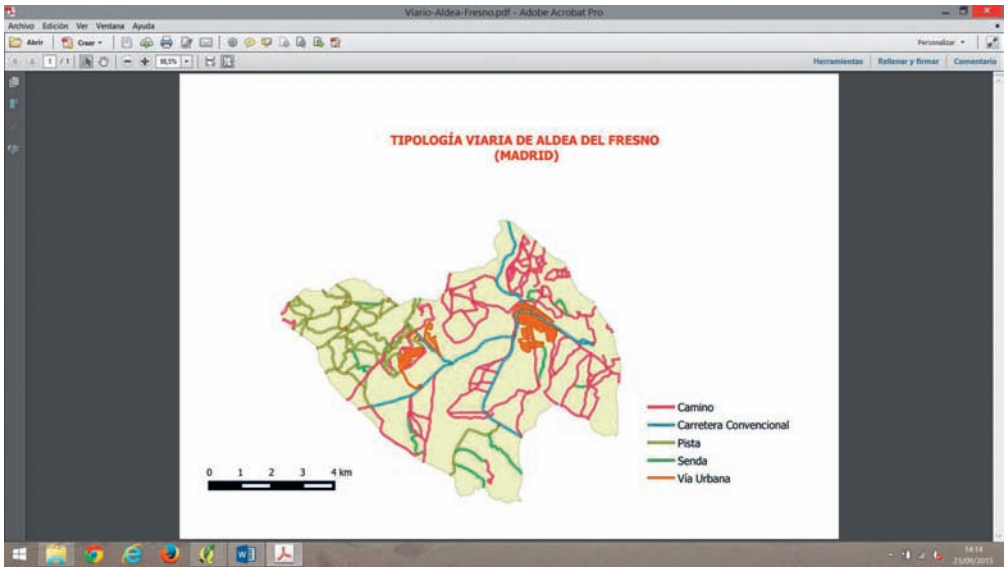
Viario-Aldea-Fresno.pdf - Adobe Acrobat Pro
TIPOLOGÍA VIARIA DE ALDEA DEL FRESNO
(MADRID)
Camino
Carretera Convencional
Pista
Senda
Vía Urbana
0 1 2 3 4 km

Figura 2.38

Capítulo 3

Interacción entre QGIS y R

3.1. Introducción

Una de las razones de haber elegido los autores de este libro a QGIS como Sistema de Información Geográfica para explicar y analizar datos espaciales, es su capacidad de interactuar con el paquete estadístico R a través de *Python*, es decir, QGIS tiene la capacidad de ejecutar programas (funciones) de R utilizando Python, cuyos resultados pueden ser incorporados como capa al mapa que habíamos establecido previamente con QGIS.

Por tanto, la mayor utilidad de esta interacción entre ambos programas gratuitos es la de utilizar primero QGIS en el Análisis Descriptivo de datos espaciales para ejecutar después programas de R, efectuando así el Análisis Inferencial de dichos datos, según iremos estudiando en capítulos posteriores. Entre las librerías que más utilizaremos en los capítulos posteriores están `maptools` y `sp`.

QGIS puede ejecutar comandos no sólo de R sino también de SAGA, GRASS, OTB (Orfeo Toolbox). Estos tres últimos programas se instalan cuando se instala QGIS mientras que R debe de ser instalado de forma independiente; véase García Pérez (2008c, 2010) para instalar R y estudiar cómo funciona.

Aunque es posible ejecutar programas en lenguaje Python directamente desde la consola Python, la cual se abre seleccionado desde la zona 1 (Figura 1.1) de botones de QGIS la secuencia

```
Complementos -> Consola de Python
```

nosotros no utilizaremos esta posibilidad ya que QGIS puede ser configurado para poder ejecutar programas de R directamente.

3.2. Configuración de QGIS

Para poder ejecutar R desde QGIS debemos primero configurar QGIS de la siguiente manera. En la zona 1 de botones de QGIS debemos seleccionar la siguiente secuencia

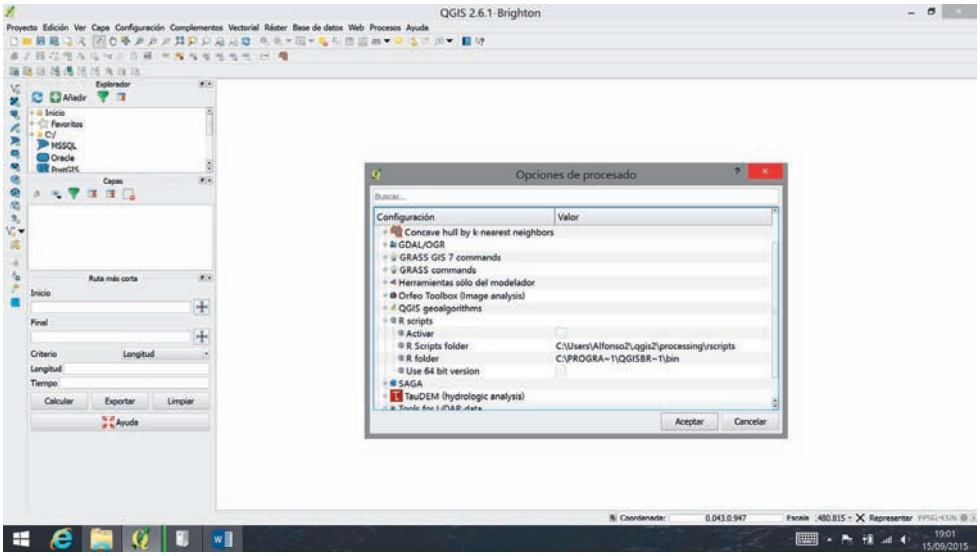

Figura 3.1

```
Procesos -> Opciones -> Proveedores -> R scripts
```

Aunque depende de los ordenadores, versiones de GIS, versiones de R, etc., es muy posible que al seguir este proceso le aparezca un cuadro como el de la Figura 3.1.

En este menú, deje como está la dirección `R Scripts folder`, rellene los dos cuadrados de `Activar` y el de `Use 64 bit version` y seleccione, con la opción del final de línea, el *path* en el que está su versión de R que seguramente será semejante a lo que aparece en la Figura 3.2, en este caso para la versión 3.1.3 de R. Pulse `Aceptar`, cierre QGIS y vuelva a abrirlo.

En principio ya tiene activado la posibilidad de ejecutar R desde QGIS. Para ello, de nuevo en la zona 1 de botones de QGIS debemos seleccionar los siguientes

```
Procesos->Caja de herramientas->R scripts->Herramientas->Create new R script
```

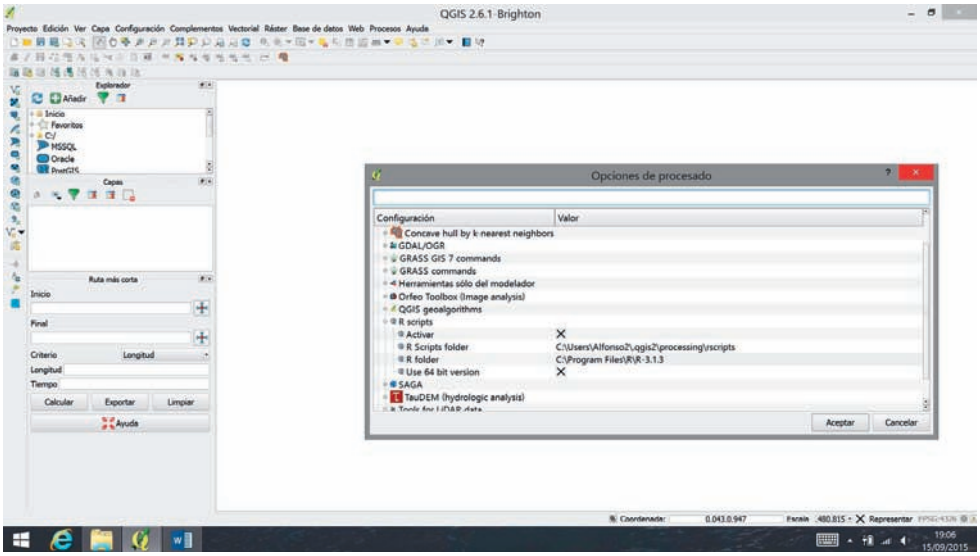

Figura 3.2

como se ve en la Figura 3.3.

Si no ve todo el desplegable de opciones de esta Figura 3.3, elija la opción `Advanced interface` de abajo a la derecha de la caja de herramientas.

En esta Figura 3.3 suponemos que hemos abierto el Ejemplo 1.1.

Clicando dos veces en este botón de `Create new R script` se abrirá un editor en el que debe teclear el script de R a ejecutar (Figura 3.4). Presione el botón de `Ejecutar algoritmo`. A continuación, dependiendo del script que esté ejecutando, le aparecerá una ventana de diálogo en la que deberá seleccionar algunos parámetros y el nuevo nombre de la capa shp que está generando con este script (o simplemente seleccione la Carpeta Temporal) y presione `Run`.

Cuando haya terminado de ejecutar el script la nueva capa aparecerá sobrepuesta a la que ya tenía (Figura 3.5) que en este caso consiste en generar (y añadir) una nueva capa con 10 puntos elegidos al azar que QGIS ha añadido en color verde, lo que implica que si el lector replica este ejemplo es posible que los 10 puntos así como los colores de estos dibujos sean algo distintos.

El script utilizado, el resultado y el de problemas si los hay, se generan en

```
c:\Usuarios\Alfonso\.qgis2\procesing
```

Es posible que QGIS, al utilizar R, trate de instalarle paquetes que no tenía o actualizar los que ya tenía y trate de grabarlos en una dirección del tipo

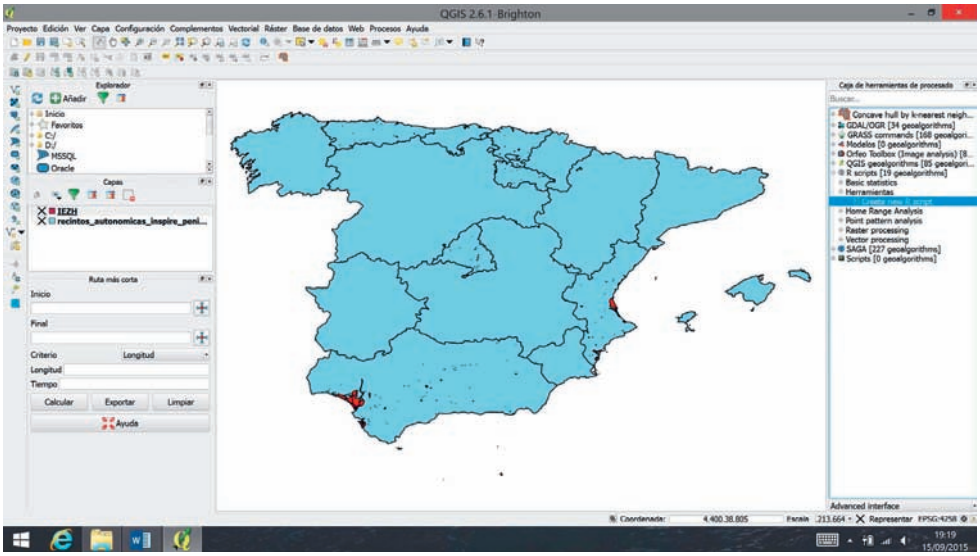

Figura 3.3

```
c:\Program Files\R\R-3.1.2\library
```

Si el programa falla deberá darle permisos al ordenador para que éste le deje escribir en un directorio tan delicado como ése. Mejor es que consulte a un informático pero si quiere arriesgarse eximiendo a los autores de este libro de toda responsabilidad, puede posicionarse sobre el directorio anterior e ir a `Propiedades` con el botón derecho del ratón, luego seleccionar `Seguridad` y en `Usuarios`, `Editar` para permitir un `Control Total`. Así, permitirá a QGIS instalar librerías en su R.

Una buena opción es comprobar qué paquetes va a necesitar para instalarlos antes en R y Actualizar R antes de utilizarlo a través QGIS utilizando el menú de R

```
Paquetes -> Actualizar paquetes
```

Las librerías de R `rgdal` y `maptools` siempre son abiertas por defecto por lo que debe de instalarlas en R. Aunque, como dijimos más arriba es posible que QGIS instale en su R las librerías que le faltan, si está seguro de necesitar alguna en concreto, lo mejor es que las instale antes de utilizar QGIS.

Como acabamos de decir, las librerías `rgdal` y `maptools` siempre son abiertas por defecto pero si va a utilizar alguna otra, lo mejor es que en su progrma

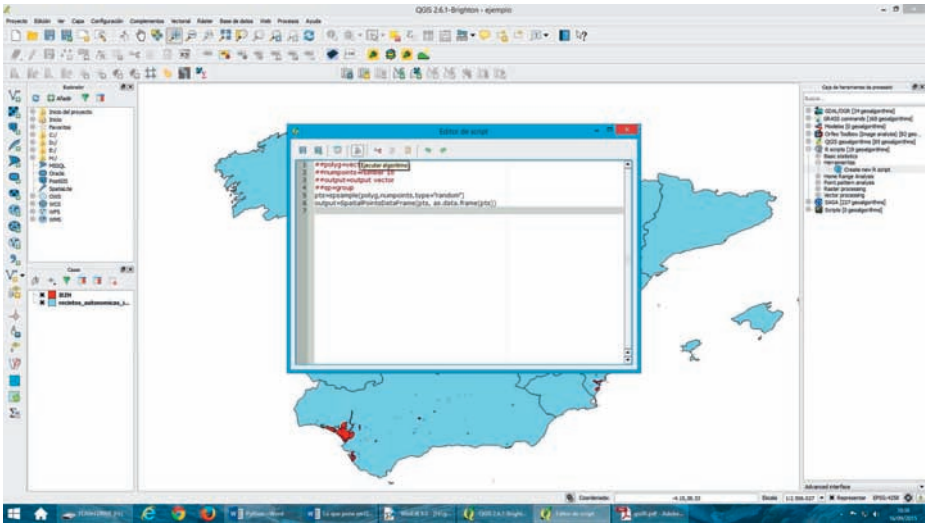

Figura 3.4

añada el comando `library( )` con el nombre de la librería que va a utilizar ya que lo mismo el programa QGIS falla al no poder abrir la librería.

3.3. Ejecución de programas de R a través de QGIS

A continuación veremos un ejemplo de ejecución de R a través de QGIS. Dado que las librerías de R sobre datos espaciales son las que habitualmente utilizaremos en esta interacción entre QGIS y R, y el estudio de estos paquetes se hace en capítulos posteriores, lo más razonable es que, aunque lea ahora esta sección, su mayor valor lo obtendrá cuando haya estudiado las librerías de R de los capítulos siguientes.

Ejemplo 3.1

Durante el desarrollo de la sección anterior hemos utilizado el Ejemplo 1.1 del capítulo 1 y al finalizar el ejemplo, sobre el mapa allí formado, hemos ejecutado como ejemplo un script un tanto peculiar (el que aparece en la Figura 3.4) pero que hemos elegido al ser este programa el que aparece en el manual de QGIS.

No obstante, este ejemplo nos va servir de guía en la explicación de cómo tenemos que ejecutar las funciones de R desde QGIS. Primero destaquemos que los programas Python necesitan comenzar con varias líneas que empiezan con el doble símbolo #. Con estas líneas definimos valores de argumentos de las funciones de R. Así, el siguiente programa Python

```
##polyg=vector                                                    (1)
##numpoints=number 10                                            (2)
##output=output vector                                           (3)
##sp=group
```

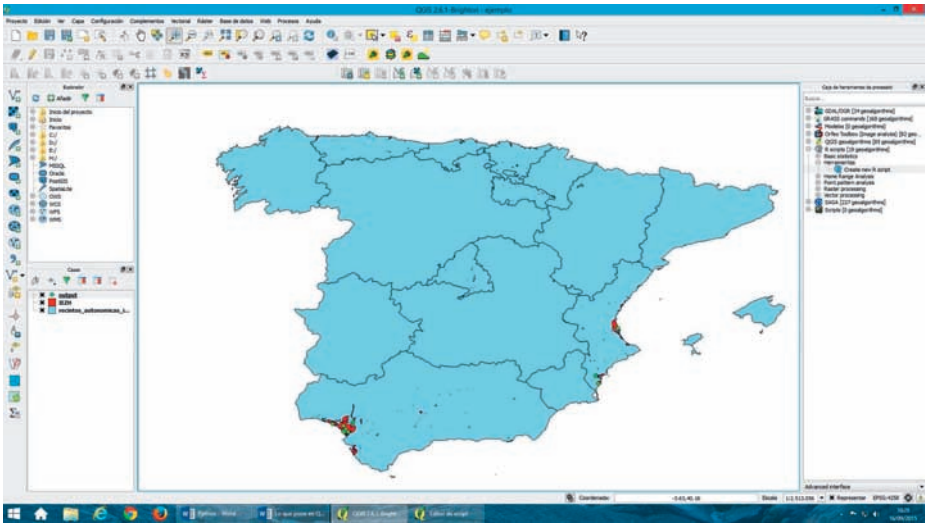

Figura 3.5

```
pts=spsample(polyg,numpoints,type="random")                    (4)
output=SpatialPointsDataFrame(pts, as.data.frame(pts))         (5)
```

con el que generamos 10 puntos al azar (indicando esta cantidad en `(2)`) que formarán una capa que se añade al mapa, le indicamos en `(1)` un *input* de tipo vector que denominamos `polyg`. Cuando ejecutemos el script anterior R abrirá la capa vectorial denominada `polyg` y la convertirá en una variable con ese nombre.
En realidad R utiliza para hacer esto las funciones `readOGR` si es una capa vectorial o `readGDAL` si es una capa raster, pero nosotros no tenemos que preocuparnos en abrir las correspondientes librerías ya que ambas funciones están en la librería `rgdal` que QGIS abre por defecto.
La salida, es decir, el *output* que generaremos le decimos en `(3)` que lo denominaremos `output` y que será una capa vectorial, lo que le indicamos con `output vector` en `(3)`.
El resultado final lo obtenemos en `(5)` ejecutando la función de R `SpatialPointsDataFrame`. Uno de sus argumentos, `pts`, lo hemos generado en `(4)` utilizando la función de R, `spsample`. Los argumentos de las funciones de R, `spsample` y `SpatialPointsDataFrame` serán estudiados en capítulos posteriores.
Si quiere ejecutar unos comandos desde R directamente sin generar una capa, debe decírselo al ordenador iniciando la línea de comandos que producen los resultados que desea ver (generalmente la última línea de comandos), con el símbolo (es decir, el *prompt*) >. La salida de todas las otras líneas no se mostrará.
Por ejemplo, si queremos calcular la media de los valores `Shape Leng` de la capa `EEZH` ejecutaríamos el programa

```
##layer=vector
##field=field layer
>mean(layer[[field]])
```

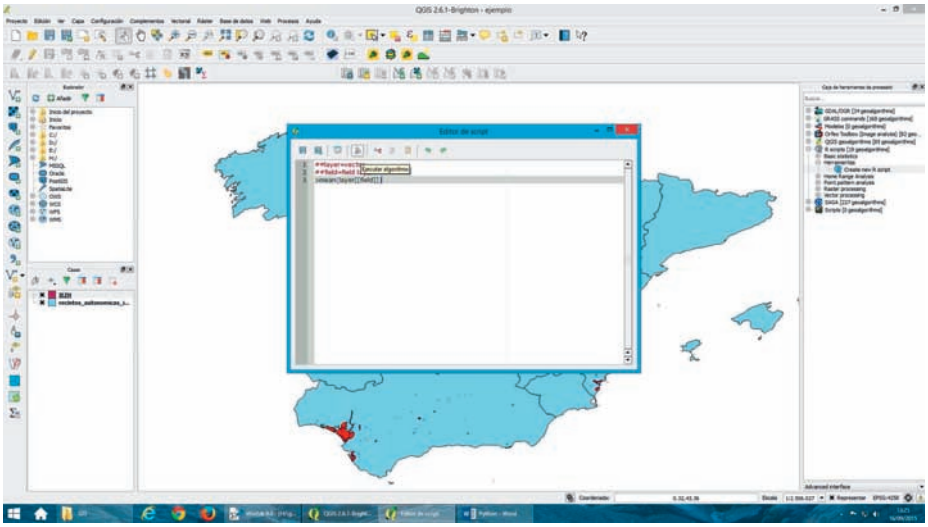

Figura 3.6

como en la Figura 3.6, efectuando la selección de la capa (Figura 3.7), obteniendo el valor medio, Figura 3.8.
Si su algoritmo crea algún tipo de gráficos, es decir, utilizando el método plot(), añada en las primeras líneas del principio la siguiente línea ##`showplots` para que QGIS redireccione los gráficos creados con R a un fichero (temporal o no). Tanto los resultados gráficos como de consola, se mostrarán cuando QGIS termine de ejecutarse.
Por ejemplo si queremos representar una nube de puntos de las dos variables cuantitativas `Shape Leng` y `Shape Area` que aparecen en la capa `EEZH` ejecutaríamos el programa

```
##layer=vector
##field1=field layer
##field2=field layer
##showplots
>plot(layer[[field1]],layer[[field2]])
```

obteniendo, después de completar el correspondiente cuadro de diálogo, la Figura 3.9, gráfico que se puede copiar con *Copy Image* y pegar.
Esta interacción entre R y QGIS permite utilizar los recursos de R (por ejemplo incorporando métodos robustos), para mejorar el análisis espacial (robusto).

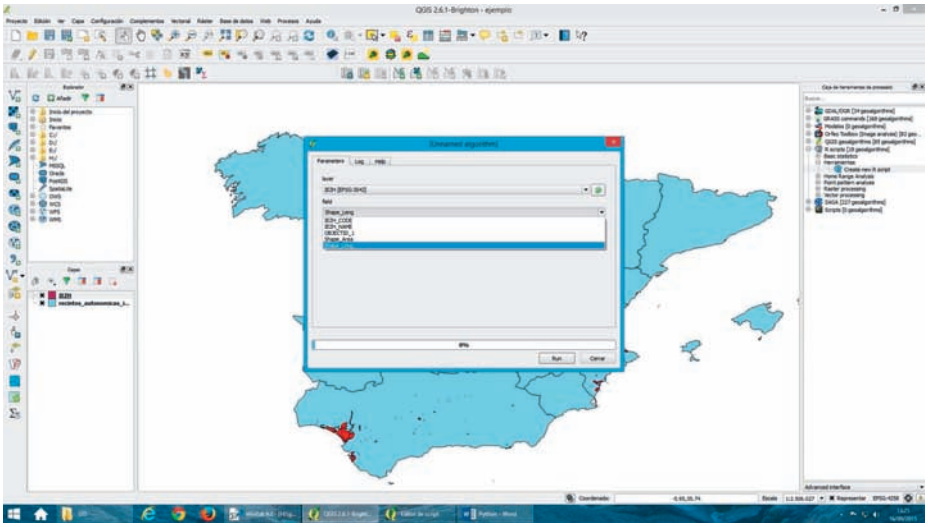

Figura 3.7

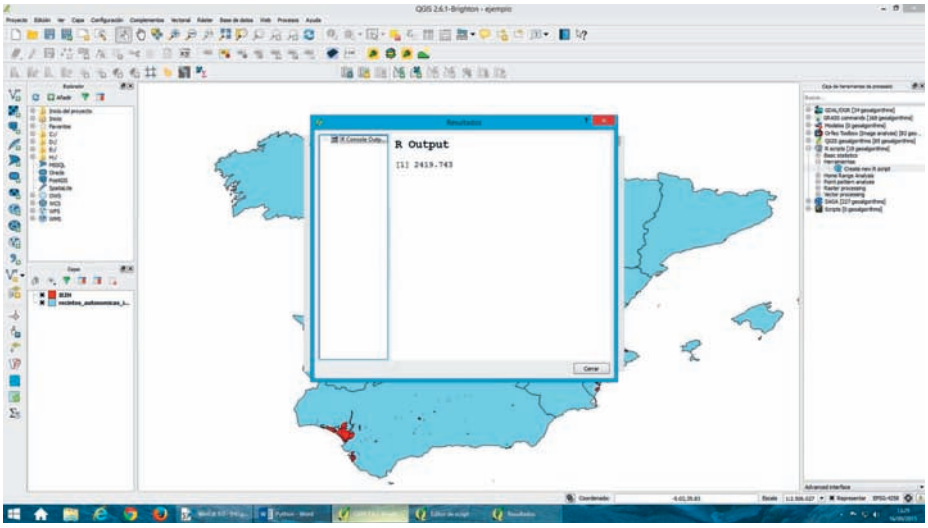

Figura 3.8

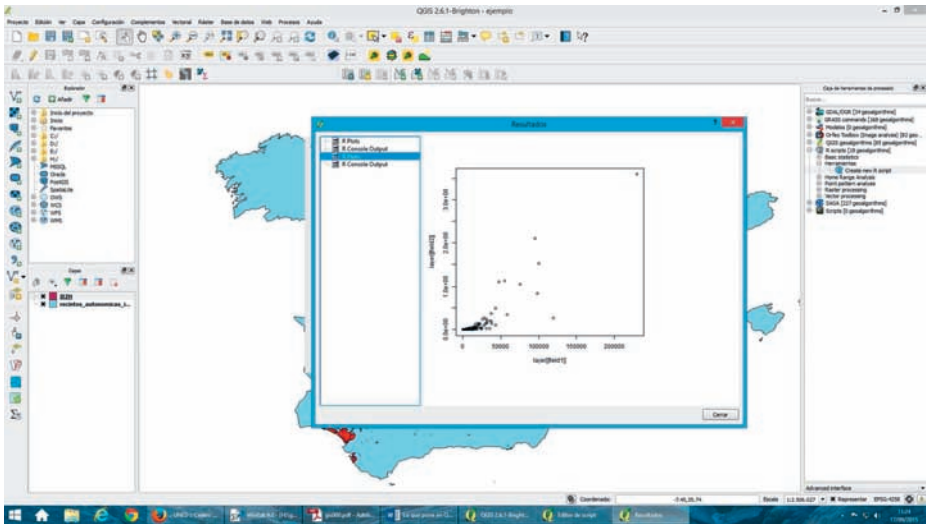

Figura 3.9

Capítulo 4

Análisis de Datos Espaciales de tipo discreto. Procesos Puntuales

4.1. Introducción

Como hemos dicho anteriormente, la localización geográfica de los lugares en donde se producen los acontecimientos observados es muy importante. No digamos ya el análisis de aspectos tan actuales como el posible cambio climático: los lugares en donde se toman las temperaturas son tan importantes como los valores de éstas.

Formalmente, los datos que se analizan con este tipo de técnicas consisten en un par de elementos, primero, unas *localizaciones* $\{\mathbf{s}_1, ..., \mathbf{s}_n\}$ sobre una superficie, generalmente La Tierra, es decir, habitualmente pares de puntos (x_i, y_i), como (Latitud , Longitud), o incluso (Menor distancia a la costa , Menor distancia a una línea imaginaria paralela a la costa) puesto que este sistema de referencia puede ser más adecuado en la modelización del fenómeno (Venables y Dichmont, 2004). Segundo, unos *datos* $\{Z(\mathbf{s}_1), ..., Z(\mathbf{s}_n)\}$ observados sobre esas localizaciones, como podrían ser precipitaciones de lluvia, o la polución aérea, etc. Supondremos que los datos son el resultado de la observación de una variable Z, unidimensional o multidimensional.

Según el tipo de localización $\mathbf{s}$ que se considere, los datos espaciales se denominan y analizan de forma diferente. Si las localizaciones $\{\mathbf{s}_1, ..., \mathbf{s}_n\}$ son fijas pero valores cualesquiera de la superficie considerada, es decir, matemáticamente valores cualesquiera de $\mathbb{R}^k$ (habitualmente $k = 2$ ó $k = 3$) se habla de *Geoestadística*. Habitualmente se conoce el valor de $Z(\mathbf{s})$ en algunos puntos y se desea conocer (interpolar) el valor de Z en otras localizaciones en donde no se conoce su valor. En este tipo de datos las localizaciones se mueven de

forma continua en $\mathbb{R}^k$.

Si las localizaciones son valores aislados y no son fijas sino que también son aleatorias (pero independientes de Z) se habla de *Procesos Puntuales* (*point patterns*) que será la situación que consideraremos en este capítulo. Ejemplos de este tipo de datos son, por ejemplo, los árboles de un bosque.

Por último, los datos pueden venir agregados, i.e., formando grupos de pequeños rectángulos, dando lugar a lo que se llama *Datos Reticulares* o *Regionales*, *lattice data*.

Es decir, que si el índice espacial $\mathbf{s}$ es discreto hablaremos de Procesos Puntuales; si el índice es continuo estaremos en el marco de la Geoestadística y, por último, si son rejillas o agregados, hablaremos de datos Regionales.

Es muy habitual en todos estos casos que la variable Z no se considere (o se considere como constante) y que, como mucho, se añada una *marca* a los datos, como por ejemplo que son de una u otra clase, o son de una población u otra, de manera que el interés en este tipo de datos se centra en las localizaciones con objeto de: *a)* Analizar la distribución que presentan los datos espaciales (por ejemplo, si están o no igualmente espaciados); *b)* Estudiar las marcas que presentan las localizaciones para, por ejemplo, comparar un par de especies, y *c)* Estudiar la densidad de las localizaciones, es decir, al número de individuos por unidad de área.

En ocasiones, las localizaciones fijas pueden ser valores aislados; más en concreto, formar un conjunto numerable como por ejemplo observaciones en puntos igualmente espaciados. Esta situación no la trataremos aquí porque es semejante a un Análisis de Series Temporales. No obstante, en todo el capítulo siempre consideraremos distinto el índice de localización de un posible índice temporal t; de hecho, si se quieren considerar datos espaciales a lo largo del tiempo, como por ejemplo el análisis de terremotos a lo largo del tiempo, hablaremos de modelos espacio-temporales.

4.2. Datos espaciales y su representación

Como dijimos más arriba, la matriz de datos espaciales habitual estará formada por columnas en donde aparecerán localizaciones, además de las columnas correspondientes a valores de variables medidas en esas localizaciones.

Siguiendo la clasificación que hicimos en el Capítulo 1, los datos espaciales puede ser de cuatro tipos distintos: Puntos, Líneas, Polígonos y Redes (*grids*).

Representación en Puntos y Polígonos

La representación en R de Puntos es la habitual de una nube de puntos, generalmente sin marco ni ejes coordenados como sucede en los mapas, utilizando la función `plot` con sus conocidos argumentos. Previamente debemos

extraer las localizaciones de la matriz de datos.

Ejemplo 1.2 (continuación)
Los datos de este ejemplo, que introdujimos anteriormente en el Ejemplo 1.2, los tenemos en el fichero de texto `meuse10.txt`. Si queremos utilizar R, los tenemos con el nombre `meuse` en la librería `sp` y corresponden a localizaciones y concentraciones de metales pesados.
La manera de incorporar estos datos a R sería en formato *data.frame*, pero como éstos ya están en la librería `sp`, basta con abrirlos con `(1)` y `(2)`.
Ahora extraemos las localizaciones con `(3)` ya que los nombres de éstas en la matriz de datos son, en este ejemplo, `x` e `y`. Luego ejecutamos `plot` con sus habituales opciones, obteniendo la Figura 4.1.

Figura 4.1 : Localizaciones de los datos del Ejemplo 1.2

```
> library(sp)                                    (1)
> data(meuse)                                    (2)
> coordinates(meuse)<-c("x","y")                 (3)
> plot(meuse,pch=16,col=2)
```

En este ejemplo, además de los datos de las localizaciones en donde se produjeron las observaciones, también se tienen las coordenadas del propio río Mosa en el fichero `meuse.riv` también en la librería `sp`. Su representación es trivial con la función `plot` obteniendo la Figura 4.2 al ejecutar

```
> data(meuse.riv)
> plot(meuse.riv,type="l",col=3,xlab=" ",ylab=" ")
```

Este tipo de representación (más semejante a un mapa) se denomina representación en Polígonos.

Representación en Líneas

Una vez que tenemos las localizaciones, podemos unirlas mediante segmentos con la función (de la librería `sp`) `SpatialLines`.

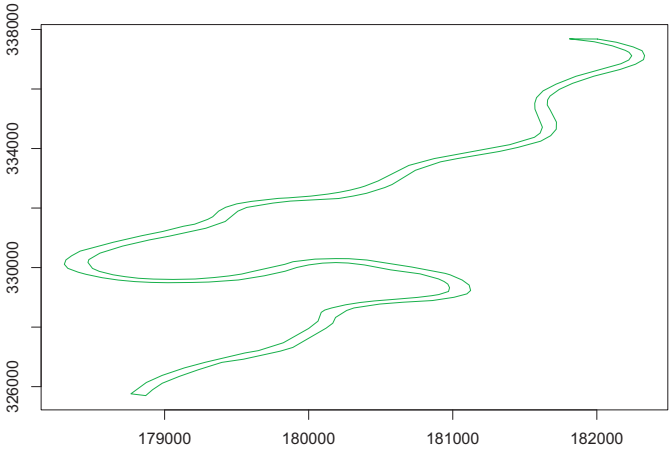

Figura 4.2 : Dibujo del río Mosa

Ejemplo 1.2 (continuación)
Ejecutando la función `SpatialLines` en las localizaciones de los datos antes extraídas, obtenemos la Figura 4.3.

```
> lineas<-SpatialLines(list(Lines(list(Line(coordinates(meuse))))))
> plot(lineas,col=4)
```

Representación en Redes (*Grids*)

Si queremos representar un área, basta con tener muchas localizaciones de ella, de manera que la representación de esa gran cantidad de puntos dará la sensación de una representación de toda la zona. Este tipo de gráfica se denomina Representación en Redes.

Ejemplo 1.2 (continuación)
Se tienen muchas coordenadas de la zona en donde se hicieron las observaciones. Éstas están en el fichero `meuse.grid`. Primero extraemos las coordenadas ejecutando `(1)`. Podríamos representar ya esta área con la función `plot` aplicada a estas coordenadas, pero la representación sería muy tosca. R tiene la posibilidad de representaciones mejores mediante la función `image`, pero esta función sólo admite *objetos*, es decir datos, del tipo `SpatialPixels`; por eso, en `(2)` obligamos a nuestras coordenadas antes extraídas con `(1)` a que se conviertan en objetos de este tipo con la función `as`. Ahora con `(3)` representamos estos objetos obteniendo la Figura 4.4.

```
> data(meuse.grid)
> coordinates(meuse.grid)<-c("x","y")                    (1)
> zona<-as(meuse.grid,"SpatialPixels")                   (2)
> image(zona,col="lightblue")                            (3)
```

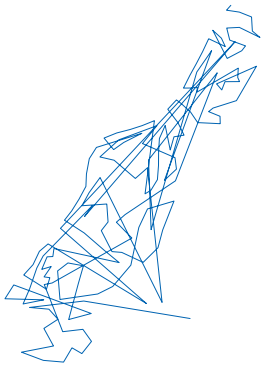

Figura 4.3 : Localizaciones de los datos, unidas por segmentos

Figura 4.4 : Zona de las localizaciones de los datos

Podemos representar en R a la vez el río, la zona en donde se produjeron las localizaciones y éstas comenzando los tres gráficos con la zona y utilizando el argumento `add=TRUE` en la función `plot`. Para representar juntos la zona y las localizaciones basta con ejecutar `(4)` y `(5)`. Si queremos que también aparezca el río debemos cambiar antes un poco el objeto a representar y ejecutar `(6)` antes de `(7)`.
Así, la Figura 4.5 se obtiene ejecutando las tres sentencias siguientes,

```
> image(zona,col="lightblue")                                                    (4)
> plot(meuse,pch=16,col=2,add=TRUE)                                              (5)
> rio<-SpatialPolygons(list(Polygons(list(Polygon(meuse.riv)),"meuse.riv")))     (6)
> plot(rio,col=3,add=TRUE)                                                       (7)
```

Vemos con este ejemplo la forma que tiene R de incorporar *capas*, distinta de la que tiene QGIS. Nuestra recomendación es que, si tiene los ficheros para poder utilizar QGIS, siempre será más sencillo e intuitivo utilizar QGIS en la representación de datos espaciales.

