Grado en Matemáticas

TRABAJO FIN DE GRADO

Análisis de la siniestralidad vial mediante modelos de datos de conteo

Estadística e Investigación Operativa

Autor: José Manuel López Jiménez

Director: José Luis Pino Mejías

Sevilla, Junio 2019

Índice general

Resumen

El objetivo de este trabajo es estudiar la siniestralidad vial mediante la aplicación de modelos de datos de conteo, los cuales previamente comentaremos para después poder aplicarlos a nuestra base de datos. Comenzaremos por el estudio del Modelo Lineal Generalizado. Luego nos adentraremos en los modelos de datos de conteo como lo son el de Poisson y el Binomial Negativa. También veremos otros tipos de modelos más complejos que se podrían aplicar (como los truncados o en dos partes). Veremos los posibles problemas que se nos puede plantear debido al uso de dichas variables de conteo, cómo detectarlos y cómo solucionarlos. Para concluir, aplicaremos todo lo estudiado a nuestra base de datos que describiremos detalladamente en el último capítulo del trabajo, siempre con ayuda del software R.

Abstract

The aim of this work is to study the road accident rate by means of the application of models for count of data, which previously will be commented in order to can be applied to our database. We shall start with the so-called Generalized Linear Model. Then we shall enter the models for count of data such as the one due to Poisson and the so-called Negative Binomial. Moreover, we shall deal with other kinds of more complex models that could be applied (as for instance, the truncate model or the two-part model). We will analyze the problems that are likely to arise due to the use of the mentioned variables of count, as well as how to detect them and how to solve them. To finish, and always with the help of the software R, all the items that have been studied along this memory will be applied to our database which will be described in detail in the final chapter of this work.

Introducción

El trabajo titulado "Análisis de la siniestralidad vial mediante datos de conteo" consta de 7 capítulos de los cuales los 6 primeros abordan los diferentes modelos estadísticos y sus características, acabando en el último capítulo con su aplicación a una base de datos.

En primer lugar, se comienza con el concepto de modelo y la necesidad de tenerlo para poder realizar estudios estadísticos para comprender el medio. Como los datos de conteo se pueden ver como un caso particular del Modelo Lineal Generalizado (MLG), dicho modelo se estudia en este primer capítulo. Así pues, se estudian sus diferentes componentes (aleatoria, sistemática y función link). Se mencionan las hipótesis realizadas sobre dicho modelo. También se realiza una estimación de los parámetros mediante el Método de Máxima Verosimilitud y se hace inferencia, es decir, se ve como se ajusta nuestro modelo a un conjunto de observaciones y se evalúa la adecuación del modelo mediante diferentes medidas de bondad de ajuste. Como el ajuste no es perfecto, se introduce el concepto de residuos, que nos marca la discrepancia entre el modelo obtenido y los datos. Se finaliza con una interpretación del modelo.

En el segundo capítulo, se introduce el concepto de dato y variable de conteo, así como los diferentes problemas que surgen al utilizar dichas variables. Por este motivo se hace una lista con los diferentes modelos que son adecuados para el estudio de dichas variables de conteo.

En el tercer capítulo se estudia el modelo de regresión de Poisson, modelo base para el estudio de variables de conteo pero con un problema, la equidispersión, es decir, la igualdad entre media y varianza, cosa que provoca que dicho modelo sea poco aplicable en la práctica ya que, por lo general, los datos poseen una mayor varianza que media (hecho que se conoce como sobredispersión). Es por ello que el trabajo continúa con el estudio de modelos que podamos aplicar. Además, se estudian estadísticos para ver la bondad de ajuste del modelo y se dan criterios para estudiar la dispersión de los datos. Se concluye el capítulo con el estudio del modelo de Regresión Bivariante de Poisson, modelo cuyo vector respuesta es bidimensional.

En el cuarto capítulo se estudia el modelo de regresión Binomial Negativa, modelo estrella para el estudio de las variables de conteo y de datos que poseen sobredispersión. En el caso de sobredispersión constante, se parte de un modelo mixto Poisson-Gamma. En el caso de sobredispersión variable, podemos obtener la binomial negativa desde dos caminos. La primera forma es viéndolo como un modelo de Poisson con heterogeneidad Gamma de media uno. La segunda forma es tratándolo como una función de probabilidad propiamente dicha, como un miembro de la familia del MLG.

En el quinto capítulo se aborda el problema del exceso de ceros en datos de conteo. Dicha cantidad de ceros no es compatible con los modelos de Poisson y Binomial Negativa. A continuación se introduce el concepto de falsos ceros y ceros auténticos. Se plantean diferentes tipos de modelos: truncados en cero, Hurdle y cero inflado.

Se concluye el estudio teórico en este sexto capítulo con dos criterios de selección de modelos, uno basándose en el estadístico AIC y el otro en el estadístico BIC.

En el séptimo y último capítulo se encuentra la parte práctica del trabajo, donde se aplican los modelos estudiados a lo largo del trabajo a nuestra base de datos elaborada en

el Apéndice C. Se comparan dichos modelos, seleccionando los más adecuados para así finalizar con las conclusiones. Todo ello se ha realizado con el software estadístico R.

La parte del código tanto del trabajo como de la depuración de los datos como la utilizada auxiliarmente se encuentran en los apéndices del trabajo.

Para concluir, se muestra la bibliografía utilizada para elaborar dicho trabajo.

Índice de figuras

Índice de cuadros

Capítulo 1

Modelo Lineal Generalizado

1.1. Motivación

El ser humano está constantemente creando "modelos" para así comprender lo que ocurre a partir de las observaciones que realizamos de nuestro entorno, pudiendo realizar incluso predicciones sobre ellos. En el ámbito científico, un modelo que explica un fenómeno se expresa en forma matemática. Este proceso se conoce como *modelización matemática*, y cuando los fenómenos son probabilísticos, hablamos de *modelado estadístico o estocástico*. Según López-González (2011), un modelo pretende explicar la variación de una respuesta a partir de la relación conjunta de dos fuentes de variabilidad, una de carácter determinista y otra aleatoria: *Respuesta = componente sistemático + componente aleatorio.*

Otros autores lo denotan por: *Datos = Modelo + Error*, asociando datos a las observaciones que se quieren realizar, modelo a la función que se introduce con objeto de explicar los datos, y dado que la variabilidad recogida en datos no termina de estar explicada, se introduce el término error, que contiene la discrepancia o falta de ajuste entre los datos y el modelo. Es deseable que el modelo sea una buena representación de los datos, de forma que el error se reduzca lo máximo posible.

De todo esto se encarga el modelado estadístico. Éste debe responder a dos criterios: (a) *bondad de ajuste:* la inclusión de parámetros en el modelo en beneficio de una mejor representación de los datos y su correspondiente disminución del error. (b) *principio de parsimonia:* la selección de los parámetros que formen parte del modelo de tal modo que éste se convierta en una representación simple y sobria de la realidad.

En esta construcción del modelo más parsimonioso que explique la variable de respuesta con el menor error posible se realiza atendiendo a unas etapas:

1. *Especificación del modelo teórico:* ¿Qué variables estudiamos? Determinamos las variables que tendrá nuestro modelo, así como sus características y las relaciones entre ellas.
2. *Estimación de parámetros:* ¿Representa el modelo teórico a nuestros datos? Dependerá del valor de los coeficientes del modelo que hemos calculado a partir del conjunto de datos observados.
3. *Selección del modelo:* ¿Aceptamos o rechazamos el modelo? Dependerá del nivel de discrepancia entre los datos observados y ajustados.

4. *Evaluación del modelo:* ¿Qué podemos decir de nuestro modelo? Se examinarán las observaciones individuales y los datos influyentes, los supuestos de normalidad, linealidad, homocedasticidad e independencia.
5. *Interpretación del modelo:* ¿Qué conclusiones obtenemos? Esta fase conlleva una explicación detallada de los parámetros del modelo (con respecto a la variable de respuesta) para comprobar si se cumplen los criterios estadísticos y lógicos y poder sacar conclusiones.

Finalmente, se acepta o no el modelo y, si es preciso, se reinicia el proceso.

Uno de los software estadísticos que reúne las características para trabajar el modelado estadísitco es R, el cual vamos a usar nosotros para elaborar nuestro trabajo.

En resumen, el MLG es una herramienta que nos permite estudiar todas las situaciones de análisis dentro de un mismo esquema general. Obviamente, esto nos facilita el aprendizaje de nuevos modelos de análisis porque se trata simplemente de contemplarlos como particularidades de un modelo más general.

1.2. Componentes

Un MLG posee 3 componentes básicos:

1. **Componente aleatoria:** Identifica la variable respuesta y su distribución de probabilidad.
2. **Componente sistemática:** Especifica las variables explicativas utilizadas en la función predictora lineal.
3. **Función link:** Es una funcion del valor esperado de Y como una combinación lineal de las variables predictoras.

1.2.1. Componente aleatoria

La **componente aleatoria** de un MLG consiste en una variable aleatoria Y.

En muchas aplicaciones, las observaciones de Y son binarias y se identifican como éxito y fracaso. Aunque de modo más general, cada Y_i indica el número de éxitos de entre un número fijo de ensayos, y se modeliza como una distribución binomial.

En otras ocasiones cada observación es un recuento, que este es el caso que vamos a estudiar en nuestro modelo, con lo que se puede asignar a Y una distribución de Poisson o una distribución binomial negativa. Finalmente, si las observaciones son continuas puede darse el caso de que Y siga una distribución normal.

1. **Función**

Todos estos modelos se engloban dentro de la llamada **familia exponencial de distribuciones**, cuya función de probabilidad (si Y es discreta) o función de densidad (si Y es continua) se escribe:

$$f(y;\theta,\phi) = exp\left\{\frac{y\theta - b(\theta)}{a(\phi)} + c(y,\phi)\right\},$$

donde θ es el parámetro natural, ϕ es el parámetro de dispersión, y las funciones $a(\phi)$, $b(\theta)$ y $c(y,\phi)$ son concretas para cada elemento de la familia.

La **función de verosimilitud** $L(y;\theta,\phi)$ es algebráicamente igual a la función de densidad $f(y;\theta,\phi)$:

$$L(y;\theta,\phi) = f(y;\theta,\phi) = \prod_{i=1}^{n} f(y_i;\theta_i,\phi),$$

donde $y = (y_1,...,y_n)$, $\theta = (\theta_1,...,\theta_n)$.

La función **log-verosimilitud** es el logaritmo de la función de verosimilitud $l(\theta,y)$ =log$L(\theta,y)$:

$$l(\theta,\phi,y) = \sum_{i=1}^{n} l_i(\theta_i,\phi,y_i) = \sum_{i=1}^{n} \left\{ \frac{y_i\theta_i - b(\theta_i)}{a(\phi)} + c(y_i,\phi) \right\},$$

Se omite la función $c(y_i,\phi)$ que no depende de θ_i y se inserta la relación $\theta_i = \theta_i(\mu_i)$ entre el parámetro natural y la esperanza de la i-ésima observación, obteniéndose:

$$l(\mu,\phi,y) = \sum_{i=1}^{n} l_i(\mu_i,\phi,y_i) = \sum_{i=1}^{n} \left\{ \frac{y\theta_i - b(\theta_i(\mu_i))}{a(\phi)} \right\}.$$

2. **Propiedades**

Los elementos de esta familia verifican las siguientes propiedades:

$$\begin{aligned} E(Y) &= \mu = b'(\theta) \\ Var(Y) &= \sigma^2 = a(\phi)V(\mu), \end{aligned}$$

donde $V(\mu)$ es la función de varianza.

3. **Momentos de la familia exponencial**

Cálculo del primer y segundo momento a partir del logaritmo de su verosimilitud.

$$l(\theta,y) = \frac{y\theta - b(\theta)}{a(\phi)} + c(y,\phi).$$

Su primera derivada es:

$$l'(\theta,y) = \frac{\partial l(\theta,y)}{\partial\theta} = \frac{y - b'(\theta)}{a(\phi)}.$$

Su segunda derivada es:

$$l''(\theta,y) = \frac{\partial l^2(\theta,y)}{\partial\theta^2} = \frac{-b''(\theta)}{a(\phi)}.$$

Como $E\left(\dfrac{\partial l(\theta,y)}{\partial\theta}\right) = 0$, entonces

$$0 = E(l'(\theta,y)) = E\left(\frac{y - b'(\theta)}{a(\phi)}\right),$$

y por lo tanto

$$\mu = E(Y) = b'(\theta).$$

Además, sabemos que

$$E(l''(\theta, y)) = -E[(l'(\theta, y))^2],$$

entonces

$$Var(l'(\theta, y)) = E[(l'(\theta, y))^2] = -E(l''(\theta, y)) = \frac{b''(\theta)}{a(\phi)}.$$

Por otro lado,

$$Var(l'(\theta, y)) = Var\left(\frac{y - b'(\theta)}{a(\phi)}\right) = \frac{1}{a^2(\phi)} Var(Y),$$

y en consecuencia

$$Var(Y) = a(\phi)b''(\theta).$$

Se comprueba que la varianza de Y se puede escribir como producto de dos funciones, una que depende del parámetro natural y otra que depende del parámetro de dispersión.

4. **Elementos de la familia exponencial**

Cuadro 1.1: Elementos de la familia exponencial

Familia exponencial						
Distribuciones	Expresión	Rango	θ	$a(\phi)$	$b(\theta)$	$V(\mu)$
Bernouilli	$B(p)$	0,1	$\ln(\frac{p}{1-p})$	1	$\ln(1 + e^{\theta})$	$p(1-p)$
Binomial	$Bi(n, p)$	[0,n]	$\ln(\frac{p}{1-p})$	1	$n\ln(1+e^{\theta})$	$np(1-p)$
Normal	$N(\mu, \sigma^2)$	$(-\infty, \infty)$	μ	σ^2	$\frac{\theta^2}{2}$	1
Gamma	$G(\mu, v)$	$(0, \infty)$	$\frac{-1}{\mu}$	$\frac{1}{v}$	-ln(-θ)	μ^2
Poisson	$P(\mu)$	$\mathbb{N} \cup \{0\}$	$\ln(\mu)$	1	e^{θ}	μ
Binomial negativa	BN(p,r)	$\mathbb{N} \cup \{0\}$	ln(1-p)	1	$-r(\ln(1 - e^{\theta}))$	$\frac{r(1-p)}{p^2}$

1.2.2. Componente sistemática

La componente sistemática, también llamada predictor lineal, se escribe como combinación lineal de las variables explicativas x_i:

$$\eta = \beta_0 + \sum_{i=1}^{p} \beta_i x_i.$$

1.2.3. Función link

Sea $\mu = E(Y)$. Se denomina función link a la función g que nos relaciona μ con η como:

$$g(\mu) = \eta = \beta_0 + \sum_{i=1}^{p} \beta_i x_i.$$

Es decir, nos relaciona la componente aleatoria con la componente sistemática, o lo que es lo mismo, el valor esperado con las variables explicativas.

La función g más simple que podemos tomar es la identidad, la cual da lugar al modelo lineal clásico:

$$\mu = \eta = \beta_0 + \sum_{i=1}^{p} \beta_i x_i.$$

Además, cada elemento de la familia exponencial viene determinado por una función link canónica, que consiste en relacionar el parámetro natural directamente con el predictor lineal:

$$\theta_i = \theta(\mu_i) = \eta_i = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} \quad g(\mu_i) = \theta(\mu_i).$$

En el cuadro 1.2 se recogen las funciones link canónicas más importantes que aparecen en el apartado 4.4 de Hilbe (2012).

Cuadro 1.2: Funciones link canónicas

Funciones Link	
Distribuciones	Expresión
Bernouilli	$g(\mu_i) = \theta(\mu_i) = \ln(\frac{\mu_i}{1-\mu_i}) = \eta_i$
Binomial	$g(\mu_i) = \theta(\mu_i) = \ln(\frac{\mu_i}{1-\mu_i}) = \eta_i$
Gamma	$g(\mu_i) = \theta(\mu_i) = \frac{-1}{\mu_i} = \eta_i$
Binomial negativa	$g(\mu_i) = \theta(\mu_i) = \ln(\frac{\alpha\mu_i}{1+\alpha\mu_i}) = \eta_i$
Normal	$g(\mu_i) = \theta(\mu_i) = \mu_i = \eta_i$
Poisson	$g(\mu_i) = \theta(\mu_i) = \ln(\mu_i) = \eta_i$

1.3. Hipótesis

En este apartado se estudian las principales hipótesis que se hacen sobre el MLG.

1. **Linealidad de los parámetros.**

Se supone que se establece una linealidad entre las variables de nuestro modelo.

2. **Grados de libertad positivos.**

Esto quiere decir que, como mínimo, necesitamos que haya mayor número de datos que de variables. En caso contrario no existe la inversa de X^TX ya que no tendría rango máximo y por tanto nuestro sistema

$$X^TX\beta = X^TY$$

sería indeterminado.

3. **Parámetros constantes.**

Suponemos que los parámetros $\beta_0, .., \beta_p \in \mathbb{R}$.

4. **Regresores no estocásticos.**

Suponemos que nuestra matriz numérica de datos X es fija.

5. **Independecia lineal entre variables explicativas.**

Esta hipótesis nos dice que cada variable explicativa contiene información adicional de la variable respuesta que no está contenida en otra.

1.4. Estimación en MLG

Hay dos formas más comunes de realizar una estimación de los parámetros, mediante **mínimos cuadrados** o mediante el **método de maxima verosimilitud**. Utilizamos el segundo método ya que nos aporta mejores propiedades.

Consideremos la función de densidad de la familia exponencial como

$$f(y;\theta) = exp\{a(y)b(\theta) + c(\theta) + d(y)\}.$$

1.4.1. Método de máxima verosimilitud

Sea $Y_1, ..., Y_n$ variables aleatorias cuya función de densidad es:

$$f(y_1, ..., y_n; \theta_1, ..., \theta_p).$$

Consideremos $y = (y_1, ..., y_n)$ y $\theta = (\theta_1, ..., \theta_p)$. Sea Ω el espacio paramétrico (conjunto de valores que puede tomar θ), según el apartado 4.2 de Dobson (2008), se define el **estimador de máxima verosimilitud** de θ como el valor $\hat{\theta}$ que maximiza la función de verosimilitud:

$$L(\hat{\theta}; y) \geq L(\theta; y) \quad \forall \theta \in \Omega.$$

Equivalentemente, también maximiza la función log-verosimilitud, la cual es el logaritmo de la función de verosimilitud ($l(\theta; y)$ =log$L(\theta; y)$), esto es:

$$l(\hat{\theta}; y) \geq l(\theta; y) \quad \forall \theta \in \Omega.$$

Este estimador se obtiene haciendo :

$$\frac{\partial l(\theta; y)}{\partial \theta_j} = 0, \quad \forall j \in \{1, ..., p\},$$

y luego viendo que la matriz de las segundas derivadas evaluadas en ese valor obtenido de $\hat{\theta}$ es definida negativa:

$$\frac{\partial^2 l(\theta; y)}{\partial \theta_j \partial \theta_k}.$$

Una propiedad a destacar que pose la estimación por máxima verosimilitud es la **propiedad de la invarianza**, esto es que si nosotros tenemos una función $g(\theta)$, entonces su estimador por máxima verosimilitud es $g(\hat{\theta})$. Otras propiedades a destacar es la consistencia, suficiencia y eficiencia asintótoca.