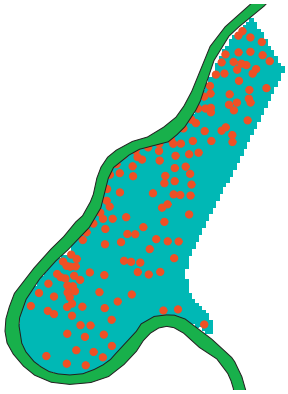

Figura 4.5 : Zona de las localizaciones junto con éstas y el río

4.3. Procesos Puntuales Espaciales

Los Modelos Espaciales Puntuales (*Spatial Point Patterns*) inicialmente fueron utilizaron por botánicos y ecólogos en la década de los 30 del siglo pasado para determinar, por ejemplo, la distribución espacial de los datos y sus causas en unas determinadas especies en estudio, o para comparar si podía admitirse que dos especies estaban igualmente distribuidas; no obstante, hoy en día son utilizados en muchos campos tales como la arqueología, la epidemiología, la astronomía o la criminología. Por ejemplo, es posible diseñar un modelo para comprender mejor la ubicación de los delitos, o bien es posible estudiar si los casos de una cierta enfermedad están distribuidos geográficamente según algún determinado modelo. En todos los casos, los datos observados serán del tipo pares (x_i, y_i) y, si se quieren comparar poblaciones, tendrán asociados una *marca* que identifique las poblaciones a comparar.

Como dijimos más arriba, los tres propósitos para los que se usan los Procesos Puntuales Espaciales son: Analizar la distribución que presentan los datos espaciales para concluir si están *distribuidos aleatoriamente*, es decir, al azar y sin ningún modelo que rija las localizaciones observadas; están *distribuidos regularmente*, es decir, están igualmente (uniformemente) espaciados; o, por último, si las localizaciones están *distribuidas formando clusters*.

El segundo objetivo es analizar la densidad espacial, es decir, el número de individuos por unidad de área.

El último objetivo de análisis es relativo a las marcas que presentan los datos para, por ejemplo, comparar dos especies.

4.3.1. Análisis de la distribución espacial

Los datos completos de los siguientes tres ejemplos están en la librería `spatstat`, respectivamente con los nombres `cells`, `japanesepines` y `redwood`.

Ejemplo 4.1

Los siguientes datos representan la localización de los centros de 42 células observadas bajo un microscopio óptico en una sesión histológica. El campo de visión del microscopio ha sido re-escalado al cuadrado unidad. Los datos fueron recogidos por F.H.C. Crick (uno de los dos descubridores de la estructura molecular del ADN) y Ripley (véase Ripley, 1977).

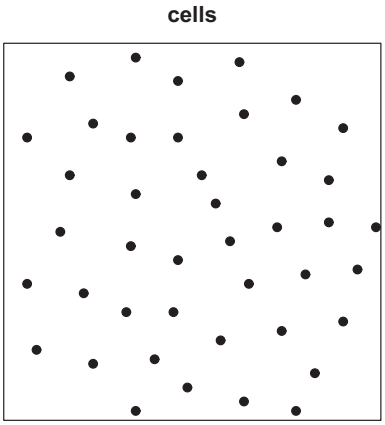

Figura 4.6 : Distribución espacial de las células

0′35	0′025
0′487	0′087
0′637	0′05
...	...
0′35	0′962
0′462	0′9
0′625	0′95

Su representación gráfica es la Figura 4.6 obtenida ejecutando `(1)`. Esta representación gráfica sugiere que los datos están distribuidos regularmente sobre el cuadrado unidad. Es decir, los datos siguen el modelo de estar igualmente espaciados.

```
> library(spatstat)
> data(cells)
> plot(cells,pch=16)                                                    (1)
```

Observe el lector que si, en lugar de importar los datos de localizaciones, quiere incorporarlos, debe hacerlo como matriz o como un par de vectores.

Ejemplo 4.2

Los siguientes datos son las localizaciones de pinos negros japoneses realizadas por Numata (1961) re-escalados a un cuadrado de lado unidad.

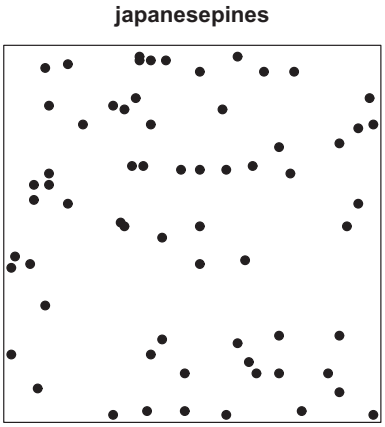

Figura 4.7 : Distribución espacial de los pinos japoneses

0′09	0′09
0′29	0′02
0′38	0′03
...	...
0′39	0′96
0′43	0′96
0′62	0′97

Su representación gráfica es la Figura 4.7 obtenida ejecutando `(1)`. De esta representación gráfica parece deducirse que éstos no se distribuyen ni regularmente ni siguiendo ningún modelo sobre el cuadrado unidad; parece que se distribuyen al azar sobre dicho cuadrado sin seguir un patrón claro.

Remarcamos que en este capítulo, *al azar*, no significará lo mismo que *uniformemente distribuidos* (situación que se presentaba en el ejemplo anterior). Lógicamente si se supone un modelo probabilístico que genera los datos, éstos se obtienen al azar según el modelo supuesto. Este modelo puede ser el modelo uniforme (CB-sección 4.5.2) u otro. En este capítulo entenderemos *distribuidos al azar* cuando no haya modelo aparente que genere los datos mientras que *uniformemente* significará que es un modelo uniforme el que los genera. Esto no es del todo cierto porque cuando más abajo analicemos si puede admitirse o no que los datos están generados al azar supondremos un proceso de Poisson homogéneo como generador de los datos, pero esto es sólo una suposición matemática para explicar situaciones como

la representada en la Figura 4.7 en donde no parece haber ni una regularidad (uniformidad) en la distribución de las localizaciones, como ocurría en el ejemplo anterior, ni una tendencia a agrupamientos (a clusters) en éstas, como ocurrirá en el ejemplo siguiente.

```
> data(japanesepines)
> plot(japanesepines,pch=16)                                    (1)
```

Ejemplo 4.3

Los siguientes datos representan las ubicaciones de 62 secuoyas de California en una región muestral cuadrada. Los datos originales era 195, procedentes de Strauss (1975), pero se suelen utilizar los 62 aquí tratados, estudiados anteriormente por Ripley (1977) en una subregión que se ha re-escalado a un cuadrado unidad.

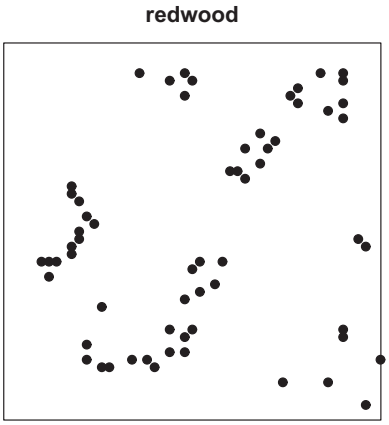

Figura 4.8 : Distribución espacial de las secuoyas californianas

$0'36$	$-0'08$
$0'44$	$-0'1$
$0'48$	$-0'08$
...	...
$0'74$	$-0'9$
$0'86$	$-0'9$
$0'96$	$-0'96$

Su representación gráfica es la Figura 4.8 obtenida ejecutando `(1)`. De esta representación gráfica se desprende que los datos aparecen distribuidos en clusters lo que indica un modelo subyacente, no regular como ocurría en el caso de las células.

```
> data(redwood)
```

```
> plot(redwood,pch=16)                                    (1)
```

Para poder abordar los tres objetivos anteriores es necesario introducir algunas herramientas matemáticas.

Proceso Puntual

Un *Proceso Estocástico* es una sucesión de observaciones de origen aleatorio. Cuando decimos *sucesión* nos estamos refiriendo a que las observaciones se obtienen siguiendo un orden que puede ser temporal (como ocurre con las Series Temporales) o espacial (el que aquí nos ocupa) o, incluso, espacio-temporal. Formalmente, un Proceso Estocástico es una sucesión de variables aleatorias X_t que evolucionan en función de otra variable (la que marca el orden) denominada índice t, que será el tiempo o el espacio. Cada una de las variables aleatorias del proceso tiene su propia distribución de probabilidad y, entre ellas, pueden estar correlacionadas o no.

Un *Proceso Puntual Espacial* es un proceso estocástico que genera localizaciones de algunos sucesos de interés dentro de una región concreta en estudio.

Denominaremos *Modelo Espacial Puntual* a las localizaciones de los sucesos generados por un proceso puntual en el área de estudio. Si las localizaciones tienen *Marcas* para distinguir varios grupos de datos, hablaremos de *Proceso* y *Modelo Espacial Puntual con Marcas*.

4.3.2. Aleatoriedad Espacial Completa (*CSR*)

Como dijimos más arriba, dentro del Análisis de la Distribución de las localizaciones, el primer objetivo es averiguar si éstas están distribuidas al azar en la región de estudio. En el ejemplo anterior de los pinos negros japoneses parecía intuirse una aleatoriedad en su distribución. Es decir, que no existe ningún patrón que regule su ubicación. Esta idea se denomina *Aleatoriedad Espacial Completa* (*Complete Spatial Randomness*) o, abreviadamente, CSR y se formaliza matemáticamente con un *Proceso de Poisson homogéneo de parámetro* λ, ya que este tipo de procesos se caracteriza por tres propiedades:

a) El número de localizaciones en una región A de área $|A|$ sigue una distribución de Poisson con media $\lambda|A|$, en donde λ es la intensidad del proceso, es decir, el número esperado de localizaciones por unidad de área.

b) Dadas n localizaciones en una región A, es decir, condicionalmente a que hay n localizaciones en A, éstas se distribuyen según una distribución uniforme sobre A.

c) En dos regiones disjuntas A y B, el número de localizaciones en A y el número de localizaciones en B son variables aleatorias independientes.

El analizar si los datos siguen o no Aleatoriedad Espacial Completa, es decir, un proceso de Poisson homogéneo, puede hacerse de dos formas: una, mediante *cuadrados* (*quadrats*), de manera que se anota el número de localizaciones acaecidas en cuadrados en los que se ha dividido la zona en estudio y se compara mediante un test χ^2 de bondad del ajuste con las que debería haber si fuera cierto el modelo Poisson, y dos, mediante *distancias*. Como es bien conocido, los tests basados en recuentos de observaciones son menos precisos que los basados en las propias observaciones. Por ello, para analizar la CSR consideraremos métodos basados en distancias.

Distancia a la localización más cercana

Hay varias posibilidades de distancia aunque suele utilizarse la distancia (Euclídea) entre una localización y la localización vecina más cercana (*nearest-neighboring*). Se puede demostrar que si las localizaciones están generadas por un proceso de Poisson homogéneo de parámetro λ, es decir, al azar, la distribución de estas distancias viene dada por la siguiente función de densidad

$$g(w) = 2\,\pi\,\lambda\,w\,e^{-\pi\,\lambda\,w^2} \qquad w > 0$$

o equivalentemente, por la siguiente función de distribución

$$G(w) = 1 - e^{-\pi\,\lambda\,w^2} \qquad w > 0.$$

Por tanto, las localizaciones observadas estarán generadas al azar, es decir, no siguiendo ningún patrón, si las diferencias entre su función de distribución empírica y este modelo teórico G no son significativas.

Si representamos por d_{ij} la distancia Euclídea entre dos localizaciones i y j, la distancia entre una localización i y la localización vecina más cercana será, lógicamente, $d_i = \text{mín}_j\{d_{ij},\ \text{con}\ j \neq i\}$, para $i = 1, ..., n$. Por tanto, fijada una distancia w, el estimador de $G(w)$ será la función de distribución empírica

$$\widehat{G}(w) = \frac{\text{número de } d_i \leq w}{n}$$

(Apuntamos el que las localizaciones i y j serán vectores, de dos o tres dimensiones habitualmente, por lo que deberían representarse por $\boldsymbol{i}$ y $\boldsymbol{j}$ aunque, por simplificar la notación, no la hemos incorporado.)

Hay varios tests de hipótesis para contrastar la aleatoriedad CSR (véase Cressie, 1993, pp. 604). En la Figura 4.9 aparecen los gráficos de los pares $(G(w), \widehat{G}(w))$ para los tres ejemplos anteriores así como las sentencias en R para obtenerlos, utilizando la librería `spatstat`.

```
> library(lattice)
> library(spatstat)
```

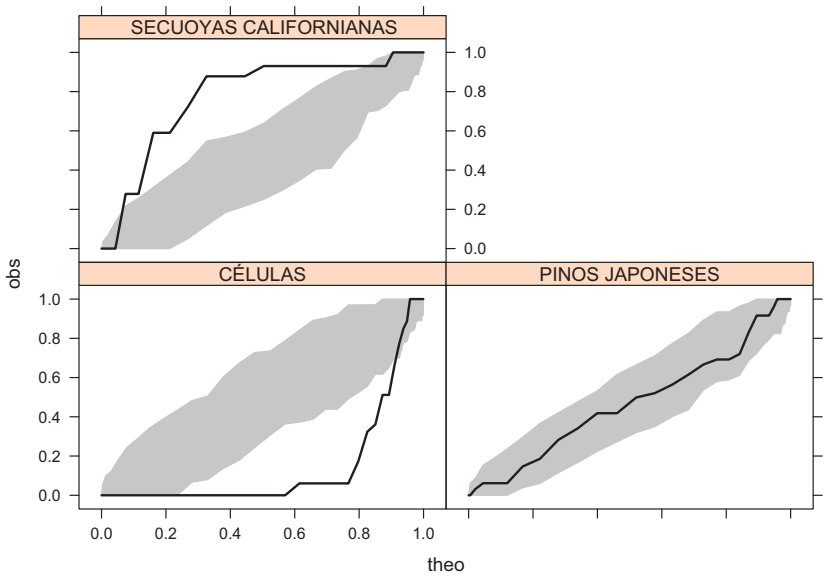

Figura 4.9 : Análisis visual de la CSR

```
> r<-seq(0,sqrt(2)/6,by=0.005)
> japo<-envelope(as(japanesepines,"ppp"),fun=Gest,r=r,nrank=2,nsim=99)
> rojo<-envelope(as(redwood,"ppp"),fun=Gest,r=r,nrank=2,nsim=99)
> celu<-envelope(as(cells,"ppp"),fun=Gest,r=r,nrank=2,nsim=99)
> resulta<-rbind(japo,rojo,celu)
> resulta<-cbind(resulta,DATASET=rep(c("PINOS JAPONESES","SECUOYAS CALIFORNIANAS",
+ "CÉLULAS"),each=length(r)))

> DATASET=rep(c("PINOS JAPONESES","SECUOYAS CALIFORNIANAS","CÉLULAS"),each=length(r))

> print(xyplot(obs~theo|DATASET,data=resulta,type="l",panel=function(x, y,subscripts)
  {lpolygon(c(x, rev(x)),c(resulta$lo[subscripts],rev(resulta$hi[subscripts])),
  border="gray",col = "gray", fill = T)
  llines(x, y, col="black", lwd=2)}
  ))
```

Como se deduce de estos tres gráficos, solamente en el caso de los pinos negros japoneses se tiene la Aleatoriedad Espacial Completa CSR.

Ejemplo 4.4

La utilización de los datos de los tres ejemplos anteriores es interesante pero habitualmente el lector estará más interesado en analizar si sus propios datos cumplen o no la hipótesis CSR. Para ello detallaremos este hipotético ejemplo en el que el autor del texto se ha inventado unos pares de datos en `(1)` y `(2)` que serían, por ejemplo, los pares reales (latitud, longitud),

para formar la matriz de datos en (3), que corresponderá a la matriz de datos reales del lector. El análisis de la CSR se hace con datos re-escalados en el cuadrado unidad; es decir, debemos cambiar la escala de éstos para que todos ellos tomen valores en [0,1]. Esto se consigue restando a cada dato x el menor de los valores, mín(x) y dividiendo el resultado de esta diferencia por la diferencia entre el máximo y el mínimo de los valores, es decir, haciendo el cálculo

$$\frac{x - \text{mín}(x)}{\text{máx}(x) - \text{mín}(x)}.$$

El re-escalamiento se hace en tres pasos a partir de (4), denominando de la misma manera la matriz resultante. Por supuesto, si el lector debe repetir este proceso varias veces, le resultará más sencillo crear una función que haga todos los pasos. Finalmente se pueden representar los datos.

```
> library(lattice)
> library(spatstat)
> x1<-c(21,22,21.2,22.4,22.8,21.7,22.3,21.5,22.4,21.9,21.2,22.2,21.4,      (1)
  22.6,23.0,21.9,22.5,21.7,22.6,22.1,21.5,22.5,21.7,22.9,23.3,22.2,
  22.8,22.0,22.9,22.4)
> x2<-c(34.1,35,33.9,34.9,35.1,33.7,33.1,33.4,33.5,33.7,33.7,34.6,33.5,    (2)
  34.5,34.7,33.3,32.7,33.0,33.1,33.3,34.8,35.7,34.6,35.6,35.8,34.4,33.8,
  34.1,34.2,34.4)
> prueba<-matrix(c(x1,x2),ncol=2)                                          (3)
> b1<-(prueba[,1]-min(prueba[,1]))/(max(prueba[,1])-min(prueba[,1]))       (4)
> b2<-(prueba[,2]-min(prueba[,2]))/(max(prueba[,2])-min(prueba[,2]))
> prueba<-matrix(c(b1,b2),ncol=2)
> plot(prueba)
```

La aleatoriedad CSR se verificará en nuestros datos si las diferencias (en este caso gráficas) entre el modelo teórico $G(w)$ y la distribución empírica $\widehat{G}(w)$ no son grandes, para un conjunto de distancias w razonable, conjunto de distancias que fijamos en (5), iguales en este caso a 50 distancias entre 0 y 0′25.

```
> w<-seq(0,0.25,len=50)                                                    (5)
```

Como el modelo teórico es muy difícil de manejar, lo que hacemos es simular, con la función `envelope` de la librería `spatstat` muchas realizaciones suyas (las que queramos con el argumento `nsim` de `envelope`) del proceso puntual, en este caso G, para lo que utilizamos el argumento `fun=Gest` de `envelope`. Esta función `envelope` sólo admite datos del tipo `ppp`, por eso transformamos antes los datos `japanesepines` con la función `as`. Los datos en forma de matriz no son de este tipo. Primero deberemos transformarlos en datos del tipo `SpatialPoints` con esta función, que pertenece a la librería `sp`, ejecutando (6) y, después en datos `ppp`, con la función `as` pero abierta la librería `maptools` ejecutando (7),

```
> library(sp)
> prueba2<-SpatialPoints(prueba)                                           (6)
> library(maptools)
> prueba3<-as(prueba2,"ppp")                                               (7)
```

Las distancias w a considerar se incluyen en la función `envelope` con el argumento r.
De esta forma, con `envelope` obtendremos unos "entornos de confianza" entre los que debería de estar las distribución empírica $\widehat{G}(w)$. En estos entornos se puede fijar el coeficiente de

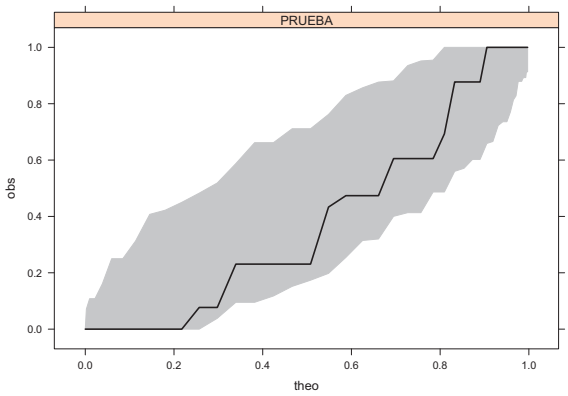

Figura 4.10 : Análisis de datos de prueba

confianza mediante el argumento `nrank` de la función `envelope`, diciéndole cuántos de los valores simulados eliminar a cada lado del entorno. Si fijamos `nrank=2` (quitamos 2 a cada lado) sobre 100 simulaciones `nsim=99`, tendremos entornos de confianza del 96 %.
Por tanto, ejecutando `(8)`, tendremos el entorno de confianza.

```
> entorno<-envelope(prueba3,fun=Gest,r=w,nrank=2,nsim=99)                    (8)
```

Ahora sólo tenemos que representarlo y sobre-impresionar en el dibujo del entorno así creado (y que, adelantamos a los lectores que tratan de replicar este ejemplo podrá cambiar de simulación en simulación) nuestra distribución empírica $\widehat{G}(w)$. Esta representación gráfica se puede hacer de varias maneras aunque utilizaremos la combinación anterior (*script* en terminología R) ejecutando

```
> entorno<-cbind(entorno,DATASET=rep(c("PRUEBA"),each=length(w)))

> DATASET=rep(c("PRUEBA"),each=length(w))

> print(xyplot(obs~theo|DATASET , data=entorno, type="l",
panel=function(x, y, subscripts)
{
lpolygon(c(x, rev(x)),
   c(entorno$lo[subscripts], rev(entorno$hi[subscripts])),
   border="gray", col="gray",fill=T
)
llines(x, y, col="black", lwd=2)
}
  ))
```

que podemos unir en una nueva función con un único argumento en el que incluyamos `entorno`, obteniendo la Figura 4.10. En ella se observa que los datos fueron generados al azar.

4.3.3. Ajuste de Modelos Espaciales Puntuales

Si hemos rechazado la Aleatoriedad Espacial Completa de una región A, es decir, que las localizaciones observadas en A no se producen al azar, el siguiente paso lógico es ajustar un modelo a las localizaciones observadas. Si hemos rechazado la CSR vimos que había dos posibilidades: Una distribución regular uniforme, como ocurría en el ejemplo de las células, que se suele modelizar mediante *Procesos de Inhibición Simple*, que no serán tratados aquí. La segunda posibilidad es que se produjeran clusters, es decir, agrupamientos de localizaciones. Esta segunda posibilidad se modeliza mediante un *Proceso de Poisson no homogéneo* (recordemos que la CSR lo era mediante un Proceso de Poisson homogéneo) o mediante un *Proceso de Cox* o mediante un *Proceso de Poisson con clusters*. Nosotros sólo analizaremos el *Proceso de Poisson no homogéneo de parámetro* $\lambda(\mathbf{s})$ que se diferencia del homogéneo estudiado más arriba porque la intensidad del proceso $\lambda(\mathbf{s})$ ya no es constante sino que depende de la localización $\mathbf{s} \in A$.

Estimación de la Intensidad

En el caso de un proceso de Poisson homogéneo la intensidad es constante en cada área considerada A, por lo que, si en ese área hay n localizaciones, un estimador suyo será $\widehat{\lambda} = n/|A|$ en donde $|A|$ representa el área de la región A.

En el caso de procesos de Poisson no homogéneos hay varias posibilidades que se resumen en dos: utilizar *Métodos Paramétricos*, consistentes en proponer una función cuyos parámetros son estimados por el método de máxima verosimilitud. Esta vía permite incluir p covariables existentes Z_j, $j = 1, ..., p$ y utilizar, por ejemplo, un modelo log-lineal de la forma

$$\log \lambda(\mathbf{s}) = \sum_{j=1}^{p} \beta_j \, Z_j(\mathbf{s})$$

siendo $Z_j(\mathbf{s})$ $j = 1, ..., p$ los valores que toman las covariables en la localización $\mathbf{s}$.

La segunda posibilidad en la estimación de la intensidad de un proceso de Poisson no homogéneo son los *Métodos no Paramétricos*, basados en el *Estimador Núcleo Suavizado* (*kernel smoothing*) dado por

$$\widehat{\lambda}(\mathbf{s}) = \frac{1}{q(||\mathbf{s}||)\, h^2} \sum_{i=1}^{n} K\left(\frac{||\mathbf{s} - \mathbf{s}_i||}{h}\right) \qquad [4.1]$$

supuesto que se han observado n localizaciones $\mathbf{s}_1$, ..., $\mathbf{s}_n$, siendo K la función núcleo considerada (habitualmente bivariante), $q(\mathbf{s})$ una corrección frontera para compensar los valores que se pierden cuando $\mathbf{s}$ está cerca de la frontera

de la región A, y siendo h una medida del nivel de suavizado (*smoothing*), también denominada ancho de banda (*bandwidth*), que se quiere considerar: valores pequeños de h conducirán a estimadores poco suaves y valores grandes a estimadores muy suaves.

La función núcleo habitualmente considerada es la denominada función cuártica (*quartic*), también denominada biponderada (*biweight*) definida, para localizaciones $\mathbf{s} \in (-1, 1)$, como

$$K(\mathbf{s}) = \frac{3}{\pi}(1 - ||\mathbf{s}||)^2$$

y como 0 para localizaciones $\mathbf{s} \notin (-1, 1)$.

Apuntamos el que $||\mathbf{s}||$ denota la *norma* (o longitud) del vector $\mathbf{s}$ que, si es bidimensional con coordenadas (s_1, s_2), es igual a $||\mathbf{s}|| = \sqrt{s_1^2 + s_2^2}$. (Análogamente con la norma de la diferencia de vectores que aparece en la fórmula anterior.)

La especificación del suavizado h es un serio problema puesto que diferentes especificaciones conducen a muy diferentes estimaciones de la intensidad.

Ejemplo 4.3 (continuación)

Vamos a estimar la intensidad del proceso de Poisson no homogéneo mediante técnicas no paramétricas utilizando el estimador núcleo suavizado dado por [4.1], ejecutado por la función `kernel2d` de la librería `splancs`. Los argumentos de esta función son, básicamente tres: el primero, los datos en formato `ppp`; el segundo, un polígono en el que queramos obtenga las estimaciones (el cuadrado de lado unidad en nuestro caso), y el tercero, el nivel de suavizado h considerado más arriba. La corrección frontera se ignora.

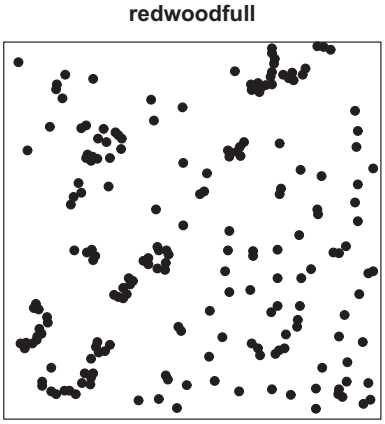

Figura 4.11 : Distribución espacial de las 195 secuoyas californianas

Todo este proceso comienza con la determinación del nivel de suavizado h, para lo que se suele utilizar el criterio propuesto por Diggle (1985) y Berman y Diggle (1989) consistente en elegir como nivel de suavizado el primer valor en el que se consigue minimizar el error cuadrático medio del estimador kernel que tratamos de construir. En este proceso se utiliza la función `mse2d` de la librería `splancs`. Los argumentos de esta función son, básicamente cuatro: el primero, los datos en formato `ppp`; el segundo un polígono en el que queramos obtener las estimaciones; el tercero, el número de iteraciones que queremos considerar y, el cuarto, el valor máximo admitido para h.
Los datos `redwood` utilizados antes en este ejemplo son una parte de los 195 datos `redwoodfull` que utilizaremos. Su representación gráfica, obtenida ejecutando

```
> library(spatstat)
> data(redwoodfull)
> plot(redwoodfull,pch=16)
```

es la Figura 4.11, en donde se aprecia la distribución de la intensidad.

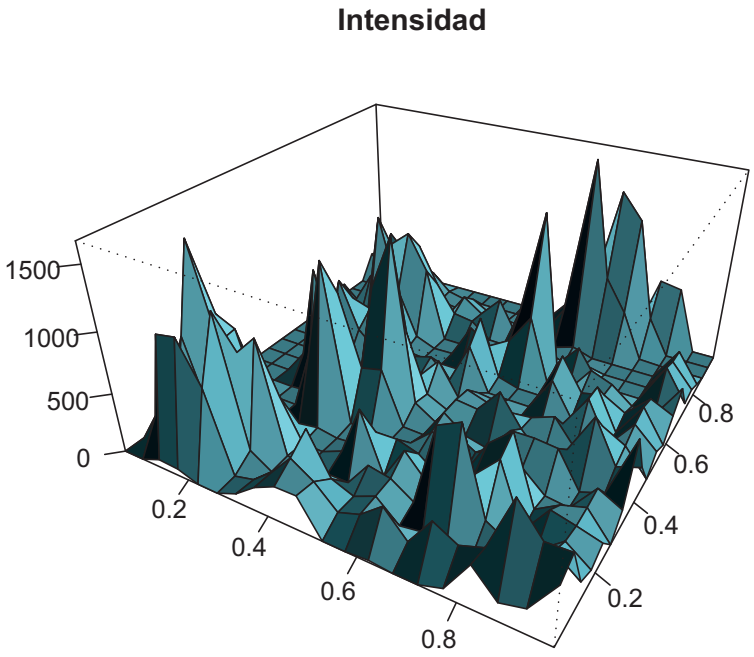

Figura 4.12 : Intensidad estimada

Como también utilizaremos el paquete `spatstat`, primero abrimos las librerías que vamos a utilizar en el ejemplo. Luego, en `(1)`, creamos el polígono en el que vamos a estimar la intensidad que es el cuadrado de lado unidad, definido dando los dos vértices extremos. Ahora, en `(2)` obtenemos 100 valores del error cuadrático medio (MSE) para 100 valores h (el máximo $h = 0'15$) utilizando la función `mse2d`, al haber considerado que el valor $0'15$ es el máximo admisible. Es decir, obtenemos 100 pares de valores (h, MSE). Podríamos representarlos para ver en qué h se alcanza el menor MSE, pero es más sencillo ejecutar `(3)`

```
> library(splancs)
```

```
> library(spatstat)

> poli<-as.points(list(x=c(0,0,1,1),y=c(0,1,1,0)))                    (1)

> suavizados<-mse2d(as.points(as(redwoodfull,"ppp")),poli,100,0.15)   (2)

> suavizados$h[which.min(suavizados$mse)]                             (3)
[1] 0.039
```

Ahora que ya sabemos que el suavizado a utilizar será $h = 0'039$ (es decir, la intensidad será poco suave), podemos obtener las estimaciones de la intensidad utilizando la función `kernel2d` ejecutando `(4)`. Por defecto elige el kernel biponderado. Lo que ocurre es que así se obtienen muchas cosas. Las coordenadas en donde se está estimando la intensidad se obtienen separadamente ejecutando `(5)` y `(6)`, cosa que no tiene mucho interés. Lo interesante son los valores estimados para esas localizaciones dadas por `(7)`.
La representación en tres dimensiones de valores z para pares de datos (x, y) la haremos con la función `persp` ejecutando `(8)` y obteniendo la Figura 4.12.

```
> kernel2d(as.points(as(redwoodfull,"ppp")),poli,h0=0.039)            (4)

> a1<-kernel2d(as.points(as(redwoodfull,"ppp")),poli,h0=0.039)$x       (5)
> a2<-kernel2d(as.points(as(redwoodfull,"ppp")),poli,h0=0.039)$y       (6)
> a3<-kernel2d(as.points(as(redwoodfull,"ppp")),poli,h0=0.039)$z       (7)

> persp(a1,a2,a3,theta=30,phi=30,expand=0.5,col="lightblue",ltheta=120,     (8)
+ shade=0.75,ticktype="detailed",xlab=" ",ylab=" ",zlab=" ",main="Intensidad")
```

Ejemplo 4.5

En Diggle (2003) se presenta un problema real de 1292 datos de niños enfermos de asma en North Derbyshire (Reino Unido) datos recogidos con objeto de explorar la relación entre esta enfermedad y la proximidad a las principales carreteras y a tres centros habitualmente contaminantes: una planta de coque (combustible sólido), una fábrica de productos químicos y un centro de tratamiento de residuos.
Para llevar a cabo este estudio se muestrearon 10 escuelas de la región y se formaron dos grupos: *Casos*, compuesto por todos los niños de esos 10 centros escolares que padecían asma y, *Control*, formado por los niños de esas escuelas que no padecían asma.
Estos datos están en los ficheros `asma1`, `asma2`, `asma3` y `asma4` (con sus respectivas extensiones), obtenidos a partir de Diggle (2003) y Bivand et al. (2013).
Si queremos hacer primero un gráfico de estos datos utilizando R como GIS, ejecutaríamos desde la línea de comandos de R la siguiente secuencia, suponiendo que nuestros ficheros con sus extensiones GIS están en `d:/datos`, ya que la función `readOGR` de la librería `rgdal` importará a R los ficheros tipo `shape` de GIS, es decir los ficheros con extensión shp, shx y dbf que aparecen en el segundo argumento de dicha función (sin poner la extensión), localizados en donde indicamos mediante su primer argumento. Es decir, que por ejemplo con `(1)` indicamos a R que importe los ficheros `asma1.shp`, `asma1.shx` y `asma1.dbf` localizados en `d:/datos`.

```
> library(rgdal)
> asma1 <- readOGR("d:/datos", "asma1")                               (1)
```

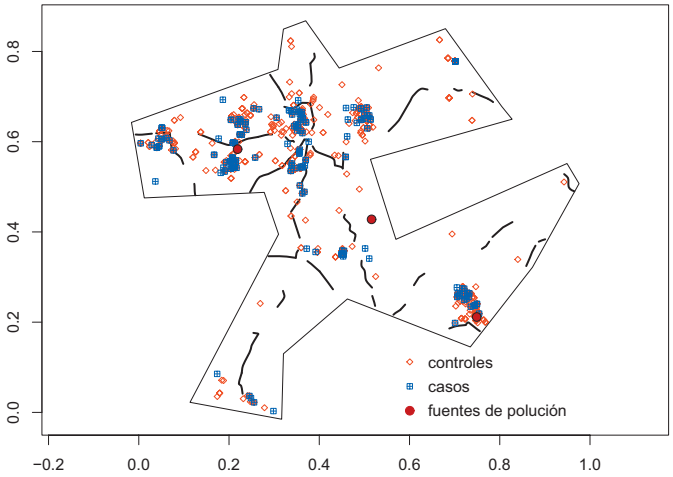

Figura 4.13 : GIS del problema con R

```
> asma2 <- readOGR("d:/datos", "asma2")
> asma3 <- readOGR("d:/datos", "asma3")
> asma4 <- readOGR("d:/datos", "asma4")

> plot(asma2, axes=TRUE, lwd=1)                                          (2)
> plot(asma4, add=TRUE, lwd=2)                                           (3)
> caso <- (asma1$Asma == "case")
> plot(asma1[caso == 0,], add=TRUE, pch=5, cex=0.6, col=2)               (4)
> plot(asma1[caso == 1,], add=TRUE, pch=12, cex=0.75, col=4)             (5)
> plot(asma3, pch=21, add=TRUE, cex=1.2, bg="brown")                     (6)

> legend("bottomright", legend=c("controles", "casos", "fuentes de polución"),
+ pch=c(5, 12, 21), pt.cex=c(0.6, 0.75, 1.2), pt.bg=c(NA, NA, "brown"),
+ col=c(2,4,"brown"), bty="n")
```

Mediante esta secuencia de comandos obtenemos la Figura 4.13. Vemos en ella representados, el contorno del gráfico, obtenido ejecutando `(2)`, las carreteras obtenidas ejecutando `(3)`, los individuos *Control* en rojo obtenidos con `(4)` y *Casos* en azul visualizados con `(5)`, y, por último, los centros contaminantes obtenidos con `(6)` en marrón. La leyenda última especifica en el gráfico los anteriores elementos.
Este ejemplo también se trabaja en Bivand et al. (2013) pero, tanto allí como aquí, la ejecución de todos estos comandos en R resulta más compleja que si hacemos la representación con QGIS.
Los ficheros utilizados están en la dirección de Internet anunciada en la introducción. Es muy interesante cargar en QGIS las cuatro capas shp anteriores, `asma1`, `asma2`, `asma2`, `asma3` y

`asma4` obteniendo fácilmente la Figura 4.14. Tendrá que clicar dos veces en la capa `asma2` para aumentar la transparencia. En este gráfico no hemos diferenciado entre Casos y Controles.

Figura 4.14 : GIS del problema con QGIS

Tanto con R como con QGIS hemos realizado un Análisis Estadístico de tipo descriptivo. Vamos a modelizar estos datos mediante el ajuste de un proceso de Poisson no homogéneo de parámetro $\lambda(\mathbf{s})$ estimando esta intensidad como hicimos antes mediante Métodos no Paramétricos, basados en el *Estimador Núcleo Suavizado* (*kernel smoothing*) dado por 4.1, es decir, por

$$\widehat{\lambda}(\mathbf{s}) = \frac{1}{q(||\mathbf{s}||)\, h^2} \sum_{i=1}^{n} K\left(\frac{||\mathbf{s} - \mathbf{s}_i||}{h}\right)$$

en donde el nivel de suavizado h se puede determinar como antes o mediante *cross-validation*. Llamaremos $\lambda_1(\mathbf{s})$ y $\lambda_0(\mathbf{s})$ a las intensidades de los procesos de Poisson no homogéneos que modelizan, respectivamente, a las poblaciones Casos, de la que observamos n_1 individuos y Control, en la que observamos n_0. La hipótesis nula de que ambas poblaciones pueden considerarse iguales se expresa diciendo que ambas intensidades son iguales salvo una constante de proporcionalidad, es decir, que es $\lambda_1(\mathbf{s}) = n_1/n_0\, \lambda_0(\mathbf{s})$.
Suele considerarse como *riesgo de enfermedad* en la comparación de dos poblaciones del tipo considerado en este ejemplo, el cociente entre las intensidades Casos y Control:

$$\rho(\mathbf{s}) = \frac{\lambda_1(\mathbf{s})}{\lambda_0(\mathbf{s})}$$

que, supuesto es cierta la hipótesis nula de *igualdad de riesgo en ambas poblaciones* se traduce en $\rho(\mathbf{s}) = n_1/n_0 = \rho_0$.
Alternativamente puede venir expresado el *riesgo de enfermedad* como el logaritmo de la razón de las densidades de Casos y Control:

$$r(\mathbf{s}) = \log \frac{f(\mathbf{s})}{g(\mathbf{s})} = \log f(\mathbf{s}) - \log g(\mathbf{s})$$

siendo

$$f(\mathbf{s}) = \frac{\lambda_1(\mathbf{s})}{\int \lambda_1(\mathbf{s}) d\mathbf{s}} \qquad \text{y} \qquad g(\mathbf{s}) = \frac{\lambda_0(\mathbf{s})}{\int \lambda_0(\mathbf{s}) d\mathbf{s}}$$

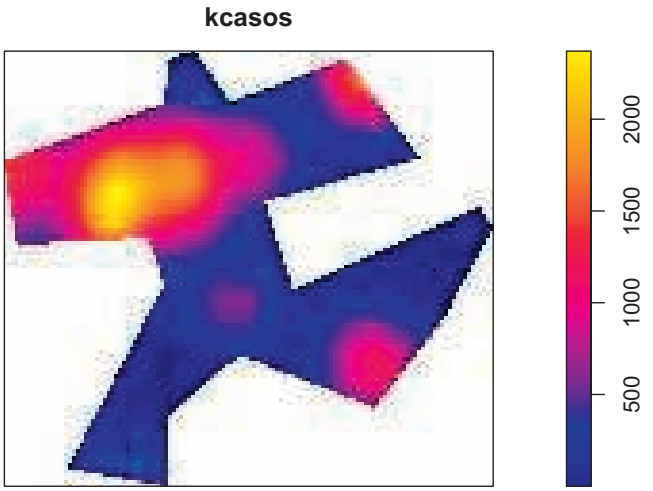

Figura 4.15 : Densidad de los Casos

y, en esta situación, la hipótesis nula significará $r(\mathbf{s}) = 0$.
En este ejemplo hemos prefijado el nivel de suavizado con `(7)` en el valor $0'06$ y estimado las densidades en `(8)` y `(9)`. La representación de ambas densidades, Figura 4.15 y Figura 4.16, parece indicarnos que éstas son bastante semejantes.

```
> library(sp)
> library(maptools)
> library(spatstat)
> library(rgeos)
```

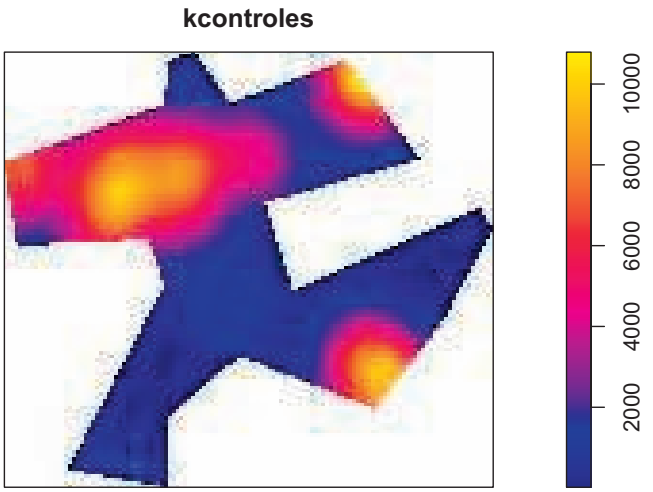

Figura 4.16 : Densidad de los Controles

```
> hasma <- 0.06                                                    (7)
> pppasma <- as(asma1, "ppp")
> pppasma$window <- as(asma2, "owin")
> marks(pppasma) <- relevel(pppasma$marks$Asma, "control")

> casos <- unmark(subset(pppasma, marks(pppasma) =="case"))
> ncasos <- npoints(cases)
> controles <- unmark(subset(pppasma, marks(pppasthma) =="control"))
> ncontroles <- npoints(controles)
> kcasos <- density(casos, hasma)                                  (8)
> kcontroles <- density(controles, hasma)                          (9)

> .iwidth <- 5
> .iheight <- 5
> .ipointsize <- 10

> plot(kcasos)
> plot(kcontroles)
```

El riesgo de enfermedad $r(\mathbf{s})$ es calculado en (10).

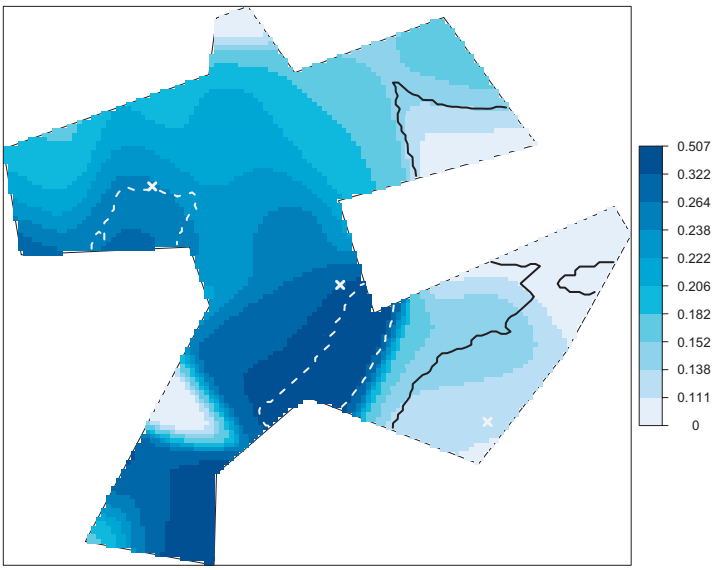

Figura 4.17 : Mapa de p-valores del test

```
> ratio0 <- as(kcasos, "SpatialGridDataFrame")
> names(ratio0) <- "kcasos"
> ratio0$kcontroles <- as(kcontroles, "SpatialGridDataFrame")$v
> ratio <- as(ratio0, "SpatialPixelsDataFrame")
> ratio$ratio <- ratio$kcasos/ratio$kcontroles
> ratio$logratio <- log(ratio$ratio) - log(ncases/ncontrols)                (10)
```

Para contrastar el test de la hipótesis nula antes mencionado, Diggle et al. (2007) proponen utilizar como estadístico de contraste

$$T = \int (\rho(\mathbf{s}) - \rho_0)^2 \, d\mathbf{s}$$

en donde $\rho(\mathbf{s})$ se estima con el cociente de los estimadores de las intensidades. La hipótesis nula de riesgo constante en toda la región de estudio se rechaza para valores grandes de este estadístico.

No obstante, suele ser de más interés contrastar la hipótesis nula de si el estimador del cociente de intensidades $\widehat{\rho}(\mathbf{s})$ es aceptable mediante el estadístico de contraste

$$T = \int (\rho(\mathbf{s}) - \widehat{\rho}(\mathbf{s}))^2 \, d\mathbf{s}$$

test que se ejecuta mediante bootstrap.

Se podría dibujar (puede verse Bivand et al., 2013), un *mapa de p-valores* de este test, que daría como gráfico la Figura 4.17 de donde parece deducirse que no hay relación significativa con las distancias a la principales carreteras y sólo uno de los tres centros contaminantes (el de abajo) parece indicar diferencias entre niños asmáticos (Casos) y no asmáticos (Controles).

En el último capítulo de este libro volveremos a este ejemplo utilizando modelos GAM para explicar estos datos.

4.3.4. Análisis de la densidad espacial

Este objetivo se consigue fácilmente con la función `summary`.

Ejemplo 4.1 (continuación)

Primero debemos abrir la librería en donde están los datos, en este caso `spatstat`, ejecutando `(1)`. Luego, ejecutando `(2)`, obtenemos la densidad en `(3)`, que es de 42 datos por unidad de área.

```
> library(spatstat)                                                    (1)
> summary(cells)                                                       (2)
Planar point pattern: 42 points
Average intensity 42 points per square unit                            (3)

Window: rectangle = [0, 1] x [0, 1] units
Window area =  1 square unit
```

Ejemplo 4.2 (continuación)

Supuesto que ya hemos abierto la librería `spatstat`, ejecutando `(1)`, obtenemos la densidad en `(2)`, que es de 65 datos por unidad de área.

```
> summary(japanesepines)                                               (1)
Planar point pattern: 65 points
Average intensity 65 points per square unit (one unit = 5.7 metres)    (2)

Window: rectangle = [0, 1] x [0, 1] units
Window area =  1 square unit
Unit of length: 5.7 metres
```

Ejemplo 4.3 (continuación)

De nuevo, abierta la librería `spatstat`, ejecutando `(1)`, obtenemos la densidad en `(2)`, que es de 62 datos por unidad de área.

```
> summary(redwood)                                                    (1)
Planar point pattern: 62 points
Average intensity 62 points per square unit                           (2)

Window: rectangle = [0, 1] x [-1, 0] units
Window area =  1 square unit
```

Un esquema-resumen del capítulo aparece a continuación.

- Localizaciones fijas: Geoestadística.
- Localizaciones aleatorias: Procesos Puntuales Espaciales
  - Analizar la distribución
    - Aleatoriamente: CSR
      - quadrants
      - distancias
    - Regularmente: Procesos de Inhibición Simple.
    - Formando Clusters
      - Proceso de Poisson no homogéneo
        - Métodos paramétricos: Modelo log-lineal
        - Métodos no paramétricos: Estimador núcleo suavizado
      - Proceso de Cox.
      - Proceso de Poisson con Clusters.
  - Estudiar las marcas: comparar poblaciones.
  - Estudiar la densidad: número de individuos por unidad de área.

Capítulo 5

Análisis de Datos Espaciales de tipo continuo. Geoestadística

5.1. Introducción

Cuando hablamos de Geoestadística generalmente nos referimos al análisis de datos espaciales $\{Z(\mathbf{s}_1), ..., Z(\mathbf{s}_n)\}$ en donde las localizaciones $\{\mathbf{s}_1, ..., \mathbf{s}_n\}$ no son aleatorias sino fijas.

Es muy habitual que no tengamos observaciones de Z en todas las localizaciones deseadas, siendo uno de los propósitos de la Geoestadística, **hacer predicciones** (generalmente mediante *interpolaciones*) de Z en algunas localizaciones $\mathbf{s}_0$ en donde se desconoce su valor. También suele ser de interés **conocer la media** de Z en alguna zona determinada, B_0.

Estos dos objetivos se suelen conseguir estimando y modelizando la *correlación espacial* (covarianza o semivarianza) y analizando la *estacionariedad*, tanto en localizaciones concretas como en un Redes (Grids).

5.2. Variograma

En todos estos objetivos acabados de mencionar, hay una función que juega un papel destacado; se trata del *Variograma*. Desde un punto de vista teórico, supuesto que las observaciones son *estacionarias* (véase TA-sección 13.3), es decir, de la forma

$$Z(\mathbf{s}) = \mu + e(\mathbf{s})$$

siendo $\mu = E[Z(\mathbf{s})]$ constante y siendo también constante la varianza de $Z(\mathbf{s})$,

se define el *variograma* como

$$\gamma(h) = \frac{1}{2} E[(Z(\mathbf{s}) - Z(\mathbf{s}+h))^2].$$

En realidad, para ser más precisos, se debería de reservar el término variograma a $2\gamma(h)$ y denominar al valor anterior *semi-variograma* pero es habitual utilizar esta denominación por lo que aquí también la seguiremos.

Bajo la suposición de estacionariedad, es decir, que tanto la media como la varianza de Z son constantes, el variograma lo que expresa es la correlación espacial, la cual admitimos (por esa estacionariedad) que no depende de la localización $\mathbf{s}$, sino sólo de la distancia h (habitualmente Euclídea) que separa a dos observaciones $Z(\mathbf{s})$ y $Z(\mathbf{s}+h)$, puesto que podemos considerar esa media constante igual a 0 (por ser equivalente a hacer los cálculos como si restáramos a todas los valores su media) y al ser también la varianza constante, $\gamma(h)$ representará (salvo constantes) a la correlación espacial.

Por tanto, lo que básicamente hacemos al calcular el variograma es formar pares de observaciones $Z(\mathbf{s}_i), Z(\mathbf{s}_j)$ que estén a una separación $h = \mathbf{s}_i - \mathbf{s}_j$ y estimar el coeficiente de correlación entre ellos.

Dado que $\gamma(h)$ raramente será conocido, lo estimaremos a partir de los datos observados mediante el *variograma muestral*

$$\hat{\gamma}(h) = \frac{1}{2n(h)} \sum_{i=1}^{n(h)} (Z(\mathbf{s}_i + h) - Z(\mathbf{s}_i))^2$$

siendo $n(h)$ el número de puntos separados por una distancia h.

5.2.1. Utilización de covariables

En algunas ocasiones (Capítulos 7 y 8) se admitirá para la variable espacial observada $Z(\mathbf{s})$ un modelo que expresa influencia lineal de k variables *independientes* o *covariables* $X_j(\mathbf{s})$ de la forma

$$Z(\mathbf{s}) = \beta_0 + \beta_1 X_1(\mathbf{s}) + ... + \beta_k X_k(\mathbf{s}) + e(\mathbf{s})$$

siendo $e(\mathbf{s})$ una variable de error sobre la que hay que estudiar las propiedades de estacionariedad supuestas. Sobre esta posibilidad volveremos cuando estudiemos estos capítulos más adelante.

5.2.2. Análisis exploratorio del Variograma

Además del variograma muestral, una forma sencilla de analizar si la correlación espacial está presente es, como dijimos más arriba, obtener diagramas de dispersión de pares de observaciones $Z(\mathbf{s}_i), Z(\mathbf{s}_j)$ que estén a una separación $h_{ij} = ||\mathbf{s}_i - \mathbf{s}_j||$ y estimar el coeficiente de correlación entre ellos.

Recordamos que ya definimos en el capítulo anterior $||\mathbf{s}||$ como la norma o longitud del vector $\mathbf{s}$ que, si es bidimensional con coordenadas $(\mathbf{s_1}, \mathbf{s_2})$, es igual a $||\mathbf{s}|| = \sqrt{s_1^2 + s_2^2}$.

Un gráfico que también se utiliza es la denominada *Nube Variograma* que es una representación gráfica que representa las posibles diferencias al cuadrado de pares de observaciones

$$(Z(\mathbf{s}_i) - Z(\mathbf{s}_j))^2$$

frente a sus distancias de separación, $\mathbf{s}_i - \mathbf{s}_j$, es decir, nos da diferencias (al cuadrado) de valores de la variable en observación Z, en función de las distancias que separa a los puntos en donde se observa.

En el Análisis de Datos Espaciales se admite la denominada *hipótesis de autocorrelación espacial positiva*, la cual quiere decir que observaciones con valores grandes (o pequeños) están rodeadas por observaciones con también valores grandes (o pequeños). Si algún valor $Z(\mathbf{s_0})$ está rodeado de otros en donde la variable Z toma valores muy distintos, $Z(\mathbf{s_0})$ es un *outlier local*, o como también se denomina, *espacial*. Los valores de la Nube Variograma en la parte superior izquierda serán outliers locales porque están a una distancia h muy pequeña y su diferencia (al cuadrado) es grande. Es decir, hay una pendiente grande entre sus valores, indicando la falta de autocorrelación espacial positiva.

Ejemplo 1.2 (continuación)

En el Ejemplo 1.2 estudiamos los datos `meuse` que están en la librería `sp` y en el fichero `meuse10.txt`. En esta matriz de datos estaban, entre otras variables, los niveles de Zinc, cuyo logaritmos se pueden obtener, por tanto, con el comando `log(meuse$zinc)`. Si se representan estos 164 datos ejecutando por ejemplo `plot(log(meuse$zinc))` se verá una dispersión muy grande y que ninguna recta de regresión suministraría un ajuste aceptable. Por esta razón podemos admitir una tendencia constante con ordenada en el origen, lo que se expresa en R con un modelo de la forma

```
log(zinc)~1
```

Supongamos que deseamos representar diagramas de dispersión de pares de observaciones de los datos `meuse` a distancias $(0, 50]$, $(50, 100]$, $(100, 150]$, $(150, 200]$, $(200, 250]$, $(250, 300]$. Para ello, primero abrimos los datos `meuse` que están en la librería `sp` con `(1)` y `(2)`, los convertimos en un objeto de la clase `SpatialPoints-DataFrame` con `(3)`, obteniendo el objeto `coordinates(meuse)`, que son las coordenadas sobre cuyos valores estableceremos, con el último argumento de la función `hscat`, los grupos o clases de valores a distancias menores que las allí establecidas. Es decir, que cuando hablemos en este ejercicio de distancias, nos referiremos a distancias Euclídeas en el plano entre dos puntos es decir, $\mathbf{s}_i - \mathbf{s}_j$. Todo esto se ejecuta en `(4)` con la función `hscat` de la librería `gstat`

```
> library(sp)                                                    (1)
> data(meuse)                                                    (2)
> coordinates(meuse) = ~x+y                                      (3)
```

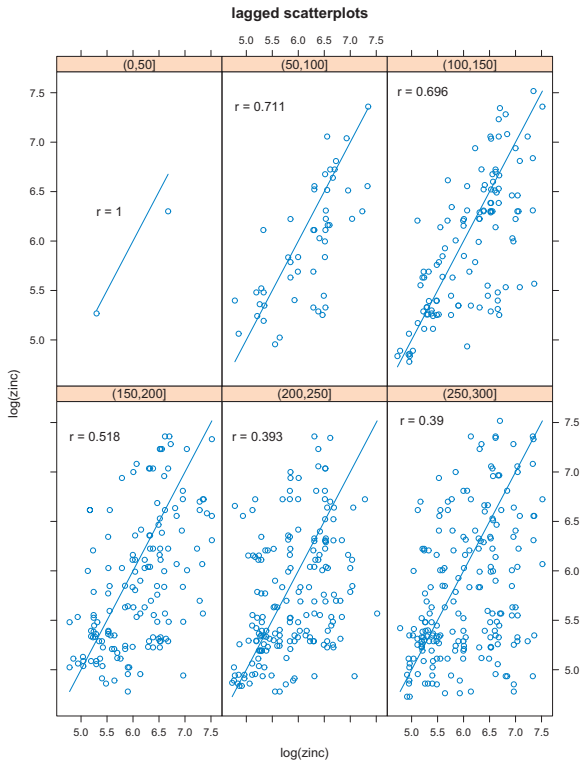

Figura 5.1 : Diagramas de dispersión por intervalos de distancias entre puntos

```
> library(gstat)
> hscat(log(zinc)~1,data=meuse,breaks=c(0,50,100,150,200,250,300))          (4)
```

obteniendo la Figura 5.1. En el primer argumento utilizado en la función `hscat` admitimos que los datos `log(zinc)` siguen una tendencia constante con ordenada en el origen y con el último argumento le indicamos los puntos de corte de los intervalos de distancias, en base a las coordenadas proporcionadas con `coordinates(meuse)`.
En la Figura 5.1 se ve que a menos de una distancia de 50 apenas hay datos y que la mayor correlación se da a una distancia entre 50 y 100.
Este gráfico anterior tiene una limitación importante y es que, para ser interpretable, no pueden utilizarse muchos datos ni un rango grande de distancias. Otra forma, más habitual, de analizar de correlación espacial es mediante el variograma muestral antes definido que, para los datos de este ejemplo se obtiene ejecutando la sentencia

```
> plot(variogram(log(zinc) ~ 1, meuse, cloud = F),pch=16)
```

obteniendo la Figura 5.2 que indica que la mayor correlación se da a distancias algo mayores de 1000.
Por último, la Nube Variograma se obtiene ejecutando la sentencia

```
> plot(variogram(log(zinc) ~ 1, meuse, cloud = T))
```

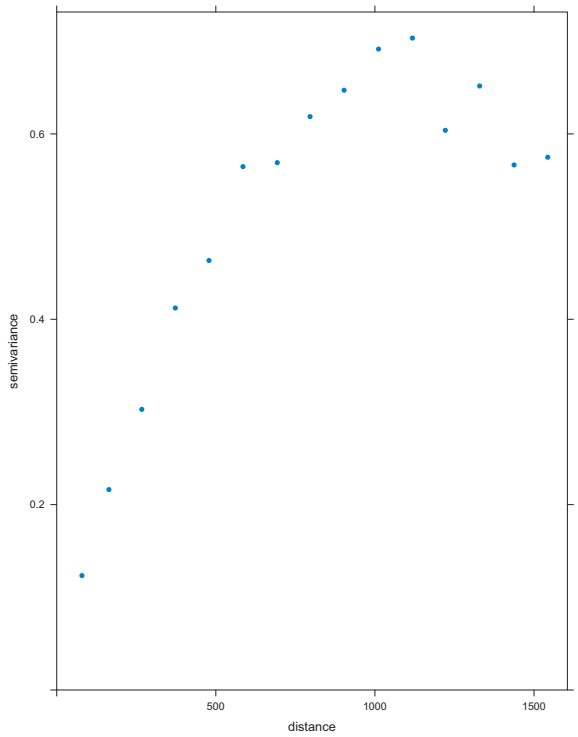

Figura 5.2 : Variograma muestral

con la que se consigue la Figura 5.3.
Se observa en esta figura que los valores de la izquierda del eje de abscisas (h pequeños) y arriba del de ordenadas (diferencias grandes) son los que muestran, por tanto, una mayor pendiente y serán outliers locales.
Si quisiéramos identificar cuáles son estos puntos de entre los datos iniciales, podemos ejecutar primero `(5)`, ir al gráfico generado y marcar con el ratón los límites de un polígono. Se para este proceso con el botón derecho del ratón obteniendo la Figura 5.4. Ahora podemos ver cuáles son los datos originales de dentro del polígono generado al ejecutar `(6)`, obteniendo así la Figura 5.5.

```
> sel <- plot(variogram(log(zinc) ~ 1, meuse, cloud = T),digitize = T)      (5)
> plot(sel, meuse)                                                          (6)
```

5.3. Interpolación espacial

Podemos definir el problema de interpolación espacial como el de determinar los valores de Z en un punto $\mathbf{s}_0$, en donde se desconoce su valor, a partir

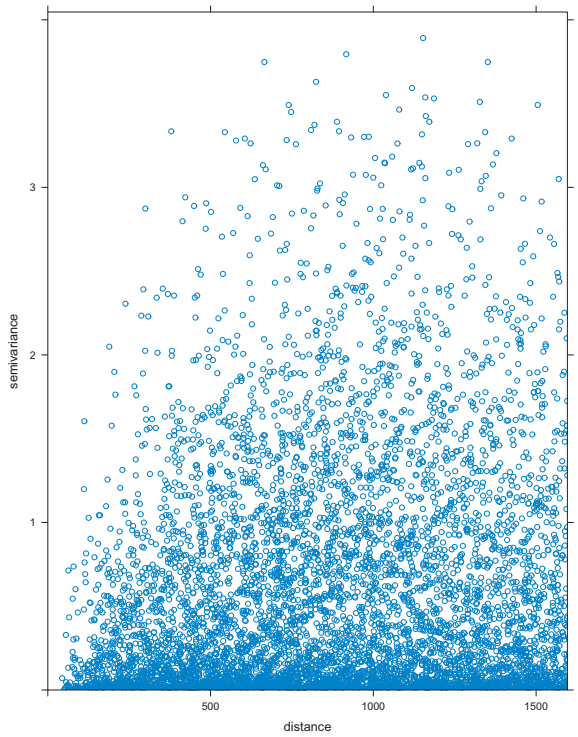

Figura 5.3 : Nube Variograma

de valores conocidos de Z en un cierto número n de *localizaciones* $\{\mathbf{s}_1, ..., \mathbf{s}_n\}$.

Uno de los métodos puede ser la *Interpolación Ponderada Inversa a la Distancia* (*Inverse Distance Weighted Interpolation*) definiendo la interpolación (estimación de $Z(\mathbf{s}_0)$ en realidad) como

$$\widehat{Z(\mathbf{s}_0)} = \frac{\sum_{i=1}^{n} w_i(\mathbf{s}_i) Z(\mathbf{s}_i)}{\sum_{i=1}^{n} w_i(\mathbf{s}_i)}$$

en donde los pesos w_i se determinan como el inverso de la distancia al punto de interpolación, generalmente como

$$w_i(\mathbf{s}_i) = ||\mathbf{s}_i - \mathbf{s}_0||^{-p}$$

siendo $||\mathbf{s}_i - \mathbf{s}_0||^{-p}$ un inverso de la distancia Euclídea entre $\mathbf{s}_i$ y $\mathbf{s}_0$.

Otra forma de interpolación es la de los *splines cúbicos* estudiados en el texto TA. Estos dos tipos de interpolación ignoran la distribución espacial de los datos.

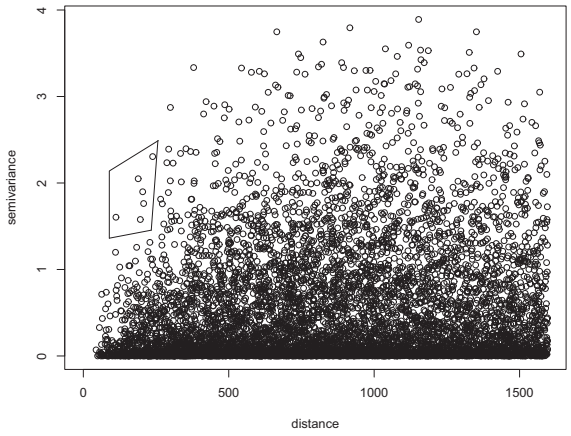

Figura 5.4 : Nube Variograma con polígono de datos extremos

Nosotros recomendamos utilizar, en lugar de estos métodos, otros de interpolación espacial que eligen estos pesos $w_i(\mathbf{s}_i)$ en función del grado de semejanza entre los valores de Z a partir de la covarianza entre los puntos en función de la distancia que lo separa. Estos métodos son el *kriging simple* (nombre debido al ingeniero de minas sudafricano que lo definió D. G. Krige), el *kriging ordinario* (o puntual) y el *kriging universal* basado en el *variograma muestral* $\hat{\gamma}(h)$.

Los dos primeros métodos mencionados requieren que la media, supuesta conocida en el primero (y muy poco utilizado por tanto) y desconocida en el segundo, y la varianza de Z sean estacionarias, es decir, que no dependan de las localizaciones sino solamente de la distancia que los separa.

En el tercer método, el kriging universal, se supone que la variable no es estacionaria, es decir, que se observa una tendencia. El kriging consiste en calcular los pesos w_i anteriores utilizando el *variograma muestral* $\hat{\gamma}(h)$.

El método de los splines cúbicos equivale un kriging utilizando una covarianza generalizada de orden 1. Estas funciones son muy utilizadas para cartografiar variables meteorológicas porque proporcionan una imagen lisa del fenómeno interpolado.

Otra posibilidad en cuanto a la predicción está basada en la modelización del variograma, útil tanto para explicar el fenómeno que estemos observando como en la predicción. Esta modelización se realiza mediante el ajuste de varios modelos al variograma muestral. Pueden estudiarse varias propuestas en Bivand et al. (2013, pp. 224-232).

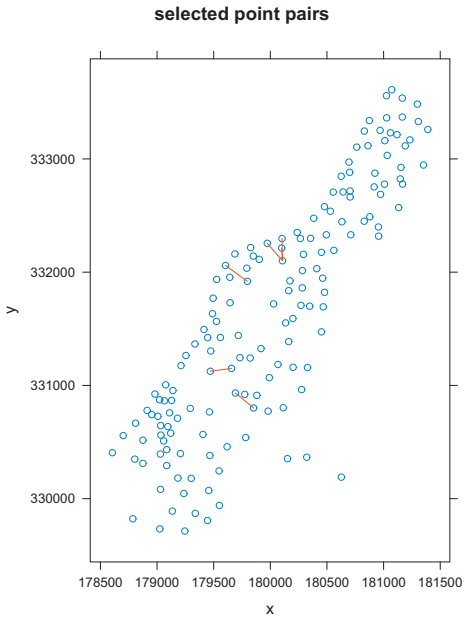

Figura 5.5 : Datos originales de dentro del polígono

Capítulo 6

Análisis de Datos Espaciales agregados o regionales

6.1. Introducción

En ocasiones, los datos espaciales observados son *áreas* o *zonas*, datos que hemos denominados en el título del capítulo como datos agregados o regionales. Estas áreas serán habitualmente las unidades de investigación, es decir, los datos observados, los cuales deben de tener límites bien definidos. No obstante, por su propia definición, varias áreas pueden unirse y formar un nuevo "dato" por lo que es de gran interés analizar si existe *autocorrelación espacial* entre varias unidades ya que, en muchas ocasiones, lo que pase en una zona está relacionado con lo que pase en la zona adyacente. Este problema se conoce como *problema de Galton* que consiste básicamente en establecer cuántas observaciones (zonas) independientes hay en la muestra cuando se han utilizado límites arbitrarios en la definición de las áreas en estudio. Por esta razón, la primera sección de este capítulo se dedica a estudiar cómo definir pesos a las áreas o zonas consideradas.

6.2. Entornos y pesos de Áreas

Asignar pesos a las zonas en estudio es necesario si queremos realizar un Análisis de datos espaciales supuesto que éstos sean Áreas, fundamentalmente con el propósito de conseguir residuos totalmente independientes (i.e., sin autocorrelación espacial) cosa que será necesaria cuando hagamos el ajuste de un modelo a este tipo de datos.

La determinación de las zonas (o clusters) a considerar ya es un problema en sí mismo aunque supondremos que las zonas están ya definidas. Después, a la hora de fijar pesos a esas zonas se recomienda asignar un peso igual a 1 a

las zonas limítrofes y un peso igual a 0 a las zonas que no son limítrofes a una dada. Es decir, si una zona A tiene sólo una zona como vecina, el peso de A será 1; si A tiene dos zonas vecinas, su peso será 2. Esta forma de fijar pesos se denomina *binaria* y, con ella, si una zona tiene a su alrededor 2 zonas, su pero será doble que el de otra zona con sólo una zona limítrofe.

La forma natural de contar cuántos vecinos tiene una zona es unir los centroides de las zonas. El número de conexiones entre los centroides nos dará su peso.

La forma binaria de asignar pesos hará que algunas áreas tengan peso 2 y otras pesos, por ejemplo, 7. Otra forma de asignar pesos es una forma *ponderada*, en donde los pesos de cada área son estandarizados de forma que todos los pesos sumen 1.

La función `nb2listw` de la librería de R, `spdep` es la que calcula los pesos de las zonas de las dos maneras antes mencionadas, las cuales se indican con el argumento `style`, asignando el valor `W` en el caso de la forma ponderada y asignando el valor `B` cuando lo hace de forma binaria.

En todo caso, representaremos por w_{ij} al *peso espacial* de la unión entre el individuo i-ésimo de la matriz de datos (fila i-ésima) y el individuo j-ésimo en donde se habrá medido la variable de interés, obteniendo respectivamente los valores y_i e y_j; es decir, por y_i representaremos el valos de Z en $\mathbf{s}_i$, es decir, $Z(\mathbf{s}_i)$.

6.3. Contraste global de autocorrelación espacial: Estadístico I de Moran

El estadístico (índice) I de Moran se define como el cociente del producto de la variable de interés y su retardo (*lag*) de la forma

$$I = \frac{n}{\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}} \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}\,(y_i - \overline{y})(y_j - \overline{y})}{\sum_{i=1}^{n}(y_i - \overline{y})^2)}.$$

Este índice toma un valor comprendido entre -1 y 1. Si es $I = 0$ interpretamos que los datos están distribucidos al azar (del estilo del CSR que vimos); si es positivo habrá concentración y, si toma un valor negativo, entendemos que hay una dispersión mayor de la que tendríamos si los datos se distribuyeran al azar.

El valor esperado del índice I, bajo la hipótesis nula de ausencia de autocorrelación espacial es $E[I] = -1/(n-1)$. Habitualmente se calculan los valores de este índice, de su media, de su varianza, de la desviación estándar $(I - E(I))/\sqrt{Var(I)}$, así como del p-valor del test de la hipótesis nula de

ausencia de autocorrelación espacial.

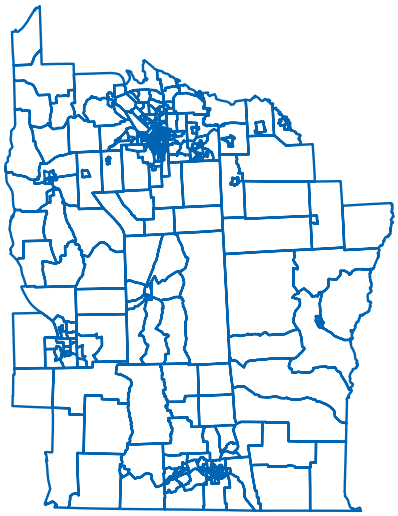

Figura 6.1 : Mapa del estado de Nueva York

Ejemplo 6.1

Supongamos que tenemos interés en estudiar el estado americano de Nueva York estado compuesto por varios condados. En el directorio `d:/datos` tenemos los 3 ficheros asociados a un GIS (el de extensión shp, el de extensión shx y el de extensión dbf), los tres con el nombre `NY` (datos basados en Waller y Gotway, 2004 y Bivand, et al., 2013).
Esta matriz de datos sobre casos de leucemia en este estado está formado por 281 individuos y 12 variables. Es la que aparece en el fichero dbf antes mencionado.
Como hicimos en el Capítulo 4, incorporamos estos datos a R mediante la función `readOGR` ejecutando `(1)`. Si queremos, podemos representar el mapa ejecutando `(2)` en donde hemos elegido un color azul utilizando el argumento `border=4` y un grosor 2 con el argumento `lwd=2` obteniendo así la Figura 6.1.
Si queremos utilizar QGIS podemos importar directamente el fichero `NY.shp` sabiendo que estamos en coordenadas geográficas UTM zona 18N. La representación sería la Figura 6.2.
Con `(3)` incorporamos a R las zonas del estado de Nueva York y con `(4)` las unimos creando la Figura 6.3.
Con `(5)` asignamos los pesos a esas zonas eligiendo aquí la forma binaria.

```
> library(spdep)
> library(rgdal)
> NY<-readOGR("d:/datos","NY")                                    (1)
> plot(NY,border=4,lwd=2)                                         (2)
```

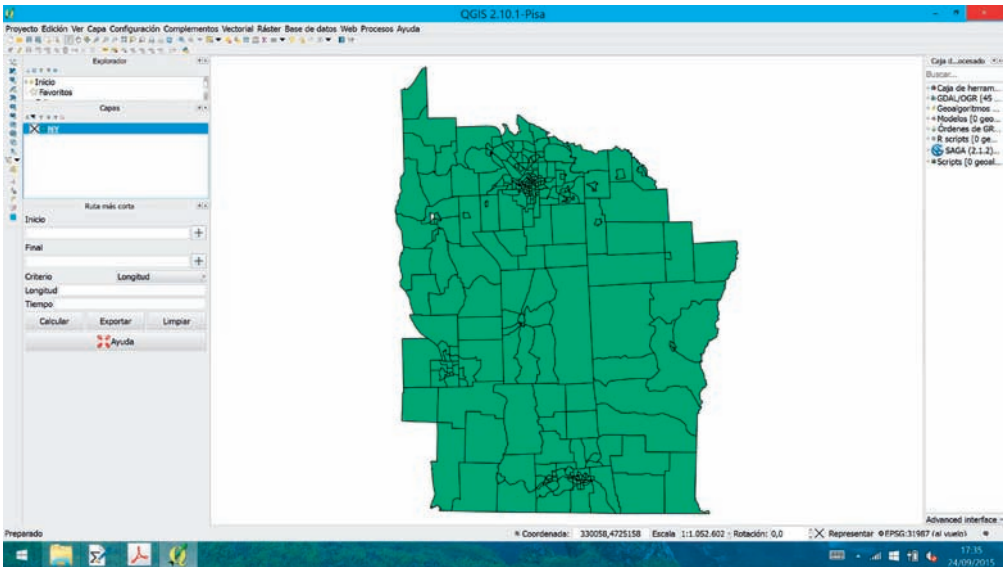

Figura 6.2 : GIS del problema con QGIS

```
> NYzonas<-read.gal("d:/datos/NY.gal",region.id=row.names(NY))                (3)
> plot(NYzonas,coordinates(NY),pch=16,cex=0.5,add=TRUE)                       (4)

> pesoszonas<-nb2listw(NYzonas,style="B")                                     (5)

> moran.test(NY$Casos,listw=pesoszonas)                                       (6)

        Moran's I test under randomisation

data:  NY$Casos
weights: pesoszonas

Moran I statistic standard deviate = 3.1862, p-value = 0.0007207            (7)
alternative hypothesis: greater
sample estimates:
Moran I statistic       Expectation          Variance
      0.110387402      -0.003571429       0.001279217
```

Entre los datos del censo está el número de casos de leucemia observados. Si queremos contrastar que estos casos se producen al azar, es decir no dependiendo del lugar (ausencia de correlación espacial) se puede ejecutar el test global de Moran. Este test sobre la hipótesis nula de ausencia de autocorrelación espacial es calculado ejecutando `(6)`, test cuyo p-valor se da en `(7)`, el cual es suficientemente pequeño como para rechazar la ausencia de autocorrelación espacial y concluir con la existencia de dicha relación. Es decir, hay una concentración espacial mayor de la cabría esperar si los casos se repartieran al azar en todo el estado.