1.4.2. Aplicación al MLG

Se quiere obtener un estimador de β del MLG. La estimación se va a obtener numéricamente mediante un método iterativo. Para ello hemos aplicado el método visto anteriormente y basado en el apartado 3.2 y 4.4 de Dobson (2008).

Sea $Y_1, ..., Y_n$ variables aleatorias independientes, podemos expresar su función log-logaritmo como:

$$l(\theta; y) = \sum a(y_i)b(\theta_i) + \sum c(\theta_i) + \sum d(y_i),$$

donde $E(Y_i) = \mu_i = -c'(\theta_i)/b'(\theta_i)$ y $g(\mu_i) = x_i^T\beta = \eta_i$, donde g es una función monótona y diferenciable.

Una propiedad de la familia exponencial es que tienen la suficiente regularidad como para asegurar que el máximo global de la función log-verosimilitud se alcanza en la solución única de las ecuaciones $\partial l/\partial\beta = 0$.

Primero calculemos la $Var(Y)$. Para ello,

$$U = \frac{\partial l}{\partial \theta} = a(y)b'(\theta) + c'(\theta),$$

y

$$U' = \frac{\partial^2 l}{\partial \theta^2} = a(y)b''(\theta) + c''(\theta).$$

Así que

$$E(U) = b'(\theta)E(a(Y)) + c'(\theta),$$

pero como $E(U) = 0$, se tiene que

$$\begin{array}{rcl} E(a(Y)) & = & -c'(\theta)/b'(\theta), \\ Var(U) & = & (b'(\theta))^2 Var(a(Y)), \\ E(-U') & = & -b''(\theta)E(a(Y)) - c''(\theta). \end{array}$$

Luego con todo esto y usando que $Var(U) = E(U^2) = E(-U')$, tenemos que

$$Var(Y) = [b''(\theta)c'(\theta) - c''(\theta)b'(\theta)]/[b'(\theta)]^3.$$

Equivalentemente tenemos que

$$Var(Y_i) = [b''(\theta_i)c'(\theta_i) - c''(\theta_i)b'(\theta_i)]/[b'(\theta_i)]^3.$$

La **función score** respecto al parámetro β_j se define como

$$U_j = \frac{\partial l(\theta; y)}{\partial \beta_j} = \sum_{i=1}^{n} \frac{\partial l_i}{\partial \beta_j},$$

donde

$$l_i = y_i b(\theta_i) + c(\theta_i) + d(y_i).$$

Para obtener U_j hacemos

$$\frac{\partial l_i}{\partial \beta_j} = \frac{\partial l_i}{\partial \theta_i} \frac{\partial \theta_i}{\partial \mu_i} \frac{\partial \mu_i}{\partial \beta_j}.$$

Derivando l_i y usando $E(Y_i)$ obtenemos

$$\frac{\partial l_i}{\partial \theta_i} = y_i b'(\theta_i) + c'(\theta_i) = b'(\theta_i)(y_i - \mu_i).$$

Derivando μ_i y usando $Var(Y_i)$

$$\frac{\partial \mu_i}{\partial \theta_i} = -\frac{c''(\theta_i)}{b'(\theta_i)} + \frac{c'(\theta_i) b''(\theta_i)}{(b'(\theta_i))^2} = b'(\theta_i) Var(Y_i).$$

Derivando $g(\mu_i)$

$$\frac{\partial \mu_i}{\partial \beta_j} = \frac{\partial \mu_i}{\partial \eta_i} \frac{\partial \eta_i}{\partial \beta_j} = x_{ij} \frac{\partial \mu_i}{\eta_i},$$

donde x_{ij} es el elemento j-ésimo de x_i^T.

Por lo que

$$\frac{\partial l_i}{\partial \beta_j} = \frac{\partial l_i}{\partial \theta_i} \frac{\partial \mu_i}{\partial \beta_j} \Big/ \frac{\partial \mu_i}{\partial \theta_i} = \frac{(y_i - \mu_i) x_{ij}}{Var(Y_i)} \left(\frac{\partial \mu_i}{\partial \eta_i} \right).$$

Y así obtenemos

$$U_j = \sum_{i=1}^{n} \frac{(y_i - \mu_i) x_{ij}}{Var(Y_i)} \left(\frac{\partial \mu_i}{\partial \eta_i} \right).$$

En general, $U_j = 0$ es no lineal, por lo que tenemos que resolverlo mediante iteración numérica. Para ello se usa el método de **Newton-Raphson**, para así escribir la m-ésima aproximación como

$$b^{(m)} = b^{(m-1)} - \left(\frac{\partial^2 l}{\partial \beta_j \partial \beta_k} \right)^{-1}_{\beta = b^{(m-1)}} U^{(m-1)},$$

donde $U^{(m-1)}$ es el vector U_j evaluado en $\beta = b^{(m-1)}$.

Un procedimiento alternativo a este es el llamado **método de score**, el cual consiste en sustituir la matriz de las segundas derivadas por

$$E \left(\frac{\partial^2 l}{\partial \beta_j \partial \beta_k} \right).$$

Se define la **matriz de información de fisher** como

$$F(\beta) = E[j(\beta; Y)],$$

donde $j(\beta; Y)$ es la **matriz de información observada** correspondiente al parámetro β, y viene determinada por

$$j(\beta; y) = -\frac{\partial^2}{\partial\beta\partial\beta^T} l(\beta; y).$$

Ahora, aplicando esto, obtenemos la ecuación

$$b^{(m)} = b^{(m-1)} + (F^{(m-1)})^{-1} U^{(m-1)},$$

donde $F^{(m-1)}$ es F evaluada en $b^{(m-1)}$. Multiplicando por $F^{(m-1)}$ en ambos lados de la ecuación obtenemos

$$F^{(m-1)} b^{(m)} = F^{(m-1)} b^{(m-1)} + U^{(m-1)}. \quad *$$

Usando que el elemento (j,k)-ésimo de la matriz F es

$$F_{jk} = E\left[\frac{\partial l_i}{\partial\beta_j}\frac{\partial l_i}{\partial\beta_k}\right] = E\left[\frac{(y_i-\mu_i)^2 x_{ij}x_{ik}}{Var(Y_i)^2}\left(\frac{\partial\mu_i}{\partial\eta_i}\right)^2\right] = \sum_{i=1}^{n}\frac{x_{ij}x_{ik}}{Var(Y_i)}\left(\frac{\partial\mu_i}{\partial\eta_i}\right)^2,$$

escribimos la matriz F como

$$F = X^T W X,$$

donde W es una matriz nxn diagonal cuyos elementos son

$$w_{ii} = \frac{1}{Var(Y_i)}\left(\frac{\partial\mu_i}{\partial\eta_i}\right)^2.$$

Usando esto en *, y evaluando en $b^{(m-1)}$, obtenemos

$$F^{(m-1)} b^{(m-1)} + U^{(m-1)} = X^T W z,$$

donde los elementos de z son

$$z_i = \sum_k x_{ik} b_k^{(m-1)} + (y_i - \mu_i)\left(\frac{\partial\eta_i}{\partial\mu_i}\right),$$

con μ_i y $\partial\eta_i/\partial\mu_i$ evaluados en $b^{(m-1)}$.

Concluimos entonces que, usando el método de score, podemos aproximar los β mediante la siguiente expresión

$$X^T W X b^{(m-1)} = X^T W z.$$

1.5. Inferencia: Ajuste del modelo

Después de haber estudiado el modelo, nos plateamos ahora ver si se ajusta o no a la realidad. Es decir, ver la relación entre los datos observados y los datos esperados. Aquí vamos a tratar el tema de selección de variables, bondad de ajuste, coste e interpretabilidad.

Tenemos que tener en cuenta que el tema de selección de variables es muy importante puesto que tener más variables de la cuenta nos provoca un mayor coste, y si tenemos menos nuestro modelo queda pobre. También afecta a la interpretabilidad del modelo el tener muchas variables ya que no sabremos si el papel de algunas de ellas está enmascarado por otras. Pero si tenemos pocas, quizás su interpretabilidad es fácil, pero puede que nuestro modelo entonces no se ajuste a la realidad.

Ahí debemos de saber seleccionar cuales son las variables que mejor ajustan nuestro modelo, mediante procesos de p-valor o de contraste de hipótesis.

Tenemos que tener cuidado a la hora de quitar variables puesto que podemos perder informacion sobre nuestro objetivo. Este proceso queda un poco a manos del investigador, que deberá establecer un criterio para quitar o no más variables sabiendo cuáles son las más influyentes en nuestro modelo.

En la búsquedad de la estabilidad, siguiendo el apartado 5.7 de Dobson (2008), nos plateamos buscar un modelo de entre los dos siguientes:

1. **Modelo saturado:** El número de parámetros es igual al número de observaciones, es decir, $\hat{\mu}_i = y_i$. Este modelo tiene la misma función link que el modelo de interés.
2. **Modelo nulo:** En este modelo no se representa lo que estamos estudiando, es solamente un modelo de referencia donde consideramos un único parámetro, el valor esperado μ, para todas las observaciones.

Lo que buscamos con este proceso es poder expresar nuestro modelo con el menor número de variables posibles (porque así tendremos un menor coste) pero manteniendo la validez del mismo.

1.5.1. Bondad de ajuste en MLG

En este apartado se han estudiado diferentes procesos, según el apartado 5.5 de Dobson, que sirven para comparar nuestro modelo con el modelo de interés, para así obtener una visión de la exactitud y validez de nuestro modelo. Es decir, estudiar lo bien que se ajusta nuestro modelo a un conjunto de observaciones. Basándonos en el apartado 1.4 de Atoche (2017), se tiene lo siguiente.

-**Estadístico de máxima verosimilitud.** Denotado por λ, este estadístico se basa en comparar la función de verosimilitud del modelo saturado con la del modelo de interés, obteniendo $L(\beta_{max}; y)$ y $L(\beta; y)$ respectivamente. Si el modelo de interés describe bien los datos, entonces $L(\beta; y)$ debe ser aproximado a $L(\beta_{max}; y)$. Si el modelo es pobre, entonces $L(\beta; y)$ será mucho menor que $L(\beta_{max}; y)$. Así, obtenemos:

$$\lambda = \frac{L(\beta_{max}; y)}{L(\beta; y)}.$$

Equivalentemente, podemos fijarnos en la diferencia entre las funciones log-logartimos de los modelos:

$$log\lambda = l(\beta_{max}; y) - l(\beta; y).$$

Grandes valores de $log\lambda$ nos indica que el modelo no representa bien los datos.

-**Estadístico de desviación.** Denotada por $D(y, \mu)$, es una medida de bondad de ajuste que compara el modelo saturado con el modelo de interés. Ésta se define como:

$$D(y, \hat{\mu}) = 2\phi(l(y, y) - l(\hat{\mu}, y)),$$

donde $l(y; y)$ es el logaritmo de verosimilitud del modelo de interés y $l(\hat{\mu}; y)$ es el logaritmo de verosimilitud del modelo saturado. Bajo ciertas condiciones en las cuales n es fijo, el parámetro de dispersión es pequeño y conocido y las observaciones individuales convergen a una distribución normal y la desviación satisface:

$$D(y, \mu) \approx \chi^2_{n-p}.$$

Este resultado provee un valor de referencia para determinar la bondad de ajuste del modelo cuando el parámetro de dispersión es conocido. $D(y, \hat{\mu})$ es no negativa y un valor grande indica un ajuste pobre del modelo. Cuando el parámetro de dispersión ϕ es desconocido, entonces el resultado de la desviación no aplica.

-**Estadístico Chi-cuadrado de Pearson.** Denotador por χ^2, este estadístico se define como:

$$\chi^2 = \sum_i \frac{(y_i - \hat{\mu}_i)^2}{Var(Y_i)}.$$

Cuando el número de observaciones es suficiente grande, ambos estadísticos, D y χ^2, son equivalentes.

-**Coeficiente de determinación.** Denotado por R^2, se define como:

$$R^2 = 1 - \frac{D(y; \mu)}{D(y; \mu_0)},$$

verifica que $0 \leqslant R^2 \leqslant 1$. La interpretación de este coeficiente es la siguiente: cuanto mayor sea el número de variables que incorporemos en nuestro modelo, mayor sera este coeficiente y por tanto, se supone que mejor será el ajuste con el modelo. Hecho que no siempre ocurre y por el cual se define el siguiente coeficiente.

-**Coeficiente de determinación ajustado.** Denotado por R^2_{adj}, se introduce con el fin de corregir el coeficiente de determinación y se define como:

$$R^2_{adj} = 1 - \frac{D(y; \mu)/(n-k)}{D(y, \mu_0)/(n-1)}.$$

Este coeficiente incorpora una penalización sobre el número de variables en el modelo. Cuantas más variables introduzcamos en el modelo, $D(y; \mu)$ decrece, pero también lo hace su correspondiente grados de libertad. Por lo tanto, el numerador puede incrementarse si la disminución de la desviación a causa del aumento del número de variables no es compensado por la pérdida de grados de libertad.

La relación entre R^2 y R^2_{adj} es la siguiente:

$$R^2_{adj} = 1 - \left[\frac{n-1}{n-k}\right] \frac{D(y;\mu)}{D(y;\mu_0)} = 1 - \left[\frac{n-1}{n-k}\right](1-R^2).$$

Intervalos de confianza y Test de Hipótesis

Dos de las herramientas más usadas de la inferencia estadística son los intervalos de confianza y los tests de hipótesis. Por ejemplo, los tests de hipótesis son necesarios para comparar el ajuste de dos modelos ajustados a los datos. Tanto para realizar tests como intervalos de confianza necesitamos las distribuciones muestrales de los estadísticos involucrados.

Vamos a realizar un contraste de hipótesis y un intervalo de confianza para nuestros coeficientes β_i.

1. **Contraste de hipótesis.**

Una hipótesis se contrasta comparando nuestras predicciones con la realidad. Si coinciden, dentro de un margen de error admisible, mantenemos la hipótesis. En caso contrario, la rechazamos y buscamos otras que sean capaces de explicar los datos. Planteamos una hipótesis nula a priori creíble y sólo la rechazamos cuando existe suficiente evidencia en los datos en contra de la misma. Si rechazamos la nula, implícitamente no rechazamos otra hipótesis llamada alternativa.

Vamos a plantearnos el siguiente contraste:

$$H_1 : \eta \in L \subset \mathbb{R}^q,$$

donde L está parametrizado como $\eta = X_1\beta$, y vamos a considerar la hipótesis

$$H_0 : \eta \in L_0 \subset \mathbb{R}^s,$$

donde $\eta = X_0\alpha$ y $s < q$, con la hipótesis alternativa $H_1 : \eta \in L \backslash L_0$.

Para este contraste, usamos el **estadístico de razón de verosimilitud**, que tiene la forma:

$$-2logl = D(\mu(\hat{\beta}); \mu(\beta(\hat{\alpha}))),$$

donde D es la desviación.

Cuando H_0 es cierto, la desviación es asintóticamente igual a la distribución $\chi^2(k-m)$.

Planteamos el siguiente contraste:

$$\begin{cases} H_0 : A\beta = S \\ H_1 : A\beta \neq S, \end{cases}$$

donde A es una matriz de rango $s \leqslant p+1$ y S un vector de dimensión s.

Para este contraste, se usa el **estadístico de Wald**:

$$S_W = (A\hat{\beta} - S)^T (AF^{-1}(\hat{\beta})A^T)(A\hat{\beta} - S).$$

2. **Intervalo de confianza.**

Para elaborar un IC (intervalo de confianza), necesitamos usar un estadístico.

- *Estadístico razón de verosimilitud* para así obtener:

$$IC(\beta_j, 1-\alpha) = \{\beta_j \in \mathbb{R} / l_1(\beta_j) \geqslant l_0\},$$

donde

$$l_0 = l(\hat{\beta}) - 0.5\chi^2_{1-\alpha}(1),$$

y $l_1(\beta_j)$ denota la función log-verosimilitud para β_j.

- *Estadístico de Wald* para así obtener:

$$IC(\beta, 1-\alpha) = \{\beta \in \mathbb{R}^{p+1} / (\hat{\beta} - \beta)(Var(\hat{\beta}))^{-1}(\hat{\beta} - \beta) < \chi^2_{p+1,n-1}\}.$$

1.5.2. Residuos

Los residuos aparecen cuando el modelo que planteamos no coincide exactamente con el problema que queremos modelar. En la realidad esto ocurre siempre, por ello los estudiamos y nos dan una indicación de las diferencias entre los valores ajustados y observados. Estos errores pueden venir a raiz de los datos o por uso de un mal modelo para su aproximación.

Un aspecto importante en los MLG es que la ortogonalidad de los residuales y los valores ajustados no se cumple. Es decir, la descomposición $datos = ajustados + residuos$ no se cumple. En general, según el apartado 4.4 de Hilbe (2007), podemos definir tres tipos de residuales:

1. **Residual básico.**

$$r_i^b = y_i - \hat{y}_i, \quad \forall i = 1, ..., n.$$

2. **Residual de Pearson.**

$$r_i^p = \frac{y_i - \hat{\mu}_i}{\sqrt{\hat{Var(\mu_i)}}}, \quad \forall i = 1, ..., n,$$

3. **Residual de Score.**

$$r_i^s = \frac{y_i - \hat{\mu}_i}{Var(\hat{\mu}_i)} \left(\frac{\partial \eta}{\partial \mu}\right)_i^{-1}$$

4. **Residual de desviación.**

$$e_i^d = \sqrt{d_i} * signo(y_i - \hat{\mu}_i), \quad \forall i = 1, ..., n,$$

donde d_i se define como la componente de desviación: $d_i = 2(l(\hat{\mu}_i, y_i) - l(\hat{\mu}_i, \mu_i))$.

Los residuales sirven para detectar datos atípicos y desvíos del modelo con relación a los datos. En MLG se deben usar gráficas similares a las discutidas en modelos lineales.

1.5.3. Interpretación

Una vez realizados los estudios anteriores, llega la hora de traducir lo realizado y poder sacar conclusiones sobre los datos. Esto hace que si relamente hemos modelado bien nuetro problema, podamos usarlo para situaciones similares y para poder hacer predicciones en el futuro si nuestro modelo esta bien planteado. Todo ello dependerá de los ajustes, estimaciones y estudio de los errores realizados durante los procesos de estimación y elección del modelo.