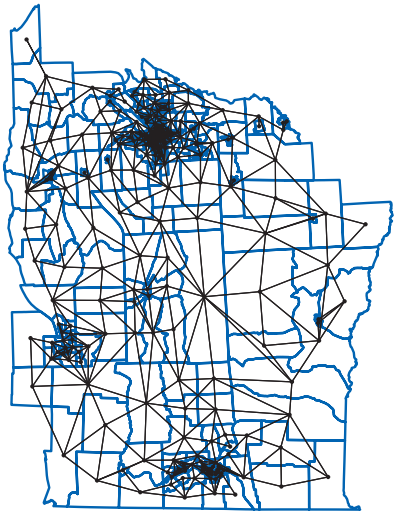

Figura 6.3 : Mapa del estado de Nueva York con zonas unidas

6.4. Contraste local de autocorrelación espacial: Gráfico de dispersión de Moran

El test de Moran estudiado más arriba es un test global de autocorrelación espacial. El valor obtenido con este test global se puede dividir para conseguir tests locales que permitan detectar clusters en donde las observaciones sean similares a las de su entorno así como detectar outliers locales, también denominados *puntos calientes* o *hotspots*.

Comencemos estudiando el *Gráfico de dispersión (scatterplot) de Moran*. Este gráfico es un gráfico de dispersión en donde aparecen en el eje de abscisas los valores de la variable de interés y en el eje de ordenadas esos mismos valores retardados espacialmente.

Este gráfico es dividido en cuatro cuadrantes recogiendo los pares de valores (bajo,bajo), (bajo,alto), (alto,bajo) y (alto,alto).

A este gráfico se añade un recta con pendiente igual al índice de Moran I tratando de expresar de esta forma una relación lineal con correlación el índice

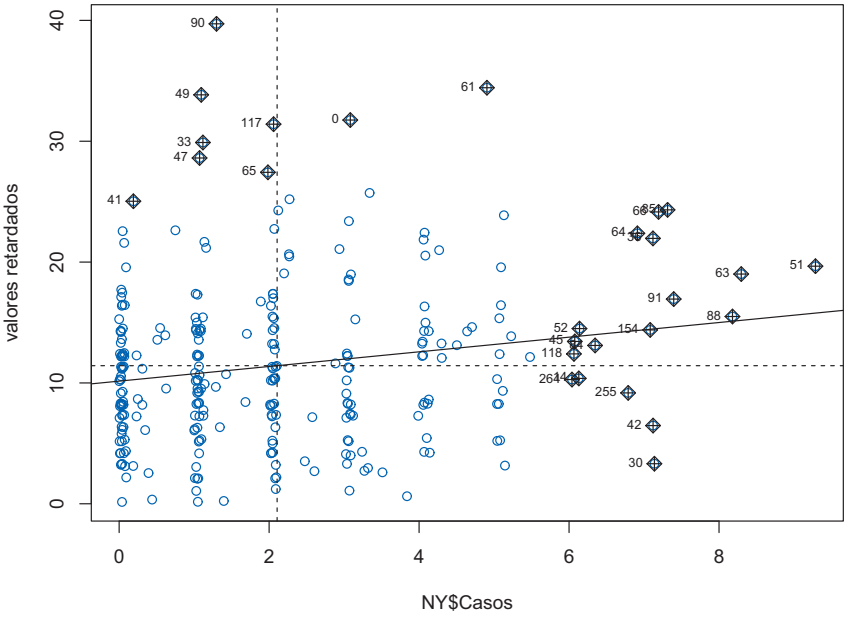

Figura 6.4 : Scatterplot de Moran

I de manera que se aprecian los puntos del gráfico que influyen en la *recta de regresión* así construida. Estos son sospechosos de autocorrelación espacial (*puntos calientes*).

Si definimos los *índices locales de Moran* como

$$I_i = n \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij}(y_i - \overline{y})(y_j - \overline{y})}{\sum_{i=1}^{n}(y_i - \overline{y})^2)}$$

se pueden hacer tests de zonas locales y contrastar esos puntos sospechosos.

Podemos escribir que es

$$\frac{1}{\sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij}} \sum_{i=1}^{n} I_i$$

con lo que, dado que el valor $\sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij}$ es una constante de normalización, podemos decir que con los índices locales de Moran descomponemos el índice global.

Ejemplo 6.1 (continuación)

El *scatterplot* de Moran se obtiene ejecutado `(1)`, obteniendo así Figura 6.4.

Los tests locales se ejecutan con (2). Los p-valores aparecen en la última columna.

```
> moran.plot(NY$Casos,listw=nb2listw(NYzonas,style="B"),col=4,
+ ylab="valores retardados")                                          (1)

> localmoran(NY$Casos, listw = nb2listw(NYzonas,style="B"))           (2)
               Ii         E.Ii     Var.Ii         Z.Ii    Pr(z > 0)
0    3.6835835873 -0.028571429  7.9485633  1.316684690 9.397217e-02
1    3.9539648449 -0.021428571  5.9606995  1.628289043 5.173181e-02
2   -2.0568022902 -0.010714286  2.9787125 -1.185523086 8.820947e-01
.............................................................
90  -4.6810881998 -0.028571429  7.9485633 -1.650226776 9.505517e-01
.............................................................
278 -0.4781876360 -0.014285714  3.9727337 -0.232745582 5.920205e-01
279  0.1852043194 -0.014285714  3.9727337  0.100086725 4.601377e-01
280  0.0512836166 -0.021428571  5.9606995  0.029782325 4.881203e-01
```

6.5. Ajuste de Modelos

En la mayoría de datos espaciales no habrá independencia, es decir, presentarán autocorrelación espacial. Una forma habitual de modelizar este problema es la de suponer que nuestras observaciones multivariantes $\mathbf{Y}$ (o $\mathbf{Z}$ si seguimos la notación anterior de este capítulo) se pueden expresar en función de covariables de la forma

$$Y = \beta_0 + \beta_1 X_1 + ... + \beta_k X_k + e$$

en donde e una variable de error con distribución normal multivariante de vector de medias cero y matriz de varianzas-covarianzas $\mathbf{V}$, aunque esta modelización podrá variar según el modelo considerado como por ejemplo los modelos Autorregresivos, similares a los utilizados en series temporales, capaces de recoger la dependencia espacial de la matriz $\mathbf{V}$

Nosotros, no obstante, nos decantamos por utilizar alguno de los modelos estudiados en los siguientes capítulos. Como muestra, vamos a utilizar una regresión lineal en el ejemplo que hemos tratado en este capítulo.

Ejemplo 6.1 (continuación)
Continuando con el ejemplo del estado americano de Nueva York, vamos a modelizar, en lugar de los valores observados Y_i los valores

$$Z_i = \log \frac{Y_i + 1}{n_i}$$

que ya están en la base de datos bajo el nombre de

```
NY$Z
```

Vamos a considerar como covariables `PEXPOSURE` (*Distancia inversa al Tricloroetileno más cercano*), `PCTAGE65P` (*Proporción de personas mayores de 65 años*) y `PCTOWNHOME`, (*Proporción de personas dueñas de su casa*).
Si ajustamos una regresión lineal múltiple ejecutando `(1)`, vemos con `(2)` y `(3)` que los residuos presentan autocorrelación espacial ya que el p-valor dado en `(4)` rechaza la hipótesis nula de ausencia de autocorrelación espacial.
Suele utilizarse mejor la sentencia `(5)`.

```
> recta <- lm(Z ~ PEXPOSURE + PCTAGE65P + PCTOWNHOME, data = NY)          (1)
> NY$residuos <- residuals(recta)                                         (2)

> library(spdep)
> moran.test(NY$residuos,list=pesoszonas)                                 (3)

        Moran's I test under randomisation

data:  NY$residuos
weights: pesoszonas

Moran I statistic standard deviate = 2.4457, p-value = 0.007229          (4)
alternative hypothesis: greater
sample estimates:
Moran I statistic       Expectation          Variance
      0.083090278      -0.003571429       0.001255603

> lm.morantest(recta,list=pesoszonas)                                     (5)

        Global Moran's I for regression residuals

data:
model: lm(formula = Z ~ PEXPOSURE + PCTAGE65P + PCTOWNHOME, data = NY)
weights: pesoszonas

Moran I statistic standard deviate = 2.638, p-value = 0.004169
alternative hypothesis: greater
sample estimates:
Observed Moran's I        Expectation           Variance
       0.083090278       -0.009891282        0.001242320
```

Observamos en `(6)` que la covariable `PEXPOSURE` no es significativa por lo que la quitamos y repetimos el análisis.

```
> summary(recta)

Call:
lm(formula = Z ~ PEXPOSURE + PCTAGE65P + PCTOWNHOME, data = NY)

Residuals:
    Min      1Q  Median      3Q     Max
```

```
-1.7417 -0.3957 -0.0326  0.3353  4.1398

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.51728    0.15856  -3.262  0.00124 **
PEXPOSURE    0.04884    0.03506   1.393  0.16480                                  (6)
PCTAGE65P    3.95089    0.60550   6.525 3.22e-10 ***
PCTOWNHOME  -0.56004    0.17031  -3.288  0.00114 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.6571 on 277 degrees of freedom
Multiple R-squared:  0.1932,    Adjusted R-squared:  0.1844
F-statistic:  22.1 on 3 and 277 DF,  p-value: 7.306e-13


> recta2 <- lm(Z ~ PCTAGE65P + PCTOWNHOME, data = NY)
> summary(recta2)

Call:
lm(formula = Z ~ PCTAGE65P + PCTOWNHOME, data = NY)

Residuals:
    Min      1Q  Median      3Q     Max
-1.8000 -0.4238 -0.0409  0.3293  4.0886

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -0.4264     0.1448  -2.946 0.003495 **
PCTAGE65P     4.0142     0.6048   6.637 1.67e-10 ***
PCTOWNHOME   -0.5795     0.1700  -3.409 0.000749 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.6583 on 278 degrees of freedom
Multiple R-squared:  0.1875,    Adjusted R-squared:  0.1817
F-statistic: 32.08 on 2 and 278 DF,  p-value: 2.915e-13

> lm.morantest(recta2,list=pesoszonas)

        Global Moran's I for regression residuals

data:
model: lm(formula = Z ~ PCTAGE65P + PCTOWNHOME, data = NY)
weights: pesoszonas

Moran I statistic standard deviate = 2.3827, p-value = 0.008593              (7)
alternative hypothesis: greater
sample estimates:
Observed Moran's I        Expectation           Variance
       0.077852682       -0.006765406        0.001261209
```

Aunque vemos en (7) que el p-valor ha disminuido, no podemos aceptar la hipótesis nula de que no hay correlación espacial. Por tanto, este modelo tan simple basado en una regresión lineal múltiple, no es válido.
Aunque hay muchas posibilidades para terminar el problema, si ajustamos una Regresión Lineal Ponderada, la cual consiste en minimizar la suma de los errores al cuadrado ponderados por unos pesos w_i, eligiendo aquí como pesos los inversos de los tamaños poblacionales, los cuales vienen incluidos en los datos en

```
NY$POP8
```

podemos ajustar la regresión ponderada ejecutando (8). Las tres covariables son significativas y

```
> recta3<-lm(Z~PEXPOSURE+PCTAGE65P+PCTOWNHOME,data=NY,weights=POP8)        (8)

> summary(recta3)
Call:
lm(formula = Z ~ PEXPOSURE + PCTAGE65P + PCTOWNHOME, data = NY,
    weights = POP8)

Weighted Residuals:
     Min       1Q   Median       3Q      Max
-129.067  -14.714    5.817   25.624   70.723

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.77837    0.14116  -5.514 8.03e-08 ***
PEXPOSURE    0.07626    0.02731   2.792  0.00560 **
PCTAGE65P    3.85656    0.57126   6.751 8.60e-11 ***
PCTOWNHOME  -0.39869    0.15305  -2.605  0.00968 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 33.5 on 277 degrees of freedom
Multiple R-squared:  0.1977,    Adjusted R-squared:  0.189
F-statistic: 22.75 on 3 and 277 DF,  p-value: 3.382e-13

> lm.morantest(recta3,list=pesoszonas)

        Global Moran's I for regression residuals

data:
model:lm(formula=Z~PEXPOSURE+PCTAGE65P+PCTOWNHOME,data=NY,weights=POP8)
weights: pesoszonas

Moran I statistic standard deviate = 0.4773, p-value = 0.3166              (9)
alternative hypothesis: greater
sample estimates:
Observed Moran's I        Expectation           Variance
```

```
       0.007533246       -0.009309771        0.001245248

> recta3

Call:
lm(formula = Z ~ PEXPOSURE + PCTAGE65P + PCTOWNHOME, data = NY,
    weights = POP8)

Coefficients:
(Intercept)    PEXPOSURE    PCTAGE65P   PCTOWNHOME
   -0.77837      0.07626      3.85656     -0.39869                          (10)
```

El p-valor dado en (9) sí confirma la ausencia de dependencia espacial en los residuos con lo que la regresión lineal dada en (10) sí parece un modelo adecuado.

Capítulo 7

Modelos Lineales Generalizados GLM

7.1. Introducción

Modelos Lineales Generalizados es una denominación genérica que engloba algunos métodos ya estudiados anteriormente, tales como la Regresión Lineal Simple (CB-capítulo 9), la Regresión Lineal Múltiple (CB-capítulo 10), la Regresión Logística (TA-capítulo 9) o la Regresión Poisson (TA-capítulo 10), así como otros Métodos de Regresión aún no estudiados y que serán analizados en este capítulo.

La razón de realizar un estudio global de estos métodos es la de obtener, de una sola vez, resultados aplicables a todos ellos. En particular en lo referente a los Métodos Robustos utilizados en dichos modelos. Esta generalización se consigue, en un primer momento, con un mayor nivel de abstracción por lo que el capítulo puede resultar, en ocasiones, demasiado técnico. Si el lector está interesado principalmente por las aplicaciones de estos métodos, encontrará más interesantes la Sección 7.4, si desea un enfoque clásico, y la Sección 7.7.3 cuando busque un análisis robusto.

A continuación aparecen tres ejemplos que serán resueltos en dichas secciones.

Ejemplo 7.1

Consideraremos el experimento realizado por Phelps (1982) en el que se anotó, para cada uno de los $i = 24$ grupos, el número de zanahorias dañadas por insectos de entre todas las del grupo. Las zanahorias fueron plantadas en tres bloques, por lo que al ser ésta una covariable de tipo cualitativo, debieron considerarse en el modelo dos covariables indicadoras, `bloque1` y `bloque2`. Además, se fumigó según ocho dosis de un determinado insecticida, considerándose la covariable cuantitativa `log(dosis)` en el modelo.

Se pretende ajustar a estos datos un Modelo de Regresión Binomial clásico y otro robusto.

Ejemplo 7.2

Feigl y Zelen (1965) analizaron datos de 33 pacientes con leucemia para los que se anotó si su tiempo de supervivencia era superior a 52 semanas (de hecho, ellos anotaron el tiempo de supervivencia y no sólo si era o no mayor a 52 semanas), asociando en ese caso un valor igual a 1, *éxito*, a una variable dependiente Y, valor que tomaría con probabilidad p. Por otro lado, asignaron el valor $Y = 0$ si ese tiempo de supervivencia era inferior o igual a 52 semanas, suceso considerado como *fracaso*, el cual tendría probabilidad $1 - p$.

Como covariables independientes que se piensa pueden explicar a Y se consideraron dos: la covariable WBC, *número de glóbulos blancos por milímetro cúbico de sangre*, (o leucocitos, o en inglés White Blood Cell Count) indicando un valor alto de esta covariable la existencia de infección, y la covariable AG, *presencia* ($AG = 1$) o *ausencia* ($AG = 0$) *de cierta característica morfológica de los glóbulos blancos*. A estos datos se ajustará en Modelo de Regresión Logística clásico y otro robusto.

Ejemplo 7.3

Los artículos de Lindenmayer y sus colaboradores (en la bibliografía damos dos de estos artículos) proporcionan multitud de datos sobre las Montañas Centrales de Victoria en Australia. Aquí trabajaremos con datos sobre diferentes especies de marsupiales arborícolas de Bosques Montano tipo Ash (*Montane Ash Forest*).

En este estudio se analizaron 151 lugares diferentes de 3ha con vegetación uniforme, observándose en cada uno de éstos la variable dependiente de respuesta, número de especies de marsupiales en el lugar (Diversidad), y las covariables siguientes: el número de arbustos (Arbustos); si había, 1, o no, 0, tocones de pasadas operaciones forestales (Tocones) que es una variable cualitativa con dos niveles; el número de árboles de porte hueco (Stags); un índice de cortezas extraídas (Cortezas); un índice de habitabilidad para marsupiales (Habitat); el área de acacias (Acacias); el tipo de Eucalipto que es una variable cualitativa con tres niveles: Eucalipto regnans (Regnans), Eucalipto delegatensis (Delegatensis) y Eucaliptus nitens (Nitens); y, por último, el aspecto del lugar que es una variable de tipo cualitativo con cuatro niveles, (NWNE), (NWSE), (SESW) y (SWNW).

Se pretende ajustar un Modelo de Regresión Poisson a estos datos, clásico y robusto.

Aunque el Modelo de Regresión Lineal Simple o Múltiple es un caso particular de Modelo Lineal General y, por tanto, también puede ser considerado como otro caso más en este capítulo, no lo haremos porque ya en el texto CB lo estudiamos con detalle desde un punto de vista clásico, y en el texto MR y en el capítulo anterior, desde un punto de vista robusto. Eso sí, utilizaremos estos modelos como punto de partida.

7.2. Definición de Modelo Lineal Generalizado univariante

Para definir los Modelos Lineales Generalizados, partiremos del Modelo de Regresión Lineal estudiado en al capítulo anterior. Vimos allí que modelizar nuestros datos con un Modelo de Regresión Lineal Múltiple supone considerar una variable *dependiente* o de *respuesta* Y sobre la que pensamos influyen

linealmente k variables *independientes* o *covariables* $X_1, ..., X_k$ de la forma

$$Y = \beta_0 + \beta_1 X_1 + ... + \beta_k X_k + e \qquad [7.1]$$

siendo e una variable de error con distribución normal $N(0, \sigma)$.

En el Modelo de Regresión Lineal [7.1] se persigue, entre otras cosas, estimar los parámetros $\beta_0, \beta_1, ..., \beta_k$ en base a una muestra aleatoria de tamaño $n(> k+1)$ de las variables independientes y de la dependiente, dando origen a los datos

$$\begin{array}{cccc} y_1 & x_{11} & ... & x_{1k} \\ \vdots & & & \\ y_i & x_{i1} & ... & x_{ik} \\ \vdots & & & \\ y_n & x_{n1} & ... & x_{nk} \end{array}$$

Si denominamos $\mathbf{y} = (y_1, ..., y_n)^t$ al vector de las observaciones de la variable dependiente, englobamos los parámetros en un vector de parámetros $\boldsymbol{\beta} = (\beta_0, ..., \beta_k)^t$, y llamamos *matriz del diseño*

$$\mathbf{X} = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1k} \\ \vdots & & & \vdots \\ 1 & x_{n1} & \cdots & x_{nk} \end{pmatrix}$$

a la matriz $n \times (k+1)$ de las observaciones de las variables independientes, el Modelo de Regresión Lineal se suele expresar de la forma

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{e}$$

en donde $\mathbf{e} = (e_1, ..., e_n)^t$ es el vector de errores.

Con este modelo estamos interesados en estimar los parámetros de $\boldsymbol{\beta}$ en base a los datos

$$(y_i, \mathbf{x}_i^t) = (y_i, 1, x_{i1}, ..., x_{ik}) \qquad , \qquad i = 1, ..., n.$$

En este Modelo de Regresión Lineal, la variable de respuesta Y es de tipo cuantitativo. Las covariables suelen ser de tipo cuantitativo (aunque también podrían considerarse de tipo cualitativo), y pueden ser *determinísticas*, es decir valores conocidos o condiciones experimentales, o pueden ser *estocásticas*, es decir valores de un vector aleatorio.

Si suponemos que las covariables son de tipo determinístico, el modelo lineal [7.1] puede reformularse diciendo que tenemos n observaciones independientes $y_1, ..., y_n$ procedentes de distribuciones $N(\mu_i, \sigma)$ en donde la media μ_i es de la forma

$$\mu_i = \mathbf{x}_i^t \boldsymbol{\beta} = \beta_0 + \beta_1 x_{i1} + ... + \beta_k x_{ik} \qquad i = 1, ..., n.$$

Si, como habitualmente sucede, las covariables se consideran estocásticas, el esquema sería el mismo aunque, ahora, condicional; en concreto, los n pares $(y_i, \mathbf{x}_i^t)$ se suponen observaciones independientes y, dadas las $\mathbf{x}_i$, las Y_i serán (condicionalmente) independientes con distribución

$$\boxed{Y_i|\mathbf{x}_i \rightsquigarrow N(\mu_i, \sigma) \qquad i = 1, ..., n}$$

con

$$\boxed{E[Y_i|\mathbf{x}_i] = \mu_i = \mathbf{x}_i^t \boldsymbol{\beta} \qquad i = 1, ..., n.}$$

En un Modelo Lineal Generalizado (univariante) ampliamos un poco la situación anterior. De nuevo suponemos que, dadas las $\mathbf{x}_i$, las n variables Y_i son (condicionalmente) independientes aunque ahora, la variable de respuesta Y_i puede ser de tipo continuo, puede ser de recuentos de observaciones, o puede ser de tipo binario.

Las dos condiciones antes recuadradas ahora también se generalizan. En este tipo de modelos suponemos que la distribución de las Y_i (condicionada por las $\mathbf{x}_i$) no es necesariamente normal, sino una familia de tipo exponencial con esperanza (condicional) $E[Y_i|\mathbf{x}_i] = \mu_i$ y, posiblemente, con un parámetro de escala (común para todas las Y_i) denominado ξ. Más en concreto, se supone que la distribución de las $Y_i|\mathbf{x}_i$ tiene por función de densidad una *familia de tipo exponencial* (Vélez y García Pérez, 1993, Ejemplo 5.17) de la forma

$$f(y_i|\theta_i, \xi) = \exp\left\{\frac{y_i\,\theta_i - b(\theta_i)}{\xi} + c(y_i, \xi)\right\} \qquad [7.2]$$

en donde θ_i se denomina *parámetro natural*, ξ es el *parámetro de escala* o *dispersión*, y b y c dos funciones que determinan el tipo de familia exponencial.

Además, en un Modelo Lineal Generalizado, la forma en que las covariables suministran información sobre la media μ_i de la variable dependiente ya no es necesariamente lineal mediante el *predictor lineal* $\eta_i = \mathbf{x}_i^t \boldsymbol{\beta}$, sino que lo hacen mediante una función de respuesta h con inversa $h^{-1} = g$, denominada esta última función *link*, es decir, de la forma

$$\mu_i = h(\eta_i) = h(\mathbf{x}_i^t \boldsymbol{\beta}) \qquad i = 1, ..., n$$

o bien,

$$\eta_i = g(\mu_i) = \mathbf{x}_i^t\,\boldsymbol{\beta} \qquad\qquad i = 1, ..., n.$$

Por tanto, un Modelo Lineal Generalizado vendrá especificado cuando fijemos el tipo de familia exponencial para las distribuciones condicionadas $Y_i|\mathbf{x}_i$, la función *link* g y el vector (o matriz) del diseño $\mathbf{x}_i$.

En estas distribuciones de $Y_i|\mathbf{x}_i$, se supone que el parámetro natural es una función w_1 de la media; es decir, $\theta_i = w_1(\mu_i)$ siendo $\mu_i = b'(\theta_i) = \partial b(\theta_i)/\partial\theta_i$.

Además, la varianza en estas distribuciones también es de una forma peculiar, $Var(Y_i|\mathbf{x}_i) = \xi\, w_2(\mu_i)$, en donde la función w_2 también se determina a partir de la función b de la forma $w_2(\mu_i) = b''(\theta_i) = \partial^2 b(\theta_i)/\partial\theta_i^2$. Es decir, suponemos que es $E(Y_i|\mathbf{x}_i) = b'(\theta_i)$ y $Var(Y_i|\mathbf{x}_i) = \xi\, b''(\theta_i)$.

Para cada familia exponencial existe una función *link natural* o *canónica* que es la que iguala al parámetro natural con el predictor lineal; es decir, $\theta_i = w_1(\mu_i) = g(\mu_i) = \eta_i = \mathbf{x}_i^t\,\boldsymbol{\beta}$; es decir, la obtenida a partir de la ecuación

$$g(\mu) \equiv w_1(\mu).$$

Ejemplo 7.4

Si las $Y_i|\mathbf{x}_i$ se distribuyen como normales $N(\mu_i, \sigma)$, su función de densidad será

$$\frac{1}{\sigma\sqrt{2\pi}}\,\exp\left\{-\frac{1}{2\sigma^2}(y_i-\mu_i)^2\right\} = \frac{1}{\sigma\sqrt{2\pi}}\,e^{-y_i^2/(2\sigma^2)}\,\exp\left\{\frac{y_i\mu_i - \mu_i^2/2}{\sigma^2}\right\}.$$

Si comparamos la expresión anterior con [7.2], podemos identificar, observando el término clave (el que involucra a las y_i y las μ_i), que es $\theta_i = \mu_i = w_1(\mu_i)$ (con lo que será $w_1(\mu) = \mu$), $b(\theta_i) = \mu_i^2/2$ y $\xi = \sigma^2$.

El término restante deberá ser $\exp\{c(y_i, \xi)\} = 1/(\sigma\sqrt{2\pi})\,e^{-y_i^2/(2\sigma^2)}$ aunque éste es irrelevante a la hora de identificar los elementos de la distribución modelo.

Como se observa, es $b'(\theta_i) = \partial b(\theta_i)/\partial\theta_i = \mu_i$ y $w_2(\mu_i) = b''(\theta_i) = \partial^2 b(\theta_i)/\partial\theta_i^2 = 1$, con lo que $Var(Y_i|\mathbf{x}_i) = \xi\, w_2(\mu_i) = \xi$.

Finalmente, de la ecuación clave

$$g(\mu) \equiv w_1(\mu) = \mu$$

se deduce que, en el caso de ser f una distribución normal (caso de Regresión Lineal), debe de ser $g(\mu) = \mu$, lo que implica una función *link* canónica igual a la identidad.

En el caso de ser f una distribución Poisson, $\mathcal{P}(\lambda_i)$ la distribución de probabilidad se puede expresar como

$$f(y_i|\theta_i, \xi) = \frac{1}{y_i!}\,\exp\{y_i\,\log\lambda_i - \lambda_i\}$$

con lo que, observando [7.2], deberá ser

$$\theta_i = \log\lambda_i \qquad \text{y} \qquad b(\theta_i) = \lambda_i$$

de la primera de estas igualdades se deduce que debe ser $\lambda_i = e^{\theta_i}$, obteniendo de la segunda, en consecuencia, que es $b(\theta_i) = \lambda_i = e^{\theta_i}$.

Por otro lado, al ser λ_i la media de Y_i, deberá ser $\theta_i = w_1(\mu_i)$, es decir, $\log \lambda_i = w_1(\lambda_i)$, por lo que la función w_1 es $w_1(\lambda) = \log \lambda$. Finalmente, de la ecuación $g(\mu) \equiv w_1(\mu)$ obtenemos $g(\lambda) = \log \lambda$, que indica a la función logaritmo como la función *link* canónica en este tipo de modelos de Regresión Poisson.

En el caso de seguir las $Y_i|\mathbf{x}_i$ una distribución binomial $B(n_i, p_i)$, será

$$f(y_i|\theta_i, \xi) = \binom{n_i}{y_i} p_i^{y_i} (1-p_i)^{n_i - y_i} = \binom{n_i}{y_i} \exp\left\{ y_i \log \frac{p_i}{1-p_i} + n_i \log(1-p_i) \right\}$$

con lo que, observando [7.2], deberá ser

$$\theta_i = \log \frac{p_i}{1-p_i} \qquad , \qquad b(\theta_i) = -n_i \log(1-p_i) \qquad \text{y} \qquad \xi = 1.$$

Como la media de la distribución binomial, $B(n_i, p_i)$, es $\mu_i = n_i\, p_i$, de la ecuación $\theta_i = w_1(\mu_i)$ obtenemos

$$w_1(\mu_i) = w_1(n_i\, p_i) = \log \frac{p_i}{1-p_i} = \log \frac{n_i\, p_i}{n_i - n_i\, p_i} = \log \frac{\mu_i}{n_i - \mu_i}$$

y, finalmente, de la ecuación $g(\mu) \equiv w_1(\mu)$, la función *link* canónica $g(\mu) = \log(\mu/(n-\mu))$. Por tanto, la ecuación que relaciona la media de la variable de respuesta con las covariables $g(\mu_i) = \mathbf{x}_i^t \boldsymbol{\beta}$, será

$$\log\left(\frac{\mu_i}{n_i - \mu_i}\right) = \log\left(\frac{n_i p_i}{n_i - n_i p_i}\right) = \log\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_1 X_1 + ... + \beta_k X_k.$$

Observemos que, en el caso de que la variable respuesta sea Bernoulli, $Y_i|\mathbf{x}_i \rightsquigarrow B(1, p_i)$ en donde ésta sólo toma los valores *éxito* y *fracaso*, tendremos un caso particular del anterior (correspondiente a la Regresión Logística) en donde la función *link* será $g(\mu) = \log(\mu/(1-\mu))$ o lo que es lo mismo, $g(p) = \log(p/(1-p))$ por ser para esta distribución $\mu = p$. La ecuación que relaciona la media de la variable de respuesta con las covariables es, en este caso, la misma de antes,

$$\log\left(\frac{\mu_i}{1-\mu_i}\right) = \log\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_1 X_1 + ... + \beta_k X_k$$

por lo que no se suele hacer distinción entre estos dos últimos casos y se habla de la función *link* canónica $g(\mu) = \log(\mu/(1-\mu))$, denominada *logit*.

En resumen, prescindiendo de la nomenclatura dada a la variable de la función considerada, hemos obtenido tres funciones *link*, la función *link identidad*, $g(\mu) = \mu$, la función *link logaritmo* o simplemente *log*, $g(\mu) = \log \mu$ y la función *link logit*, $g(\mu) = \log(\mu/(1-\mu))$, funciones *link* naturales o canónicas de los modelos, respectivamente, normal, Poisson y binomial (Bernoulli).

Se utilizan también otras funciones *link*, la función *link inversa*, $g(\mu) = -1/\mu$ y la función *link gaussiana-inversa*, $g(\mu) = -2/\mu^2$, funciones *link* canónicas de los modelos, respectivamente, gamma y gaussiano-inverso.

Otras funciones *link* no canónicas, pero que se pueden utilizar en algún modelo son, la función *link probit*, $g(\mu) = \Phi^{-1}(\mu)$, es decir, la inversa de la función de distribución de una normal estándar $N(0,1)$, la función *link*

complementaria log-log, $g(\mu) = log(-log(1-\mu))$ y la función *link raíz cuadrada*, $g(\mu) = \sqrt{\mu}$.

Con el paquete R podemos trabajar con los cinco modelos antes mencionados, formando la Tabla 7.1 en la que aparece una **C** indicando la función *link* canónica. Las opciones marcadas con una **p** indican que también pueden elegirse como funciones *link*, pero que no son las canónicas.

	Modelos				
	Normal	Poisson	Binomial	Gamma	Gaussiano-inverso
Funciones *link*					
identidad	**C**	**p**	–	**p**	–
logaritmo	–	**C**	–	**p**	–
logit	–	–	**C**	–	–
inversa	–	–	–	**C**	–
gaussiana-inversa	–	–	–	–	**C**
probit	–	–	**p**	–	–
complementaria log-log	–	–	**p**	–	–
raíz cuadrada	–	**p**	–	–	–

Tabla 7.1: Modelos y funciones *link*

7.2.1. Dispersión excesiva (*Overdispersion*)

Supongamos que queremos modelizar nuestros datos mediante un Modelo de Regresión Logístico. En este caso, la distribución asociada a las Y_i en el Modelo Lineal Generalizado sería la Bernoulli $B(1, p)$, con media p y varianza $p(1-p)$.

Si quisiéramos modelizar los datos con un Modelo de Regresión Poisson, la distribución sería Poisson, $\mathcal{P}(\lambda)$, de media λ y varianza λ.

Supongamos ahora que, al observar nuestros datos, vemos que, en uno u otro caso, su varianza es mayor de la que debería ser. En estos casos, modelizaremos los datos, para la primera situación, con un Modelo de Regresión Logística, de varianza $\varsigma\, p(1-p)$ y, en el segundo caso, mediante un Modelo de Regresión Poisson, pero con varianza $\varsigma\lambda$.

En estas situaciones decimos que nuestros datos presentan una *dispersión excesiva* (*overdispersion*), con parámetro de *overdispersion* ς, problema que trataremos más adelante al hablar de cada uno de los dos modelos.

7.3. Estimación y Contrastes basados en la verosimilitud

La estimación de los parámetros del Modelo Lineal Generalizado (así como contrastes de hipótesis referentes a éstos), además de dos tests de bondad del ajuste, se pueden realizar siguiendo métodos basados en la *verosimilitud*. En posteriores secciones estudiaremos Métodos basados en la cuasi-verosimilitud y Métodos Bayesianos.

7.3.1. Estimador de máxima verosimilitud de los β_i

En esta sección determinaremos la forma en la que estimar los parámetros β_i del modelo; es posible que los diferentes parámetros y funciones que intervienen en el Modelo Lineal Generalizado puedan entorpecer la comprensión del proceso, pero hemos querido desgranar éste puesto que la ecuación de verosimilitud resultante (en realidad, sistema de ecuaciones) es clave en las posteriores generalizaciones y robustificación.

La manera en la que habitualmente estimamos los parámetros de un modelo es mediante la utilización del Método de la Máxima Verosimilitud (CB-sección 5.2). Para ello, primero debemos expresar la función de verosimilitud como función del parámetro. Si observamos [7.2] los parámetros del modelo serán θ_i y ξ; de momento supondremos ξ conocido (aunque más abajo volveremos sobre ello). La función de verosimilitud será, por tanto,

$$L(\theta_1, ..., \theta_n) = \prod_{i=1}^{n} f(y_i|\theta_i) = \exp\left\{\sum_{i=1}^{n}\left(\frac{y_i\theta_i - b(\theta_i)}{\xi} - c(y_i, \xi)\right)\right\}.$$

El Método de la Máxima Verosimilitud indica asignar como estimadores de los parámetros a aquellos valores que hagan máxima dicha función de verosimilitud. Como el máximo de una función y de su logaritmo se alcanzan en el mismo punto, determinaremos el máximo del logaritmo de $L(\theta_1, ..., \theta_n)$,

$$\log L(\theta_1, ..., \theta_n) = \sum_{i=1}^{n}\left(\frac{y_i\theta_i - b(\theta_i)}{\xi}\right) - \sum_{i=1}^{n} c(y_i, \xi).$$

Como suponemos ξ conocido y vamos a maximizar esta función derivando respecto al parámetro e igualando a cero la derivada, el segundo sumando de la expresión anterior se anulará por lo que prescindiremos de él en lo que sigue considerándolo, simplemente, como una constante, cte.

Si reparametrizamos la función anterior (es decir, cambiamos los parámetros), al ser $\theta_i = w_1(\mu_i)$ tendremos, (la última igualdad es sólo notación)

$$\log L(\mu_1, ..., \mu_n) = \sum_{i=1}^{n} \left(\frac{y_i w_1(\mu_i) - b(w_1(\mu_i))}{\xi} \right) + \text{cte} = \sum_{i=1}^{n} l_i(\mu_i) + \text{cte} \quad [7.3]$$

y si volvemos a reparametrizar, expresando la verosimilitud anterior en términos de las β_i y las covariables, por ser $\mu_i = h(\mathbf{x}_i^t \boldsymbol{\beta})$ tendremos

$$\log L(\boldsymbol{\beta}) = \sum_{i=1}^{n} \left(\frac{y_i \, w_1(h(\mathbf{x}_i^t \boldsymbol{\beta})) - b(w_1(h(\mathbf{x}_i^t \boldsymbol{\beta}))}{\xi} \right) + \text{cte.} \quad [7.4]$$

La derivada de esta expresión la debemos obtener teniendo en cuenta las funciones que aparecen en ella y la denominación que hemos dado a sus variables.

Conviene recordar también que, como $\boldsymbol{\beta}$ es un vector, al hablar de la derivada de $\log L(\boldsymbol{\beta})$ con respecto a $\boldsymbol{\beta} = (\beta_0, \beta_1, ..., \beta_k)^t$, la cual representamos por $\partial \log L(\boldsymbol{\beta})/\partial \boldsymbol{\beta}$, nos referimos al vector de derivadas parciales

$$(\partial \log L(\boldsymbol{\beta})/\partial \beta_0, ..., \partial \log L(\boldsymbol{\beta})/\partial \beta_k)^t$$

el cual igualaremos al vector de ceros, dando origen a un sistema de ecuaciones de verosimilitud, de $k+1$ ecuaciones con $k+1$ incógnitas, $\beta_0, \beta_1, ..., \beta_k$.

Observamos también que derivar [7.4] respecto a $\boldsymbol{\beta}$ va a consistir, básicamente, en aplicar reiteradamente la derivada de una función de función por lo que expresaremos cada una de las funciones de la composición con respecto a su variable; además, como el mismo lector puede comprobar fácilmente, es

$$\frac{\partial \mathbf{x}_i^t \boldsymbol{\beta}}{\partial \boldsymbol{\beta}} = \mathbf{x}_i$$

Derivando en [7.4] será

$$\frac{\partial \log L(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = \frac{1}{\xi} \sum_{i=1}^{n} \left[y_i \cdot \left(\frac{\partial w_1(\mu_i)}{\partial \mu_i} \Big|_{\mu_i = h(\mathbf{x}_i^t \boldsymbol{\beta})} \right) \cdot \boldsymbol{\mu_i}' - b'(w_1(\mu_i)) \cdot \left(\frac{\partial w_1(\mu_i)}{\partial \mu_i} \Big|_{\mu_i = h(\mathbf{x}_i^t \boldsymbol{\beta})} \right) \cdot \boldsymbol{\mu_i}' \right]$$

$$= \frac{1}{\xi} \sum_{i=1}^{n} \left(\frac{\partial w_1(\mu_i)}{\partial \mu_i} \Big|_{\mu_i = h(\mathbf{x}_i^t \boldsymbol{\beta})} \right) \boldsymbol{\mu_i}' \, (y_i - \mu_i(\boldsymbol{\beta}))$$

por ser $b'(w_1(\mu_i)) = \mu_i(\boldsymbol{\beta}) = \mu_i$, y siendo

$$\boldsymbol{\mu_i}' = \frac{\partial \mu_i}{\partial \boldsymbol{\beta}} = \left(\frac{\partial h(\eta)}{\partial \eta} \Big|_{\eta = h(\mathbf{x}_i^t \boldsymbol{\beta})} \right) \cdot \frac{\partial \mathbf{x}_i^t \boldsymbol{\beta}}{\partial \boldsymbol{\beta}} = \left(\frac{\partial h(\eta)}{\partial \eta} \Big|_{\eta = h(\mathbf{x}_i^t \boldsymbol{\beta})} \right) \cdot \mathbf{x}_i = D_i(\boldsymbol{\beta}) \, \mathbf{x}_i$$

en donde la última igualdad sólo se ha introducido como notación para definir $D_i(\boldsymbol{\beta})$.

Como es $\mu_i = b'(\theta_i)$ será $\theta_i = (b')^{-1}(\mu_i)$ y, como era $\theta_i = w_1(\mu_i)$, será $w_1(\mu_i) = (b')^{-1}(\mu_i)$ por lo que, utilizando la fórmula para la derivada de la función inversa, será

$$\frac{\partial w_1(\mu_i)}{\partial \mu_i} = \frac{\partial (b')^{-1}(\mu_i)}{\partial \mu_i} = \frac{1}{b''((b')^{-1}(\mu_i))} = \frac{1}{b''(\theta_i)} = \frac{1}{w_2(\mu_i)} = \frac{\xi}{Var(Y_i|\mathbf{x}_i)}.$$

Por tanto, la derivada buscada se podrá expresar en cualquiera de las siguientes dos maneras,

$$\frac{\partial \log L(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = \sum_{i=1}^{n} \frac{z_i\, D_i(\boldsymbol{\beta})}{Var(Y_i|\mathbf{x}_i)}\ (y_i - \mu_i(\boldsymbol{\beta})) = \sum_{i=1}^{n} \frac{\boldsymbol{\mu_i}'}{\xi\, w_2(\mu_i)}\ (y_i - \mu_i)$$

como aparece, respectivamente, en Fahrmeir y Tutz (1994, pp. 38) o en Cantoni y Ronchetti (2001, pp. 1022).

El sistema de ecuaciones de verosimilitud

$$\frac{\partial \log L(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = \sum_{i=1}^{n} \frac{\boldsymbol{\mu_i}'}{\xi\, w_2(\mu_i)}\ (y_i - \mu_i) = \mathbf{0} \qquad [7.5]$$

no va a tener habitualmente una solución analítica y debe de resolverse de forma numérica mediante un método iterativo. R utiliza el más habitual, el de mínimos cuadrados ponderados IWLS (*iteratively reweighted least squares*), también denominado de las marcas de Fisher (*Fisher scoring*). Otras alternativas son el Método de Newton-Raphson o, mejor, los Métodos Quasi-Newton.

El estimador de máxima verosimilitud $\widehat{\boldsymbol{\beta}}$ obtenido mediante alguno de los métodos anteriores, cuando exista y sea único, tendrá una distribución asintótica normal multivariante,

$$\widehat{\boldsymbol{\beta}} \rightsquigarrow N(\boldsymbol{\beta}, V)$$

siendo la matriz de covarianzas V aproximadamente igual a la inversa de la matriz de información de Fisher

$$V \approx A^{-1}(\widehat{\boldsymbol{\beta}})$$

siendo dicha matriz de información igual a

$$A(\widehat{\boldsymbol{\beta}}) = \sum_{i=1}^{n} \mathbf{x}_i\, \mathbf{x}_i^t\, D_i^2(\widehat{\boldsymbol{\beta}})\, \frac{1}{w_2(h(\mathbf{x}_i^t\, \widehat{\boldsymbol{\beta}}))\ \xi}$$

7.3.2. Estimador del parámetro de escala ξ

Si el parámetro de escala ξ no fuese conocido podría estimarse, a partir del estimador $\widehat{\boldsymbol{\beta}}$, por la expresión,

$$\widehat{\xi} = \frac{1}{n-(k+1)} \sum_{i=1}^{n} \frac{(y_i - \widehat{\mu}_i)^2}{w_2(\widehat{\mu}_i)} \qquad [7.6]$$

en donde $\widehat{\mu}_i = h(\mathbf{x}_i^t \widehat{\boldsymbol{\beta}})$, obteniéndose de esta manera un estimador consistente para ξ, el cual puede utilizarse en la expresión de $A^{-1}(\widehat{\boldsymbol{\beta}})$.

Obsérvese que, en un Modelo de Regresión Normal, el estimador anterior del parámetro de escala coincide con el obtenido para la varianza σ^2 mediante la suma de residuos al cuadrado.

7.3.3. Contrastes de hipótesis sobre los parámetros

Una vez obtenidos los estimadores para los β_i, podemos considerar el realizar tests de hipótesis sobre ellos de la forma $H_0 : \mathbf{C}\boldsymbol{\beta} = \mathbf{c_0}$ frente a la alternativa $H_1 : \mathbf{C}\boldsymbol{\beta} \neq \mathbf{c_0}$. (En esta sección supondremos que el parámetro de escala ξ es conocido o reemplazado por el valor [7.6].)

Un caso particular de estas hipótesis, muy importante, es el contraste de $H_0 : \boldsymbol{\beta_r} = \mathbf{0}$ frente a $H_0 : \boldsymbol{\beta_r} \neq \mathbf{0}$ siendo $\boldsymbol{\beta_r}$ un subvector de $\boldsymbol{\beta}$; es decir, el contraste de ser cero algunas β_i frente a la alternativa de modelo completo, en el que todas las β_i son distintas de cero.

Se consideran tres tipos de tests de hipótesis. El primero es el *test de razón de verosimilitudes* (Vélez y García Pérez, 1993, Sección 9.2) basado en el estadístico de contraste

$$\Lambda = \frac{\sup_{\boldsymbol{\beta}\in\Theta_0} L(\boldsymbol{\beta})}{\sup_{\boldsymbol{\beta}\in\Theta} L(\boldsymbol{\beta})} = \frac{L(\widetilde{\boldsymbol{\beta}})}{L(\widehat{\boldsymbol{\beta}})}$$

siendo Θ el espacio paramétrico y Θ_0 la parte de este espacio definido por la hipótesis nula; es decir, el cociente entre el máximo de la función de verosimilitud $L(\boldsymbol{\beta})$ alcanzado cuando las variables $\boldsymbol{\beta}$ varían en la región definida por la hipótesis nula, $L(\widetilde{\boldsymbol{\beta}})$, y el máximo alcanzado por esta función cuando los parámetros toman cualquier valor posible, $L(\widehat{\boldsymbol{\beta}})$, por la definición de estimador de máxima verosimilitud.

Como todo test de hipótesis, éste requiere para su ejecución de la distribución del estadístico de contraste bajo la hipótesis nula. Aunque la distribución exacta no es fácilmente calculable, no obstante, sí se sabe (Vélez y García Pérez, 1993, Teorema 9.1) que, para tamaños muestrales suficientemente grandes, se tiene aproximadamente una distribución χ^2

$$-2\log\Lambda = -2\left[\log L(\widetilde{\boldsymbol{\beta}}) - \log L(\widehat{\boldsymbol{\beta}})\right] = 2\left[\log L(\widehat{\boldsymbol{\beta}}) - \log L(\widetilde{\boldsymbol{\beta}})\right] \rightsquigarrow \chi^2_{k+1-q}$$

siendo q la dimensión del espacio paramétrico bajo la hipótesis nula. Por ejemplo, si la hipótesis nula fuera que uno sólo de los β_i fuera cero, la dimensión del espacio paramétrico sería k ya que H_0 sólo fija una restricción (que sea $\beta_i = 0$), por lo que deja libres de tomar cualquier valor a los otros k parámetros. En este caso, los grados de libertad de la χ^2 con los que buscar puntos críticos y calcular p-valores serían $k + 1 - q = k + 1 - k = 1$.

Otro test de hipótesis muy utilizado es el *test de Wald* basado en el estadístico de contraste

$$Wald = \left(\mathbf{C}\widehat{\boldsymbol{\beta}} - \mathbf{c_0}\right)^t \left[\mathbf{C}A^{-1}(\widehat{\boldsymbol{\beta}})\mathbf{C}^t\right]^{-1} \left(\mathbf{C}\widehat{\boldsymbol{\beta}} - \mathbf{c_0}\right)$$

siendo $A^{-1}(\widehat{\boldsymbol{\beta}})$ la inversa de la matriz de información de Fisher definida más arriba.

Por último, si llamamos función *score* a la función

$$s(\boldsymbol{\beta}) = \frac{\partial \log L(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}}$$

el tercer test de hipótesis considerado es el *test score* basado en el estadístico

$$score = s(\widetilde{\boldsymbol{\beta}})^t A^{-1}(\widetilde{\boldsymbol{\beta}}) s(\widetilde{\boldsymbol{\beta}}).$$

Estos dos últimos estadísticos de contraste también tienen, bajo la hipótesis nula, la misma distribución asintótica χ^2_{k+1-q} que tenía el estadístico de razón de verosimilitudes. Mientras que cualquiera de los tres tests es aceptable para modelos sin *overdispersion*, es muy recomendable utilizar estos dos últimos cuando ésta está presente.

Sobre esta cuestión de selección de modelos, otra posibilidad es determinar, como en el capítulo anterior, el valor del Criterio de Información de Akaike AIC y, entre varios posibles modelos, elegir el que proporcione un menor valor AIC.

7.3.4. Contraste de bondad de ajuste del modelo

Como es habitual, los dos estadísticos utilizados para contrastar la hipótesis nula de adecuarse correctamente nuestros datos a un modelo concreto, son el *Estadístico de Pearson*

$$\lambda = \sum_{i=1}^{n} \frac{(y_i - \widehat{\mu}_i)^2}{\xi\, w_2(\widehat{\mu}_i)}$$

en donde, como más arriba, es $\widehat{\mu_i} = h(\mathbf{x}_i^t \widehat{\boldsymbol{\beta}})$, la media estimada, y $\xi\, w_2(\widehat{\mu_i})$ la varianza estimada, y el *estadístico desviación* (*deviance*, o con más precisión, *residual deviance*)

$$G^2 = -2\sum_{i=1}^{n} [l_i(\widehat{\mu_i}) - l_i(y_i)] = \sum_{i=1}^{n} d_i$$

donde de nuevo aparece la media estimada $\widehat{\mu_i}$ y las contribuciones l_i de cada uno de los valores muestrales al logaritmo de la verosimilitud, definidas en [7.3] y que es equivalente en los GLM a la suma residual de cuadrados en regresión lineal.

Ambos estadísticos siguen, aproximadamente, una distribución $\chi^2_{n-(k+1)}$.

7.3.5. Diagnóstico del Modelo

Al igual que en la regresión lineal debíamos de comprobar que

$$y_i|\mathbf{x}_i \rightsquigarrow N(\mu_i, \sigma) \qquad\qquad i = 1, ..., n$$

aquí, estas distribuciones condicionadas por los $x_1, ...x_n$ deberían de ser Poisson, etc. Pero al igual que allí pasaba, aquí tampoco lo podremos comprobar porque tenemos pocas observaciones y_i para $\mathbf{x}_i$ concretos; por esta razón, debemos analizar en su lugar, los residuos. En concreto, el modelo ajustado debe de cumplir que los n *Residuos de Pearson*

$$r_i^p = \frac{(y_i - \widehat{\mu_i})^2}{\xi\, w_2(\widehat{\mu_i})}$$

llamados así por ser los sumandos del Estadístico de Pearson definido en la Sección 7.3.4, deben de tener, aproximadamente, media cero y varianza ξ. Es decir, no deben de mostrar ninguna tendencia, ni en media ni en varianza, cuando se representan frente a los valores ajustados o frente a cualquier covariable.

En la práctica, la distribución de los residuos de Pearson suele ser asimétrica y no tiene la misma interpretación que sus análogos en el correspondiente del modelo de regresión lineal múltiple. Por esta razón, se prefiere determinar los n residuos *deviance*, que son los n sumandos d_i del estadístico con este nombre (con su signo), ya que éstos sí juegan el mismo papel que la suma residual de cuadrados juega en el modelo lineal; de hecho, en el modelo lineal ordinario, la *deviance* es la suma residual de cuadrados y se calcula como la suma de los residuos al cuadrado. En definitiva se determinan los n residuos *deviance*

$$r_i^d = signo\,(y_i - \widehat{\mu_i})\,\sqrt{d_i}$$

para los cuales se admite, si el ajuste del correspondiente GLM es adecuado, una distribución normal $N(0,1)$.

7.4. Cálculo con R

Con R se pueden estimar los parámetros en un Modelo de Regresión Lineal Generalizado mediante la función

```
glm(modelo,family,data)
```

en donde el argumento `modelo` debe indicar el modelo lineal que queremos contrastar, expresado mediante variables indicadoras para aquellas variables que sean de tipo cualitativo.

En el caso de datos binomiales, los de la variable respuesta aparecen habitualmente en forma de matriz de dos columnas en donde entenderemos que la primera se corresponde con el número de éxitos y la segunda columna con el de fracasos (ver el ejemplo de más abajo).

En el argumento `family` debemos indicar la familia que utilizaremos en la construcción del modelo lineal de entre las cinco que aparecen en la Tabla 7.1, así como la función `link` si no es la canónica; por ejemplo, en el caso de un modelo de Regresión Logística, en este segundo argumento, teclearemos el comando `family=binomial` o, equivalentemente, teclearíamos el comando `family=binomial(link=logit)` ya que ésta es la función *link* canónica correspondiente a esta familia.

Los datos, incluidos en el tercer argumento `data`, deben venir en modo *estructura de datos* (*data frame*).

El análisis de la significación de los coeficientes de regresión estimados se hará, como en el caso de la Regresión Lineal, con la función `summary`.

7.4.1. Regresión Logística y Regresión Binomial

Dado lo habitual que es la utilización de este tipo de modelos, hemos preferido extraerlo en un apartado independiente y que complementa lo estudiado en TA-capítulo 9.

Es razonable pensar en ajustar un Modelo de Regresión Logística cuando, en los datos observados, la variable de respuesta es dicotómica del tipo *éxito-fracaso*, o es proporción de *éxitos* de entre un número determinado de pruebas, situación esta última que se suele denominar Regresión Binomial.

Primero resolvamos, desde un punto de vista clásico, los dos primeros ejemplos considerados en la Introducción. El análisis robusto de éstos se verá al final del capítulo.

Ejemplo 7.1 (continuación)

Los datos del experimento de Phelps (1982) vienen recogidos en el fichero de datos `zanaho`, que aparece en la página web de datos del libro.
El objetivo que se persigue es ajustar un Modelo Lineal Generalizado (en esta sección, clásico) para datos binomiales $B(n_i, p_i)$ (con lo que es $\mu_i = n_i\, p_i$), de la forma

$$\log\left(\frac{\mu_i}{n_i - \mu_i}\right) = \log\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_1\ \log(\texttt{dosis}) + \beta_2\, \text{bloque2} + \beta_3\, \text{bloque1}$$

Como los datos a utilizar deben de estar en forma de *estructura de datos*, ejecutamos `(1)` para incluirlos en R con ese formato al utilizar la función `read.table`. A continuación lo comprobamos.

```
> zanahorias<-read.table("d:\\datos\\zanaho",header=T)                    (1)

> zanahorias
   dañadas total logdosis bloque bloque1 bloque2
1       10    35     1.52      1       1       0
2       16    42     1.64      1       1       0
.............................................
23       3    22     2.24      3       0       0
24       2    31     2.36      3       0       0
```

Al trabajar con datos binomiales, como dijimos más arriba, la variable de respuesta debe estar formada por una matriz en la que la primera columna sea los *éxitos* y la segunda columna los *fracasos* (=al número de pruebas-éxitos). Los datos de esta variable respuesta (que hemos denominado `respuesta`) la obtenemos en `(2)` utilizando la función de R, `cbind`, que *pega* columnas. A continuación comprobamos que lo ha hecho bien.

```
> respuesta<-cbind(zanahorias[,1],zanahorias[,2]-zanahorias[,1])          (2)
> respuesta
      [,1] [,2]
 [1,]   10   25
 [2,]   16   26
.................
[23,]    3   19
[24,]    2   29
```

Ahora ya podemos utilizar la función `glm` en `(3)`, apareciendo los resultados en `(4)`, los cuales valoramos ejecutando `(5)`.

```
> resultado<-glm(respuesta~logdosis+bloque2+bloque1,                      (3)
+ family=binomial,data=zanahorias)

> resultado                                                               (4)

Call:  glm(formula = respuesta ~ logdosis + bloque2 + bloque1,
       family = binomial, data = zanahorias)

Coefficients:
(Intercept)     logdosis      bloque2      bloque1
     1.4802      -1.8174       0.8433       0.5424
```

```
Degrees of Freedom: 23 Total (i.e. Null);  20 Residual
Null Deviance:     83.34
Residual Deviance: 39.98        AIC: 128.6

> summary(resultado)                                                    (5)

Call:
glm(formula = respuesta ~  logdosis + bloque2 + bloque1,
    family = binomial, data = zanahorias)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.9200  -1.0215  -0.3239   1.0602   3.4324

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   1.4802     0.6554   2.258 0.023918 *
logdosis     -1.8174     0.3434  -5.293 1.20e-07 ***                    (8)
bloque2       0.8433     0.2257   3.736 0.000187 ***
bloque1       0.5424     0.2315   2.343 0.019118 *

                (6)        (7)
---
Signif. codes:  0  ‘***’  0.001  ‘**’  0.01  ‘*’  0.05  ‘.’  0.1  ‘’  1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 83.344  on 23  degrees of freedom
Residual deviance: 39.976  on 20  degrees of freedom
                     (9)
AIC: 128.61

Number of Fisher Scoring iterations: 3
```

Los estimadores de los coeficientes aparecen en (6), sus errores estándar en (7) (iguales a los que aparecen en la columna izquierda de la Tabla 1 del artículo de Cantoni y Ronchetti, 2001) y los p-valores de los contrastes de la hipótesis nula de ser éstos cero, indican en (8) que son significativas las tres covariables independientes consideradas, quedando como modelo ajustado el siguiente,

$$\log\left(\frac{\mu_i}{n_i-\mu_i}\right) = 1'4802 - 1'8174\ \log(\texttt{dosis}) + 0'8433\,\text{bloque2} + 0'5424\,\text{bloque1}$$

El valor del estadístico *deviance* que aparece en (9), igual a $G^2 = 39'976$, se utiliza en el contraste de la hipótesis nula de adecuarse correctamente el modelo anterior a los datos observados y que corresponde a una $\chi^2_{n-(k+1)} = \chi^2_{24-4} = \chi^2_{20}$. El p-valor de este test será, por tanto,

```
> 1-pchisq(39.976,20)
[1] 0.005030426
```

indicando, de forma sorprendente, que debe rechazarse la bondad del ajuste del modelo obtenido cuando los contrastes individuales para los parámetros β_i indicaban que las covariables sí explicaban a la variable respuesta.

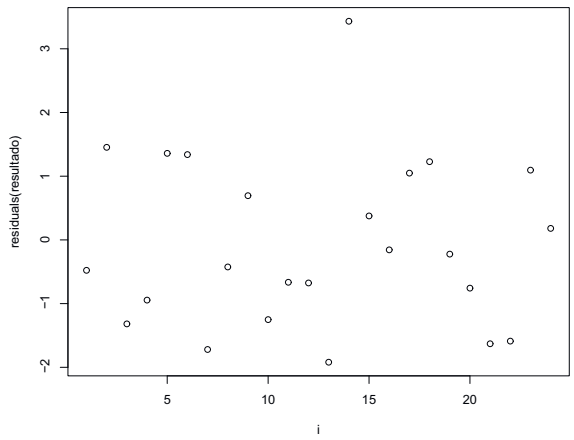

Figura 7.1 : Gráfico de los Residuos

Si representamos los residuos del modelo ajustado en la Figura 7.1 mediante la siguiente secuencia,

```
> i<-seq(1,24)
> plot(i,residuals(resultado))
```

observamos que la observación número 14 (y quizás la 13) es un outlier. Es más conveniente, por tanto, utilizar métodos robustos como veremos más adelante.
Los habituales cuatro gráficos diagnósticos se pueden obtener ejecutando

```
> par(mfrow=c(2,2))
> plot(resultado)
```

obteniendo la Figura 7.2 pero aquí, los residuos r_i se sustituyen por los residuos *deviance* r_i^d antes definidos.
En esta Figura 7.2 se observa que los residuos deviance del dato 14, y en menor medida el 13, pueden ser considerados outliers.

Ejemplo 7.2 (continuación)
Para los datos de Feigl y Zelen (1965) se pretende ajustar un Modelo de Regresión Logística (clásico en esta sección) de la forma

$$\log \frac{p}{1-p} = \beta_0 + \beta_1 \, WBC + \beta_2 \, AG.$$

Los datos observados aparecen en el fichero de datos `leucemia`, proporcionado en la página web del libro. (Los valores de WBC del fichero fueron divididos por 10^4 con lo que habrá que multiplicarlos por esta cantidad en la fórmula del modelo ajustado.)
Como los datos a utilizar deben de estar en forma de *estructura de datos*, ejecutamos `(1)` para incluirlos en R con ese formato al utilizar la función `read.table`. A continuación lo comprobamos.

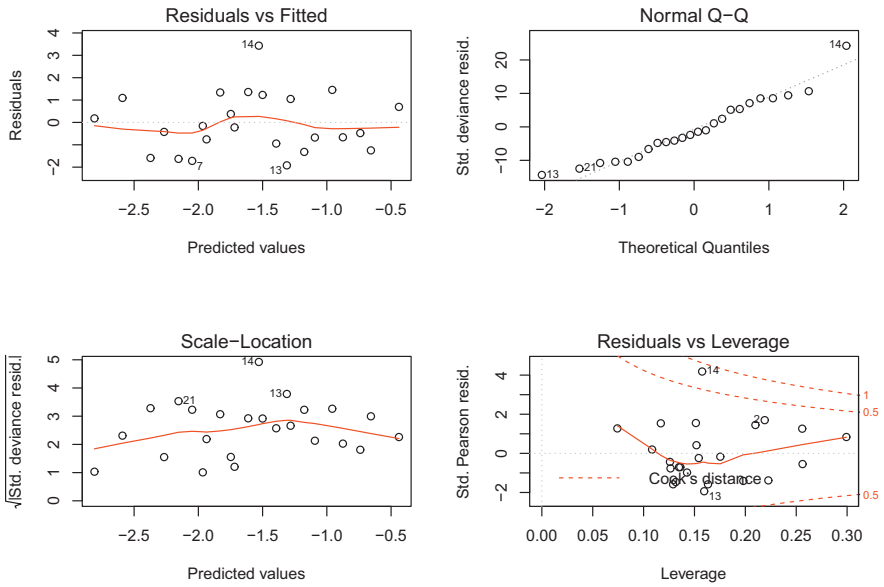

Figura 7.2 : Gráficos diagnósticos del Ejemplo 7.1

```
> leucemia<-read.table("d:\\datos\\leucemia",header=T)                    (1)

> leucemia
   Super    WBC AG
1      1  0.230  1
2      1  0.075  1
3      1  0.430  1
.................................................
32     0 10.000  0
33     0 10.000  0
```

Ahora, en (2), utilizamos la función `glm` apareciendo los resultados en (3), los cuales valoramos ejecutando (4).

```
> solu<-glm(Super~WBC+AG,family=binomial,data=leucemia)                   (2)
> solu                                                                    (3)

Call:  glm(formula = Super ~ WBC + AG, family = binomial, data = leucemia)

Coefficients:
(Intercept)          WBC           AG
    -1.3073      -0.3177       2.2611

Degrees of Freedom: 32 Total (i.e. Null);  30 Residual
Null Deviance:      42.01
```

```
Residual Deviance: 31.06        AIC: 37.06


> summary(solu)                                                    (4)


Call:
glm(formula = Super ~ WBC + AG, family = binomial, data = leucemia)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.5224  -0.6417  -0.4534   0.8362   2.1570

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -1.3073     0.8145  -1.605   0.1085
WBC          -0.3177     0.1863  -1.705   0.0881 .                 (7)
AG            2.2611     0.9522   2.375   0.0176 *
                (5)        (6)
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 42.010  on 32  degrees of freedom
Residual deviance: 31.062  on 30  degrees of freedom
                   (8)
AIC: 37.062

Number of Fisher Scoring iterations: 5
```

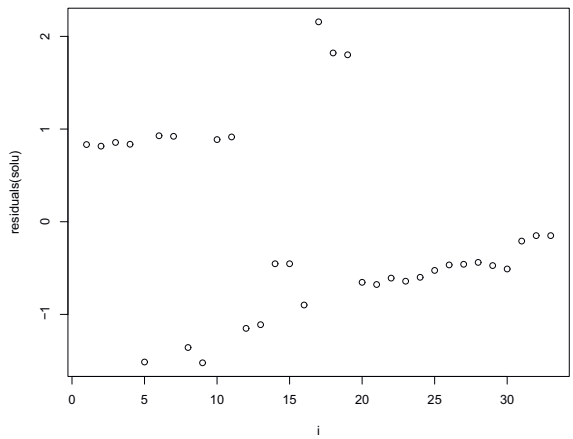

Figura 7.3 : Gráfico de los Residuos

Los estimadores de los coeficientes aparecen en `(5)`, sus errores estándar en `(6)` (iguales a los que aparecen en la Tabla 7.1 del texto de Maronna, Martin y Yohai, 2006, pp. 237) y los p-valores de los contrastes de la hipótesis nula de ser éstos cero, parecen indicar en `(7)` que no son significativas (es decir, que no se deberían de aceptar) las dos covariables independientes consideradas (con dudas podría serlo AG). Si se aceptaran ambas, quedaría como modelo ajustado el siguiente,

$$\log \frac{p}{1-p} = -1'3074 - 0'3177\, WBC(\times 10000) + 2'2611\, AG.$$

El valor del estadístico *deviance* que aparece en `(8)`, igual a $G^2 = 31'062$, se utiliza en el contraste de la hipótesis nula de adecuarse correctamente el modelo anterior a los datos observados y que corresponde a una $\chi^2_{n-(k+1)} = \chi^2_{33-3} = \chi^2_{30}$. El p-valor de este test será, por tanto,

```
> 1-pchisq(31.062,30)
[1] 0.4123636
```

indicando que debe aceptarse, por contra, la bondad del ajuste del modelo obtenido.
Si representamos los residuos del modelo ajustado en la Figura 7.3 mediante la siguiente secuencia,

```
> i<-seq(1,33)
> plot(i,residuals(solu))
```

observamos que el dato número 17 es una observación influyente (un outlier). De hecho corresponde a un individuo con cien mil glóbulos blancos (lo que parece indicar que existe infección), pero que sorprendentemente sobrevivió más de 52 semanas. Las observaciones 18 y 19 son también un tanto atípicas puesto que son individuos que han sobrevivido mucho tiempo y tienen un valor $AG = 0$.
Veremos al final del capítulo qué ocurre con este ejemplo utilizando métodos robustos.

Interpretación de los coeficientes del Modelo de Regresión Logística ajustado

Como vimos más arriba, en la Regresión Lineal Simple modelizamos el promedio de la variable dependiente Y como una función lineal de la covariable (aleatoria) continua X de la forma

$$E[Y|x] = \beta_0 + \beta_1\, x$$

representando β_1 el cambio en promedio de la variable Y por el incremento en una unidad de la covariable X.

En Regresión Logística (no binomial) la variable Y es dicotómica (es decir, toma sólo los valores 0 y 1, correspondientes a dos posibles resultados observables denominados *éxito* y *fracaso*), por lo que su media será $0 \cdot P(Y = 0) + 1 \cdot P(Y = 1) = P(Y = 1)$ y la ecuación de la regresión anterior quedará como

$$(E[Y|x] =) \qquad P(Y = 1|x) = \beta_0 + \beta_1\, x. \qquad [7.7]$$

Si la covariable X fuera también dicotómica (codificada de nuevo como 0,1), el incremento en una unidad de ésta sería el paso de $X = 0$ a $X = 1$, quedando

$$\beta_1 = P(Y = 1|1) - P(Y = 1|0)$$

diferencia que suele denominarse *Exceso de Riesgo*. Por tanto, si modelizamos la variable dependiente de la misma manera que lo habíamos hecho con la Regresión Lineal Simple, modelos que suelen denominarse *Modelos de Riesgo Aditivos*, ampliamente estudiados en Clayton y Hills (1993), el coeficiente β_1 o pendiente de la recta de regresión así ajustada, es el Exceso de Riesgo que se produce con el incremento en una unidad de la covariable independiente X.

Existen, no obstante, dos problemas importantes en la consideración de los Modelos de Riesgo Aditivos [7.7]. El primero es que, por analogía con los Modelos de Regresión Lineal, la variable dependiente debería de seguir una distribución normal, cosa inadmisible con los modelos [7.7] al ser la "variable dependiente" la probabilidad de una variable dicotómica, es decir, la de una variable que toma sólo dos valores. En segundo lugar, los estimadores de los coeficientes deben de respetar esta restricción de estar la predicción entre cero y uno.

Con respecto a la primera cuestión, si consideramos el promedio $E[Y|x]$ como una función de la covariable independiente x, supuesta ésta de tipo continuo, y denotamos este promedio por $P(x) = E[Y|x]$, en un modelo de Regresión Lineal, suponemos que $P(x)$ es una función lineal de la covariable de tipo continuo. En Regresión Logística esta suposición es poco creíble ya que una probabilidad (por ejemplo de toxicidad) sea una función lineal de la covariable, que suele ser una variable biomédica, no es muy razonable. Es más lógico modelizar esta probabilidad por una *función logística*

$$P(x) = \frac{\exp(\beta_0 + \beta_1\, x)}{1 + \exp(\beta_0 + \beta_1\, x)}$$

que, por su forma, Figura 7.4, recoge con más acierto la manera en la que va creciendo esa probabilidad.

Despejando de la ecuación anterior, el logaritmo de la denominada *odds ratio*, será

$$\log \frac{P(x)}{1 - P(x)} = \beta_0 + \beta_1\, x$$

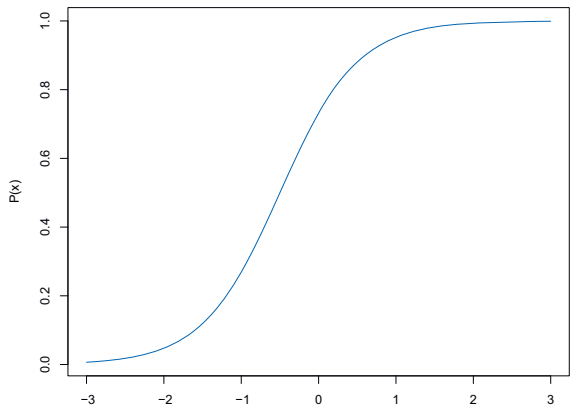

Figura 7.4 : Función logística

lo que expresa que el miembro de la izquierda, denominado en ocasiones *transformación logit de la variable de respuesta*, ahora sí es una función lineal de la covariable.

Si esta covariable predictora X tomara sólo los valores 0 , 1, la *odds ratio* sería

$$OR = \frac{P(1)/(1-P(1))}{P(0)/(1-P(0))} = \exp(\beta_1)$$

obteniéndose así, una clara interpretación del coeficiente β_1: el logaritmo de la *odds ratio*.