Capítulo 2

Modelos de conteo

2.1. Variable de conteo

Un **dato de conteo** se define como una observación de un fenómeno el cual podemos contar.

Una **variable de conteo** o variable de recuento se define como aquella variable que toma valores enteros no negativos. Estas variables representan el número de sucesos ocurridos en un determinado periodo de tiempo, cosa que ocurre en nuestro estudio y por ello usamos dichas variables.

2.2. Modelos para variables de conteo

Teniendo en cuenta lo definido anteriormente, no todos los modelos nos van a servir para nuestro estudio, debido a que puede ocurrir que:

- La variable respuesta puede salirse del rango de valores que puede tomar.
- Si acumulamos los datos en una variable binaria vamos a perder información, ya que al agrupar todos los números naturales mayores que uno en el uno estamos dando igual valor a números que son totalmente distintos y distantes.
- No tienen porqué cumplirse las hipótesis de normalidad u homocedasticidad.

Para datos de conteo se suele utilizar la distribución Poisson como componente aleatorio en el proceso de ajuste de un modelo lineal generalizado. Esta distribución se caracteriza por la igualdad entre su media y su varianza, propiedad que se conoce como **equidispersión**, aunque en la práctica la varianza suele ser mayor que la esperanza para una variable del tipo Poisson. Se denomina **sobredispersión** a este hecho. En estos casos se suele usar la distribución binomial negativa. Cuando la varianza es menor que la esperanza se denomina **infradispersión**.

2.2.1. Sobredispersión

Tratamos el tema de la sobredispersión, qué es, cómo se detecta y cómo solucionarla.

En primer lugar, la sobredispersión aparece en los modelos de conteo y no en los modelos continuos estudiados anteriormente, ya que en ellos aparece el parámetro de dispersión ϕ, y requieren de una función que relacione la varianza y la media.

El problema de la sobredispersión es que puede causar una subestimación del error de β. Por lo que una variable puede aparecer como significativa cuando realmente ésta no lo es.

Nosotros podemos **detectar** la posible sobredispersión viendo si al dividir el estadístico de Pearson (χ^2) o la desviación por los grados de libertad ($n - p$) es mayor que 1. Para valores pequeños, no suelen ser de preocupación.

La sobredispersión puede corregirse por alguno de estos métodos:

- Incorporación de un predictor apropiado.
- Inclusión de interacciones significativas.
- Utilizando otra función link.

Como consecuencia, la incorporación de estos remedios podría mejorar el ajuste de nuestro modelo. Sin embargo, hay que tener en cuenta que al realizar un análisis de regresión con estos datos, si la varianza muestral es más del doble de la media, probablemente los datos permanezcan sobredispersos aún después de la inclusión de dichos remedios.

Una de las causas de la sobredispersión es la presencia excesiva de ceros en las observaciones y en tal caso hay una variedad de métodos utilizados para manejarla, entre los que destacamos:

1. *Modelo Binomial Negativa*: abreviado por BN2, es el modelo que se usa cuando hay sobredispersión en los datos. También su variante puede ser usada, el abreviado como BN-P.
2. *Modelo Poisson Inversa Gaussiana*: abreviado por PIG, es el modelo que se usa cuando hay infradispersión en los datos.
3. *Modelo de Poisson cero truncado*: para el caso en que los datos no admiten el conteo cero.
4. *Modelo Poisson cero inflado*: para el caso en el que hay más valores ceros de los esperados para una distribución de Poisson para una media dada o los conteos cero provienen de una fuente diferente que los conteos mayores que cero. Los conteos cero se admiten en ambas componentes del modelo.
5. Versiones con la Binomial Negativa de los puntos anteriores.
6. *Modelo Hurdle*: para el caso en el que hay más valores cero o menos valores cero basados en la distribución Poisson para una media dada o los conteos cero provienen de una fuente diferente que los conteos mayores que cero. En este modelo, los conteos cero se admiten solo en el componente binario, mientras que el componente truncado en cero no presenta ese valor en el recorrido.

Todos estos modelos los estudiaremos particularmente más adelante.

Se recogen los modelos de conteo en el cuadro (2.1).

Cuadro 2.1: Modelos de conteo con sus medias y varianzas

Modelo	Abreviatura	Media	Varianza
Poisson	P	μ	μ
Binomial Negativa	BN2	μ	$\mu(1+\alpha\mu)$
Binomial Negativa	BN-P	μ	$\mu(1+\alpha\mu^{p-1})$
Poisson Inversa Gaussiana	PIG	μ	$\mu(1+\alpha\mu^2)$

Capítulo 3

Modelo de Regresión de Poisson

Se estudia el modelo de Poisson de un parámetro. Dicho modelo es simple debido sobre todo a la equidispersión.

Según el apartado 13.2 de Hilbe (2012), se dirá que $Y \approx P(\mu)$ si su función de probabilidad es de la forma

$$P(Y = y) = e^{-\mu}\mu^y/y!,$$

o en forma exponencial

$$f(y;\mu) = exp\{yln(\mu) - \mu - ln(y!)\}.$$

Luego de aquí se deduce que:

$$\begin{array}{rcl} \theta & = & ln(\mu) \\ b(\theta) & = & \mu \end{array},$$

La función link canónica es la función logaritmo. Su función inversa es $\exp(\eta)$, con η el predictor lineal. A partir de aquí, escribimos el modelo exponencial como

$$E(Y_i|x_i) = exp(\beta_0 + \beta_1 x_{i1} + ... + \beta_n x_{in}).$$

La media y varianza se calculan mediante la primera y segunda derivada respecto a θ

$$\begin{array}{rcl} b'(\theta) & = & \underbrace{\dfrac{\partial b}{\partial \mu}}_{1}\dfrac{\partial \mu}{\partial \theta} \\ & = & \mu \\ b''(\theta) & = & \underbrace{\dfrac{\partial^2 b}{\partial \mu^2}}_{0}\left(\dfrac{\partial \mu}{\partial \theta}\right)^2 + \dfrac{\partial b}{\partial \mu}\dfrac{\partial^2 \mu}{\partial \theta^2} \\ & = & \mu \end{array},$$

luego la media y la varianza son idénticas a μ, hecho que se conoce como **equidispersión**.

La distribución de Poisson condicionada a x_i es

$$P(Y_i = y_i|x_i) = \frac{e^{-\mu_i(x_i)}\mu_i(x_i)^{y_i}}{y_i!}.$$

A partir de esto, obtenemos lo que se conoce como **Modelo de Regresión de Poisson**, que se escribe

$$E(Y_i|x_i) = e^{\beta_0+\beta_1 x_{i1}+...+\beta_n x_{in}}.$$

La función log-verosimilitud es

$$l(\mu; y) = \sum_{i=1}^{n}\{y_i ln(y_i) - \mu_i - ln(y_i!)\}.$$

En el caso particular en el que $Y = 0$, esta expresión se reduce a

$$l_i(\mu; 0) = -\mu_i.$$

Además, se puede escribir la función log-verosimilitud en términos de $X\beta$

$$\begin{array}{rcl} l(X\beta; y) & = & \sum_{i=1}^{n}\{y_i ln(exp(X_i\beta)) - exp(X_i\beta) - ln(y_i!)\} \\ & = & \sum_{i=1}^{n}\{y_i X_i\beta - exp(X_i\beta) - ln(y_i!)\} \\ l_i(X\beta; y_i = 0) & = & -exp(X_i\beta) \end{array} .$$

3.1. Características

La distribución de Poisson Y posee diversas características:

1. Sean $Y_1, ..., Y_n \in P(\mu)$ independientes, entonces

$$Y = \sum_{i=1}^{n} Y_i \in P(n\mu).$$

2. Equidispersión.
3. Como consecuencia del **teorema central del límite**, para valores grandes de μ, una variable Poisson puede aproximarse por una normal de la siguiente forma

$$\frac{Y - \mu}{\sqrt{\mu}} \approx N(0, 1).$$

4. La distribución de Poisson es el caso límite de una distribución binomial en el caso en el que $n \longrightarrow \infty$ y $p \longrightarrow 0$ de tal forma que $\mu = pn = cte$.

3.2. Propiedades

Estudiamos qué propiedades deben cumplir los datos para el modelo de Poisson:

1. La probabilidad de que ocurra un solo suceso sobre un intervalo pequeño es aproximadamente proporcional al tamaño del intervalo.
2. La probabilidad de que ocurran dos sucesos en un mismo intervalo cuya longitud tiende a cero es despreciable.

3. La probabilidad de que ocurra un suceso en un intervalo de una cierta longitud no varía en cualquier otro intervalo con esa misma longitud.
4. La probabilidad de que ocurra un suceso en un intervalo es independiente de la probabilidad de que ocurra un suceso en otro intervalo que no se superpone al anterior.

Cuando las propiedades 3 y 4 no se cumplen, los datos pueden tener mayor varianza que media, que es lo que se define como sobredispersión.

3.3. Bondad de ajuste

Una medida de discrepancia entre los valores observados y los predichos es la llamada **desviación**, la cual se escribe como

$$\begin{aligned} D &= 2\sum_{i=1}^{n}\{y_i ln(y_i) - y_i - y_i ln(\hat{\mu}_i) + \hat{\mu}_i\} \\ &= 2\sum_{i=1}^{n}\{y_i ln\left(\frac{y_i}{\hat{\mu}_i}\right) - (y_i - \hat{\mu}_i)\} \qquad . \\ D_i(y_i = 0) &= 2\hat{\mu}_i \end{aligned}$$

El estadístico χ^2 de Pearson también nos sirve para estudiar esta bondad, el cual se escribe como

$$\chi^2 = \sum_{i=1}^{n} \frac{(y_i - \hat{\mu}_i)^2}{\hat{\mu}_i}.$$

Es de destacar el estadístico

$$G^2 = 2\sum_{i=1}^{n} y_i ln(y_i/\hat{\mu}_i),$$

el cual nos dice que para $G^2 = 0$, el ajuste de nuestro modelo es perfecto.

3.4. Dispersión

Cuando se trabaja con una base de datos de conteo puede ocurrir que el modelo Poisson puede parecer sobredisperso y en realidad no lo es, o puede efectivamente presentar sobredispersión. En el primer caso, una simple corrección al modelo puede hacer desaparecer la variabilidad no deseada. Si después de los ajustes realizados sobre el modelo el problema de sobredispersión no desaparece, se presenta el segundo caso, donde se deben buscar modelos alternativos, que puede ser el modelado a partir de la distribución Binomial Negativa (no tiene restricción de igualdad en media y varianza), o el uso de modelos más complejos como son los modelos compuestos y modelos en 2 partes.

Podemos usar, según el apartado 13.4 de Hilbe (2012), un test basado en este supuesto siguiendo el siguiente procedimiento:

1. Obtener $\hat{\mu}_i$.
2. Calcular $z = \dfrac{(y_i - \hat{\mu}_i)^2 - y_i}{\hat{\mu}_i\sqrt{2}}$.

3. Escribir z como un modelo de constante única.

El test de hipótesis que vamos a usar es el siguiente

$$\begin{cases} H_0 : V(y) &= E(y) \\ H_1 : V(y) &= E(y) + \alpha g\{E(y)\} \end{cases},$$

el cual se lleva a cabo a través del t test y α es el nivel de significación. El estadístico que se suele usar para este contraste de hipótesis es $CT = -2(l_{Poisson} - l_{BN})$. Se tiene que la distribución asintótica del CT es χ^2_1.

Si existe evidencia suficiente para probar que los datos no siguen una distribución Poisson, entonces será necesario emplear un modelo de conteo alternativo que se ajuste al tipo de supuesto violado en la distribución de los datos, los cuales aparecen nombrados en el apartado de dispersión del capítulo 2.

3.5. Modelo de Regresión Bivariante de Poisson

El anterior modelo estudiado hacía referencia al caso en el que solamente teníamos una variable respuesta con una distribución de Poisson. Ahora vamos a tratar el caso en el que tenemos un vector respuesta de dos dimensiones que sigue una distribucion bivariante Poisson.

3.5.1. Distribución de Poisson Bivariada

Sean X_1, X_2, X_3 variables aleatorias que siguen una distribución de Poisson cuyos respectivos parámetros son $\lambda_1, \lambda_2, \lambda_3 > 0$. Según el apartado 3.2 de Quinteiro (2014), se dice que $X = X_1 + X_3$ e $Y = X_2 + X_3$ siguen cojuntamente una distribución Bivariada Poisson ($BP(\lambda_1, \lambda_2, \lambda_3)$).

La función de probabilidad es

$$f_{BP}(x, y) = exp\{-(\lambda_1 + \lambda_2 + \lambda_3)\}\frac{\lambda_1^x}{x!}\frac{\lambda_2^y}{y!}\sum_{k=0}^{min(x,y)}\binom{x}{k}\binom{y}{k}k!\left(\frac{\lambda_3}{\lambda_1\lambda_2}\right)^k, \qquad x, y \in \mathbb{N}\cup\{0\}.$$

Marginalmente cada variable aleatoria sigue una distribución de Poisson con $E(X) = \lambda_1 + \lambda_3$ y $E(Y) = \lambda_2 + \lambda_3$. Además, $cov(X, Y) = \lambda_3$. En el caso en el que $\lambda_3 = 0$ ambas distribuciones son independientes y la distribución Bivariada de Poisson se reduce al producto de dos distribuciones Poisson independientes.

3.5.2. Modelo de Regresión Bivariada de Poisson

Sean $i = 1, ..., n$ el número de observaciones. Consideramos el modelo para la i-ésima observación

$$(X_i, Y_i) \sim BP(\lambda_{1i}, \lambda_{2i}, \lambda_{3i}),$$
$$log(\lambda_{ki}) = w'_{ki}\beta_k \qquad k = 1, 2, 3,$$

donde $w_{ki}^{'}$ denota el vector de las variables explicativas para la observación i-ésima usada para el modelo λ_{ki} (donde para $k = 1, 2, 3$ nos referimos al modelo para X, Y y el modelo que nos relaciona ambas variables ,respectivamente) y β_k vector de coeficientes. Las variables explicativas usadas para modelizar cada parámetro λ_{ki} no tienen por qué ser las mismas. Se utiliza el algoritmo EM para obtener la estimación por máxima verosimilitud de los coeficientes beta.

Capítulo 4

Modelo de Regresión Binomial Negativa

La **distribución binomial negativa** se utiliza para modelos con datos de conteo. Principalmente para aquellos casos en los que aparece sobredispersión y no podemos abarcarlo con el modelo de Poisson. La distribución binomial negativa es una distribución discreta que toma valores no negativos. Esta distribución modela la probabilidad de observar y fallos antes del r-ésimo éxito en un número finito de sucesos independientes e idénticamente distribuidas (i.i.d.) Bernoulli. Sea $Y \sim BN(r, p)$, con $0 < p < 1$ y $0 < r$. La función de probabilidad del modelo binomial negativo se escribe como

$$P(Y = y) = \binom{y + r - 1}{r - 1} p^r (1 - p)^y.$$

El modelo de regresión binomial negativo comenzó a estudiarse en 1949 por Anscombe. En 1987, Lawless caracterizó la binomial negativa como un modelo mixto, dando fórmulas para su media, varianza, log-verosimilitud y momentos. Más tarde, en 1990, Breslow citó el trabajo de Lawless mientras manipulaba el modelo de Poisson para ajustar los parámetros de la binomial negativa. Desde principios hasta finales de los 80, se contruyó el modelo binomial negativo como un modelo mixto para estudiar aquellos datos con sobredispersión.

McCullagh y Nelder (1989) ajustando el valor de τ, podría ser considerado como un MLG. Los autores mencionaron de la existencia de una función canónica link, rompiendo con el concepto de modelo mixto, pero no consiguieron explicarlo.

No fue hasta la mitad de los años 90 cuando la binomial negativa fue construida como un elemento de la familia MLG. Hilbe (2011) detalla como los modelos de regresión binomial negativo y geométrico pueden ser derivados de sus respectivas distribuciones de probabilidad. La idea que hay detrás de todo es que el modelo tradicionalmente llamado como binomial negativo no es mas que una log-link binomial negativa. La función canónica link puede tener propiedades de no convergencia y esto ocurre cuando τ aparece en la función link y su inversa.

La binomial negativa está basada en el modelo mixto Poisson-Gamma. Hay dos métodos para motivar el modelo de regresión binomial negativo, ambos se estudian a continuación.

4.1. Sobredispersión constante

En primer lugar, basado en el apartado 14.1 de Hilbe (2012), consideremos una Poisson-Gamma, donde $Y \sim P(\lambda)$ en la que

$$P[Y_i = y_i] = e^{-\lambda_i}\lambda_i^{y_i}/y_i!, \qquad y_i \in \mathbb{Z}^+, \quad \lambda_i > 0, \quad \forall i = 1, ..., n,$$

donde el parámetro λ_i es una variable aleatoria. Suponemos que su distribución es $Ga(\delta, \mu_i)$ con función de densidad

$$f_{\lambda_i}(\lambda_i) = \frac{1}{(\mu_i - 1)!}\delta^{\mu_i}\lambda_i^{\mu_i - 1}e^{-\delta\lambda_i},$$

donde

$$\mu_i = e^{X_i\beta + offset_i}.$$

Así, podemos expresar la esperanza y varianza de la variable de Poisson como

$$\begin{aligned} E(\lambda_i) &= e^{X_i\beta + offset_i}/\delta \\ V(\lambda_i) &= e^{X_i\beta + offset_i}/\delta^2. \end{aligned}$$

Con lo visto hasta ahora, se puede escribir la función de densidad como

$$\begin{aligned} f_{Y_i}(y_i|x_i) &= \int_0^\infty \frac{e^{-\lambda_i}\lambda_i^{y_i}}{y_i!}\frac{\delta^{\mu_i}}{(\mu_i - 1)!}\lambda_i^{\mu_i - 1}e^{-\lambda_i\delta}d\lambda_i \\ &= \frac{\delta^{\mu_i}}{y_i!(\mu_i - 1)!}\int_0^\infty \lambda_i^{(y_i + \mu_i) - 1}e^{-\lambda_i(\delta + 1)}d\lambda_i \\ &= \frac{\delta^{\mu_i}}{y_i!(\mu_i - 1)!}\frac{(y_i + \mu_i - 1)!}{(\delta + 1)^{y_i + \mu_i}}\underbrace{\int_0^\infty \frac{(\delta + 1)^{y_i + \mu_i}}{(y_i + \mu_i - 1)!}\lambda_i^{(y_i + \mu_i) - 1}e^{-\lambda_i(\delta + 1)}d\lambda_i}_{1} \\ &= \frac{(y_i + \mu_i - 1)!}{y_i!(\mu_i - 1)!}\left(\frac{\delta}{1 + \delta}\right)^{\mu_i}\left(\frac{1}{1 + \delta}\right)^{y_i}. \end{aligned}$$

Los momentos vienen dados por

$$\begin{aligned} E(\lambda_i) &= e^{X_i\beta + offset_i}/\delta, \\ V(\lambda_i) &= e^{X_i\beta + offset_i}/\delta^2. \end{aligned}$$

Llamamos **índice de sobredispersión** al cociente entre la varianza y la media como

$$\frac{V(Y_i)}{E(Y_i)} = \frac{1 + \delta}{\delta},$$

el cual es constante para todas las observaciones. Es decir, la varianza se puede expresar como un múltiplo de la media.