Como, de la ecuación anterior, obtenemos que es

$$\frac{P(1)}{1-P(1)} = \exp(\beta_1)\ \frac{P(0)}{1-P(0)}$$

en ocasiones se dice que el Modelo de Regresión Logística es un *Modelo de Riesgo Multiplicativo*.

Como conclusión, apuntamos que las suposiciones en las que está basado un Modelo de Regresión Logística y que deberemos comprobar antes de aplicarlo son: *(a)* La variable de respuesta Y es de tipo dicotómico, *(b)* La media de esta variable dicotómica, condicionada por la covariable observada x es la función logística $P(x)$, si X es de tipo continuo, *(c)* Los valores de la variable de respuesta son independientes unos de otros.

Ejemplo 7.5

Los datos del fichero `cryptoDATA.txt` corresponden a un estudio (Korich et al., 2000) de

cómo los ratones de laboratorio responden a la exposición a parásitos microscópicos del tipo Cryptosporidium mediante un modelo denominado *dosis-respuesta* (*dose-response*) que no es más que un modelo de regresión en donde la variable dependiente es la *respuesta* y la independiente, diferentes cantidades de *dosis*, modelo que suele ser del tipo Regresión Logística en donde se modeliza la variable dependiente, probabilidad de ser infectado p en función del número de parásitos d de la forma

$$logit(p) = \log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 \log_{10}(d)$$

En el fichero de datos aparecen las columnas `Y`= número de infectados (*éxitos*) y `N`=Número de pruebas, así como la covariable independiente `Dose`, dosis en 98 ratones.
Este estudio es muy importante ya que el Cryptosporidium es uno de los parásitos más comunes; causa diarrea, conocida como Cryptosporidiosis (más conocida por crypto) y está presente en el agua potable y las piscinas siendo muy resistente al cloro. Suele desactivarse (porque no muere) con rayos ultravioleta aunque así, alterando los ácidos nucleicos del parásito, se le impide replicarse e infectar. El ajuste de un modelo dosis-respuesta es, por tanto, de gran importancia y en él se inocularon un número `N` de ratones, con número `Dose` de cryptos, observándose un número `Y` de infectados.
Para ajustar el modelo logístico con R, después de incorporar los datos en `(1)`, dado que la variable `Y` no toma valores enteros, los redondeamos en `(2)` antes de ajustar el modelo en `(3)`.

```
> crypto<-read.table("d:\\datos\\cryptoDATA.txt",header=T)                    (1)

> crypto$Y <- round(crypto$Y)                                                 (2)

> crypto.glm1<-glm(cbind(Y,N-Y)~log10(Dose),data=crypto,family=binomial)     (3)

> summary(crypto.glm1)

Call:
glm(formula = cbind(Y, N - Y) ~ log10(Dose), family = binomial,
    data = crypto)
Deviance Residuals:
     Min        1Q    Median        3Q       Max
-3.81106  -1.25896  -0.08834   1.70009   5.12062
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -4.8648     0.3286  -14.80   <2e-16 ***
log10(Dose)   2.6163     0.1619   16.16   <2e-16 ***                          (4)
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
(Dispersion parameter for binomial family taken to be 1)
    Null deviance: 692.99  on 97  degrees of freedom
Residual deviance: 368.05  on 96  degrees of freedom
AIC: 588.05
Number of Fisher Scoring iterations: 4
```

El p-valor del test sobre el coeficiente de regresión es significativo por lo que el modelo ajustado será

$$\log\left(\frac{\hat{p}}{1-\hat{p}}\right) = -4'8648 + 2'6163 \log_{10}(d)$$

o bien

$$\hat{p} = \frac{e^{-4'8648+2'6163\ \log_{10}(d)}}{1+e^{-4'8648+2'6163\ \log_{10}(d)}} \qquad [7.8]$$

siendo $\hat{p}$ la probabilidad de ser infectado con una dosis d de cryptos. Así por ejemplo, la probabilidad de infectarse con una dosis $d = 30$ será

$$\hat{p} = \frac{e^{-4'8648+2'6163\ \log_{10} 30}}{1+e^{-4'8648+2'6163\ \log_{10} 30}} = 0'2689$$

ya que

```
> exp(-4.8648 + 2.6163 * log10(30))/(1+exp(-4.8648+2.6163*log10(30)))
[1] 0.2689006
```

En los modelos dosis-respuesta suele desearse determinar la dosis, $LD_{0'5}$, necesaria para que la probabilidad anterior sea $0'5$. Para ello sólo hay que despejar d de la ecuación [7.8] igualada a $0'5$

$$0'5 = \frac{e^{-4'8648+2'6163\ \log_{10} LD_{0'5}}}{1+e^{-4'8648+2'6163\ \log_{10} LD_{0'5}}}$$

de donde se obtiene el valor

$$LD_{0'5} = 10^{-\widehat{\beta}_0/\widehat{\beta}_1}$$

en nuestro ejemplo igual a

$$LD_{0'5} = 10^{4'8648/2'6163} = 72'35.$$

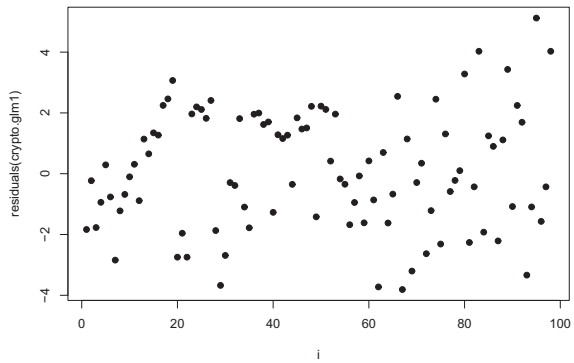

Figura 7.5 : Gráfico de los Residuos

Un gráfico de los residuos del modelo ajustado, obtenido ejecutando las dos siguientes sentencias

```
> i<-seq(1,98)
> plot(i,residuals(crypto.glm1),pch=16)
```

permite obtener la Figura 7.5 en donde no se aprecia ningún dato influyente.
Después de derivar en [7.8] se obtiene que el máximo valor de la pendiente de $\widehat{p}$ es $\widehat{\beta}_1/4$. Es decir, éste es el máximo cambio en la probabilidad de *éxito* que se produce por unidad de cambio en la variable predictora x. En nuestro ejemplo, podemos decir que $\widehat{\beta}_1/4 = 2'6163/4 = 0'654$ es el máximo cambio que se se puede producir en la probabilidad de infectarse por un aumento de una unidad de la covariable $\log_{10} d$, es decir, por un aumento de diez unidades de dosis.
Si editamos los datos `crypto` vemos que éstos proceden de cuatro laboratorios por lo que incorporar una covariable adicional, `Source`, parece razonable. Para ello ejecutamos `(1)` en donde con `factor(Source)` ya le indicamos a R que `Source` es una variable cualitativa (con cuatro clases, `Finch, UA, SPDL-HE` y `SPDL-TH`) por lo que debe de considerar tres covariables indicadoras en el modelo (las cuales elige no considerando la primera por orden alfabético, `Finch`), modelo que es analizado con `(2)`

```
> crypto.glm2<-glm(cbind(Y,N-Y)~log10(Dose)+factor(Source),data=crypto,
+ family=binomial)                                                    (1)

> crypto.glm2

Call:glm(formula=cbind(Y,N-Y)~log10(Dose)+factor(Source),family=binomial,data=crypto)

Coefficients:
          (Intercept)            log10(Dose)  factor(Source)SPDL-HE
             -5.00656               2.63312                0.04836
factor(Source)SPDL-TH       factor(Source)UA
              0.32151               0.07143

Degrees of Freedom: 97 Total (i.e. Null);  93 Residual
Null Deviance:      693
Residual Deviance: 363.8        AIC: 589.8

> summary(crypto.glm2)                                               (2)
Call:
glm(formula = cbind(Y, N - Y) ~ log10(Dose) + factor(Source),
    family = binomial, data = crypto)
Deviance Residuals:
     Min        1Q    Median        3Q       Max
-3.81429  -1.48396   0.01570   1.75416   4.62493
Coefficients:
                     Estimate Std. Error z value Pr(>|z|)
(Intercept)           -5.00656    0.35039 -14.288   <2e-16 ***
log10(Dose)            2.63312    0.16241  16.213   <2e-16 ***
factor(Source)SPDL-HE  0.04836    0.17655   0.274   0.7842                (3)
factor(Source)SPDL-TH  0.32151    0.17856   1.801   0.0718 .              (3)
factor(Source)UA       0.07143    0.16053   0.445   0.6564                (3)
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
(Dispersion parameter for binomial family taken to be 1)
    Null deviance: 692.99  on 97  degrees of freedom
Residual deviance: 363.84  on 93  degrees of freedom
AIC: 589.84
Number of Fisher Scoring iterations: 4
```

Los p-valores dados en `(3)` indican que es razonable no considerar esta covariable, es decir, que el laboratorio en donde se recogieron los datos no parece influir significativamente en los resultados.

Dispersión excesiva (*Overdispersion*)

En Regresión Logística se supone que la variable de respuesta sigue (condicionalmente) una distribución de Bernoulli, o en general binomial $B(n,p)$, con lo que $V(Y|\mathbf{x}) = n\,p\,(1-p)$. Si los n ensayos de Bernoulli no son independientes, o la probabilidad p no es constante a lo largo de estos ensayos, o alguna covariable importante no se ha incluido en el modelo, la varianza esperada anterior será mayor. Este fenómeno, al que denominamos en la Sección 7.2.1 Dispersión excesiva (*overdispersion*) puede detectarse mediante los *Residuos Estandarizados*

$$rest_i = \frac{r_i}{\sqrt{n_i\,\widehat{p}_i\,(1-\widehat{p}_i)}}$$

en donde $r_i = \widehat{y}_i - y_i$, $i = 1, ..., n$, son los residuos habituales y $n_i = 1$ en el caso de la Regresión Logística. Si planteamos la hipótesis nula H_0 :*Los datos no muestran overdispersion*, frente a la hipótesis alternativa, H_1 :*Los datos muestran overdispersion*, se tiene que, si H_0 es cierta, el estadístico

$$Over = \sum_{i=1}^{n} rest_i^2$$

sigue una distribución χ^2_{n-k} siendo $n-k$ los grados de libertad de la *Residual deviance*.

Un buen estimador del parámetro de *overdispersion*, ς, parámetro definido en la Sección 7.2.1, es

$$\widehat{\varsigma} = Over/(n-k).$$

Cuando existe *overdispersion*, la desviación típica de los estimadores hay que multiplicarla por $\widehat{\varsigma}$ y, por consiguiente, los p-valores de los tests también deben modificarse. Con R esto se hace simplemente cambiando `family=binomial` por `family=quasibinomial` en la ejecución de la función `glm`.

Ejemplo 7.5 (continuación)

En este ejemplo vimos más arriba que la *Residual deviance* tiene 96 grados de libertad, cosa que podemos averiguar ejecutando `(1)`. Los residuos estandarizados los obtenemos en `(2)`, cuya suma vemos en `(3)`. El p-valor del test de *overdispersion* lo ejecutamos en `(4)`, el cual concluye claramente que sí existe *overdispersion* en los datos.

```
> summary(crypto.glm1)$df[2]                                    (1)
```

```
[1] 96

> re<-(crypto$Y-crypto$N*fitted(crypto.glm1))/
+ sqrt(crypto$N*fitted(crypto.glm1)*(1-fitted(crypto.glm1)))                (2)

> over<-sum(re^2)

> over
[1] 333.9296                                                               (3)

> 1-pchisq(333.926,96)                                                     (4)
[1] 0
```

La estimación del parámetro de *overdispersion* es

$$\widehat{\varsigma} = Over/(n-k) = \frac{333'93}{96} = 3'4784$$

y el ajuste adecuado a esta situación el obtenido ejecutando `(5)`, que conduce a un p-valor, `(6)`, corregido por la situación de *overdispersion*, aunque con las mismas conclusiones que antes (dado lo significativo de los datos), y los mismos estimadores puesto que este análisis sólo afecta a la varianza de los estimadores de los coeficientes de regresión y, en consecuencia, a su posible significación.

```
> crypto.glm3<-glm(cbind(Y,N-Y)~log10(Dose),data=crypto,family=quasibinomial)

> summary(crypto.glm3)                                                     (5)

Call:
glm(formula = cbind(Y, N - Y) ~ log10(Dose), family = quasibinomial,
    data = crypto)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-3.81106  -1.25896  -0.08834   1.70009   5.12062

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -4.8648     0.6129  -7.938 3.85e-12 ***
log10(Dose)   2.6163     0.3020   8.664 1.10e-13 ***                        (6)
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for quasibinomial family taken to be 3.478531)

    Null deviance: 692.99  on 97  degrees of freedom
Residual deviance: 368.05  on 96  degrees of freedom
AIC: NA

Number of Fisher Scoring iterations: 4
```

7.4.2. Regresión Logística Multinomial

En Regresión Logística la variable dependiente toma sólo dos valores asimilados a *éxito* y *fracaso*. Es como decir que la variable dependiente toma sólo los valores, 0 y 1.

En el siguiente apartado veremos la situación en la que la variable dependiente toma los valores de una distribución de Poisson, 0, 1, 2, ... Una situación intermedia es la que veremos en este apartado en donde la variable dependiente presenta un número pequeño y limitado de distintas clases, es decir, que se puede admitir una variable dependiente que toma, digamos, los valores 1, 2, 3, ..., y_u.

En este caso se pueden formar $y_u - 1$ modelos de regresión logística como las antes estudiados, en donde se ajusta el logaritmo de la probabilidad de una clase, dividida por la probabilidad de la clase *baseline*, generalmente la primera, de la forma

$$\log\left(\frac{p_{i2}}{p_{i1}}\right) = \beta_{20} + \sum_{j=1}^{k} \beta_{2j} X_{ji}$$

$$\log\left(\frac{p_{i3}}{p_{i1}}\right) = \beta_{30} + \sum_{j=1}^{k} \beta_{3j} X_{ji}$$

$$\dots$$

$$\log\left(\frac{p_{iy_u}}{p_{i1}}\right) = \beta_{y_u 0} + \sum_{j=1}^{k} \beta_{y_u j} X_{ji}$$

En el capítulo de Problemas Resueltos Avanzados del texto *Cuadernos de Estadística Aplicada: Biología y Ciencias Ambientales*, hay algún caso de ajuste de este tipo de modelos, el cual se hace con la función `multinom` de la librería `nnet` ejecutando

```
multinom(modelo,data)
```

en donde `modelo` se especifica como siempre y `data` debe venir en formato *data frame*.

7.4.3. Regresión Poisson

Cabe pensar en ajustar un Modelo de Regresión Poisson a los datos cuando hay un incremento o reducción exponencial en la tasa media de ocurrencias de un determinado suceso, por ejemplo,

$$\lambda_i = E[Y|x] = c_0\, e^{\beta_1\, x_i}$$

ya que sería

$$\log(\lambda_i) = \beta_0 + \beta_1\, x_i$$

suponiendo que las observaciones y_i sigue una distribución de Poisson de parámetro λ_i y, por tanto, el logaritmo sería la función *link* asociada que relaciona la media de la variable dependiente con las covariables independientes como establece el modelo de Regresión Poisson.

A continuación aparece la resolución del ejemplo de Regresión Poisson clásica introducido al comienzo del capítulo, cuya versión robusta posponemos hasta el final.

Ejemplo 7.3 (continuación)

Para los datos de Lindenmayer sobre marsupiales, que vienen recogidos en el fichero de datos `marsu` proporcionado entre el Material Didáctico del curso, se pretende ajustar un Modelo de Regresión Poisson (en esta sección clásico) que tendrá 11 covariables, puesto que las cualitativas incorporan al modelo tantas covariables indicadoras como clases presentan menos una. Serán, 5 covariables cuantitativas, `Arbustos, Stags, Cortezas, Habitat` y `Acacias`, una indicador correspondiente a `Tocones`, dos covariables indicador correspondientes al tipo de Eucalipto, `Delegatensis` y `Nitens`, y tres covariables indicador correspondientes al aspecto del lugar, `NWSE`, `SESW` y `SWNW`, quedando el modelo de la forma

$$\log \texttt{Diversidad} = \beta_0 + \beta_1\, \texttt{Arbustos} + \beta_2\, \texttt{Stags} + \beta_3\, \texttt{Cortezas} + \beta_4\, \texttt{Habitat} + \beta_5\, \texttt{Acacias}$$

$$\beta_6\, \texttt{Tocones} + \beta_7\, \texttt{Delegatensis} + \beta_8\, \texttt{Nitens} + \beta_9\, \texttt{NWSE} + \beta_{10}\, \texttt{SESW} + \beta_{11}\, \texttt{SWNW}$$

Como los datos a utilizar deben de estar en forma de *estructura de datos*, ejecutamos `(1)` para incluirlos en R con este formato al utilizar la función `read.table`

```
> marsu<-read.table("d:\\datos\\marsu",header=T)                    (1)
```

Ahora, en `(2)`, utilizamos la función `glm` apareciendo los resultados en `(3)`, los cuales valoramos ejecutando `(4)`.

```
> respu<-glm(Diversidad ~ Arbustos+Stags+Cortezas+Habitat+Acacias+          (2)
+ Tocones+Delegatensis+Nitens+NWSE+SESW+SWNW,family=poisson,data=marsu)

> respu                                                             (3)

Call:  glm(formula = Diversidad ~ Arbustos + Stags + Cortezas + Habitat +
Acacias + Tocones + Delegatensis + Nitens + NWSE + SESW + SWNW,
family = poisson, data = marsu)

Coefficients:
 (Intercept)      Arbustos         Stags      Cortezas       Habitat
    -0.94694       0.01192       0.04023       0.03989       0.07173
```

```
     Acacias       Tocones  Delegatensis        Nitens          NWSE
     0.01764      -0.27241      -0.01534       0.11492       0.06676

        SESW          SWNW
     0.11695      -0.48891

Degrees of Freedom: 150 Total (i.e. Null);  139 Residual
Null Deviance:      187.5
Residual Deviance: 118.9        AIC: 423.7


> summary(respu)                                                    (4)

Call:
glm(formula = Diversidad ~ Arbustos + Stags + Cortezas + Habitat +
    Acacias + Tocones + Delegatensis + Nitens + NWSE + SESW +
    SWNW, family = poisson, data = marsu)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.04444  -0.97981   0.05174   0.44497   1.78912

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.94694    0.26550  -3.567 0.000362 ***
Arbustos      0.01192    0.02195   0.543 0.587005
Stags         0.04023    0.01120   3.590 0.000330 ***
Cortezas      0.03989    0.01439   2.772 0.005571 **
Habitat       0.07173    0.03814   1.881 0.059998 .
Acacias       0.01764    0.01060   1.664 0.096044 .
Tocones      -0.27241    0.28592  -0.953 0.340727
Delegatensis -0.01534    0.19161  -0.080 0.936176
Nitens        0.11492    0.27242   0.422 0.673131
NWSE          0.06676    0.19016   0.351 0.725554
SESW          0.11695    0.19029   0.615 0.538840
SWNW         -0.48891    0.24746  -1.976 0.048193 *
                (5)        (6)              (7)
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 187.49  on 150  degrees of freedom
Residual deviance: 118.87  on 139  degrees of freedom
AIC: 423.67

Number of Fisher Scoring iterations: 5
```

Los estimadores de los coeficientes aparecen en (5) y sus errores estándar en (6) (iguales ambos a los que aparecen en la corrección al artículo de Cantoni y Ronchetti en la página web de la primera) y los p-valores de los contrastes de la hipótesis nula de ser éstos cero,

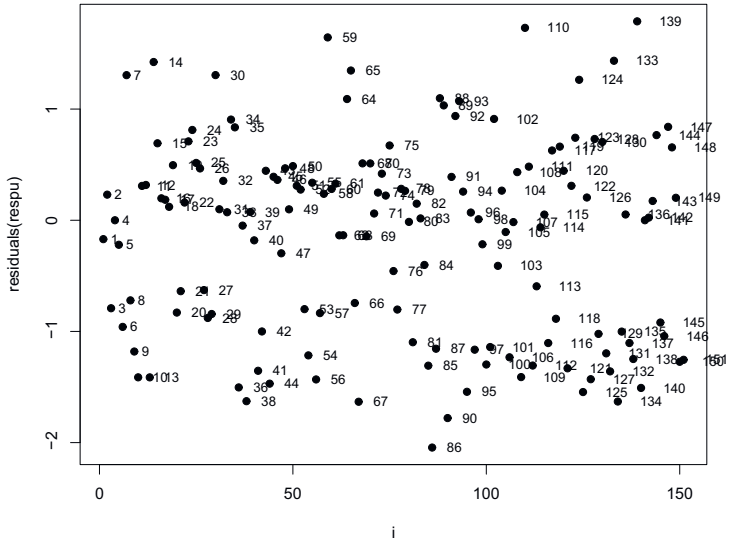

Figura 7.6 : Gráfico de los Residuos

aparecen en (7). Éstos parecen indicar que son significativas (es decir, que deberían de utilizarse) `Stags` y `Cortezas`; con dudas, el aspecto del lugar `SWNW` y, con muchas más dudas, `Habitat` y `Acacias`. Si nos quedáramos con estas cinco covariables, el modelo de Regresión Poisson clásico ajustado se obtendría ejecutando

```
> glm(Diversidad ~ Stags+Cortezas+Habitat+Acacias+SWNW,family=poisson,
+ data=marsu)$coeff

(Intercept)       Stags    Cortezas     Habitat     Acacias        SWNW
-0.82126194  0.04095916  0.04064335  0.07820461  0.01363318 -0.59675913
```

es decir, obtendríamos el modelo

$$\begin{aligned}\log \texttt{Diversidad} \;=\; & -0'8213 + 0'0410\,\texttt{Stags} + 0'0406\,\texttt{Cortezas} + 0'0782\,\texttt{Habitat} \\ & +0'0136\,\texttt{Acacias} - 0'5967\,\texttt{SWNW} \end{aligned} \qquad [7.9]$$

el mismo (segunda columna de la tabla 5) de las correcciones al artículo de Cantoni y Ronchetti.
Obsérvese que si representamos los residuos del modelo ajustado en la Figura 7.6 mediante la siguiente secuencia,

```
> i<-seq(1,151)
> plot(i,residuals(respu),pch=16)
> text(i,residuals(respu),1:151,adj=-1,cex=0.8)
```

no vemos aparentemente casi ninguna observación influyente. Cantoni y Ronchetti dicen que lo son la 59, la 110, la 139 y la 133, pero esto es un tanto arriesgado. Lo que sí pone de manifiesto este ejemplo es que el método de observar, quitar las observaciones anómalas y utilizar métodos clásicos para las observaciones restantes, no es operativo. Más adelante aplicaremos a estos datos métodos robustos.
Observemos por último, que hemos utilizado como variable dependiente de respuesta el número de especies marsupiales del lugar y no una tasa de éstas como hacemos habitualmente con los Modelos de Regresión Poisson. No debemos preocuparnos ya que el modelo sigue siendo válido al estar considerando, de hecho, una tasa hipotética consistente en dividir el número observado por 10 ó 100, y hablar de número de especies de marsupiales de cada 10 ó, de cada 100. Lo importante es tenerlo en cuenta cuando si hiciéramos predicciones con el modelo ajustado.

7.5. Métodos basados en la cuasi-verosimilitud

La definición de Modelo Lineal Generalizado, establecida en la sección segunda, lleva a suponer un distribución concreta de tipo exponencial para las observaciones $Y_i|\mathbf{x}_i$ (Poisson, normal, etc).

Además, una estructura para la media $E[Y_i|\mathbf{x}_i] = \mu_i = \mathbf{x}_i^t\,\boldsymbol{\beta}$, la cual implica una forma concreta para la varianza, ya que ésta está relacionada con la media a través de la expresión $Var(Y_i|\mathbf{x}_i) = \xi\, w_2(\mu_i) = \xi\, w_2(\mathbf{x}_i^t\,\boldsymbol{\beta})$.

La estimación y contrastes basados en la cuasi-verosimilitud (Wedderburn, 1974; McCullagh y Nelder, 1989; Heyde, 1997) relajan la suposición de una familia de tipo exponencial para las observaciones y, también, relajan algo la anterior ligadura entre la media y la varianza, ya que siguen suponiendo para la media la forma

$$E[Y_i|\mathbf{x}_i] = \mu_i = \mathbf{x}_i^t\,\boldsymbol{\beta}$$

pero para la varianza

$$Var(Y_i|\mathbf{x}_i) = \xi\, w_2(\mu_i)$$

se deja libertad a la función w_2.

El *estimador de cuasi-verosimilitud* es, de nuevo, la solución del sistema de ecuaciones de cuasi-verosimilitud

$$\sum_{i=1}^{n} \frac{\partial Q(y_i, \mu_i(\boldsymbol{\beta}))}{\partial \boldsymbol{\beta}} = \sum_{i=1}^{n} \frac{\boldsymbol{\mu_i}'}{\xi\, w_2(\mu_i)}\,(y_i - \mu_i) = \mathbf{0} \qquad [7.10]$$

denominado ahora así porque la forma de $w_2(\mu_i)$ es arbitraria. Los sumandos de la anterior ecuación, que serían los *scores* individuales en el método de cuasi-verosimilitud, suelen representarse como

$$\frac{\partial Q(y_i,\mu_i)}{\partial \boldsymbol{\beta}} = \frac{(y_i-\mu_i)}{\xi\, w_2(\mu_i)}\, \boldsymbol{\mu_i}'$$

7.6. Métodos Bayesianos

Si existe información previa sobre los parámetros $\boldsymbol{\beta}$ suministrada a través de una distribución *a priori* $\pi(\boldsymbol{\beta})$, por el teorema de Bayes, la distribución a posteriori de los parámetros será

$$\pi(\boldsymbol{\beta}|y_1,...,y_n) = \frac{L(\boldsymbol{\beta})\,\pi(\boldsymbol{\beta})}{\int L(\boldsymbol{\beta})\,\pi(\boldsymbol{\beta}) d\boldsymbol{\beta}}$$

Si se considera una función de pérdida cuadrática, el *estimador Bayes* sería la media de la distribución a posteriori anterior (véase, por ejemplo, Vélez y García Pérez, 1993, Sección 5.5.2).

El problema radica (además de la habitual subjetividad en la elección de la distribución a priori, lo que duplica los problemas de sensibilidad en la distribución modelo) en los cálculos, que deben de ser numéricos y las integrales a resolver, por ejemplo la media de la anterior distribución a posteriori,

$$E[\boldsymbol{\beta}|y_1,...,y_n] = \int \boldsymbol{\beta}\,\pi(\boldsymbol{\beta}|y_1,...,y_n)\, d\boldsymbol{\beta}$$

tendrían la dimensión de $\boldsymbol{\beta}$ siendo su cálculo numérico muy complejo.

Una alternativa es considerar la moda de esta distribución a posteriori como *estimador Bayes* de los parámetros $\boldsymbol{\beta}$ (véase, por ejemplo, Vélez y García Pérez, 1993, Sección 7.5), es decir, como estimador $\widehat{\boldsymbol{\beta}}_{\boldsymbol{Ba}}$ el valor de $\boldsymbol{\beta}$ que maximiza la densidad a posteriori $\pi(\boldsymbol{\beta}|y_1,...,y_n)$ o, equivalentemente, su logaritmo, igual (salvo constantes) a

$$\log L(\boldsymbol{\beta}) + \log \pi(\boldsymbol{\beta})$$

en donde el primer sumando es el logaritmo de la verosimilitud del modelo lineal generalizado, expresado, por ejemplo, por [7.4], y el segundo sumando el logaritmo de la distribución a priori. Por ejemplo, si esta distribución a priori fuera normal multivariante,

$$\boldsymbol{\beta} \rightsquigarrow N_k(\boldsymbol{\alpha}, \mathbf{B})$$

la función anterior a maximizar sería

$$\log L(\boldsymbol{\beta}) - \frac{1}{2}\,(\boldsymbol{\beta}-\boldsymbol{\alpha})^t\, \mathbf{B^{-1}}\, (\boldsymbol{\beta}-\boldsymbol{\alpha})$$

la cual puede maximizarse iterativamente, por ejemplo, mediante el algoritmo *EM* (*E*xpectation-*M*aximizing).

7.7. Métodos robustos

Es conocido que los estimadores de máxima verosimilitud son, en general, bastante sensibles a la presencia de datos anómalos. En concreto, la falta de robustez en la Regresión Logística fue puesta de manifiesto por Pregibon (1982), y, en general, para todos los modelos lineales generalizados por autores como Stefanski, Carroll y Ruppert (1986); Künsch, Stefanski y Carroll (1989); o Morgenthaler (1992).

Si comparamos el sistema [7.5] ó [7.10] (de ecuaciones de verosimilitud o cuasi-verosimilitud) con el que proporciona los M-estimadores multidimensionales (sistema [6.2] de la sección 6.5.2 del texto MR) o, en primera instancia, se compara con la situación unidimensional (ecuación [2.6] de la sección 2.5 del texto MR), se pueden considerar los estimadores de máxima verosimilitud o cuasi-verosimilitud, como M-estimadores con función ψ (función *score*) asociada, la función

$$\psi(y_i, \mu_i) = \frac{(y_i - \mu_i)}{\xi\, w_2(\mu_i)}\, \boldsymbol{\mu_i}'$$

Como la función de influencia de tales estimadores es proporcional a esta función (véase la ecuación [6.3] del texto MR), si esta función *score* no es acotada (como función de las observaciones y_i o de las funciones $\mathbf{x}_i$ a través de μ_i) el estimador resultante no será robusto. Aquí, como puede observarse, la diferencia $y_i - \mu_i$ del numerador nos dice que no es acotada y que, por tanto, los estimadores de máxima verosimilitud y cuasi-verosimilitud, no van a ser robustos frente a observaciones y_i distantes de su media μ_i o frente a la presencia de datos anómalos en las covariables $\mathbf{x}_i$.

Aunque existen varios trabajos sobre Regresión Logística Robusta, principalmente del grupo *Agoras* liderado por Peter Rousseeuw, aquí expondremos la solución propuesta por Elvezio Ronchetti (y Eva Cantoni) en su trabajo de 2001 para todo modelo lineal generalizado.

7.7.1. *M*-estimadores basados en la cuasi-verosimilitud

Como dijimos más arriba, la forma de las ecuaciones de verosimilitud [7.5] y cuasi-verosimilitud [7.10] sugiere buscar el estimador robusto entre los M-estimadores (MR-secciones 2.5 y 6.5.2), uno de los cuales es el estimador de máxima verosimilitud y otro el estimador basado en la cuasi-verosimilitud. En concreto, Cantoni y Ronchetti (2001) sugieren M-estimadores para los parámetros $\boldsymbol{\beta}$ con función ψ asociada, de la forma

$$\psi(y_i, \mu_i) = w(\mathbf{x}_i)\, \nu(y_i, \mu_i)\, \boldsymbol{\mu_i}' - a(\boldsymbol{\beta})$$

es decir, soluciones en $\boldsymbol{\beta}$ de las ecuaciones

$$\sum_{i=1}^{n} \frac{\partial Q(y_i, \mu_i(\boldsymbol{\beta}))}{\partial \boldsymbol{\beta}} = \sum_{i=1}^{n} \left[w(\mathbf{x}_i)\, \nu(y_i, \mu_i)\, \boldsymbol{\mu_i}' - a(\boldsymbol{\beta}) \right] = \mathbf{0} \qquad [7.11]$$

de manera que se pueda separar la influencia de datos anómalos en dos funciones (M-estimadores tipo-Mallows como se indica en MR, pp. 182) una, $w(\mathbf{x}_i)$, que recoja la influencia en el espacio de las covariables y otra, $\nu(y_i, \mu_i)$ que haga lo propio en el de las observaciones dependientes y_i. Eligiendo una y otra acotadas obtendremos estimadores robustos.

Como función $a(\boldsymbol{\beta})$ se elige la función

$$a(\boldsymbol{\beta}) = \frac{1}{n} \sum_{i=1}^{n} E_{y_i|\mathbf{x}_i}[\nu(y_i, \mu_i)]\, w(\mathbf{x}_i)\, \boldsymbol{\mu_i}'$$

(en donde $E_{y_i|\mathbf{x}_i}$ representa la esperanza o media con respecto a la distribución condicionada $y_i|\mathbf{x}_i$) con objeto de que el estimador resultante sea Fisher-consistente[1].

Como funciones $w(\mathbf{x}_i)$ y $\nu(y_i, \mu_i)$ se suelen elegir funciones que han dado buenos resultados en Regresión Lineal, desde el punto de vista de la robustez. Obsérvese que si elegimos

$$w(\mathbf{x}_i) = 1 \qquad \text{y} \qquad \nu(y_i, \mu_i) = \frac{(y_i - \mu_i)}{\xi\, w_2(\mu_i)} \qquad \forall\, i = 1, ..., n$$

obtendremos como M-estimadores los basados en la cuasi-verosimilitud.

Para los modelos lineales generalizados, Regresión Logística y Regresión Poisson, Cantoni y Ronchetti (2001) proponen utilizar como función $\nu(y_i, \mu_i)$ la función

$$\nu(y_i, \mu_i) = \frac{\psi_b(r_i)}{\sqrt{\xi\, w_2(\mu_i)}}$$

en donde es

$$r_i = \frac{(y_i - \mu_i)}{\sqrt{\xi\, w_2(\mu_i)}}$$

y ψ_b la función de Huber (ya definida en el Ejemplo 2.8 de MR)

[1] Propiedad definida como $T(F_\theta) = \theta$ sea cual sea el valor del parámetro θ dentro del espacio paramétrico, y que significa que el estimador, con funcional asociado T, toma, asintóticamente, el valor correcto del parámetro.

$$\begin{aligned}\psi_b(x) &= \text{mín}\{b, \text{máx}\{x, -b\}\} = x \cdot \text{mín}\left\{1, \frac{b}{|x|}\right\} \\ &= \begin{cases} -b & \text{si} \quad x < -b \\ x & \text{si} \quad -b \leq x \leq b \\ b & \text{si} \quad x > b \end{cases}\end{aligned}$$

por lo que llamaremos *estimador cuasi-verosímil tipo-Mallows* a la solución en $\boldsymbol{\beta}$ del sistema de ecuaciones

$$\sum_{i=1}^{n} \left[w(\mathbf{x}_i) \, \frac{\psi_b(r_i)}{\sqrt{\xi \, w_2(\mu_i)}} \, \boldsymbol{\mu_i}' - a(\boldsymbol{\beta}) \right] = \mathbf{0}.$$

Como ocurría con los M-estimadores en la Regresión Lineal (MR-sección 7.3), si tomamos además $w(\mathbf{x}_i) = 1$, el estimador resultante recibe el nombre de *estimador cuasi-verosímil de Huber*.

7.7.2. Contraste robusto de bondad de ajuste del modelo

Más arriba basamos el contraste de bondad de ajuste de un modelo lineal generalizado a unos datos en el estadístico de contraste *desviación* (*deviance*)

$$G^2 = -2 \sum_{i=1}^{n} \left[l_i(\widehat{\mu}_i) - l_i(y_i) \right]$$

en donde las l_i son (salvo constantes irrelevantes en la obtención del máximo) las contribuciones de cada uno de los valores muestrales al logaritmo de la verosimilitud, $\log L(\mu_1, ..., \mu_n) = \sum_{i=1}^{n} l_i(\mu_i)$, pero ahora evaluadas en la media estimada $\widehat{\mu}_i$ y en los datos observados y_i, lo que permite comparar el máximo obtenido con los estimadores de máxima verosimilitud y el obtenido con los datos.

Mediante los M-estimadores basados en la cuasi-verosimilitud resolvemos el sistema [7.11], es decir, minimizamos (hay un cambio de signo irrelevante al estar la derivada igualada a cero) la función $\sum_{i=1}^{n} Q_i(y_i, \mu_i)$, por lo que una medida de la *cuasi-verosimilitud* alcanzada por los estimadores obtenidos será $\sum_{i=1}^{n} Q_i(y_i, \widehat{\mu}_i)$.