Si consideramos ahora $\alpha = 1/\delta$ ($\alpha > 0$), podemos reescribir el modelo como

$$f_{Y_i}(y_i|x_i) = \frac{(y_i + \mu_i - 1)!}{y_i!(\mu_i - 1)!}\left(\frac{1}{1 + \alpha}\right)^{\mu_i}\left(\frac{\alpha}{1 + \alpha}\right)^{y_i},$$

y

$$\frac{V(Y_i)}{E(Y_i)} = 1 + \alpha.$$

Sin embargo, su distribución no se puede escribir como un elemento de la familia exponencial.

4.2. Sobredispersión variable

En segundo lugar, basado en el apartado 14.2 de Hilbe (2012), se puede obtener la binomial negativa desde dos caminos. La primera forma es viéndolo en su sentido tradicional, es decir, como un modelo de Poisson con heterogeneidad Gamma de media uno. La segunda forma es tratándolo como una función de probabilidad propiamente dicha, independientemente de la distribución de Poisson. Lo que hacemos es ver su función de probabilidad como la probabilidad de observar y fallos antes del r-ésimo éxito en un número finito de sucesos independientes e idénticamente distribuidas (i.i.d.) Bernoulli. Ambos métodos convergen a la misma función log-verosimilitud.

4.2.1. Derivación en términos de Poisson-Gamma

La primera forma, como dijimos antes, consiste en verlo como un modelo de Poisson con heterogeneidad Gamma de media uno. Primero, tenemos que $Y_i \sim P(\mu_i)$ donde $\mu_i = \lambda_i u_i$ de tal forma que

$$\begin{aligned} ln\mu_i &= x_i\beta + \epsilon_i \\ &= ln\lambda_i + lnu_i, \end{aligned}$$

donde $u_i \sim G(\nu, \nu)$ es el llamado **efecto no observado**, con $\nu > 0$.

La distribución de y condicionada a x_i y u_i quedaría

$$f(y_i|u_i) = \frac{e^{-\lambda_i u_i}(\lambda_i u_i)^{y_i}}{y_i!}.$$

La función de densidad de y condicionada a x_i quedaría

$$f(y_i|x_i) = \int_0^\infty \frac{e^{-\lambda_i u_i}(\lambda_i u_i)^{y_i}}{y_i!} g(u_i)du_i,$$

cuyo resultado depende de la función $g()$ que elijamos para modelar u_i. Aquí lo hacemos para el caso en el que $u_i = exp(\epsilon_i)$. Gracias a que la media de Gamma es uno, se tiene el siguiente resultado

$$\begin{aligned} f(y_i|x_i) &= \int_0^\infty \frac{e^{-\lambda_i u_i}(\lambda_i u_i)^{y_i}}{y_i!} \frac{\nu^\nu}{\Gamma(\nu)} u_i^{\nu-1} e^{-\nu u_i} du_i \\ &= \frac{\lambda_i^{y_i}}{\Gamma(y_i+1)} \frac{\nu^\nu}{\Gamma(\nu)} \int_0^\infty e^{-(\lambda_i+\nu)u_i} u_i^{(y_i+\nu)-1} du_i \\ &= \frac{\lambda_i^{y_i}}{\Gamma(y_i+1)} \frac{\nu^\nu}{\Gamma(\nu)} \frac{\Gamma(y_i+\nu)}{(\lambda_i+\nu)^{y_i+\nu}} \underbrace{\int_0^\infty \frac{(\lambda_i+\nu)^{y_i+\nu}}{\Gamma(y_i+\nu)} e^{-(\lambda_i+\nu)u_i} u_i^{(y_i+\nu)-1} du_i}_{1} \\ &= \frac{\lambda_i^{y_i}}{\Gamma(y_i+1)} \frac{\nu^\nu}{\Gamma(\nu)} \frac{\Gamma(y_i+\nu)}{(\lambda_i+\nu)^{y_i+\nu}} \\ &= \frac{\lambda_i^{y_i}}{\Gamma(y_i+1)} \frac{\nu^\nu}{\Gamma(\nu)} \Gamma(y_i+\nu) \left(\frac{\nu}{\lambda_i+\nu}\right)^\nu \frac{1}{\nu^\nu} \left(\frac{\lambda_i}{\lambda_i+\nu}\right)^{y_i} \frac{1}{\lambda_i^{y_i}} \\ &= \frac{\Gamma(y_i+\nu)}{\Gamma(y_i+1)\Gamma(\nu)} \left(\frac{\nu}{\lambda_i+\nu}\right)^\nu \left(\frac{\lambda_i}{\lambda_i+\nu}\right)^{y_i} \\ &= \frac{\Gamma(y_i+\nu)}{\Gamma(y_i+1)\Gamma(\nu)} \left(\frac{1}{1+\lambda_i/\nu}\right)^\nu \left(1-\frac{1}{1+\lambda_i/\nu}\right)^{y_i}. \end{aligned}$$

Reparametrizando $\nu = 1/\alpha$ obtenemos el **modelo de Regresión**

$$\begin{aligned} f(y_i|x_i) &= \frac{\Gamma(y_i + 1/\alpha)}{\Gamma(y_i+1)\Gamma(1/\alpha)} \left(\frac{1}{1+\alpha\mu_i}\right)^{1/\alpha} \left(1 - \frac{1}{1+\alpha\mu_i}\right)^{y_i}, \\ \mu_i &= \lambda_i = exp(x_i\beta + offset_i). \end{aligned}$$

En la parametrización hemos supuesto que $\alpha > 0$. Para el caso $\alpha = 0$ obtenemos el modelo de Poisson. A medida que α aumenta, aumenta también la sobredispersión del modelo.

Escrito como miembro de la familia exponencial sería

$$f(y;\mu,\alpha) = exp\left\{yln\left(\frac{\alpha\mu}{1+\alpha\mu}\right) + \frac{1}{\alpha}ln\left(\frac{1}{1+\alpha\mu}\right) + ln\Gamma\left(y+\frac{1}{\alpha}\right) - ln\Gamma(y+1) - ln\Gamma\left(\frac{1}{\alpha}\right)\right\}.$$

De aquí, podemos obtener las componentes del MLG binomial negativo

$$\begin{aligned} \theta &= ln\left(\frac{\alpha\mu}{1+\alpha\mu}\right), \\ b(\theta) &= -\frac{1}{\alpha}ln\left(\frac{1}{1+\alpha\mu}\right), \\ b'(\theta) &= \frac{\partial b}{\partial \mu}\frac{\partial \mu}{\partial \theta} \\ &= \left(\frac{1}{1+\alpha\mu}\right)\{\mu(1+\alpha\mu)\} \\ &= \mu, \\ b''(\theta) &= \frac{\partial^2 b}{\partial \mu^2}\left(\frac{\partial \mu}{\partial \theta}\right)^2 + \frac{\partial b}{\partial \mu}\frac{\partial^2 \mu}{\partial \theta^2} \\ &= \left\{-\frac{\alpha}{(1+\alpha\mu)^2}\right\}\{\mu^2(1+\alpha\mu)^2\} + \left(\frac{1}{1+\alpha\mu}\right)(1+\alpha\mu)(\mu+2\alpha\mu^2) \\ &= -\alpha\mu^2 + \mu + 2\alpha\mu^2 \\ &= \mu + \alpha\mu^2, \\ V(\mu) &= \mu + \alpha\mu^2, \\ \frac{\partial V(\mu)}{\partial \mu} &= 1 + 2\alpha\mu. \end{aligned}$$

La función log-verosimilitud y la desviación se obtienen

$$\begin{aligned} l(\mu;y,\alpha) &= \sum_{i=1}^n \left\{y_i ln\left(\frac{\alpha\mu_i}{1+\alpha\mu_i}\right) - \frac{1}{\alpha}ln(1+\alpha\mu_i) + ln\Gamma\left(y_i+\frac{1}{\alpha}\right) - ln\Gamma(y_i+1) - ln\Gamma\left(\frac{1}{\alpha}\right)\right\}, \\ D &= 2\sum_{i=1}^n \left\{y_i ln\left(\frac{y_i}{\mu_i}\right) - \left(y_i+\frac{1}{\alpha}\right) ln\left(\frac{1+\alpha y_i}{1+\alpha\mu_i}\right)\right\}. \end{aligned}$$

4.2.2. Derivación en términos de Binomial Negativa

La segunda forma, como dijimos antes, consiste en ver su función de probabilidad como la probabilidad de observar y fallos antes del r-ésimo éxito en un número finito de sucesos independientes e idénticamente distribuidas (i.i.d.) Bernoulli.

Componente aleatoria

La función de probabilidad del modelo binomial negativo se escribe como

$$f(y;r,p) = \binom{y+r-1}{r-1} p^r (1-p)^y.$$

Escrito como miembro de la familia exponencial sería

$$f(y;r,p) = exp\left\{ yln(1-p) + rln(p) + ln\binom{y+r-1}{r-1} \right\}.$$

Así tenemos

$$\begin{aligned} \theta &= ln(1-p), \\ e^\theta &= 1-p, \\ p &= 1-e^\theta, \\ b(\theta) &= -rln(p) = -rln(1-e^\theta), \end{aligned}$$

con $a(\phi) = 1$. La primera y segunda derivada se pueden obtener derivando $b(\phi)$ respecto de θ

$$\begin{aligned} b'(\theta) &= \frac{\partial b}{\partial p}\frac{\partial p}{\partial \theta} \\ &= \left(\frac{-r}{p}\right)\{-(1-p)\} = \frac{r(1-p)}{p} \\ &= \mu, \\ b''(\theta) &= \frac{\partial^2 b}{\partial p^2}\left(\frac{\partial p}{\partial \theta}\right)^2 + \frac{\partial b}{\partial p}\frac{\partial^2 p}{\partial \theta^2} \\ &= \left(\frac{r}{p^2}\right)(1-p)^2 + \frac{r}{p}(1-p) \\ &= \frac{r(1-p)^2 + rp(1-p)}{p^2} = \frac{r(1-p)}{p^2}. \end{aligned}$$

En términos de la media, la varianza se puede escribir como

$$V(\mu) = b''(\theta) = \mu + \frac{\mu^2}{r}.$$

Reparametrizando $\alpha = 1/r$, podemos expresar lo escrito anteriormente como

$$\begin{aligned} p &= 1-e^\theta = \frac{1}{1+\alpha\mu}, \quad * \\ \theta &= ln(1-p) = ln\left(\frac{\alpha\mu}{1+\alpha\mu}\right), \\ b(\theta) &= -\frac{1}{\alpha}ln(p) = \frac{1}{\alpha}ln(1+\alpha\mu), \\ b'(\theta) &= \frac{1-p}{\alpha p} = \mu = \frac{1}{\alpha(e^{-\theta}-1)}, \\ b''(\theta) &= \frac{1-p}{\alpha p^2} = \mu + \alpha\mu^2, \\ g'(\theta) &= \frac{\partial}{\partial \mu}ln\left(\frac{\alpha\mu}{1+\alpha\mu}\right) = \frac{1}{\mu + \alpha\mu^2,} \end{aligned}$$

donde la varianza escrita en términos de la media viene dada por

$$V(\mu) = b''(\theta) = \mu + \alpha\mu^2.$$

La función log-verosimilitud y la desviación se obtienen sustituyendo * en la expresión exponencial de f

$$\begin{aligned} l(\mu; y, \alpha) &= \sum_{i=1}^{n} \{y_i ln\left(\frac{\alpha\mu_i}{1+\alpha\mu_i}\right) - \frac{1}{\alpha} ln(1+\alpha\mu_i) + ln\Gamma\left(y_i + \frac{1}{\alpha}\right) - ln\Gamma(y_i+1) - ln\Gamma\left(\frac{1}{\alpha}\right)\}, \\ D &= 2\sum_{i=1}^{n} \{y_i ln\left(\frac{y_i}{\mu_i}\right) - \left(y_i + \frac{1}{\alpha}\right) ln\left(\frac{1+\alpha y_i}{1+\alpha\mu_i}\right)\}. \end{aligned}$$

Función de enlace

El modelo de Regresión Binomial Negativa se deriva a partir de la función enlace, la cual viene dada por:

$$g(\mu) = \theta = ln((\alpha\mu)/(1+\alpha\mu)) = -ln(1/(1+\alpha\mu))$$

Usando la función enlace canónica el modelo tiene la siguiente forma:

$$g(\mu) = \eta = x\beta = ln(1/(1+\alpha\mu))$$

Usando el enlace logarítmico, $g(\mu) = ln(\mu)$, obtenemos el modelo de regresión binomial negativa.

Capítulo 5

Exceso de ceros en datos de conteo

Las distribuciones que hemos estudiado hasta ahora asumen que pueden existir datos iguales a cero. Algunas variables de conteo que describen datos reales muestran un porcentaje de ceros muy alto. Esa cantidad de ceros no es compatible con las distribuciones Poisson o Binomial Negativa.

La gran diferencia entre el número esperado y el número observado de ceros es un problema en nuestro modelo, ya que la estimación de los coeficientes puede no ser fiable. Este hecho nos crea intervalos de confianza más chicos de lo que corresponde, obteniendo como consecuencia variables significativas que no lo son. Además, la precisión en las inferencias se verán altamente afectadas. Para corregir este problema se debe hacer un ajuste a la función o usar otro modelo diferente.

Es por ello que es de vital importancia determinar correctamente la procedencia de los ceros, los cuales pueden venir de dos fuentes distintas: **ceros auténticos** y **falsos ceros**.

Podemos entender esta diferencia entre ambos ceros con el siguiente ejemplo: el estudio del número de accidentes de coche en el último mes, aquellos que no han tenido ningún accidente puede ser porque

1. Falso cero: El coche no ha sido utilizado en ningún momento.
2. Cero auténtico: El coche ha sido utilizado pero no ha tenido ningún accidente.

Según los apuntes Pino-Mejías (2017), hay dos estrategias para lidiar con el problema de los ceros:

1. La primera, denominada **ceros truncados**, asumimos que todos los ceros son iguales (tanto los falsos como los auténticos). El modelo de cero truncado consiste en separar los ceros del resto de observaciones, de modo que usamos un modelo de poisson o binomial negativa para modelizar la probabilidad de cero, mientras que para las demás observaciones se emplearía una distribución truncada en el cero. Estos modelos reciben el nombre de **modelos truncados en cero**.
2. La segunda posibilidad, denominada **ceros inflados**, se caracteriza porque los ceros provienen de dos procesos distintos: el proceso binomial y el proceso de Poisson. Primero usamos un modelo de poisson o binomial negativa para modelizar la probabilidad de medir un 0 (los falsos ceros). Posteriormente, se modeliza la probabilidad de obtener el resto de valores, incluyendo ceros (los ceros auténticos).

Estos modelos reciben el nombre de **modelos mezclados** o **modelos de cero inflados**.

Ambos tipos de modelos nos permiten abordar el exceso de ceros. La elección de uno u otro modelo debe basarse en el conocimiento a priori de las fuentes de ceros en nuestro problema.

A continuación, tomando como referencia el apartado 2.5 de Calcaterra (2017), estudiamos los modelos para trabajar con estas variables de conteo que describen datos con alta cantidad de ceros.

5.1. Modelos de regresión truncados en ceros

Los modelos truncados implican que en algún punto del recorrido de la variable, un determinado valor está totalmente ausente. Si el valor que no se observa es el cero entonces se dice que es un modelo "Truncado en Cero". Este tipo de modelos no admite conteos ceros, por lo que la distribución no debe tener este valor en su recorrido para poder modelar los datos adecuadamente. Las distribuciones Poisson y Binomial Negativa pueden ser modificadas para llegar a sus versiones truncadas.

5.1.1. Poisson cero truncado

El modelo de Poisson asume la posibilidad de haber ceros incluso cuando no hay ningun cero en el registro de los datos. Cuando se excluye la posibilidad de que la variable respuesta no pueda tomar el valor cero, el modelo de Poisson no es adecuado. En este caso, se usa el modelo de Poisson cero truncado, el cual consiste en reajustar la función de probabilidad de la Poisson adecuadamente para excluir el valor cero. Esto se realiza sustituyendo la probabilidad de que la variable respuesta tome el valor cero ($f(0;\mu) = exp(-\mu)$) por 1.

La función de probabilidad de este modelo sería

$$f(y_i;\mu|y_i > 0) = \frac{e^{-\mu}\mu^{y_i}}{y_i!(1-e^{-\mu})}.$$

La función log-verosimilitud de este modelo sería

$$l(\mu;y|y>0) = \sum_{i=1}^{n} \left\{ y_i X_i\beta - exp(X_i\beta) - ln(y_i!) - ln(1-exp(-exp(X_i\beta))) \right\}.$$

5.1.2. Binomial negativa cero truncado

La lógica usada en esta distribución es la misma que empleamos a la hora de elaborar la distribución de Poisson cero truncado, se trunca la distribución en $y = 0$, es decir, se sustituye la probabilidad de que la variable respuesta tome el valor cero ($f(0;r,p) = p^r$) por 1.

La función de distribución de este modelo sería

$$f(y;r,p|y>0)=\frac{\binom{y+r-1}{y}\left(\frac{\alpha}{\alpha+1}\right)^r\left(\frac{1}{\alpha+1}\right)^\alpha}{1-\left(\frac{\alpha}{\alpha-1}\right)^r},\qquad \alpha=\frac{p}{1-p}.$$

5.2. Modelos de regresión Hurdle

Otro modelo que maneja el exceso de ceros es el denominado modelo "Hurdle" que difiere del modelo anterior en cómo entienden el origen o generación de los ceros extras. Este modelo es un modelo de dos componentes, o **modelo en dos partes** que combina:

1. Un proceso binario para los valores que están por encima o por debajo del valor de selección, modelado por medio de un proceso logit, para describir la probabilidad de que se cruce el "obstáculo". Dicho proceso modela datos que toman dos valores: éxito o fracaso. Esta componente del modelo solo genera conteos cero.

Sea y_i la observación i, $y_i \sim B(p_i)$ siendo $p_i = E(y_i/x_i)$ la probabilidad de éxito.

$$\begin{aligned} E(Y/X) &= \pi_i = \frac{e^{X\beta}}{1+e^{X\beta}}, \\ \pi_i &= \frac{1}{1+e^{-X\beta}}. \end{aligned}$$

Con lo anterior llegamos a

$$\frac{\pi_i}{1-\pi_i}=\frac{1+e^{X\beta}}{1+e^{-X\beta}}=e^{X\beta}.$$

Tomando logaritmo en ambos miembros

$$log\left(\frac{\pi_i}{1-\pi_i}\right)=X\beta,$$

que es lo que se conoce como transformación **logit** de π_i.

El conciente entre la probabilidad de que ocurra el suceso y que no ocurra es lo que se conoce como *odds*, cuya expresión es $\frac{\pi_i}{1-\pi_i}$. Cuanto mayor sea esta razón, mayor probabilidad habrá de que ocurra el suceso.

2. Un proceso que genera solo los conteos mayores que cero mediante un modelo cero truncado. Esta componente se puede modelar mediante un modelo Poisson o Binomial Negativo como hemos visto anteriormente.

Según el apartado 2.2 de Achim Zeileis, el modelo Hurdle tiene la forma

$$f(y;x,z,\beta,\gamma)=\begin{cases} f_{cero}(0;z,\gamma), & y=0 \\ (1-f_{cero}(0;z,\gamma))f_{cont}(y;x,\beta)/(1-f_{cont}(0;x,\beta)), & y>0. \end{cases}$$

En este modelo, se considera que los datos son generados de tal forma que un proceso genera conteos positivos después de cruzar un obstáculo. Hasta que dicha barrera es cruzada, el proceso genera conteos cero. El vector de parámetros β y γ del modelo se estiman por máxima verosimilitud y pueden ser maximizados por separado.

5.3. Modelos de regresión cero inflado

Frecuentemente la sobredispersión se produce por la presencia de un número mayor de conteos nulos que el esperado bajo la distribución Poisson supuesta para las observaciones. En estos casos, una manera de modelar datos de conteo con excesivos ceros es la denominada Regresión "Zero-inflated", o regresión cero inflado, introducida por Lambert (1992). La misma asume que los conteos nulos pueden provenir de dos fuentes diferentes, por lo que considera una mezcla de dos procesos estadísticos, uno que genera sólo conteos iguales a cero y otro que genera tanto conteos ceros como distintos de cero. A diferencia de los modelos truncados en cero, la primera componente genera solo conteos cero, pero la segunda genera el rango completo de conteos, incluyendo los ceros.

5.3.1. Poisson cero inflado

Es un modelo mixto de dos componentes que da mayor peso a la probabilidad de que la variable sea igual a cero, por lo que la función de probabilidad para un modelo de regresión cero inflado es una mezcla de una función de probabilidad concentrada en cero y un modelo perteneciente a la familia exponencial.

El modelo nos dice que existe una probabilidad p_i de que la i-ésima observación sea siempre igual a cero, es decir, que la variable respuesta tome siempre el valor cero, y una probabilidad $1 - p_i$ de que el valor de la i-ésima observación proceda de una distribución de Poisson.

La distribución viene dada por

$$P(Y_i = y_i) = \begin{cases} p_i + (1-p_i)e^{-\lambda_i} & si \quad y_i = 0 \\ \dfrac{(1-p_i)e^{-\lambda_i}\lambda_i^k}{k!} & si \quad y_i = k > 0, \end{cases}$$

donde los vectores de parámetros λ_i y p_i satisfacen las condiciones

$$\begin{aligned} log(\lambda_i) &= Z_i\beta, \\ logit(p_i) &= log(\frac{p_i}{1-p_i}) = W_i\gamma, \end{aligned}$$

donde Z_i y W_i son vectores de variables explicativas.

5.3.2. Binomial negativa cero inflado

Un modelo de regresión cero inflado se obtiene como resultado de mezclar una distribución binomial y una distribución binomial negativa degenerada en cero. Sea $Z \sim BN(\mu, \phi)$.

La observación Y con ceros inflados tiene la siguiente función de probabilidad

$$P(Y_i = y_i) = \begin{cases} p_i + (1-p_i)\left(\dfrac{\phi}{\mu_i + \phi}\right)^{\phi}, & y_i = 0 \\ (1-p_i)\dfrac{\Gamma(\phi + y_i)}{\Gamma(y_i+1)\Gamma(\phi)}\left(\dfrac{\mu_i}{\mu_i+\phi}\right)^{y_i}\left(\dfrac{\phi}{\mu_i+\phi}\right)^{\phi}, & y_i = k > 0. \end{cases}$$

Capítulo 6

Criterios de selección de modelos

El objetivo de este apartado es obtener test que comparen diferentes modelos y nos permitan seleccionar cual de ellos se ajusta mejor a nuestro modelo y por tanto, deberíamos de usar. Según el apartado 2.6.5 de Calcaterra (2017), estos tests de criterios de selección del modelo son tests comparativos. Aquellos tests que presentan valores menores son los que indican un mejor ajuste. Los principales tests de criterio de la Información son **Akaike Information Criterion** (AIC) y **Bayesian Information Criterion** (BIC). A continuación se estudian ambos estadísitcos.

6.1. Criterio de información de Akaike

El estadístico AIC tiene la forma

$$AIC = \frac{-2(L-k)}{n},$$

donde L representa la función de verosimilitud del modelo, k el número de variables y n el número de observaciones.

El término $2k$ penaliza la cantidad de variables, dado que al aumentar la cantidad de los mismos el modelo es más verosímil, por lo que $-2L$ se vuelve más chico. Por el principio de parsimonia, en igualdad de condiciones, el modelo más sencillo, suele ser el mejor.

6.2. Criterio de información Bayesiana

El estadístico BIC tiene la forma

$$BIC = -2L + klog(n),$$

donde L representa la función de verosimilitud del modelo, k el número de variables y n el número de observaciones.

Este estadístico da un mayor peso al término de ajuste $klog(n)$ que el AIC.

Capítulo 7

Aplicación a los accidentes de tráfico

En este capítulo se procede a aplicar todos los modelos vistos anteriormente para el estudio de variables de conteo a una base de datos.

7.1. Estadística en los accidentes

Cada vez que ocurre algún tipo de accidente, los cuerpos de seguridad del Estado son los encargados de, a parte de solucionar todo lo posible, anotar y dejar constancia de lo ocurrido. Estos datos pasan a digitalizarse y se hacen muchos estudios con ellos. La finalidad de estos puede ser variada, pero siempre con la intención de disminuir el número de fallecidos. Por ejemplo:

1. Dentro de la ciudad, el ayuntamiento puede considerar necesario modificar los límites de velocidad permitidos dentro de ella.
2. Dentro de la ciudad, el ayuntamiento puede considerar necesario incorporar diversos tipos de obstáculos (como resaltos y badenes) o luces (actualmente se colocan en los pasos de peatones y los semáforos) con la intención de disminuir ese número de fallecidos.
3. La DGT (Dirección General de Tráfico) puede considerar necesario modificar los límites de velocidad permitidos, poner zonas de prohibido adelantar, incorporar radares y controles en zonas de riesgo y días de mayor movimiento de personas.
4. Modificar la carretera o redirigir el tráfico en aquellos lugares donde existen puntos negros de las carreteras (puntos donde se concentran gran cantidad de accidentes).
5. También los seguros necesitan de estos datos para subsistir.
6. Incorporación de cada vez más sistemas de seguridad en los vehículos.

Como podemos observar, la capacidad de obtener todos estos datos tras los accidentes pueden servir de mucho para poner medidas y evitar o reducir los accidentes y por tanto, las víctimas en las carreteras. Esto es, que gracias a los avances de la tecnología nos permiten hacer estudios para poder sacar conclusiones y extraer medidas más reales y efectivas, las cuales poder aplicar a las carreteras, vehículos y conciencia de las personas para ese número de fallecidos que, desgraciadamente, es positivo.

7.2. Aplicación a nuestra base de datos

Todos los datos necesarios para realizar nuestro estudio han sido cogidos de las paginas de la DGT, del INE y del Ministerio de Fomento. A partir de ellos se ha creado una base de datos en el fichero datos.RData.

7.2.1. Lectura de los datos

Nuestra base de datos está formada por 780 observaciones y 14 variables. En nuestro estudio tenemos dos variables objetivo y 9 variables explicativas (3 más que se explicarán más adelante). Las variables objetivo son: número de fallecidos en carreteras interurbanas y urbanas, representadas por fallecidos_interurbana y fallecidos_urbana, respectivamente. Las variables explicativas son: provincia, año, comunidades autónomas, población, número de vehículos, km de carreteras totales y precipitación, temperatura y horas de sol media en un año. Las variables aparecen recogidas por provincia, representadas por provincia, year, CCAA, poblacion, numero_vehiculos, km_carretera, precipitacion_media, temperatura_media y horas_sol_media, respectivamente.

Es de notar que cada vez que estudiemos un modelo vamos a hacerlo para cada una de nuestras dos variables objetivo.

En primer lugar, cargamos nuestra base de datos ya previamente depurada en el script Base_de_datos.R, y luego vemos un resumen de los valores que toman cada variable.

```
load(file="datos.RData")
summary(datos_f)
```

```
##   provincia         fallecidos_interurbana fallecidos_urbana
##  Length:780         Min.   :  0.0          Min.   :  0.00
##  Class :character   1st Qu.: 22.0          1st Qu.:  3.00
##  Mode  :character   Median : 37.0          Median :  7.00
##                     Mean   : 48.0          Mean   : 12.23
##                     3rd Qu.: 63.5          3rd Qu.: 13.00
##                     Max.   :253.0          Max.   :149.00
##       year      numero_vehiculos    poblacion       precipitacion_media
##  Min.   :2002   Min.   :  39411   Min.   :  66526   Min.   :  6.658
##  1st Qu.:2005   1st Qu.: 202288   1st Qu.: 321755   1st Qu.: 26.600
##  Median :2009   Median : 382112   Median : 584128   Median : 32.104
##  Mean   :2009   Mean   : 569163   Mean   : 863809   Mean   : 40.236
##  3rd Qu.:2013   3rd Qu.: 645352   3rd Qu.:1002091   3rd Qu.: 43.019
##  Max.   :2016   Max.   :4474787   Max.   :6466996   Max.   :133.567
##  temperatura_media horas_sol_media  km_carretera         CCAA
##  Min.   :10.68     Min.   :128.4   Min.   :  26.83   Length:780
##  1st Qu.:13.16     1st Qu.:207.4   1st Qu.:2369.99   Class :character
##  Median :15.53     Median :236.7   Median :3322.39   Mode  :character
##  Mean   :15.50     Mean   :227.0   Mean   :3184.52
##  3rd Qu.:18.25     3rd Qu.:255.7   3rd Qu.:3984.99
##  Max.   :21.75     Max.   :283.4   Max.   :6364.66
```

```
head(datos_f)
```

```
##          provincia fallecidos_interurbana fallecidos_urbana year
## 1      Araba/Álava                    40                 9 2002
## 2         Albacete                    61                 7 2002
## 3 Alicante/Alacant                   155                 8 2002
## 4          Almería                    97                 8 2002
## 5            Ávila                    45                 3 2002
## 6          Badajoz                    74                19 2002
##   numero_vehiculos poblacion precipitacion_media temperatura_media
## 1           174235    288558              44.375            11.808
## 2           202325    356661              33.150            14.842
## 3           998821   1492669              20.933            18.342
## 4           339226    514237              17.058            18.583
## 5            93558    163305              28.358            11.258
## 6           336611    663723              26.525            16.917
##   horas_sol_media km_carretera        CCAA
## 1         169.133      1463.33      PVasco
## 2         261.133      3710.45    CLMancha
## 3         265.158      2704.29 CValenciana
## 4         268.508      2368.19   Andalucía
## 5         226.983      2545.57       CLeón
## 6         250.375      4882.83 Extremadura
```

Calculamos la media y la varianza tanto de los fallecidos en carreteras interurbanas como urbanas.

```
mean(datos_f$fallecidos_interurbana)
```

```
## [1] 48.00256
```

```
var(datos_f$fallecidos_interurbana)
```

```
## [1] 1446.339
```

```
mean(datos_f$fallecidos_urbana)
```

```
## [1] 12.22564
```

```
var(datos_f$fallecidos_urbana)
```

```
## [1] 347.2712
```

Se observa que la media y la varianza de ambas variables son bastante diferentes, por lo que ya tenemos que esperar que haya sobredispersión en los datos.

7.2.2. Estudio descriptivo

En primer lugar, realizamos el estudio descriptivo con los fallecidos en vías interurbanas.

Se estudian las relaciones existentes entre fallecidos_interurbana y las demás variables (menos fallecidos_urbana) con el siguiente gráfico:

```
attach(datos_f)
pairs(fallecidos_interurbana~year+numero_vehiculos+poblacion+
        precipitacion_media+temperatura_media+horas_sol_media+
        km_carretera,panel=panel.smooth)
```

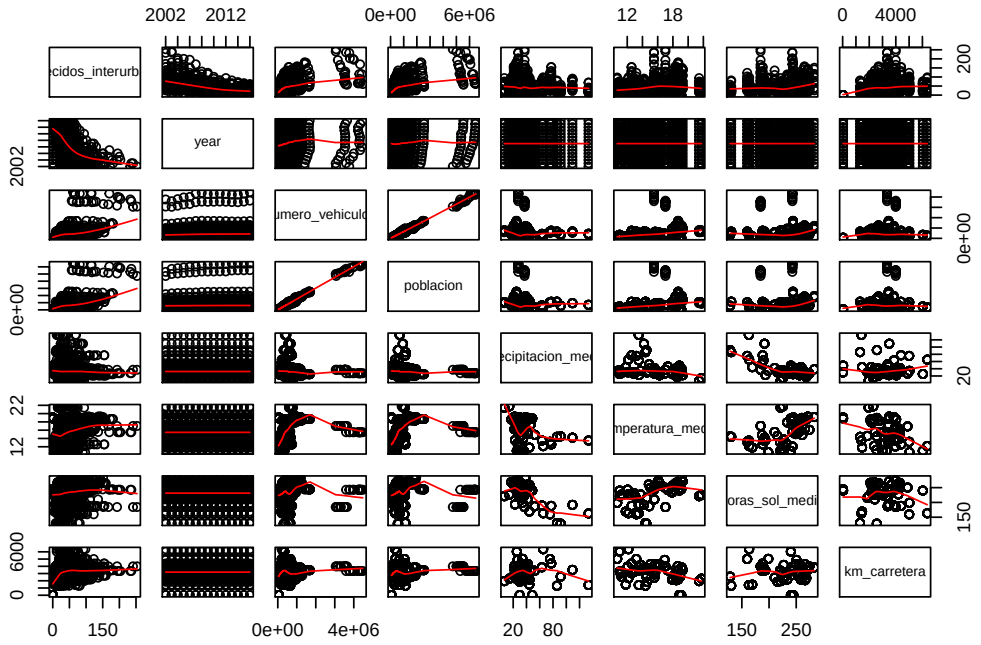

Figura 7.1: Figura con todos los pares de variables para fallecidos interurbana

El número de fallecidos en vías interurbanas desciende a lo largo de los años, aumenta con la población y el número de vehículos, aumenta levemente con respecto a los km de carretera y horas media de sol, mientras que la precipitación y temperatura media no destaca. Es de notar que aquellas gráficas donde hay huecos es debido a las dos provincias con más habitantes (Madrid y Barcelona). Además, viendo el cuadro referente a la población y el número de vehículos, podemos intuir que ambas están altamente correladas, cosa que puede afectar a nuestro estudio.

Se dibujan, a modo de ejemplo, 2 gráficas referentes a la provincia de Sevilla.

```
library(ggplot2)
ggplot( datos_f[datos_f$provincia=="Sevilla",],
        aes(x=year, y=fallecidos_interurbana)) +
        geom_point(size=1)+ geom_line()
```

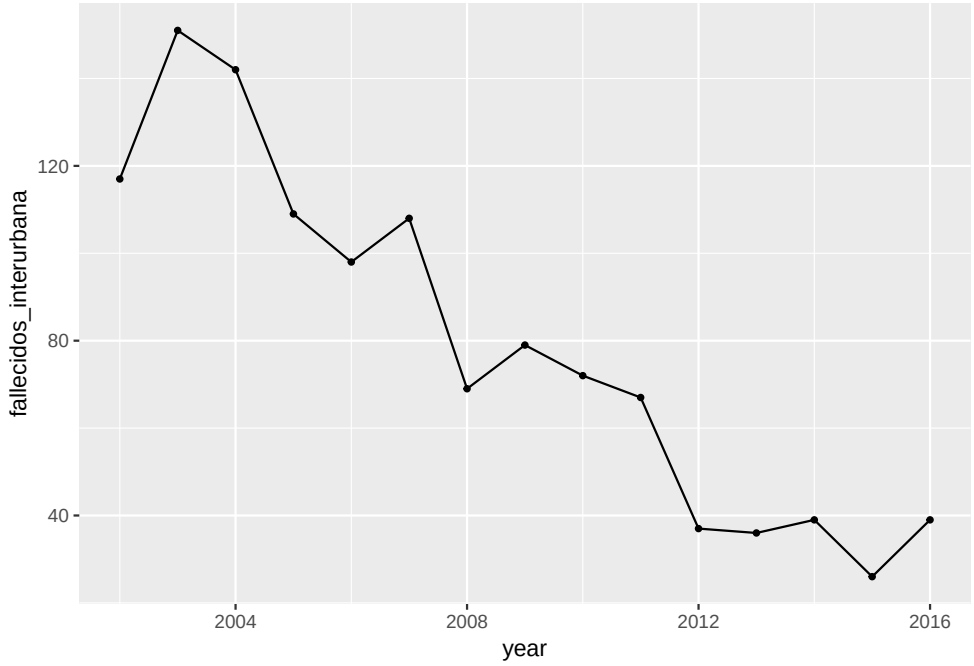

Figura 7.2: Número de fallecidos en vías interurbanas en Sevilla por año

En la gráfica se ve como el número de fallecidos, que al inicio de nuestro estudio superaba los 110 fallecidos por año, ha disminuido notablemente hasta mantenerse por debajo de los 40 fallecidos por año, menos de la mitad.

```
ggplot( datos_f[datos_f$provincia=="Sevilla",],
        aes(x=poblacion, y=fallecidos_interurbana)) +
        geom_point(size=1)+ geom_line()
```

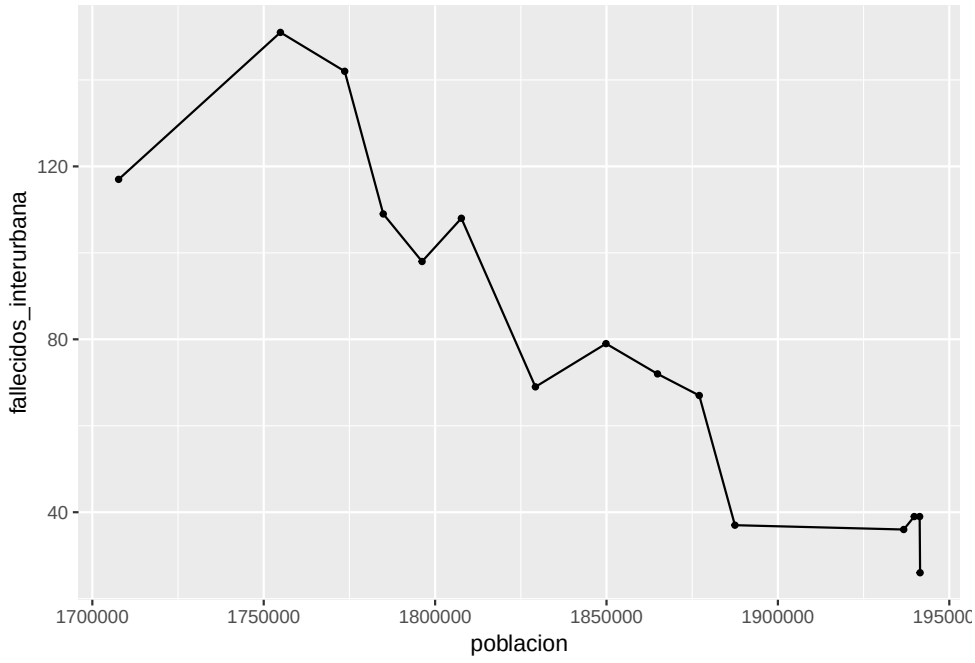

Figura 7.3: Número de fallecidos en vías interurbanas en Sevilla por población

Esta gráfica es muy similar a la anterior, cosa que nos inclina a pensar que tanto la población como el año están correladas.