De esta manera podemos comparar dos modelos determinados, al igual que lo hacíamos en TA-sección 8.4.1, considerando como hipótesis nula un modelo con $k + 1 - q$ términos (es decir, con q determinados $\beta_i = 0$) al que podemos denominar *submodelo*, frente a la hipótesis alternativa de un modelo con más términos, digamos con $k+1$ parámetros $\beta_i \neq 0$. Si $\widetilde{\mu}_i$ y $\widehat{\mu}_i$ son, respectivamente, los estimadores de μ_i bajo los modelos con los $k + 1 - q$ y $k + 1$ parámetros

estimados, Cantoni y Ronchetti (2001) proponen un test robusto de bondad de ajuste basado en el estadístico de contraste

$$Q^2 = 2\left[\sum_{i=1}^{n} Q_i(y_i, \widetilde{\mu}_i) - \sum_{i=1}^{n} Q_i(y_i, \widehat{\mu}_i)\right]$$

el cual, para tamaños muestrales suficientemente grandes, sigue aproximadamente una distribución combinación lineal de q variables independientes Y_i, cada una de ellas con distribución χ^2_1

$$Q^2 \rightsquigarrow \sum_{i=1}^{q} d_i\, Y_i$$

siendo $d_1, ..., d_q$ los q autovalores positivos de una determinada matriz.

7.7.3. Cálculo con R^{mo}

Cantoni y Ronchetti (2001) proporcionan apoyo informático para la obtención de los M-estimadores cuasi-verosímiles tipo-Mallows robustos antes estudiados, así como los estimadores cuasi-verosímiles de Huber, en Modelos Lineales Generalizados con distribuciones Bernoulli (es decir, Regresión Logística Robusta), Poisson (es decir, Regresión Poisson Robusta) y Binomial. Además, también proporcionan códigos para ejecutar el test robusto de bondad del ajuste Q^2 definido más arriba.

Para la estimación robusta de los parámetros utilizaremos la función

```
glm.rob(x,y,choice,ni)
```

en donde bajo el argumento `x` incluimos la matriz de datos de las covariables, incorporando los datos de éstas en las columnas. En el argumento `y` incluimos los datos de la variable respuesta en una matriz con una columna. Con `choice` elegimos cuál de los tres análisis queremos realizar, logístico con `logit`, binomial con `binom` y de Regresión Poisson con `poisson`. El argumento `ni` se utiliza sólo si se eligió la Regresión Binomial y, en este caso, debe ser una matriz de una columna (de igual tamaño que `y`), en donde indicamos el número de ensayos n_i correspondientes al número de éxitos y_i antes fijado en `y`.

Para la ejecución del contraste robusto de bondad del ajuste basado en el estadístico Q^2, utilizaremos la función

```
quasi.rob(x,y,out.col,choice,ni)
```

con idéntico significado de los argumentos que en la función antes considerada `glm.rob`, y donde el nuevo argumento `out.col` debe indicar las columnas a omitir en el submodelo. (Sobre este punto ver el ejemplo que sigue).

En la elección del estimador tipo-Mallows debemos fijar previamente el valor de la constante de Huber c. Esto lo haremos, por tanto, con anterioridad y con la precaución de que si se guardan los resultados al salir de R^{mo} éste será el valor de dicha constante en sesiones sucesivas y de que, si no se guarda, deberá volver a definirse. Si se hace c igual a infinito obtendremos los mismos resultados que con el método clásico.

En la librería `robustbase` del R de Internet se puede utilizar la función `glmrob` en lugar de la que el equipo docente ha implementado en R^{mo} antes descrita, `glm.rob`. En esa función de R uno de los argumentos es `control` con el que se fija el valor de la constante de Huber. Al resultado obtenido utilizándola se le puede aplicar la función `summary` para analizar la significación de los estimadores de regresión robustos así obtenidos.

También figura en R la función `anova.glmrob` que hará el papel de la función `quasi.rob` antes mencionada, pero esta función aún no está implementada en el R de Internet. De ahí la necesidad de seguir utilizando R^{mo} en algunas ocasiones.

Comencemos con un ejemplo de Análisis de Regresión Binomial robusto.

Ejemplo 7.1 (continuación)
Primero fijamos el valor de la constante de Huber en `(1)`, ejecutando a continuación la función que nos proporciona las estimaciones robustas. En `(2)` obtenemos éstas y en `(3)` sus errores estimados, iguales a los obtenidos en la columna derecha de la Tabla 1 del trabajo de Cantoni y Ronchetti (2001), con una pequeña diferencia ya que nosotros trabajamos con R^{mo} y ellos con S-Plus.

```
> chuber<-1.2                                                          (1)
> salida.robusta<-glm.rob(as.matrix(zanahorias[,c(3,6,5)]),
+ as.matrix(zanahorias[,1]), choice="binom",ni=as.matrix(zanahorias[,2]))

> salida.robusta$coeff                                                 (2)
[1]  1.9301522 -2.0497142  0.6897909  0.4613198

> salida.robusta$sd.coeff                                              (3)
[1] 0.6984066 0.3689728 0.2366980 0.2413989
```

Si ahora queremos validar el modelo con el que nos quedaremos, podemos hacer contrastes anidados como los que se indicaban más arriba, consistentes en establecer como hipótesis alternativa un modelo con un número determinado de covariables y como hipótesis nula un submodelo de éste. Si rechazamos la hipótesis nula, con un p-valor bajo, podemos concluir que la covariable no incluida en el modelo de la hipótesis nula (en el submodelo) es relevante a la hora de explicar a la variable dependiente. Todo esto lo haremos con la función anterior `quasi.rob`

Primero plantearemos la hipótesis alternativa de un modelo con las tres covariables consideradas, `logdosis`, `bloque1` y `bloque2` frente a la hipótesis nula del submodelo sin la covariable `bloque2`. Para ello ejecutamos la secuencia siguiente en donde destacamos que, en la línea marcada con `(4)` incluimos, como primer argumento de la función, un modelo con las tres covariables que aparecen en las columnas 3, 5 y 6 de la matriz de datos, y que en la línea `(5)` le decimos, con el argumento `out.col=3`, que como hipótesis nula considere el submodelo sin

la que aparece en la columna 3 de las anteriores, es decir, en la columna 6 de la matriz de datos, es decir, sin `bloque2`.
El p-valor de este test lo obtenemos ejecutando `(6)` que claramente indica que rechazamos la hipótesis nula del submodelo, lo que indica cierta significación (i.e., algo explica) la covariable `bloque2`.

```
> resultado<-quasi.rob(as.matrix(zanahorias[,c(3,5,6)]),                    (4)
+ as.matrix(zanahorias[,1]),out.col=3,choice="binom",                       (5)
+ ni=as.matrix(zanahorias[,2]))

> resultado$pvalue                                                          (6)

             [,1]
 [1,] 0.003565751
```

Podemos considerar el siguiente árbol de posibles modelos en una primera tanda de comparaciones

H_0 : `logdosis, bloque1`
H_1 : `logdosis, bloque1, bloque2`

H_0 : `logdosis, bloque2`
H_1 : `logdosis, bloque1, bloque2`

H_0 : `bloque1, bloque2`
H_1 : `logdosis, bloque1, bloque2`

En el primer test obtuvimos el p-valor $0'0036$. Los otros dos p-valores los obtenemos ejecutando

```
> quasi.rob(as.matrix(zanahorias[,c(3,5,6)]),as.matrix(zanahorias[,1]),
+ out.col=2,choice="binom",ni=as.matrix(zanahorias[,2]))$pvalue

             [,1]
  [1,] 0.05600116
```

y

```
> quasi.rob(as.matrix(zanahorias[,c(3,5,6)]),as.matrix(zanahorias[,1]),
+ out.col=1,choice="binom",ni=as.matrix(zanahorias[,2]))$pvalue

             [,1]
  [1,] 2.773081e-08
```

p-valores que llevan a la conclusión de ser muy significativa (muy explicativa) la covariable `logdosis`, algo significativa (como dijimos más arriba) `bloque2` y poco relevante `bloque1`.
Como el único posible modelo sería el que contiene a las covariables `logdosis` y `bloque2` surgen ahora dos posibles tests,

H_0 : `logdosis`
H_1 : `logdosis, bloque2`

H_0 : `bloque2`
H_1 : `logdosis, bloque2`

cuyos p-valores obtenemos ejecutando, respectivamente, las secuencias,

```
> quasi.rob(as.matrix(zanahorias[,c(3,6)]),as.matrix(zanahorias[,1]),
+ out.col=2,choice="binom",ni=as.matrix(zanahorias[,2]))$pvalue

             [,1]
  [1,] 0.01178241
```

y

```
> quasi.rob(as.matrix(zanahorias[,c(3,6)]),as.matrix(zanahorias[,1]),
+ out.col=1,choice="binom",ni=as.matrix(zanahorias[,2]))$pvalue

               [,1]
  [1,] 3.961684e-08
```

los cuales indican, de nuevo, la significación de `bloque2` y, de nuevo, lo significativo que resulta la covariable `logdosis`.
Parece, por tanto, razonable utilizar estas dos covariables, para cuya estimación de parámetros ejecutamos la siguiente secuencia

```
> glm.rob(as.matrix(zanahorias[,c(3,6)]),as.matrix(zanahorias[,1]),
+ choice="binom",ni=as.matrix(zanahorias[,2]))$coeff

[1]  2.1187526 -2.0355601  0.4759153
```

que lleva a quedarnos, finalmente, con el modelo

$$\log\left(\frac{\mu_i}{n_i-\mu_i}\right) = 2'119 - 2'036\ \log(\texttt{dosis}) + 0'476\,\text{bloque2}$$

Observemos que si en `(1)` hacemos la constante de Huber igual a infinito, obtendremos, en lugar de `(2)`, los resultados clásicos obtenidos cuando hicimos este ejemplo con Métodos Clásicos. Veámoslo,

```
> chuber<-Inf

> a<-glm.rob(as.matrix(zanahorias[,c(3,6,5)]),as.matrix(zanahorias[,1]),
+ choice="binom",ni=as.matrix(zanahorias[,2]))
There were 26 warnings (use warnings() to see them)

> a$coeff
[1]  1.4540106 -1.8078152  0.8497862  0.5524021
```

Veamos a continuación dos ejemplos de Análisis de Regresión Logística Robusta.

Ejemplo 7.2 (continuación)
Después de fijar el valor de la constante de Huber en $1'2$ utilizamos la función `glm.rob` en la estimación robusta de los parámetros de la Regresión Logística, los cuales obtenemos en `(1)`.

```
> chuber<-1.2

> B<-glm.rob(as.matrix(leucemia[,c(2,3)]),as.matrix(leucemia[,c(1)]),
+ choice="logit")

> B$coeff
[1]  0.1646176 -2.0318031  2.4926958                                          (1)
```

Si ahora queremos analizar con cuál modelo nos quedamos, podemos hacer contrastes anidados, como los que hicimos en el ejemplo anterior, en los que estableceremos como hipótesis alternativa un modelo con un número determinado de covariables y como hipótesis nula un submodelo de éste. Si rechazamos la hipótesis nula, con un p-valor bajo, podemos concluir que la covariable no incluida en el modelo de la hipótesis nula (en el submodelo) es relevante a la hora de explicar a la variable dependiente. Todo esto lo haremos con la función anterior `quasi.rob`
Primero plantearemos la hipótesis alternativa de un modelo con las dos covariables consideradas, WBC y AG frente a la hipótesis nula del submodelo sin la covariable AG. Es decir, contrastaremos las hipótesis

$H_0 : WBC$
$H_1 : WBC, AG$

Para ello ejecutamos la secuencia siguiente en donde destacamos que en la línea marcada con `(2)` incluimos, como primer argumento de la función, un modelo con las dos covariables que aparecen en las columnas 2 y 3 de la matriz de datos, y que en la línea `(3)` le decimos, con el argumento `out.col=2`, que como hipótesis nula considere el submodelo sin la covariable que aparece en la columna 2 de las anteriores, es decir, en la columna 3 de la matriz de datos, es decir, sin AG.
El p-valor de este test lo obtenemos ejecutando `(4)` que no es concluyente en cuanto al rechazo de la hipótesis nula del submodelo (desde luego la rechaza para un nivel de significación $0'05$), indicando cierta significación (i.e., algo explica) la covariable AG.

```
> a1<-quasi.rob(as.matrix(leucemia[,c(2,3)]),as.matrix(leucemia[,c(1)]),   (2)
out.col=2,choice="logit")                                                  (3)

> a1$pvalue                                                                (4)
             [,1]
  [1,] 0.04645812
```

Ahora contrastaremos la otra posibilidad cual es la de eliminar la covariable WBC, es decir, contrastar las hipótesis

$H_0 : AG$
$H_1 : WBC, AG$

Para ello ejecutamos la siguiente sentencia, indicándole en `(5)` que ahora no considere la covariable que aparece en el lugar 1 de la matriz previa de datos de las covariables; es decir, la de la columna 2 de la matriz de datos, es decir, que prescinda en la hipótesis nula de WBC.
El p-valor lo obtenemos ejecutando `(6)`, el cual indica que se puede aceptar la hipótesis nula y prescindir de la covariable WBC.

```
> a2<-quasi.rob(as.matrix(leucemia[,c(2,3)]),as.matrix(leucemia[,c(1)]),
out.col=1,choice="logit")                                                    (5)

> a2$pvalue                                                                  (6)
            [,1]
  [1,] 0.1371982
```

Por tanto, como ya hemos decidido quedarnos sólo con la covariable AG, volvemos a ajustar el modelo de Regresión Logístico Robusto ejecutando

```
> glm.rob(as.matrix(leucemia[,c(3)]),as.matrix(leucemia[,c(1)]),
+ choice="logit")$coeff

[1] -1.945900  2.063683
```

quedándonos, por tanto, con el modelo de Regresión Logística Robusto

$$\log \frac{p}{1-p} = -1'9459 + 2'063683\, AG.$$

Ejemplo 7.6 (TA-ejemplo 9.1)
En el texto TA resolvimos un ejercicio (el 9.1 de la sección 9.4 de ese texto) en el que realizábamos un Análisis de Regresión Logística a unos datos. Allí lo resolvíamos utilizando Métodos Clásicos. A continuación utilizaremos Métodos Robustos.
Para ello primero volvemos a fijar, en `(1)`, el valor de la constante de Huber y luego ejecutamos `(2)` sólo con la covariable `presión` que era la significativa.

```
> chuber<-1.2                                                               (1)

> A<-glm.rob(as.matrix(valores[,c(10)]),as.matrix(valores[,c(6)]),          (2)
+ choice="logit")                                                           (2)

> A$coeff
[1]  1.335000 -1.180849
```

Observemos que obtenemos las mismas estimaciones para los coeficientes que obteníamos allí (al final de la Sección 9.4 de TA) puesto que no había datos anómalos entre las observaciones.

Para finalizar veamos la versión robusta de un ejemplo de Regresión Poisson antes considerado.

Ejemplo 7.3 (continuación)
Primero fijamos el valor de la constante de Huber en $1'6$ que es el valor establecido en Cantoni y Ronchetti (2001). Después utilizamos la función `glm.rob` en la estimación robusta de los parámetros de la Regresión Poisson, los cuales obtenemos en `(1)`.

```
> chuber<-1.6

> C<-glm.rob(as.matrix(marsu[,c(2,3,4,5,6,7,9,10,12,13,14)]),
```

```
+ as.matrix(marsu[,c(1)]),choice="poisson")

> C$coeff                                                                    (1)
 [1] -0.89780510  0.00994289 -0.25141328  0.04016733  0.03999019
 [6]  0.07141413  0.01777746 -0.02022772  0.12693237  0.06009973
[11]  0.09492416 -0.50792232
```

Si aceptáramos este modelo de Regresión Poisson Robusto, nos quedaría por tanto,

$$\begin{aligned}\log \texttt{Diversidad} &= -0'8978 + 0'0099\,\texttt{Arbustos} + 0'0402\,\texttt{Stags} + 0'04\,\texttt{Cortezas} \\ &\quad +0'0714\,\texttt{Habitat} + 0'0178\,\texttt{Acacias} - 0'2514\,\texttt{Tocones} \\ &\quad -0'0202\,\texttt{Delegatensis} + +0'1269\,\texttt{Nitens} + 0'0601\,\texttt{NWSE} \\ &\quad +0'0949\,\texttt{SESW} - 0'5079\,\texttt{SWNW}\end{aligned}$$

que son los mismos valores que aparecen en la corrección del trabajo de Cantoni y Ronchetti. Ahora deberíamos realizar tests condicionales para ver con qué modelo nos quedamos finalmente. Como hay muchas covariables y muchos datos, el programa da errores en algunos contrastes anidados. Si nos limitamos a ajustar el Modelo de Regresión Poisson Robusto para las cinco covariables con las que nos quedamos en los métodos clásicos, ejecutaríamos

```
> glm.rob(as.matrix(marsu[,c(4,5,6,7,14)]),as.matrix(marsu[,c(1)]),
+ choice="poisson")$coeff

[1] -0.79811068  0.04057311  0.04099017  0.07762185  0.01429919 -0.60443908
```

con lo que nos quedaríamos con el Modelo de Regresión Poisson Robusto,

$$\begin{aligned}\log \texttt{Diversidad} &= -0'7981 + 0'0406\,\texttt{Stags} + 0'0410\,\texttt{Cortezas} + 0'0776\,\texttt{Habitat} \\ &\quad +0'0143\,\texttt{Acacias} - 0'6044\,\texttt{SWNW}\end{aligned}$$

el mismo obtenido en la corrección del artículo de Cantoni y Ronchetti y casi idéntico al clásico [7.9] como era de esperar, ya que allí comentamos que no veíamos observaciones influyentes.

7.8. Ajuste de modelos GLM para datos espaciales

En Kelsall y Diggle (1998) se propone un modelo *logit* en el caso de que la variable de respuesta Y sea dicotómica, típica de situaciones en las que queramos comparar dos poblaciones: Casos y Controles, de datos espaciales en donde el interés principal sea la localización $\mathbf{s}_i$ ($\mathbf{x}_i$ siguiendo la notación de este capítulo) y no el valor observado allí. En concreto se modeliza el problema

con una variable de respuesta Y que tome el valor $Y = 1$, cuando se observe un Caso, e $Y = 0$ cuando se observe un Control, de la forma

$$P\{Y_i = 1|X_i = \mathbf{x}_i\} = p(\mathbf{x}_i) = \frac{\lambda_1(\mathbf{x}_i)}{\lambda_0(\mathbf{x}_i) + \lambda_1(\mathbf{x}_i)}$$

siendo $\lambda_1(\mathbf{x})$ y $\lambda_0(\mathbf{x})$ las intensidades de dos procesos de Poisson homogéneos de las poblaciones, respectivamente, Casos y Controles. Es decir,

$$logit(p(\mathbf{x})) = \log\left(\frac{p(\mathbf{x})}{1 - p(\mathbf{x})}\right) = \log\frac{\lambda_1(\mathbf{x}_i)}{\lambda_0(\mathbf{x}_i)} = r(\mathbf{x}_i) + \log\frac{n_1}{n_0}$$

siendo

$$r(\mathbf{x}_i) = \log\frac{f(\mathbf{x}_i)}{g(\mathbf{x}_i)}$$

el riesgo de enfermedad definido en el Capítulo 4.

Capítulo 8

Modelos Aditivos Generalizados GAM

8.1. Introducción

En este capítulo estudiaremos los *Modelos Aditivos Generalizados* GAM, debidos a Hastie y Tibshirani (1986, 1990). Estos modelos son una extensión del Modelo de Regresión Lineal Múltiple y de los Modelos Aditivos.

En el Modelo de Regresión Lineal Múltiple explicamos la media de la variable de respuesta Y con k covariables de forma lineal

$$E[Y|\mathbf{X}] = \beta_0 + \beta_1 X_1 + ... + \beta_k X_k.$$

En los denominados *Modelos Aditivos* permitimos a las covariables X_i una expresión más general que la anterior mediante unas funciones h_i, aunque manteniendo la linealidad del modelo,

$$E[Y|\mathbf{X}] = h_0 + h_1(X_1) + ... + h_k(X_k).$$

La incorporación de las funciones h_i hace que el modelo sea más flexible y capaz de adaptarse a datos más complejos que no muestren una estricta linealidad en las covariables.

No obstante, los modelos aditivos tienen que verificar todas las suposiciones que exigíamos a los modelos de regresión lineal: normalidad de los residuos, homocedasticidad, etc. y, además, la variable de respuesta Y debe de tener distribución normal. Los denominados *Modelos Aditivos Generalizados* GAM generalizan los Modelos Aditivos de la misma manera que los Modelos Lineales Generalizados GLM generalizaban los Modelos de Regresión Lineal Múltiple, permitiendo que la variable de respuesta dependiente Y tenga una distribución más general que la normal, en concreto una familia exponencial como las

estudiadas en el capítulo anterior, por lo que este tipo de técnicas serán muy adecuadas en el caso de que la suposición de normalidad para la variable dependiente Y no se pueda mantener.

Los modelos GAM generalizan también los modelos GLM (y por tanto a los modelos lineales) ya que los GLM corresponden a modelos GAM en el caso en las que las funciones h_i sean todas iguales a la función identidad. Es decir, los modelos GAM son los que tienen una mayor flexibilidad entre todas las clases de modelos analizados hasta ahora, ya que permiten obtener como modelo final ajustado el caso particular de un modelo aditivo, o un GLM (o incluso un modelo lineal), si fuera de esta clase el modelo que mejor se ajustara a los datos.

Ejemplo 8.1

Si consideramos los datos `RIKZ.txt`, incorporados ejecutando `(1)`, cuya nueva variable *riqueza* obtenemos con `(2)` y representamos esta nueva variable como función dependiente de la covariable *tamaño del grano*, vemos en la Figura 8.1, obtenida ejecutando `(3)`, que no puede pensarse en una relación lineal entre ellas.

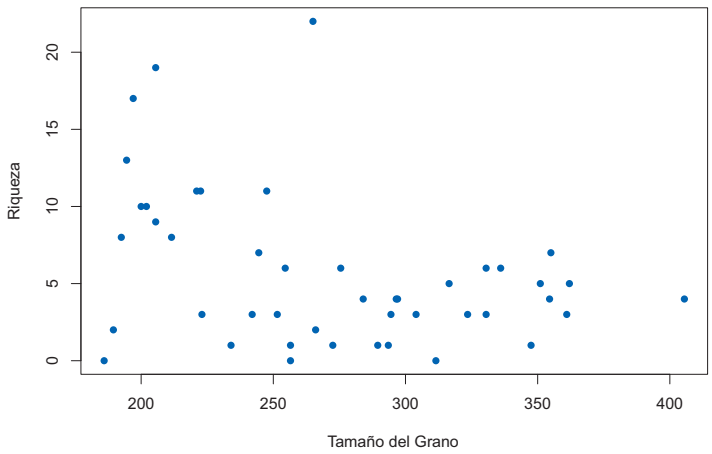

Figura 8.1 : Diagrama de dispersión de la riqueza de las especies frente al tamaño del grano

```
> RIKZ<-read.table("d:\\datos\\RIKZ.txt",header=T)                    (1)
> RIKZ$Richness <- rowSums(RIKZ[, 2:76] > 0)                          (2)
> plot(RIKZ$grainsize,RIKZ$Richness,ylab="Riqueza",                   (3)
+ xlab="Tamaño del Grano",pch=16,col=4)                               (3)
```

Una opción que suele adoptase es transformar los datos de una o ambas variables buscando

la linealidad. En este sentido, las transformaciones logarítmica y raíz cuadrada suelen dar buenos resultados.
Otra posibilidad es considerar una relación no lineal entre ellas, como las consideradas en TA-capítulo 11 pero es mejor considerar modelos más generales GAM, de manera que el resultado final pueda ser un modelo GLM o, incluso lineal aunque desde luego, no en este ejemplo.

8.2. Modelos GAM clásicos

La generalización de los Modelos Aditivos a los GAM es similar a la generalización que se hizo en el capítulo anterior de los Modelos Lineales a los Modelos Lineales Generalizados. En los Modelos Lineales, el estimador de Mínimos Cuadrados, es decir, el que minimiza la suma de los residuos al cuadrado, coincide con el estimador de Máxima Verosimilitud en el caso de que la variable Y tenga distribución normal. En los Modelos Lineales Generalizados extendíamos esta situación a distribuciones para Y *más generales* que la distribución normal. Aquí ocurre lo mismo admitiendo una situación para los modelos GAM en la que

$$g(E[Y|\mathbf{X}]) = g(\mu) = h_0 + h_1(X_1) + ... + h_k(X_k)$$

siendo g la *función link*. No obstante, el proceso de estimación numérica es más complejo debido al modelo supuesto. Las funciones *link* y las distribuciones posibles para Y son las mismas que se consideraron en el capítulo anterior de los GLM.

Como las funciones h_i no serán conocidas las estimaremos, siendo entonces un estimador natural de la media μ

$$\widehat{\mu} = g^{-1}\left(\hat{h}_0 + \hat{h}_1(X_1) + ... + \hat{h}_k(X_k)\right).$$

Esta situación tan general de los modelos GAM hace que en ocasiones se combinen términos paramétricos con términos en las funciones h_i, por ejemplo de la forma

$$g(E[Y|\mathbf{X}]) = g(\mu) = \beta_0 + \beta_1 X_1 + h_1(X_2) + h_2(X_3).$$

Cuando así ocurra, hablaremos de *modelos semi-paramétricos*.

8.2.1. Estimación

Existen dos maneras de estimar las funciones h_i, funciones que en la mayoría de las ocasiones serán la misma para todo $i = 0, 1, ..., k$. Una manera es

mediante el *generalized local scoring algorithm* GLSA según la propuesta de Hastie y Tibshirani (1986), técnica muy dependiente del ordenador.

La otra, que explicamos a continuación, consiste en elegir las funciones h_i dentro de un grupo de *funciones suaves* y estimarlas por métodos no paramétricos.

Para simplificar, consideraremos de momento una sola función h con una sola covariable X, de manera que el modelo sea de la forma

$$Y = h(X) + e$$

En los modelos GAM, la variable de error e se supone que sigue una distribución normal $N(0, \sigma)$.

Si queremos estimar h utilizando las técnicas estudiadas en el capítulo anterior, necesitamos expresar h como un modelo lineal o lineal generalizado. Esto se puede hacer eligiendo una *base* de funciones de la que h (o una aproximación cercana suya) sea miembro. En concreto, eligiendo un conjunto de funciones que suponemos conocidas, $\{b_1, ..., b_q\}$ tal que podamos expresar h de la forma

$$h(x) = \sum_{j=1}^{q} b_j(x)\,\beta_j$$

siendo $\{\beta_1, ..., \beta_q\}$, q parámetros desconocidos. De esta manera, las observaciones y_i se podrán modelizar como

$$y_i = \sum_{j=1}^{q} b_j(x_i)\,\beta_j + e_i$$

con e_i variables independientes e idénticamente distribuidas como una $N(0, \sigma)$, estando hablando ya de un modelo lineal (o lineal generalizado) puesto que, recuérdese, el término lineal se aplica cuando el modelo lo sea en los parámetros β_j.

Entre las bases más utilizadas están los polinomios, aunque éstas se emplean cuando el interés en la aproximación está en un punto concreto de la función h. No obstante, lo más frecuente es que queramos aproximar h en todo su dominio, en cuyo caso, lo habitual es utilizar la Regresión Spline (véase TA-sección 11.4); en concreto, las bases con splines cúbicos son las más frecuentes.

Con las bases suavizamos la función h. El grado de suavizado (*smoothing*), es decir, el número q de covariables significativas utilizado en la aproximación anterior, se podría determinar mediante técnicas de selección de modelos, pero este método tendría muchas dificultades de funcionar bien debido a que los nodos (*knots*) utilizados en la Regresión Spline están espaciados a una misma distancia.

La alternativa que se utiliza, denominada *suavizado con regresión de splines penalizada* (*smoothing with penalized regression splines*), es mantener el grado de suavizado de la base; es decir, el valor de q, y añadir una componente de penalización de manera que, en lugar de minimizar los residuos,

$$\sum_{i=1}^{n} (y_i - y_{t_i})^2$$

como habitualmente hacemos, minimicemos la función

$$\sum_{i=1}^{n} (y_i - y_{t_i})^2 + \lambda \int [h''(x)]^2 dx$$

siendo λ el *parámetro de suavizado* que permite controlar éste.

Al haber supuesto que h es lineal en los parámetros $\boldsymbol{\beta}$, se puede demostrar que la penalización siempre se puede expresar en términos de $\boldsymbol{\beta}$ de la forma

$$\int [h''(x)]^2 dx = \boldsymbol{\beta}^t \mathbf{S} \boldsymbol{\beta}$$

siendo $\mathbf{S}$ una matriz de coeficientes conocidos dependientes de h.

El problema es, por tanto, determinar los $\boldsymbol{\beta}$ que minimicen

$$\sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{q} b_j(x_i)\, \beta_j \right)^2 + \lambda\, \boldsymbol{\beta}^t S \boldsymbol{\beta}$$

supuesto que el parámetro de suavizado λ se ha fijado previamente. En este caso, el estimador de $\boldsymbol{\beta}$ es

$$\widehat{\boldsymbol{\beta}} = \left(\mathbf{X}\mathbf{X}^t + \lambda \mathbf{S} \right)^{-1} \mathbf{X}\mathbf{y}$$

en donde $\mathbf{X}$ es la matriz del diseño.

8.2.2. Validación Cruzada (*Cross validation*)

Cuando un modelo requiere más información que la suministrada por los datos disponibles se suele decir que hay *sobre-ajuste* (*over-fitting*). Esto ocurre cuando, por ejemplo, con unos mismos datos, ajustamos un modelo y luego utilizamos esos mismos datos para analizar la bondad del ajuste de ese modelo, cosa que hacemos al analizar el modelo con los residuos. No obstante, nuestro propósito en ese caso, es realmente averiguar lo bueno que sería el modelo ajustado para predecir nuevos datos.

Este *sesgo* causado por el sobre-ajuste se puede evitar con la *Validación Cruzada* (*Cross validation*) que permite analizar la bondad del ajuste de un

modelo de mejor manera que analizando sólo los residuos, lo cual no permitiría valorar lo bien (o mal) que el modelo ajustado haría nuevas predicciones de casos no observados.

La técnica de la Validación Cruzada consiste en no incluir todos los datos en el ajuste del modelo sino separar unos cuantos antes de efectuar el ajuste, datos que se denominan *conjunto prueba* (*testing set*) y, una vez efectuado el ajuste con los datos restantes, denominados *conjunto de entrenamiento* (*training set*), los datos del conjunto prueba son utilizados para contrastar la bondad del modelo ajustado.

Si sólo dejamos un dato fuera para formar el conjunto prueba, la técnica se denomina *leave-one-out cross validation* y será la que utilicemos para estimar el grado de suavizado en la regresión spline penalizada.

Matemáticamente, estos conceptos se formalizan mediante la adecuada elección del parámetro de suavizado λ. Esta elección es muy importante puesto que si λ es muy pequeño (cercano a 0), la base de splines utilizada tendrá muchas oscilaciones, es decir, el término adicional $\lambda \int [h''(x)]^2 dx$ casi no intervendrá en el proceso de estimación y $\hat{h}$ tendrá muchos máximos y mínimos.

Si por contra λ es muy grande (tendente a ∞) $\hat{h}$ tenderá a ser una recta. El ideal sería encontrar el valor de λ que minimizara

$$\frac{1}{n}\sum_{i=1}^{n}(\hat{h}(x_i) - h(x_i))^2$$

pero como h no es conocida, se elige el valor de λ que minimice la *validación cruzada ordinaria* (*ordinary cross validation*)

$$CV_o = \frac{1}{n}\sum_{i=1}^{n}(\hat{h}_i^{[-i]} - y_i)^2$$

en donde $\hat{h}_i^{[-i]}$ representa el modelo ajustado a todos los datos menos el y_i. Pero calcular CV_o dejando una observación fuera cada vez es ineficaz. Afortunadamente, es

$$CV_o = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{h}(x_i))^2/(1 - h_{ii})^2$$

siendo $\hat{h}$ la estimación de h de ajustar todos los datos y h_{ii} los elementos de la diagonal de la matriz sombrero o de influencia $\widehat{\mathbf{H}}$ (véase TAEA-sección 1.3) con lo que la elección de λ resultaría más sencilla. (¡Cuidado con la notación! Los valores h_{ii} no tienen nada que ver con las funciones h_i.)

En la práctica, las ponderaciones (*weights*) $1 - h_{ii}$ se sustituyen por la ponderación media, $traza(\mathbf{I} - \widehat{\mathbf{H}})/n$, obteniendo la denominada *validación cruzada generalizada* (*generalized cross validation*)

$$CV_g = \frac{n \sum_{i=1}^{n} (y_i - \hat{h}(x_i))^2}{traza(\mathbf{I} - \widehat{\mathbf{H}})^2}$$

eligiendo el valor de λ que minimice CV_g.

8.2.3. Cálculo con R

La estimación y ajuste de modelos GAM lleva implícita la estimación de las funciones h_i. Básicamente hay dos propuestas para estimar estas funciones h_i, usar el *generalized local scoring algorithm* GLSA que ejecuta la función `gam` de la librería `gam`, y el que nosotros utilizaremos aquí, de las *penalized regression splines*, que corresponde con el método descrito en los apartados anteriores, y que se ejecuta con la función `gam` de la librería `mgcv`,

```
gam(modelo,data,family,gamma)
```

en donde los argumentos principales de esta función son `modelo`, para indicar las variables de la regresión. Éstas se expresan como argumentos de la función `s`, función que se utiliza para que el programa haga un *smoothing* de ellas (es decir, utilice una función h de suavizado como las antes mencionadas). En esta función `s` se utilizan, fundamentalmente, tres argumentos: las covariables; el tipo de suavizado que se indica con `bs` (base para el suavizado) y, aunque se le pueden dar varios posibles valores, recomendamos sea `ts` que corresponde a un suavizado por *penalized regression spline* (ver TA-sección 11.4.1); si no se especifica se toma `tp`, muy parecido a `ts`. El tercer argumento de la función `s` es la dimensión de la base, especificado con `k`, y que nosotros recomendamos sea un número alrededor de $20 \cdot n^{2/9}$ siendo n el número de datos observados; si no se fija, se toma igual a 12.

Es decir, habitualmente expresaremos las covariables $x_1, x_2, ...$ en el `modelo` de la forma `s(x1,x2,...,bs="ts",k=12)`, supuesta una dimensión 12 para la base. El grado de suavizado λ lo estimaremos a partir de los datos utilizando la técnica de *validación cruzada* (*cross validation*) y es fijado con el argumento `gamma` de la función; por defecto se toma igual a 1 y, este valor se puede mantener en principio. Algunos autores proponen calcular el valor AIC para diversos λ y elegir aquel λ que proporcione el menor valor AIC.

Los otros argumentos de la función `gam` que utilizaremos son: `data` para indicar el nombre de los datos, en formato data frame, y `family` con los mismos posibles modelos que utilizamos en la función `glm` en el capítulo anterior sobre GLM. Por tanto, utilizando el argumento `family=gaussian`, o lo que es lo mismo, no utilizando este argumento, podremos ajustar un Modelo Aditivo, no generalizado.