Para acabar con el estudio descriptivo de los fallecidos_interurbana, se dibuja:

```
library(lattice)
xyplot(fallecidos_interurbana~year|CCAA,col=1)
```

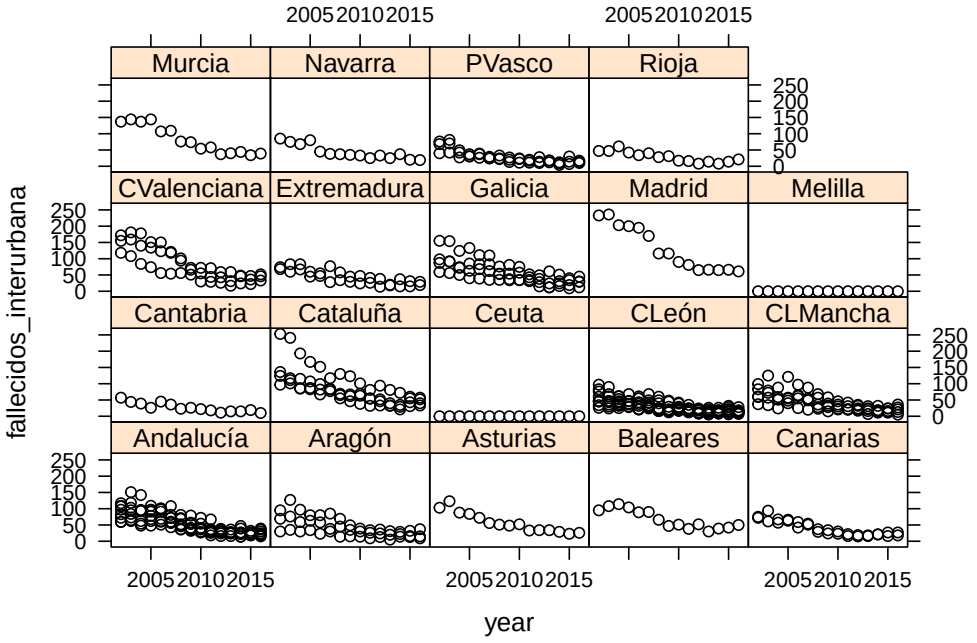

Figura 7.4: Número de fallecidos en vías interurbanas por año en cada Comunidad Autónoma

En esta gráfica se muestra por cada comunidad autónoma como el número de fallecidos en vías interurbanas desciende a lo largo de los años, pudiendose notar aquellas comunidades donde hay más y menos provincias.

En segundo lugar, realizamos el estudio descriptivo con los fallecidos en vías urbanas.

Se estudian las relaciones existentes entre fallecidos_urbana y las demás variables (menos fallecidos_interurbana) con el siguiente gráfico:

```
pairs(fallecidos_urbana~year+numero_vehiculos+poblacion+precipitacion_media+
      temperatura_media+horas_sol_media+km_carretera,panel=panel.smooth)
```

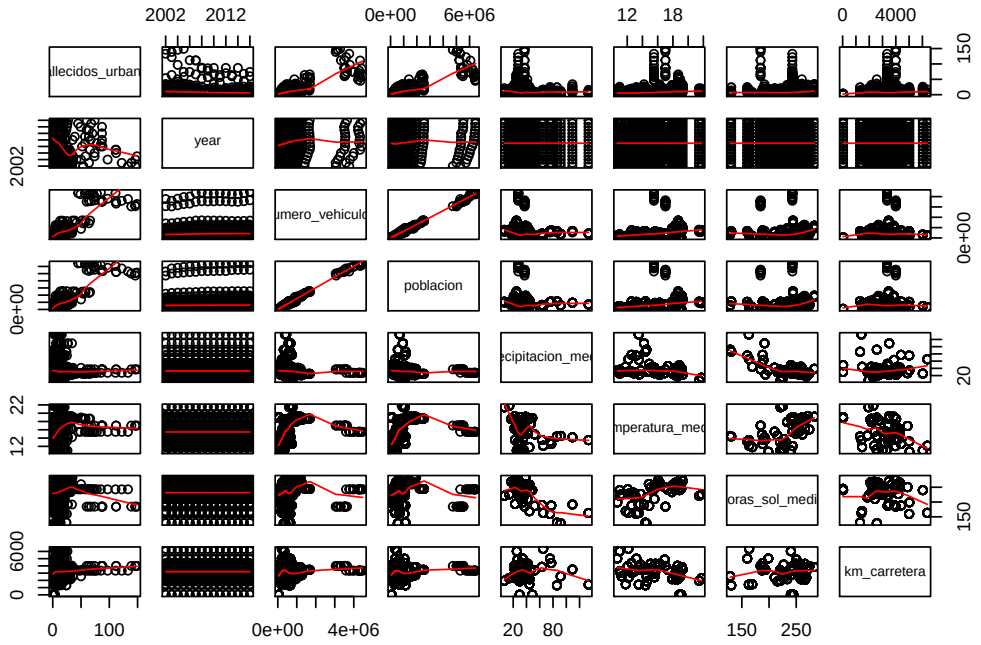

Figura 7.5: Figura con todos los pares de variables en vías urbanas

El número de fallecidos en vías urbanas aumenta con la población y el número de vehículos, la interacción con las demás variables no es de destacar. Al igual que ocurría con los fallecidos en vías interurbanas, aquellas gráficas donde hay huecos es debido a las dos provincias con más habitantes (Madrid y Barcelona). Además, viendo el cuadro referente a la población y el número de vehículos, podemos intuir que ambas están altamente correladas, cosa que puede afectar a nuestro estudio.

Se dibujan, a modo de ejemplo, 2 gráficas referentes a la provincia de Sevilla.

```
ggplot( datos_f[datos_f$provincia=="Sevilla",],
        aes(x=year, y=fallecidos_urbana)) +
        geom_point(size=1)+ geom_line()
```

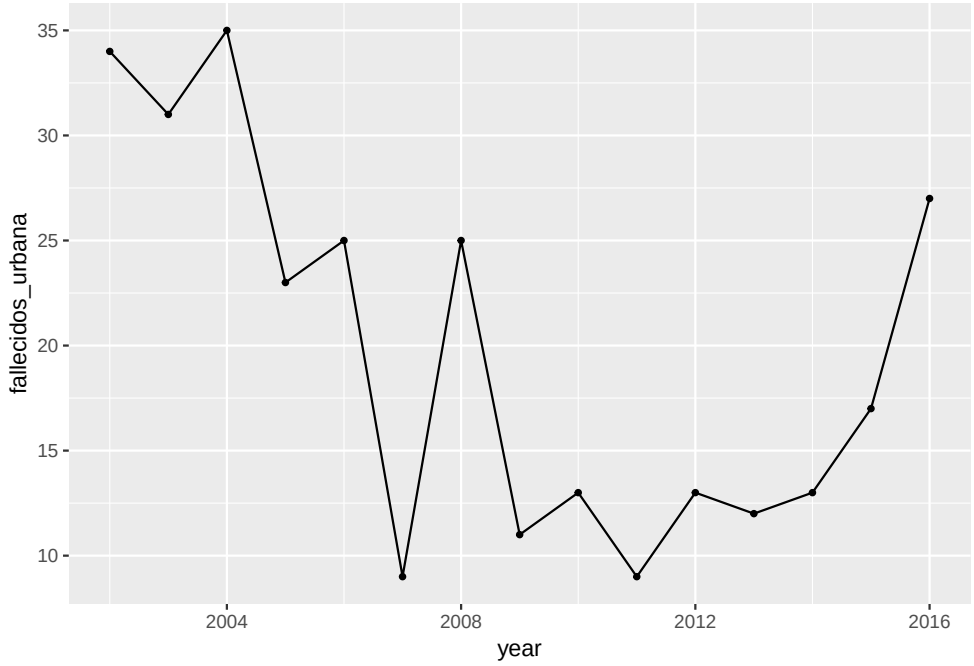

Figura 7.6: Número de fallecidos en vías urbanas en Sevilla por año

En la gráfica se ve como el número de fallecidos, excepto por un par de años donde ha habido un pico con mayor número de fallecidos, ha ido descendiendo desde estar entorno a los 33 fallecidos por año hasta los 12.

```
ggplot( datos_f[datos_f$provincia=="Sevilla",],
        aes(x=poblacion, y=fallecidos_urbana)) +
        geom_point(size=1)+ geom_line()
```

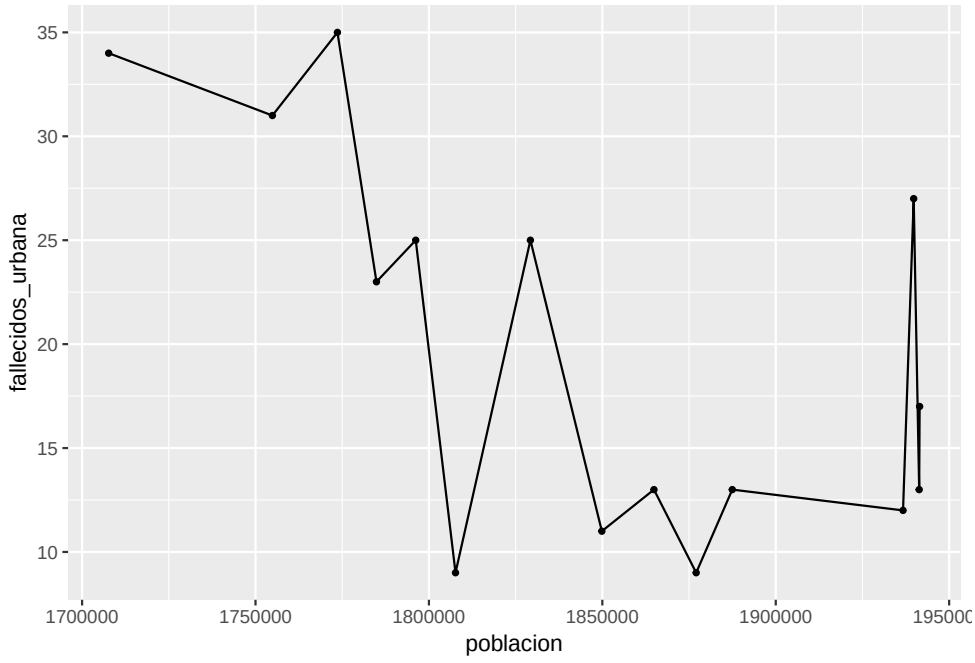

Figura 7.7: Número de fallecidos en vías urbanas en Sevilla por población

Esta gráfica es muy similar a la anterior, cosa que nos inclina a pensar que tanto la población como el año están correladas.

Para acabar con el estudio descriptivo de los fallecidos_urbana, se dibuja:

```
xyplot(fallecidos_urbana~year|CCAA,col=1)
```

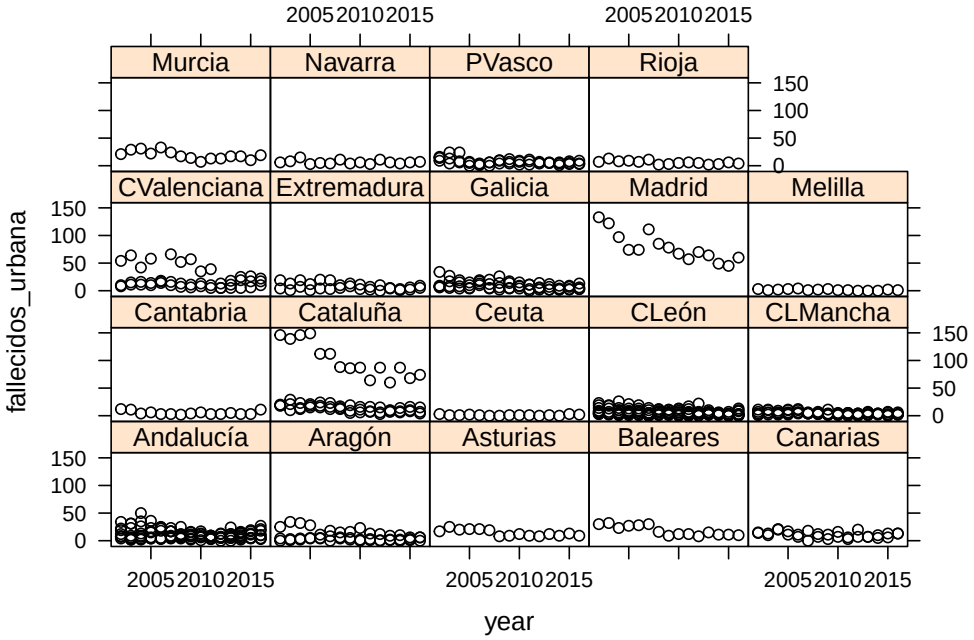

Figura 7.8: Número de fallecidos en vías urbanas por año en cada Comunidad Autónoma

En esta gráfica se muestra por cada comunidad autónoma como el número de fallecidos en vías urbanas desciende a lo largo de los años, pudiendose notar aquellas comunidades donde hay más y menos provincias.

7.2.3. Aplicación del modelo de Poisson

Comenzamos el estudio con el modelo de Poisson. Se ha ultilizado las siguientes variables explicativas: year, numero_vehiculos, poblacion, precipitacion_media, temperatura_media, horas_sol_media, km_carretera, CCAA.

7.2.3.1. Vías interurbanas

```
modeloTotal_i_p1<-glm(fallecidos_interurbana~year+
                  numero_vehiculos+poblacion+precipitacion_media+
                  temperatura_media+horas_sol_media+
                  km_carretera+CCAA,family="poisson",data=datos_f)
```

```
summary(modeloTotal_i_p1)
```

```
##
## Call:
```

```
## glm(formula = fallecidos_interurbana ~ year + numero_vehiculos +
##     poblacion + precipitacion_media + temperatura_media + horas_sol_media +
##     km_carretera + CCAA, family = "poisson", data = datos_f)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -5.5873  -1.3360  -0.1337   0.8937   5.2869
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)          2.289e+02  2.856e+00  80.127  < 2e-16 ***
## year                -1.133e-01  1.425e-03 -79.472  < 2e-16 ***
## numero_vehiculos     8.105e-08  9.735e-08   0.833 0.405105
## poblacion            1.620e-07  6.182e-08   2.620 0.008788 **
## precipitacion_media  3.154e-03  5.049e-04   6.246 4.22e-10 ***
## temperatura_media    6.631e-03  7.571e-03   0.876 0.381114
## horas_sol_media      6.618e-03  4.348e-04  15.222  < 2e-16 ***
## km_carretera         1.313e-04  8.466e-06  15.505  < 2e-16 ***
## CCAAAragón          -1.387e-01  3.388e-02  -4.093 4.25e-05 ***
## CCAAAsturias         3.817e-01  6.170e-02   6.186 6.16e-10 ***
## CCAABaleares         5.378e-01  4.253e-02  12.646  < 2e-16 ***
## CCAACanarias         1.162e-01  5.189e-02   2.239 0.025171 *
## CCAACantabria        2.044e-01  7.901e-02   2.587 0.009682 **
## CCAACataluña         4.548e-01  2.677e-02  16.990  < 2e-16 ***
## CCAACeuta           -1.948e+01  5.136e+02  -0.038 0.969739
## CCAACLeón           -2.359e-01  4.645e-02  -5.079 3.80e-07 ***
## CCAACLMancha        -1.200e-01  3.025e-02  -3.966 7.30e-05 ***
## CCAACValenciana      3.695e-01  2.383e-02  15.508  < 2e-16 ***
## CCAAExtremadura     -2.696e-01  3.441e-02  -7.833 4.75e-15 ***
## CCAAGalicia          3.596e-01  4.922e-02   7.306 2.75e-13 ***
## CCAAMadrid          -6.658e-02  6.131e-02  -1.086 0.277462
## CCAAMelilla         -1.945e+01  5.136e+02  -0.038 0.969791
## CCAAMurcia           2.652e-01  3.383e-02   7.839 4.56e-15 ***
## CCAANavarra          1.925e-01  5.617e-02   3.426 0.000612 ***
## CCAAPVasco           1.816e-01  6.507e-02   2.791 0.005258 **
## CCAARioja            2.503e-02  5.762e-02   0.434 0.663980
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 20905.0  on 779  degrees of freedom
## Residual deviance:  2312.1  on 754  degrees of freedom
## AIC: 6489.9
##
## Number of Fisher Scoring iterations: 14
```

Es de notar que lo coeficientes beta están calculados en función de la comunidad autónoma Andalucía. Antes de sacar cualquier conclusión, vamos a comprobar si nuestra

percepción de la correlación entre variables hecha en el estudio descriptivo finalmente se da. Esto se hace mediante la siguiente función:

```
library(car)
vif(modeloTotal_i_p1)
```

```
##                            GVIF Df GVIF^(1/(2*Df))
## year                   1.239854  1        1.113488
## numero_vehiculos     392.820261  1       19.819694
## poblacion            359.159640  1       18.951508
## precipitacion_media    4.814127  1        2.194112
## temperatura_media     15.141814  1        3.891248
## horas_sol_media        9.243337  1        3.040286
## km_carretera           2.828548  1        1.681829
## CCAA                2276.478791 18        1.239531
```

Efectivamente, se comprueba que las variables poblacion y numero_vehiculos estan muy correladas. ¿Cómo solucionamos este problema? Pues bien, se crean las siguientes 3 variables:

```
datos_f$fallecidos_int_milhab<-
  100000*datos_f$fallecidos_interurbana/datos_f$poblacion
datos_f$fallecidos_urb_milhab<-
  100000*datos_f$fallecidos_urbana/datos_f$poblacion
datos_f$numero_vehiculos_milhab<-
  100000*datos_f$numero_vehiculos/datos_f$poblacion
```

Además, como estamos trabajando con variables de conteo, tenemos que redondearlas:

```
datos_f$fallecidos_int_milhabr<-round(datos_f$fallecidos_int_milhab)
datos_f$fallecidos_urb_milhabr<-round(datos_f$fallecidos_urb_milhab)
datos_f$numero_vehiculos_milhabr<-round(datos_f$numero_vehiculos_milhab)
```

Procedemos a rehacer el estudio sustituyendo las respectivas variables por estas nuevas y no incluyendo la variable poblacion, teniendo en cuenta ahora que los resultados van a venir dados por cada cien mil habitantes.

```
modeloTotal_i_p<-glm(fallecidos_int_milhabr~year+
                  numero_vehiculos_milhabr+precipitacion_media+
                  temperatura_media+horas_sol_media+
                  km_carretera+CCAA,family="poisson",data=datos_f)
```

```
summary(modeloTotal_i_p)
```

```
##
## Call:
## glm(formula = fallecidos_int_milhabr ~ year + numero_vehiculos_milhabr +
##     precipitacion_media + temperatura_media + horas_sol_media +
##     km_carretera + CCAA, family = "poisson", data = datos_f)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
```

```
## -4.1870  -0.6238  -0.0793   0.4212   4.3803
##
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)               2.853e+02  8.024e+00  35.553  < 2e-16 ***
## year                     -1.414e-01  4.075e-03 -34.692  < 2e-16 ***
## numero_vehiculos_milhabr  2.601e-05  2.481e-06  10.485  < 2e-16 ***
## precipitacion_media      -1.729e-03  1.201e-03  -1.440 0.149969
## temperatura_media        -1.230e-01  1.601e-02  -7.686 1.52e-14 ***
## horas_sol_media           4.339e-03  8.695e-04   4.990 6.03e-07 ***
## km_carretera             -3.180e-05  1.673e-05  -1.901 0.057326 .
## CCAAAragón                2.030e-01  7.586e-02   2.677 0.007436 **
## CCAAAsturias             -4.330e-02  1.576e-01  -0.275 0.783565
## CCAABaleares             -4.225e-01  1.217e-01  -3.473 0.000514 ***
## CCAACanarias             -1.002e-01  1.341e-01  -0.747 0.454790
## CCAACantabria            -2.260e-02  1.732e-01  -0.130 0.896233
## CCAACataluña              8.226e-02  6.729e-02   1.222 0.221522
## CCAACeuta                -1.915e+01  8.303e+02  -0.023 0.981603
## CCAACLeón                 1.466e-02  1.001e-01   0.146 0.883562
## CCAACLMancha              2.101e-01  6.790e-02   3.094 0.001974 **
## CCAACValenciana          -7.314e-02  7.321e-02  -0.999 0.317773
## CCAAExtremadura           1.108e-01  8.079e-02   1.371 0.170353
## CCAAGalicia               2.091e-01  1.066e-01   1.962 0.049807 *
## CCAAMadrid               -1.483e+00  1.866e-01  -7.945 1.93e-15 ***
## CCAAMelilla              -1.911e+01  8.443e+02  -0.023 0.981938
## CCAAMurcia                1.125e-02  1.150e-01   0.098 0.922124
## CCAANavarra              -1.150e-01  1.290e-01  -0.892 0.372622
## CCAAPVasco               -3.509e-01  1.414e-01  -2.482 0.013048 *
## CCAARioja                 2.228e-01  1.105e-01   2.016 0.043816 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 3772.90  on 779  degrees of freedom
## Residual deviance:  647.38  on 755  degrees of freedom
## AIC: 3508.5
##
## Number of Fisher Scoring iterations: 15
```

Como se preveía, un aumento en las variables año o la temperatura producen una disminución del número esperado de fallecidos por cien mil habitantes (hecho que se puede explicar mediante el argumento de que al haber mas temperatura sale menos gente a conducir). Un aumento del número de vehículos por cien mil habitantes y de las horas de sol media producen un aumento del número de fallecidos por cien mil habitantes. Particularizando el caso de Andalucía, podemos ver como lugares como Cataluña o Galicia hay un mayor número esperado de fallecidos por cien mil habitantes, mientras que en lugares como País Vasco o Ceuta hay un menor número.

Aunque luego lo comprobaremos, se puede estudiar si hay sobredispersión viendo si el cociente entre el estadístico de desviación (647.38) y su grados de libertad (755) es mayor que uno, cosa que no se cumple.

Comprobemos que está solucionado el problema de la correlación entre las variables.

```
vif(modeloTotal_i_p)
```

```
##                                GVIF Df GVIF^(1/(2*Df))
## year                       1.658695  1        1.287903
## numero_vehiculos_milhabr   2.639121  1        1.624537
## precipitacion_media        3.829963  1        1.957029
## temperatura_media         11.956158  1        3.457768
## horas_sol_media            5.160573  1        2.271689
## km_carretera               2.087479  1        1.444811
## CCAA                     237.490453 18        1.164100
```

Obtenemos que nuestro modelo no se puede reducir más, en el sentido de disminuir el número de variables del estudio, mediante la función:

```
modeloTotal_i_p_reducido=step(modeloTotal_i_p)
```

```
## Start:  AIC=3508.49
## fallecidos_int_milhabr ~ year + numero_vehiculos_milhabr + precipitacion_media +
##     temperatura_media + horas_sol_media + km_carretera + CCAA
##
##                            Df Deviance    AIC
## <none>                          647.38 3508.5
## - precipitacion_media       1   649.46 3508.6
## - km_carretera              1   651.02 3510.1
## - horas_sol_media           1   672.65 3531.8
## - temperatura_media         1   706.48 3565.6
## - numero_vehiculos_milhabr  1   756.16 3615.3
## - CCAA                     18  1194.12 4019.2
## - year                      1  1968.42 4827.5
```

Se muestran 2 gráficas relacionadas con los residuos y observaciones.

```
plot(modeloTotal_i_p,which=c(1,4))
```

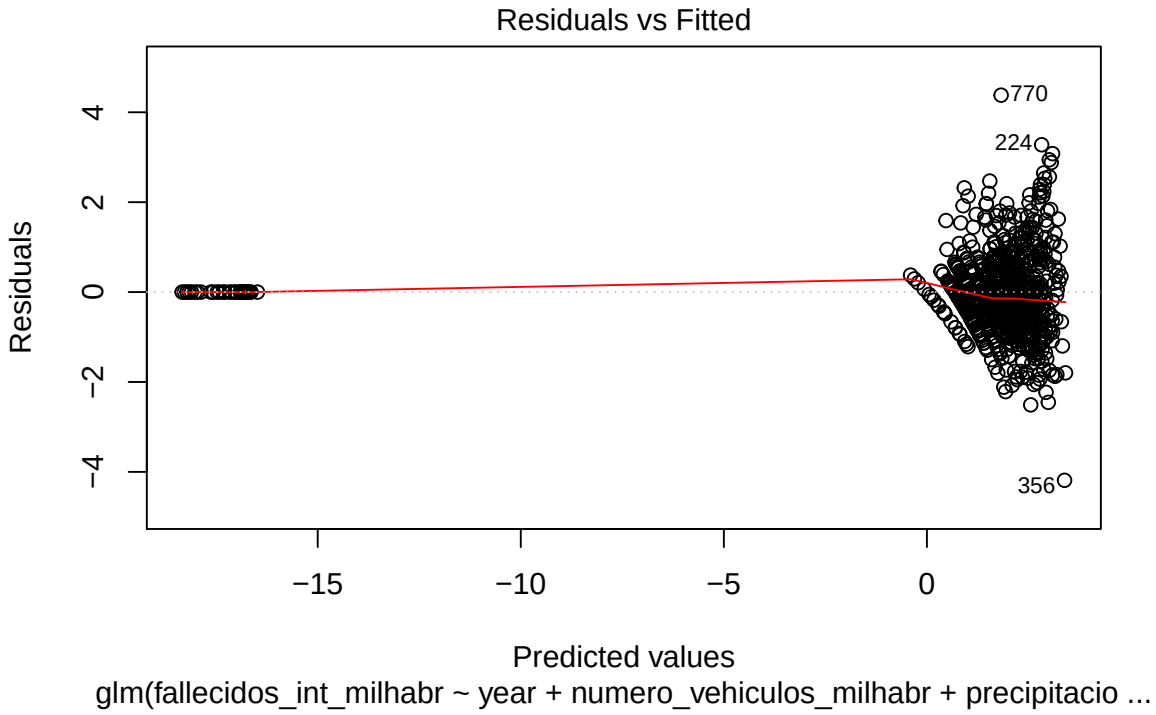

Figura 7.9: Residuos y observaciones influyentes para fallecidos en vías interurbanas con Poisson

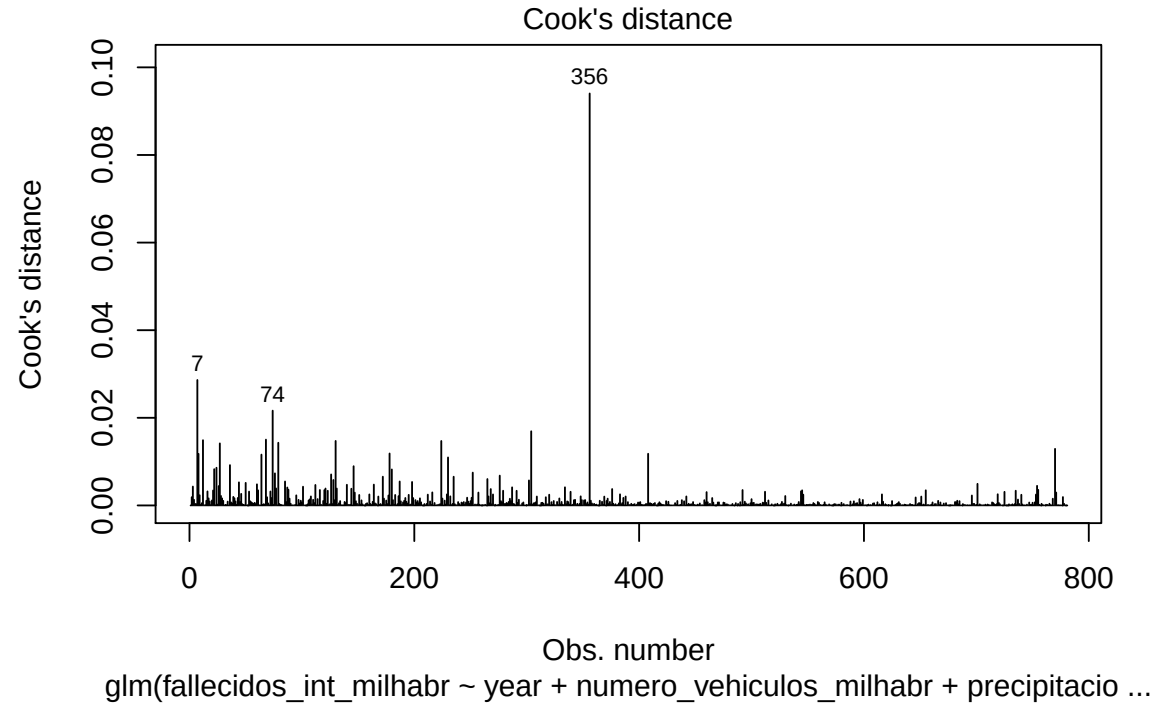

Figura 7.10: Residuos y observaciones influyentes para fallecidos en vías interurbanas con Poisson

En la primera gráfica podemos ver como nuestro modelo aproxima bien a los valores pequeños, pero una vez que vamos alejándonos, como la varianza es mayor, los datos aparecen mas dispersos, y de ahí un mayor residuo, lo cual es coherente con el modelo

de Poisson. En la segunda podemos ver que hay 3 observaciones influyentes en nuestro modelo, pero sobre todo la 356, observación que corresponde a Teruel en el año 2008 con 14 fallecidos en vías interurbanas.

Comprobamos si se cumple la hipótesis de la equidispersión, viendo así que no hay sobredispersión de los datos mediante la función:

```
library(AER)
dispersiontest(modeloTotal_i_p,trafo=1)
```

```
##
##  Overdispersion test
##
## data:  modeloTotal_i_p
## z = -1.6639, p-value = 0.9519
## alternative hypothesis: true alpha is greater than 0
## sample estimates:
##     alpha
## -0.103754
```

Nos da un p-valor cercano a uno, por lo que se acepta la hipótesis de la equidispersión, y por tanto aceptamos el estudio mediante el modelo de Poisson.

7.2.3.2. Vías urbanas

```
modeloTotal_u_p<-glm(fallecidos_urb_milhabr~year+
                numero_vehiculos_milhabr+precipitacion_media+
                temperatura_media+horas_sol_media+
                km_carretera+CCAA,family="poisson",data=datos_f)
```

```
summary(modeloTotal_u_p)
```

```
##
## Call:
## glm(formula = fallecidos_urb_milhabr ~ year + numero_vehiculos_milhabr +
##     precipitacion_media + temperatura_media + horas_sol_media +
##     km_carretera + CCAA, family = "poisson", data = datos_f)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.24278  -0.41637  -0.01771   0.32226   2.57912
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 1.520e+02  1.700e+01   8.939  < 2e-16 ***
## year                       -7.678e-02  8.643e-03  -8.884  < 2e-16 ***
## numero_vehiculos_milhabr    1.189e-05  6.083e-06   1.954 0.050705 .
## precipitacion_media        -1.265e-04  2.839e-03  -0.045 0.964470
## temperatura_media           5.845e-02  3.908e-02   1.496 0.134707
## horas_sol_media             9.457e-04  2.024e-03   0.467 0.640332
```

```
## km_carretera              1.132e-04  4.006e-05   2.826 0.004710 **
## CCAAAragón                3.920e-01  1.858e-01   2.110 0.034894 *
## CCAAAsturias              4.126e-01  3.521e-01   1.172 0.241276
## CCAABaleares              3.975e-01  2.525e-01   1.574 0.115384
## CCAACanarias             -1.995e-01  2.894e-01  -0.689 0.490538
## CCAACantabria             3.676e-01  3.979e-01   0.924 0.355537
## CCAACataluña              6.449e-01  1.548e-01   4.166 3.10e-05 ***
## CCAACeuta                 3.359e-01  2.774e-01   1.211 0.225907
## CCAACLeón                 9.005e-01  2.521e-01   3.572 0.000355 ***
## CCAACLMancha              1.506e-01  1.760e-01   0.855 0.392291
## CCAACValenciana           1.885e-01  1.605e-01   1.175 0.239980
## CCAAExtremadura          -3.054e-02  2.010e-01  -0.152 0.879215
## CCAAGalicia               3.573e-01  2.669e-01   1.338 0.180746
## CCAAMadrid                2.217e-01  2.568e-01   0.863 0.387953
## CCAAMelilla               8.206e-01  2.377e-01   3.452 0.000556 ***
## CCAAMurcia               -2.309e-02  2.518e-01  -0.092 0.926939
## CCAANavarra               2.352e-01  3.230e-01   0.728 0.466414
## CCAAPVasco                5.261e-01  3.530e-01   1.490 0.136177
## CCAARioja                 1.063e+00  2.555e-01   4.161 3.17e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 587.86  on 779  degrees of freedom
## Residual deviance: 397.59  on 755  degrees of freedom
## AIC: 2101.8
##
## Number of Fisher Scoring iterations: 5
```

Como podemos observar, un aumento de la variable km_carretera provoca un aumento del número esperado de fallecidos en vías urbanas por cien mil habitantes; mientras que ocurre lo contrario con las variable year, hecho que se puede explicar gracias a que con los años las personas se van concienciando más, hay más avances. Es de destacar un dato bastante relevante y es que Andalucía tiene un número esperado de fallecidos en vías urbana por cien mil habitantes menor que prácticamente todas las comunidades autónomas, excepto en Extremadura, Canarias y Murcia, y estas no son influyentes en nuestro modelo.