Ejemplo 8.2

Los datos `trees` (que están dentro de la librería `mgcv`), son datos de 31 árboles de cerezo negro en los que se midió, a 4 pies del suelo, el diámetro de la circunferencia de su grosor (*Girth*) en pulgadas, su altura (*Height*) en pies, y su volumen de madera (*Volume*) en pies cúbicos. Se cree, como es razonable, que el volumen de madera sea el producto de una función del diámetro y otra de la altura, por lo que se piensa en ajustar un modelo GAM para explicar el volumen en función de las covariables diámetro y altura, expresando el logaritmo del volumen como suma de estas funciones del diámetro y altura, admitiendo una distribución gamma para el volumen. Es decir, se pretende ajustar el modelo

$$\log E[\texttt{Volume}] = h_0 + h_1(\texttt{Height}) + h_2(\texttt{Girth}) + e$$

suponiendo que la variable `Volume` sigue una distribución gamma y una función *link* sea el logaritmo (que hay que especificar puesto que no es la canónica). Es decir, las tres suposiciones que se hacen es, que la variable de respuesta Y sigue, en este caso, una distribución gamma, que los errores e_i siguen una $N(0, \sigma)$ y que la función link que liga la variable de respuesta con las covariables es el logaritmo.
Para ajustar este modelo GAM ejecutamos `(1)`

```
> library(mgcv)
> respuesta<-gam(Volume~s(Height,bs="ts")+s(Girth,bs="ts"),data=trees,      (1)
+ family=Gamma(link=log))                                                  (1)

> summary(respuesta)
Family: Gamma
Link function: log
Formula:
Volume ~ s(Height, bs = "ts") + s(Girth, bs = "ts")
Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.27571    0.01496   218.9   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Approximate significance of smooth terms:
            edf Ref.df     F  p-value
s(Height) 1.023  1.155  25.2 1.48e-05 ***                                  (2)
s(Girth)  2.382  2.994 227.8  < 2e-16 ***                                  (2)
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.974   Deviance explained = 97.8%
              (3)                          (4)
GCV score = 0.0080891  Scale est. = 0.0069395  n = 31
                                    (5)
```

Los p-valores, dados en `(2)`, parecen indicarnos que ambas covariables son adecuadas. Además, el valor de R^2 dado en `(3)` o mejor, el valor de `Deviance` dado en `(4)`, parecen indicar que el modelo es adecuado. Es decir, que el modelo ajustado

$$\log E[\texttt{Volume}] = 3'27571 + \texttt{s(Height)} + \texttt{s(Girth)}$$

es adecuado. Un estimador de la varianza aparece en `(5)`, siendo $\widehat{\sigma^2} = 0'00694$.
No obstante, si analizamos qué covariables tiene residuos dentro de la banda de confianza ejecutando

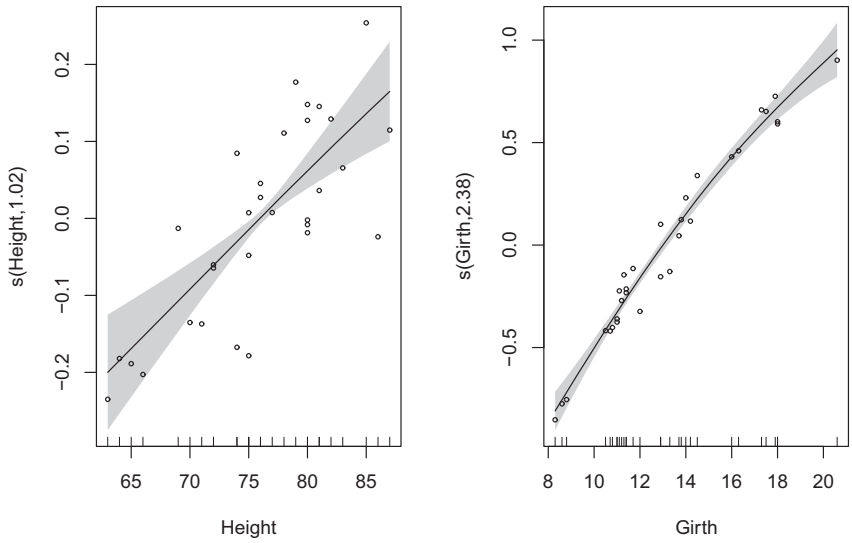

Figura 8.2 : Gráfico de bondad del ajuste del modelo GAM Gamma

```
> par(mfrow=c(1,2))
> plot(respuesta,select=1,scale=0,residuals=T,shade=T,pch=1,cex=0.5,xlab="Height")
> plot(respuesta,select=2,scale=0,residuals=T,shade=T,pch=1,cex=0.5,xlab="Girth")
```

obtenemos la Figura 8.2 que nos señala que los residuos de `s(Height)` se salen mucho de las bandas de confianza. Los de `s(Girth)` aún se podrían aceptar. Parece necesario, por tanto, utilizar Métodos Robustos.
Por decir también cómo ejecutaría el ajuste la librería `gam`, ponemos a continuación los pasos que debería de efectuar con esta librería y la función `gam` de ella. Anunciamos que el resultado es muy similar al obtenido con el paquete `mgcv`

```
> library(gam)
> summary(gam(Volume~s(Height)+s(Girth),data=trees,family=Gamma(link=log)))

Call: gam(formula = Volume ~ s(Height) + s(Girth), family = Gamma(link = log),
    data = trees)
Deviance Residuals:
      Min        1Q    Median        3Q       Max
-0.146302 -0.050607  0.007386  0.047357  0.142059

(Dispersion Parameter for Gamma family taken to be 0.0073)

    Null Deviance: 8.3172 on 30 degrees of freedom
Residual Deviance: 0.1611 on 21.9998 degrees of freedom
```

```
AIC: 147.8686

Number of Local Scoring Iterations: 5

DF for Terms and F-values for Nonparametric Effects

            Df Npar Df Npar F   Pr(F)
(Intercept)  1
s(Height)    1       3 0.7190 0.55127
s(Girth)     1       3 3.5679 0.03051 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

8.3. Modelos GAM robustos

Una de las aplicaciones más habituales de los modelos GAM en Ciencias de la Salud es la de modelizar recuentos semanales de enfermedades infecciosas; más en concreto, detectar incrementos repentinos en el número de casos comunicados de una determinada enfermedad, u otras desviaciones de un modelo previamente establecido para pasados recuentos.

Aunque estas desviaciones se pueden detectar habitualmente mediante un Análisis Exploratorio de Datos, el propósito que se persigue es establecer un procedimiento automático de detección. Para ello se ajusta un modelo y se comparan los valores observados con los predichos con el modelo para detectar desviaciones significativas entre ambas. Como los recuentos semanales de enfermedades infecciosas (gripe, Sida, Hepatitis C, etc.) muestran una fuerte y no lineal variación estacional, es habitual utilizar modelos GAM para ajustar los datos.

Desgraciadamente, los modelos GAM son muy sensibles a la presencia de datos anómalos y unos pocos de ellos pueden afectar seriamente a los estimadores de las funciones h, por lo que un *buen ajuste* de un modelo GAM, puede hacer que los valores predichos estén cercanos a los incrementos anómalos (pero no erróneos) haciéndolos imperceptibles, produciéndose así un indeseable efecto de enmascaramiento. Además, puede ocurrir que datos correctos se desvíen significativamente del modelo. Es decir, necesitamos un modelo insensible a datos extremos que, después de ajustado, permita identificar datos extremos.

Existen Métodos Robustos que generalizan los dos métodos clásicos antes estudiados. En un reciente trabajo de Croux et al. (2012) se robustifica el método antes considerado de splines penalizadas, que era el segundo método clásico, desarrollado en Wood (2006). Pero este trabajo no tiene asociada, de momento, una librería de R para poder ejecutar los resultados obtenidos allí.

El primer método basado en el algoritmo GLSA, desarrollado en Hastie y

Tibshirani (1986), sí tiene análogo robusto. Se trata del trabajo de Alimadad y Salibian-Barrera (2011), en donde los autores definen unos M-estimadores robustos para modelos GAM y crean una librería en R para obtenerlos: la librería `rgam`. La función para hacer los ajustes del modelo robusto lleva el mismo nombre que la librería.

```
rgam(x,y,family,cv.method="rcv",k,epsilon=1e-5,alpha,max.it=500)
```

en donde los argumentos principales de esta función son `x` e `y` que representan los datos en formato vector (o matriz) para la variable independiente y dependiente respectivamente; `k` es utilizado para indicar la *tuning constant* de Huber; `alpha` es un parámetro a determinar como veremos más adelante que sea óptimo en el proceso de suavizado, y `family` es un argumento con los mismos posibles modelos que utilizamos en la función `glm` en el capítulo anterior sobre GLM. Los demás se dejan como están.

Ejemplo 8.3

Consideremos los datos `ili.visits` de la librería de R, `rgam`, en los que aparecen datos del número de visitas semanales (*visits*), desde la semana 40 a la semana 20 del año siguiente de los años 2006, 2007, 2008 y 2009; es decir, de 33 semanas (*week*) de cada una de las cuatro temporadas (*season*), de pacientes con síntomas de gripe, incluida la gripe A (N1H1).
Alimadad y Salibian-Barrera (2011) consideran este ejemplo ajustando un modelo GAM robusto. Como la variable de respuesta es el número de visitas, parece razonable utilizar una distribución Poisson para la variable de respuesta Y, pero lo que los autores del trabajo no consideran es que la media de los datos y_i es 10.026'28 y la varianza 36.389.127, lo que deja sin sentido un ajuste considerando la distribución Poisson en donde la media y la varianza deben de ser similares. Nosotros lo haremos para la variable dependiente $Y/1000$ que sí cumple este requisito aproximadamente.
Si realizamos un ajuste estimando las funciones h con el algoritmo GLSA y un suavizado por *penalized regression spline* clásico, utilizando la función `gam` ejecutaríamos `(1)`, obteniendo a continuación los coeficientes estimados.

```
> library(mgcv)
> expla2000<-gam(visits/1000~s(week),data=ili.visits,family=poisson)       (1)

> expla2000$coefficients
(Intercept)   s(week).1   s(week).2   s(week).3   s(week).4
  2.2180353  -0.1194091  -1.1112055  -0.9697781   0.4541091
  s(week).5   s(week).6   s(week).7   s(week).8   s(week).9
  0.1624136   0.1424738  -0.1701618  -0.4335908   0.5957964
```

Para utilizar la función `rgam` de un ajuste robusto de GAM, primero debemos determinar el valor de `alpha` en donde es óptima. Para ello elegimos, en este caso, el intervalo de valores `14:25/80` en `(2)` y ejecutamos `(3)`, supuesto que estamos considerando un valor de la *tuning constant* igual a $0'5$,

```
> library(rgam)
```

```
> x <- ili.visits$week
> y <- ili.visits$visits

> rgam(x=x,y=y/1000,family="poisson",cv.method="rcv",k=0.5,                    (3)
+ epsilon=1e-5,alpha=14:25/80,max.it=500)$opt.alpha
                        (2)
[1] 0.2875                                                                    (4)
```

y vemos en (4) que el valor óptimo es $0'2875$. Si fuera un extremo del intervalo en donde buscamos el óptimo, deberíamos de volver a ejecutar la función anterior, moviendo el intervalo, de manera que el óptimo alcanzado no quede nunca en el extremo. En este caso no es necesario porque es $14/80 < 0'2875 < 25/80$.
Ahora, que ya sabemos en donde ejecutar la función que nos da las estimaciones robustas, ejecutamos (5)

```
> a12<-rgam(x=x,y=y/1000,family="poisson",cv.method="rcv",k=0.5,              (5)
+ epsilon=1e-5,alpha=0.2875,max.it=500)                                       (5)
```

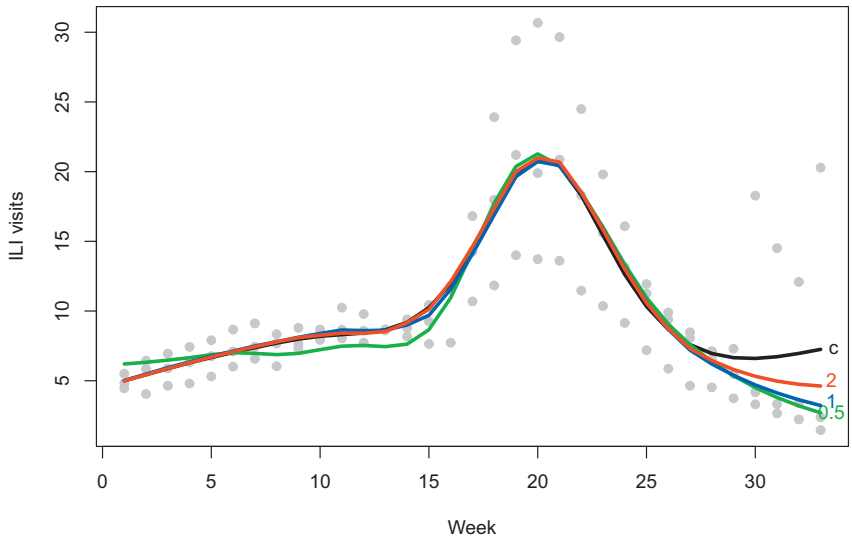

Figura 8.3 : Ajuste clásico y robusto para varias tuning constants

Un par más de valores de la *tuning constant*, $k = 1$ y $k = 2$, buscando primero el óptimo, se ajustan a continuación.

```
> rgam(x=x,y=y/1000,family="poisson",cv.method="rcv",k=1,
+ epsilon=1e-5,alpha=29:40/80,max.it=500)$opt.alpha
```

```
[1] 0.3875
> a13<-rgam(x=x,y=y/1000,family="poisson",cv.method="rcv",k=1,
+ epsilon=1e-5,alpha= 0.3875,max.it=500)

> rgam(x=x,y=y/1000,family="poisson",cv.method="rcv",k=2,
+ epsilon=1e-5,alpha=24:35/80,max.it=500)$opt.alpha
[1] 0.4125
> a15<-rgam(x=x,y=y/1000,family="poisson",cv.method="rcv",k=2,
+ epsilon=1e-5,alpha= 0.4125,max.it=500)
```

Si representamos tanto el ajuste clásico como estos tres ajustes robustos tenemos la Figura 8.3 obtenida ejecutando

```
> plot(x,y/1000,xlab="Week",ylab="ILI visits",pch=19,col="grey75")

> prediccion2000 <- predict(expla2000, type="response")        #clásico
> lines(x[order(x)],prediccion2000[order(x)],lwd=3,col=1)
> text(33.6,7.3,"c")

> pr.rgam.a12 <- predict(a12, type="response")                 # k=0.5
> lines(x[order(x)], pr.rgam.a12[order(x)], lwd=3, col="3")
> text(33.5,2.7,"0.5",col=3)

> pr.rgam.a13 <- predict(a13, type="response")                 # k=1
> lines(x[order(x)], pr.rgam.a13[order(x)], lwd=3, col="4")
> text(33.5,3.5,"1",col=4)

> pr.rgam.a15 <- predict(a15, type="response")                 # k= 2
> lines(x[order(x)], pr.rgam.a15[order(x)], lwd=3, col="2")
> text(33.5,5,"2",col=2)
```

En esta figura se observa que, si seguimos aumentando la tuning constant nos acercamos al ajuste clásico, el de color negro, muy afectado por los outliers del final. El azul, correspondiente a $k = 1$ proporciona un ajuste óptimo por la izquierda (igual al clásico) y todavía no se ve afectado por los datos anómalos. Nos quedamos con este ajuste, el `a13`. Sobre la elección de la constante de Huber se puede leer el artículo de García Pérez (2014).

Ejemplo 8.4

Los datos `brain` de la librería `gamair` procedentes de Landau et al. (2003), corresponden a $n = 1567$ vóxeles, siendo el vóxel la unidad cúbica que compone un objeto tridimensional y que constituye la unidad mínima procesable de una matriz tridimensional, equivalente al píxel en un objeto de tres dimensiones. En cada uno de estos vóxel se midió, entre otras cosas, su localización (columnas `X` e `Y` de los datos) y una medida de la actividad cerebral denominada `medFPQ` y que es la mediana de tres observaciones del *Fundamental Power Quotient*, por supuesto en cada (`X,Y`).

El propósito es explicar la actividad cerebral `medFPQ` en función de las coordenadas `X,Y`. Dado que el modelo GAM es el más general, comenzaremos ajustando `(1)`. Dado que tenemos $n = 1567$ datos, el valor de la dimensión de la base sugerido sería $k = 20 \cdot 1567^{2/9} = 102'5759$. Cogeremos 100 que además es el máximo valor admitido. Con estas sentencias obtenemos la Figura 8.4 ejecutando `(2)`.

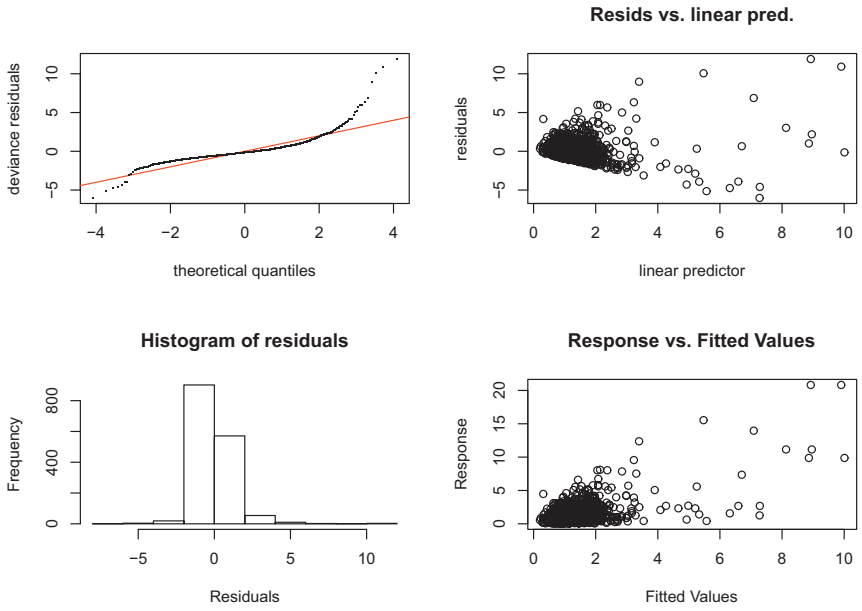

Figura 8.4 : Gráficos de análisis del modelo `m0`

```
> library(mgcv)
> library(gamair)
> data(brain)
> m0<-gam(medFPQ~s(Y,X,k=100),data=brain)                    (1)
> gam.check(m0)                                               (2)
```

Los gráficos de la izquierda ya muestran problemas con la normalidad de los residuos. Los dos de la derecha muestran que no es posible mantener la suposición de que la varianza sea constante.
Para analizar una posible relación entre la varianza de las y_i (que es la de los residuos) y la de sus medias μ_i, que son los valores ajustados, digamos `ajusm0<-m0$fitted`, dado que sí puede admitirse una media cero para los residuos `residuals(m0)` (ya que sí parecen agruparse alrededor de 0), su varianza será igual a sus cuadrados, porque, en definitiva, si hay una relación de potencia η entre la varianza de las y_i y sus medias del tipo

$$V(Y) \propto {\mu_i}^{\eta}$$

deberá ser también cierta una relación del tipo

$$\texttt{residuals(m0)}^2 \propto \texttt{ajusm0}^{\eta}$$

con lo que del ajuste lineal

```
> lm(log(residuals(m0)^2)~log(ajusm0))

Call:
```

```
lm(formula = log(residuals(m0)^2) ~ log(ajusm0))

Coefficients:
(Intercept)  log(ajusm0)
     -1.961        1.922
```

el valor razonable para η es 2. Es decir, la varianza de los datos se incrementa con el cuadrado de la media. La distribución Gamma(a, b) en el caso $a = 1$ cumple esta condición de ser la varianza proporcional al cuadrado de la media (CB-sección 4.5.4). Por tanto, parece razonable considerar el ajuste

```
> m01<-gam(medFPQ~s(Y,X,k=100),data=brain,family=Gamma(link=log))
> gam.check(m01)
```

observándose en la Figura 8.5 un par de datos anómalos. Por eso eliminamos estos dos datos anómalos ejecutando otra vez

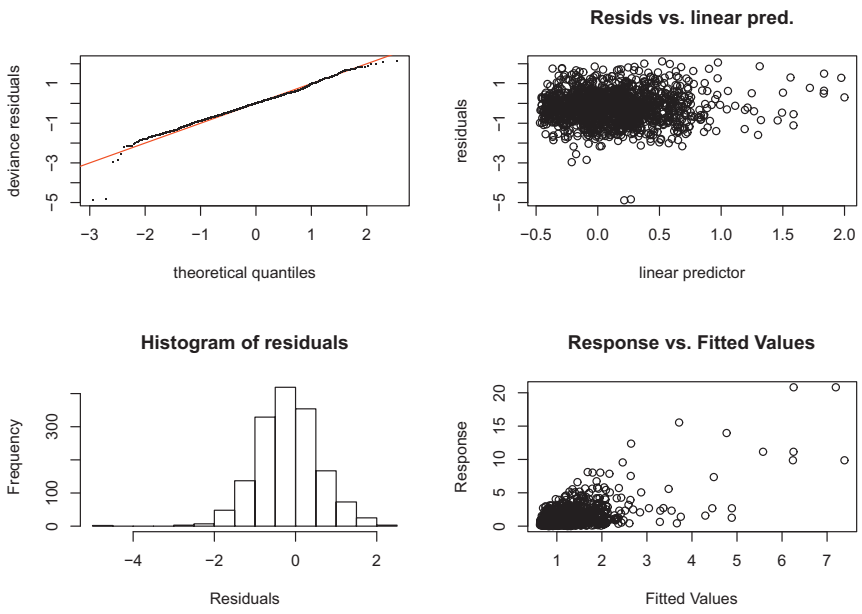

Figura 8.5 : Gráficos de análisis del modelo `m01`

```
> brain2<-brain[brain$medFPQ>5e-3,]
> m02<-gam(medFPQ~s(Y,X,k=100),data=brain2,family=Gamma(link=log))
> gam.check(m02)
```

obteniendo ahora la Figura 8.6 en donde se aprecia un mejor ajuste.
Si queremos representar el modelo `m02` ajustado podemos ejecutar `(3)`, después de ver con detalle el modelo,

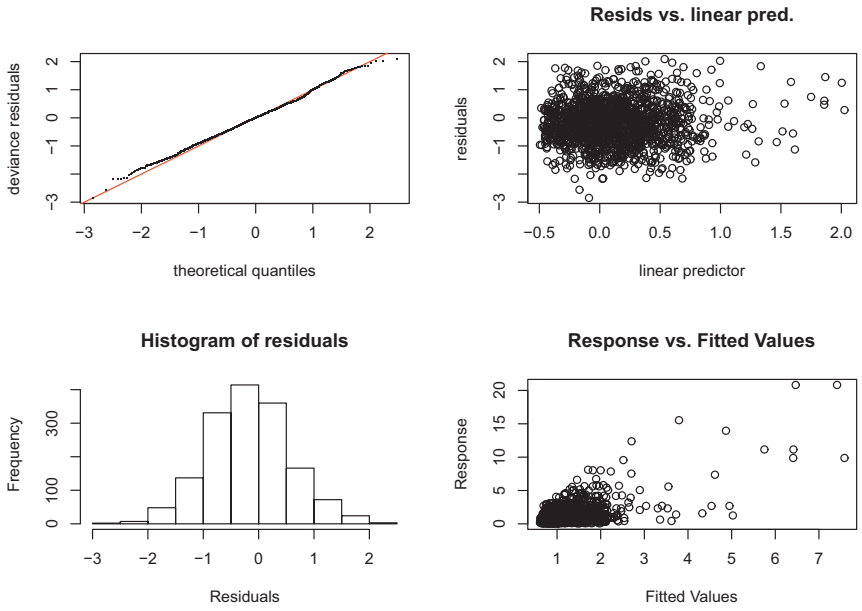

Figura 8.6 : Gráficos de análisis del modelo `m02`

```
GCV score: 0.6216871
> summary(m02)

Family: Gamma
Link function: log

Formula:
medFPQ ~ s(Y, X, k = 100)

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.12031    0.01954   6.157  9.5e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
          edf Ref.df     F p-value
s(Y,X) 60.61  77.82 4.212  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.307   Deviance explained = 26.4%
GCV score = 0.62169  Scale est. = 0.5972    n = 1564
```

```
> vis.gam(m02,plot.type="contour",too.far=0.03,color="gray",n.grid=60,zlim=c(-1,2))     (3)
```

obteniendo la Figura 8.7.

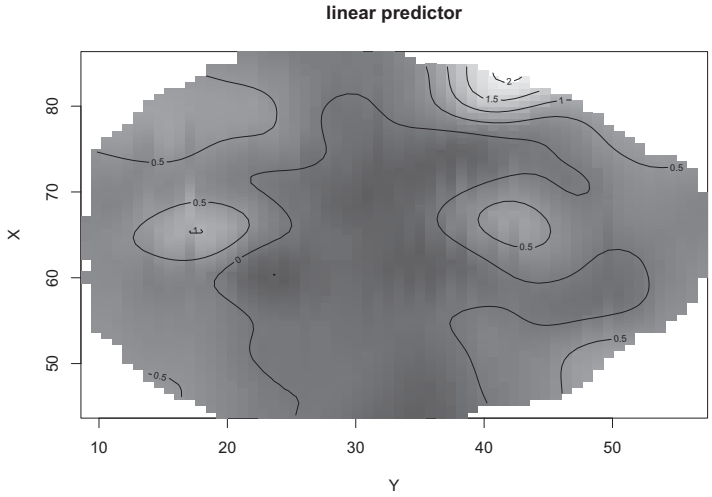

Figura 8.7 : Gráfico de contornos del modelo estimado `m02`

El modelo ajustado ha sido, por tanto,

$$\log E[\texttt{medFPQ}] = 0'12031 + h(\texttt{X}, \texttt{Y}) + e$$

suponiendo que la variable `medFPQ` sigue una distribución gamma y una función *link* sea el logaritmo (que hay que especificar puesto que no es la canónica).

Ejemplo 4.5 (continuación)

Si queremos ajustar un modelo GAM (semi-paramétrico) a los datos del Ejemplo 4.5 considerando como covariables de riesgo: la distancia a los centros de polución y a las principales carreteras, además de sexo, edad, previos casos de fiebre alta, que haya al menos un fumador en la casa y la latitud (`x1` y longitud (`x2` como covariables que deben suavizarse mediante regresión spline `tp` (la que toma por defecto) ejecutaríamos la siguiente secuencia.

```
> asma1$y <- as.integer(!as.integer(asma1$Asma)-1)
> ccasma <- coordinates(asma1)
> asma1$x1 <- ccasma[,1]
> asma1$x2 <- ccasma[,2]
> asma1$dist1 <- sqrt(asma1$Fuente1)
> asma1$dist2 <- sqrt(asma1$Fuente2)
> asma1$dist3 <- sqrt(asma1$Fuente3)
> asma1$discarr <- sqrt(asma1$distcarr2)
> asma1$fuma <- as.factor(as.numeric(asma1$Nfumadores>0))
```

```
> asma1$Generof<- as.factor(asma1$Genero)
> asma1$FiAltaf<- as.factor(asma1$FiALta)

> library(mgcv)
> gasma<-gam(y~1+dist1+dist2+dist3+discarr+Generof+Age+FiAltaf+fuma+s(x1,x2),
+ data=asma1[asma1$Genero==1 | asma1$Genero==2, ], family=binomial)

> summary(gasma)

Family: binomial
Link function: logit

Formula:
y ~ 1 + dist1 + dist2 + dist3 + discarr + Generof + Age + FiAltaf +
    fuma + s(x1, x2)

Parametric coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.0326698  0.9196842  -2.210   0.0271 *
dist1        0.9822455  6.0721436   0.162   0.8715
dist2       -9.5791492  5.7719999  -1.660   0.0970 .
dist3       11.2247362  7.8743643   1.425   0.1540
discarr      0.0001479  0.0001717   0.861   0.3890
Generof2    -0.3476860  0.1562020  -2.226   0.0260 *
Age         -0.0679032  0.0382349  -1.776   0.0757 .
FiAltaf1     1.1881330  0.1875415   6.335 2.37e-10 ***
fuma1        0.1651213  0.1610364   1.025   0.3052
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
           edf Ref.df Chi.sq p-value
s(x1,x2) 2.001  2.001  7.004  0.0302 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.0403   Deviance explained = 4.94%
UBRE = -0.12348  Scale est. = 1         n = 1283
```

De este análisis se deduce que son significativas las covariables Antecedentes de Fiebre Alta (p-valor 2.37e-10) y Sexo (p-valor 0.0260). El residuo del término de covariables (latitud,longitud) suavizadas también es significativo (p-valor 0.0302) lo que sugiere que puede haber alguna variación espacial no explicada.

Capítulo 9

Bibliografía

Alimadad, A. y Salibian-Barrera, M. (2011). An Outlier-Robust fit for Generalized Additive Models with applications to disease outbreak detection. *Journal of the American Statistical Association*, **106**, 719-731.

Berman, M. y Diggle, P.J. (1989). Estimating weighted integrals of the second-order intensity of a spatial point process. *Journal of the Royal Statistical Society*, Serie B, **51**, 81–92.

Bivand, R.S., Pebesma, E.J. y Gómez-Rubio, V. (2013). *Applied Spatial Data Analysis with R*, 2a edición. Springer.

Burrough, P.A. y McDonnell, R.A. (1998). *Principles of Geographical Information Systems*. Oxford University Press.

Cantoni, E. y Ronchetti, E. (2001). Robust inference for generalized linear models. *Journal of the American Statistical Association* **96**, 1022-1030.

Clayton, D. y Hills, M. (1993). *Statistical Models in Epidemiology*. Oxford University Press, Oxford.

Cressie, N.A.C. (1993). *Statistics for Spatial Data*. Wiley

Croux, C., Gijbels, I. y Prosdocimi, I. (2012). Robust estimation of mean and dispersion functions in Extended Generalized Additive Models. *Biometrics*, **68**, 31-44.

Diggle, P.J. (1985). A kernel method for smoothing point process data. *Applied Statistics*, **34**, 138–147.

Diggle, P.J. (2003). *Statistical Analysis of Spatial Point Patterns*, 2a edición. Arnold.

Diggle, P. J., Gómez-Rubio, V., Brown, P. E., Chetwynd, A., y Gooding, S. (2007). Second-order analysis of inhomogeneous spatial point processes using case-control data. *Biometrics*, **63**, 550–557.

Fahrmeir, L. y Tutz, G. (1994). *Multivariate Statistical Modelling Based on Generalized Linear Models*. Springer-Verlag.

Feigl, P. y Zelen, M. (1965). Estimation of exponential probabilities with concomitant information. *Biometrics* **21**, 826-838.

García Pérez, A. (1998). *Problemas Resueltos de Estadística Básica*. UNED. Colección Educación Permanente.

García Pérez, A. (2005a). *Métodos Avanzados de Estadística Aplicada. Técnicas Avanzadas*. UNED. Colección Educación Permanente.

García Pérez, A. (2005b). *Métodos Avanzados de Estadística Aplicada. Métodos Robustos y de Remuestreo*. UNED. Colección Educación Permanente.

García Pérez, A. (2008a). *Estadística Aplicada: Conceptos Básicos*. Segunda edición. UNED. Colección Educación Permanente.

García Pérez, A. (2008b). *Ejercicios de Estadística Aplicada*. UNED. Colección Cuadernos de la UNED.

García Pérez, A. (2008c). *Estadística Aplicada con R*. Editorial UNED. Colección Varia.

García Pérez, A. (2010). *Estadística Básica con R*. Editorial UNED. Colección Grado.

García Pérez, A. (2014). *La Interpretación de los Datos. Una Introducción a la Estadística Aplicada*. Editorial UNED. Colección Temática.

García Pérez, A. (2014). The p-value line. A way to choose the tuning constant in tests based on the Huber M-estimator. *Test*, **23**, 536-555.

García Pérez, A. y Cabrero Ortega, Y. (2010). Técnicas estadísticas clásicas y robustas aplicadas a la investigación histórica: Análisis de datos reales. En *Actas del XXXII Congreso Nacional de Estadística e Investigación Operativa*.

García Pérez, A. y Cabrero Ortega, Y. (2015). A spatial influence function and local outliers: Applications to Geographical Information Systems. En *Proceedings of the 8th International Conference of the ERCIM Working Group on Computational and Methodological Statistics*.

García Pérez, A. y Cabrero Ortega, Y. (2015). On robustness for spatial data. Enviado.

Hastie, T. y Tibshirani, R. (1986). Generalized additive models. *Statistical Science*, **1**, 297-318.

Hastie, T. y Tibshirani, R. (1990). *Generalized additive models*. Chapman and Hall.

Heyde, C.C. (1997). *Quasi-likelihood and its Applications*. Springer-Verlag.

Kelsall, J. E. y Diggle, P. J. (1998). Spatial variation in risk: a nonparametric binary regression approach. *Applied Statistics*, **47**, 559-573.

Korich, D.G., Marshall, M.M., Smith, H.V, O'Grady, J., Bukhari, C.R., Fricker, Z., Rosen, J.P., y Clancy, J.L. (2000). Inter-laboratory comparison of the cd-1 neonatal mouse logistic dose-response model for *Cryptosporidium parvum* oocysts. *Journal of Eukaryotic Microbiology* **47**, 294-298.

Künsch, H.R., Stefanski, L.A., y Carroll, R.J. (1989). Conditionally unbiased bounded-influence estimation in general regression models, with applications to generalized linear models. *Journal of the American Statistical Association* **84**, 460-466.

Landau, S., Ellison-Wright, I.C. y Bullmore, E.T. (2003). Tests for a difference in timing of psysiological response between two brain regions measured by using functional magnetic resonance imaging. *Applied Statistics*, **53**, 63-82.

Lindenmayer, D.B., Cunningham, R.B., Tanton, M.T., Smith, A.P., y Nix, H.A. (1990). The conservation of arboreal marsupials in the montane ash forest of the central highlands of Victoria, south-east Australia: I. Factors influencing the occupancy of trees with hollows. *Biological Conservation* **54**, 111-131.

Lindenmayer, D.B., Cunningham, R.B., Tanton, M.T., Nix, H.A., y Smith, A.P. (1991). The conservation of arboreal marsupials in the montane ash forest of the central highlands of Victoria, south-east Australia: III. The habitat requirements of Leadbeater's possum Gymnobelideus leadbeateri and models of the diversity and abundance of arboreal marsupials. *Biological Conservation* **56**, 295-315.

Maronna, R.A., Martin, R.D. y Yohai, V.J. (2006). *Robust Statistics. Theory and Methods*. Wiley.

McCullagh, P. y Nelder, J.A. (1989). *Generalized Linear Models*, 2a edición. Chapman and Hall.

Morgenthaler, S. (1992). Least-absolute-deviations fits for generalized linear models. *Biometrika* **79**, 747-754.

Nelder, J.A. y Wedderburn, R.W.M. (1972). Generalized linear models. *Journal of Royal Statistical Society*, A **135**, 370-384.

Numata, M. (1961). Forest vegetation in the vicinity of Choshi. Coastal flora and vegetation at Choshi, Chiba Prefecture. IV. *Bulletin of Choshi Marine Laboratory, Chiba University*, **3**, 28–48 (en Japonés).

Phelps, K. (1982). Use of the complementary log-log function to describe dose-response relationships in insecticide evaluation field trials. En *Lecture Notes in Statistics, 14. GLIM.82: Proceedings of the International Conference on Generalized Linear Models*, ed. R. Gilchrist. Springer-Verlag.

Pregibon, D. (1982). Resistant fits for some commonly used logistic models with medical applications. *Biometrika* **38**, 485-498.

Rhind, D. (1990). Global Database and GIS. En *The Association for Geographic Information Yearbook 1990*, Foster, M.J. and Shand, P.J. (eds). Londres, Taylor and Francis and Miles Arnold.

Rikken, M.G.J. y van Rijn, R.P.G. (1993). *Soil pollution with heavy metals - an inquiry into spatial variation, cost of mapping and the risk evaluation of copper, cadmium, lead and zinc in the floodplains of the Meuse west of Stein, the Netherlands*. Tesis Doctoral, Dept. de Geografía Física, Universidad de Utrecht.

Ripley, B.D. (1977). Modelling spatial patterns (with discussion). *Journal of the Royal Statistical Society*, Serie B, **39**, 172–212.

Stefanski, L.A., Carroll, R.J. y Ruppert, D. (1986). Optimally bounded score functions for generalized linear models with applications to logistic regression. *Biometrika* **73**, 413-424.

Strauss, D.J. (1975). A model for clustering. *Biometrika*, **63**, 467–475.

Venables, W. N. y Dichmont, C. M. (2004). A generalised linear model for catch allocation: an example from Australia's northern prawn fishery. *Fisheries Research*, **70**, 409–426.

Vélez, R. y García Pérez, A. (1993). *Principios de Inferencia Estadística*. UNED.

Waller, L.A. y Gotway, C.A. (2004). *Applied Spatial Statistics for Public Health Data*. John Wiley and Sons.

Wedderburn, R.W.M. (1974). Quasi-likelihood functions, generalized linear models, and the Gauss-Newton method. *Biometrika* **61**, 439-447.

Wood, S.N. (2006). *Generalized Additive Models. An Introduction with R*. Chapman and Hall/CRC.

UNED Editorial

Juan del Rosal, 14
28040 MADRID
Tel. Dirección Editorial: 913 987 521