Comprobamos que no hay gran correlación entre las variables:

```
vif(modeloTotal_u_p)
```

```
##                               GVIF Df GVIF^(1/(2*Df))
## year                      1.554399  1        1.246755
## numero_vehiculos_milhabr  2.715358  1        1.647834
## precipitacion_media       4.338545  1        2.082917
## temperatura_media        15.222225  1        3.901567
## horas_sol_media           5.578642  1        2.361915
## km_carretera              3.194961  1        1.787445
```

```
## CCAA                       567.294223 18        1.192600
```

Mostramos que nuestro modelo se puede reducir más, en el sentido de disminuir el número de variables del estudio, mediante la función:

```
modeloTotal_u_p_reducido=step(modeloTotal_u_p)
```

```
## Start:  AIC=2101.8
## fallecidos_urb_milhabr ~ year + numero_vehiculos_milhabr + precipitacion_media +
##     temperatura_media + horas_sol_media + km_carretera + CCAA
##
##                            Df Deviance    AIC
## - precipitacion_media       1   397.59 2099.8
## - horas_sol_media           1   397.80 2100.0
## <none>                          397.59 2101.8
## - temperatura_media         1   399.83 2102.1
## - numero_vehiculos_milhabr  1   401.39 2103.6
## - km_carretera              1   405.37 2107.6
## - CCAA                     18   457.42 2125.6
## - year                      1   479.78 2182.0
##
## Step:  AIC=2099.81
## fallecidos_urb_milhabr ~ year + numero_vehiculos_milhabr + temperatura_media +
##     horas_sol_media + km_carretera + CCAA
##
##                            Df Deviance    AIC
## - horas_sol_media           1   397.81 2098.0
## <none>                          397.59 2099.8
## - temperatura_media         1   399.87 2100.1
## - numero_vehiculos_milhabr  1   401.39 2101.6
## - km_carretera              1   405.37 2105.6
## - CCAA                     18   461.23 2127.4
## - year                      1   479.79 2180.0
##
## Step:  AIC=2098.03
## fallecidos_urb_milhabr ~ year + numero_vehiculos_milhabr + temperatura_media +
##     km_carretera + CCAA
##
##                            Df Deviance    AIC
## <none>                          397.81 2098.0
## - temperatura_media         1   400.82 2099.0
## - numero_vehiculos_milhabr  1   401.81 2100.0
## - km_carretera              1   405.48 2103.7
## - CCAA                     18   465.28 2129.5
## - year                      1   480.72 2178.9
```

Reduzcamos nuestro modelo:

```
modeloTotal_u_p_reducido<-glm(fallecidos_urb_milhabr~year+
                      numero_vehiculos_milhabr+temperatura_media+
                      km_carretera+CCAA,family="poisson",data=datos_f)
```

```r
summary(modeloTotal_u_p_reducido)
```

```
##
## Call:
## glm(formula = fallecidos_urb_milhabr ~ year + numero_vehiculos_milhabr +
##     temperatura_media + km_carretera + CCAA, family = "poisson",
##     data = datos_f)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.26563  -0.41525  -0.01642   0.32266   2.58170
##
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)                1.525e+02  1.696e+01   8.993  < 2e-16 ***
## year                      -7.699e-02  8.632e-03  -8.920  < 2e-16 ***
## numero_vehiculos_milhabr   1.214e-05  6.060e-06   2.003 0.045190 *
## temperatura_media          6.375e-02  3.688e-02   1.729 0.083842 .
## km_carretera               1.124e-04  4.007e-05   2.804 0.005040 **
## CCAAAragón                 3.952e-01  1.858e-01   2.128 0.033362 *
## CCAAAsturias               3.405e-01  3.003e-01   1.134 0.256804
## CCAABaleares               3.667e-01  2.427e-01   1.511 0.130779
## CCAACanarias              -2.531e-01  2.418e-01  -1.047 0.295301
## CCAACantabria              2.534e-01  2.918e-01   0.868 0.385327
## CCAACataluña               6.138e-01  1.404e-01   4.371 1.23e-05 ***
## CCAACeuta                  3.052e-01  2.692e-01   1.134 0.256909
## CCAACLeón                  9.017e-01  2.517e-01   3.582 0.000341 ***
## CCAACLMancha               1.523e-01  1.763e-01   0.864 0.387826
## CCAACValenciana            1.819e-01  1.547e-01   1.176 0.239637
## CCAAExtremadura           -2.365e-02  2.002e-01  -0.118 0.905953
## CCAAGalicia                3.018e-01  2.139e-01   1.411 0.158296
## CCAAMadrid                 2.182e-01  2.562e-01   0.852 0.394442
## CCAAMelilla                7.959e-01  2.295e-01   3.468 0.000525 ***
## CCAAMurcia                -1.856e-02  2.479e-01  -0.075 0.940319
## CCAANavarra                1.882e-01  2.972e-01   0.633 0.526501
## CCAAPVasco                 4.434e-01  2.754e-01   1.610 0.107376
## CCAARioja                  1.039e+00  2.501e-01   4.155 3.25e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 587.86  on 779  degrees of freedom
## Residual deviance: 397.81  on 757  degrees of freedom
## AIC: 2098
##
## Number of Fisher Scoring iterations: 5
```

Observamos que la interpretación coincide con la del modelo sin reducir, aunque con coeficientes distintos lógicamente. Ahora sí, prácticamente todas las variables son influyentes mientras que antes solo lo eran 2.

Aunque luego lo comprobaremos, podemos estudiar si hay sobredispersión viendo si el cociente entre el estadístico de desviación (397.59) y su grados de libertad (755) es mayor que uno, cosa que no se cumple.

Dibujamos 2 gráficas relativas a los residuos y observaciones.

```
plot(modeloTotal_u_p_reducido,which=c(1,4))
```

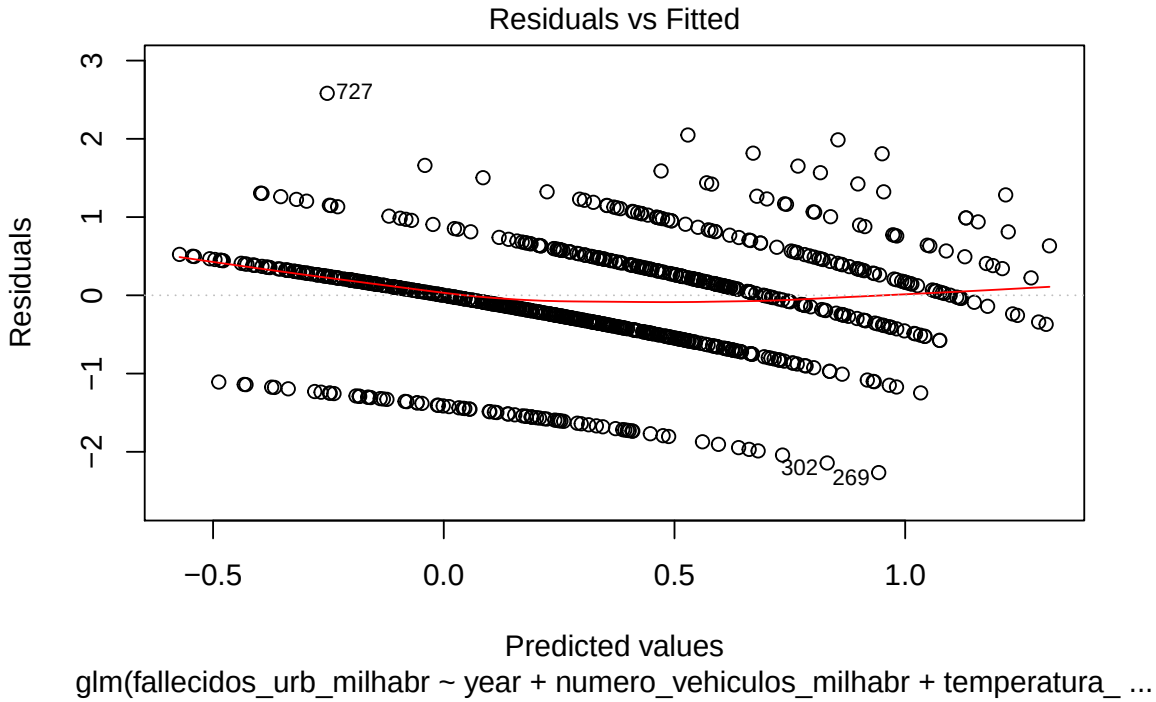

Figura 7.11: Residuos y observaciones influyentes para fallecidos en vías urbanas con Poisson

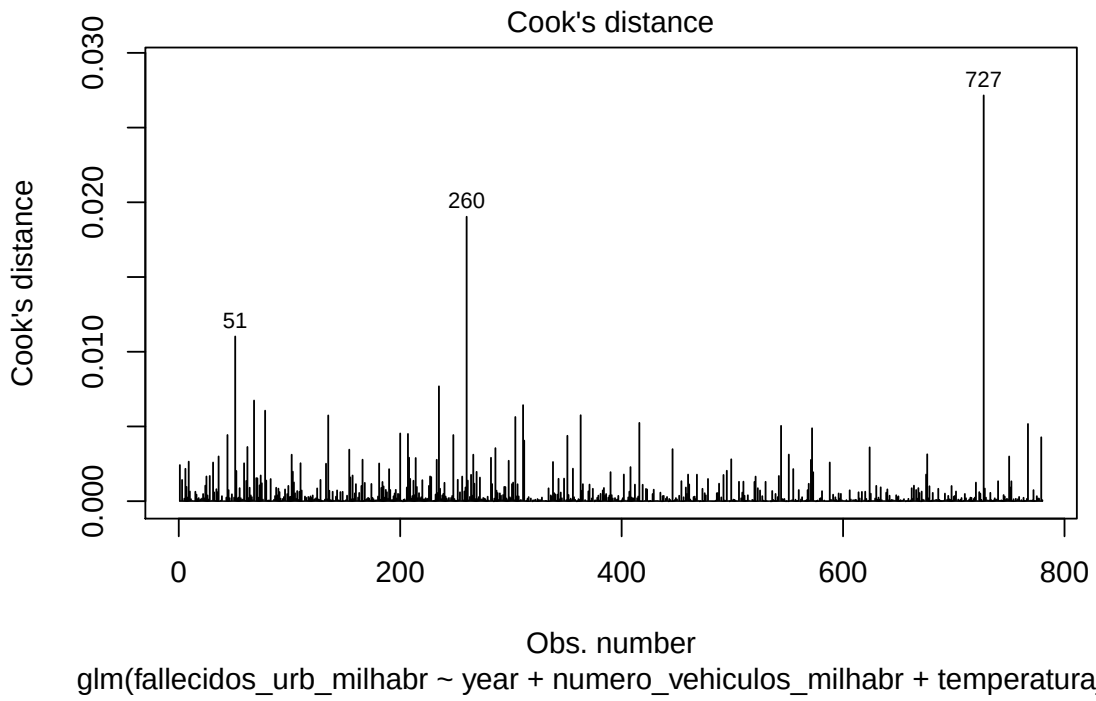

Figura 7.12: Residuos y observaciones influyentes para fallecidos en vías urbanas con Poisson

En la primera gráfica podemos ver como nuestro modelo aproxima bien a valores entre 0 y 1 fallecidos, pero no ocurre lo mismo una vez que aumenta este número, puesto que como la varianza es mayor, los datos aparecen más dispersos, y de ahí un mayor residuo, lo cual es coherente con el modelo de Poisson. En la segunda podemos ver que hay 3 observaciones influyentes en nuestro modelo, pero sobre todo la 727, observación que corresponde a Ceuta en el año 2015 con 3 fallecidos en vías urbanas.

Con la siguiente función vemos si se cumple la hipótesis de la equidispersión, comprobando así también que no hay sobredispersión de los datos mediante la función:

```
dispersiontest(modeloTotal_u_p_reducido,trafo=1)
```

```
##
##  Overdispersion test
##
## data:  modeloTotal_u_p_reducido
## z = -19.116, p-value = 1
## alternative hypothesis: true alpha is greater than 0
## sample estimates:
##      alpha
## -0.5706904
```

Nos da un p-valor de uno, por lo que se acepta la hipótesis de la equidispersión, y por tanto aceptamos el estudio mediante el modelo de Poisson.

7.2.4. Aplicación del modelo Binomial Negativa

Comenzamos el estudio con el modelo Binomial Negativo como miembro de la familia del MLG. Hemos ultilizado las siguientes variables explicativas: year, numero_vehiculos_milhabr, precipitacion_media, temperatura_media, horas_sol_media, km_carretera, CCAA.

7.2.4.1. Vías interurbanas

```
library(MASS)
modeloTotal_i_bn<-glm.nb(fallecidos_int_milhabr~year+
                         numero_vehiculos_milhabr+precipitacion_media+
                         temperatura_media+horas_sol_media+
                         km_carretera+CCAA,data=datos_f)
```

```
summary(modeloTotal_i_bn)
```

```
##
## Call:
## glm.nb(formula = fallecidos_int_milhabr ~ year + numero_vehiculos_milhabr +
##     precipitacion_media + temperatura_media + horas_sol_media +
##     km_carretera + CCAA, data = datos_f, init.theta = 87.47198285,
##     link = log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.8288  -0.5985  -0.0789   0.4125   4.1309
##
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)               2.849e+02  8.437e+00  33.766  < 2e-16 ***
## year                     -1.412e-01  4.285e-03 -32.947  < 2e-16 ***
## numero_vehiculos_milhabr  2.645e-05  2.665e-06   9.924  < 2e-16 ***
## precipitacion_media      -1.826e-03  1.264e-03  -1.445  0.14835
## temperatura_media        -1.249e-01  1.706e-02  -7.321 2.47e-13 ***
## horas_sol_media           4.464e-03  9.276e-04   4.812 1.49e-06 ***
## km_carretera             -3.272e-05  1.797e-05  -1.821  0.06865 .
## CCAAAragón                2.028e-01  8.055e-02   2.517  0.01182 *
## CCAAAsturias             -3.285e-02  1.656e-01  -0.198  0.84276
## CCAABaleares             -4.211e-01  1.280e-01  -3.290  0.00100 **
## CCAACanarias             -9.556e-02  1.403e-01  -0.681  0.49582
## CCAACantabria            -7.604e-03  1.820e-01  -0.042  0.96667
## CCAACataluña              8.012e-02  7.139e-02   1.122  0.26178
## CCAACeuta                -3.515e+01  2.475e+06   0.000  0.99999
## CCAACLeón                 5.080e-03  1.065e-01   0.048  0.96197
## CCAACLMancha              2.010e-01  7.216e-02   2.785  0.00536 **
## CCAACValenciana          -7.815e-02  7.672e-02  -1.019  0.30836
## CCAAExtremadura           1.089e-01  8.513e-02   1.280  0.20063
```

```
## CCAAGalicia                2.131e-01  1.133e-01   1.882  0.05986 .
## CCAAMadrid                -1.486e+00  1.900e-01  -7.823 5.14e-15 ***
## CCAAMelilla               -3.512e+01  2.517e+06   0.000  0.99999
## CCAAMurcia                 9.972e-03  1.203e-01   0.083  0.93392
## CCAANavarra               -1.116e-01  1.364e-01  -0.819  0.41303
## CCAAPVasco                -3.451e-01  1.499e-01  -2.303  0.02130 *
## CCAARioja                  2.217e-01  1.177e-01   1.883  0.05975 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(87.472) family taken to be 1)
##
##     Null deviance: 3426.70  on 779  degrees of freedom
## Residual deviance:  571.68  on 755  degrees of freedom
## AIC: 3503.6
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  87.5
##           Std. Err.:  37.2
##
##  2 x log-likelihood:  -3451.636
```

Los coeficientes de este modelo son practicamente iguales que los del modelo de Poisson excepto los referidos a las comunidades autónomas, de los cuáles las más significativas son Baleares, Madrid y País Vasco con menos fallecidos por cien mil habitantes y Aragón y Castilla La Mancha con más (todos respecto a Andalucía).

Comprobamos que no hay problemas destacables de las correlaciones entre variables.

```
vif(modeloTotal_i_bn)
```

```
##                               GVIF Df GVIF^(1/(2*Df))
## year                      1.650018  1        1.284530
## numero_vehiculos_milhabr  2.628555  1        1.621282
## precipitacion_media       3.846280  1        1.961194
## temperatura_media        12.101229  1        3.478682
## horas_sol_media           5.309612  1        2.304259
## km_carretera              2.129291  1        1.459209
## CCAA                    250.812237 18        1.165866
```

Comprobemos que nuestro modelo no se puede reducir más, en el sentido de disminuir el número de variables del estudio, mediante la función:

```
modeloTotal_i_bn_reducido=step(modeloTotal_i_bn)
```

```
## Start:  AIC=3501.64
## fallecidos_int_milhabr ~ year + numero_vehiculos_milhabr + precipitacion_media +
##     temperatura_media + horas_sol_media + km_carretera + CCAA
##
##                            Df Deviance    AIC
```

```
## <none>                           571.68 3501.6
## - precipitacion_media       1   573.78 3501.7
## - km_carretera              1   574.99 3502.9
## - horas_sol_media           1   594.90 3522.8
## - temperatura_media         1   625.26 3553.2
## - numero_vehiculos_milhabr  1   668.08 3596.0
## - CCAA                     18  1079.63 3973.6
## - year                      1  1762.78 4690.7
```

Por último, se compara el modelo de Poisson y el Binomial Negativo. Para ello se usó la siguiente prueba de razón de verosimilitud:

```
tt=2*(logLik(modeloTotal_i_bn)-logLik(modeloTotal_i_p))
```

```
pchisq(tt,df=1,lower.tail=FALSE)
```

```
## 'log Lik.' 0.008846886 (df=26)
```

El hecho que salga un valor del χ^2 tan próximo a cero nos indica que es más apropiado utilizar para nuestro estudio el modelo Binomial Negativo que el modelo de Poisson con el mismo número de variables.

7.2.4.2. Vías urbanas

```
library(MASS)
modeloTotal_u_bn<-glm.nb(fallecidos_urb_milhabr~year+
                      numero_vehiculos_milhabr+precipitacion_media+
                      temperatura_media+horas_sol_media+
                      km_carretera+CCAA,data=datos_f)
```

```
summary(modeloTotal_u_bn)
```

```
##
## Call:
## glm.nb(formula = fallecidos_urb_milhabr ~ year + numero_vehiculos_milhabr +
##     precipitacion_media + temperatura_media + horas_sol_media +
##     km_carretera + CCAA, data = datos_f, init.theta = 55483.63471,
##     link = log)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.24275  -0.41636  -0.01771   0.32226   2.57908
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 1.520e+02  1.700e+01   8.939  < 2e-16 ***
## year                       -7.678e-02  8.643e-03  -8.884  < 2e-16 ***
## numero_vehiculos_milhabr    1.189e-05  6.083e-06   1.954 0.050709 .
## precipitacion_media        -1.264e-04  2.839e-03  -0.045 0.964478
## temperatura_media           5.845e-02  3.908e-02   1.496 0.134712
```

```
## horas_sol_media           9.457e-04  2.024e-03   0.467 0.640348
## km_carretera              1.132e-04  4.006e-05   2.826 0.004711 **
## CCAAAragón                3.920e-01  1.858e-01   2.110 0.034897 *
## CCAAAsturias              4.126e-01  3.521e-01   1.172 0.241285
## CCAABaleares              3.975e-01  2.525e-01   1.574 0.115393
## CCAACanarias             -1.995e-01  2.894e-01  -0.689 0.490542
## CCAACantabria             3.676e-01  3.979e-01   0.924 0.355549
## CCAACataluña              6.449e-01  1.548e-01   4.166 3.10e-05 ***
## CCAACeuta                 3.359e-01  2.774e-01   1.211 0.225917
## CCAACLeón                 9.005e-01  2.521e-01   3.572 0.000355 ***
## CCAACLMancha              1.506e-01  1.760e-01   0.855 0.392295
## CCAACValenciana           1.885e-01  1.605e-01   1.175 0.239985
## CCAAExtremadura          -3.054e-02  2.010e-01  -0.152 0.879218
## CCAAGalicia               3.573e-01  2.669e-01   1.338 0.180755
## CCAAMadrid                2.217e-01  2.568e-01   0.863 0.387960
## CCAAMelilla               8.206e-01  2.377e-01   3.452 0.000556 ***
## CCAAMurcia               -2.309e-02  2.518e-01  -0.092 0.926941
## CCAANavarra               2.352e-01  3.230e-01   0.728 0.466420
## CCAAPVasco                5.261e-01  3.531e-01   1.490 0.136186
## CCAARioja                 1.063e+00  2.555e-01   4.161 3.17e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(55483.63) family taken to be 1)
##
##     Null deviance: 587.85  on 779  degrees of freedom
## Residual deviance: 397.58  on 755  degrees of freedom
## AIC: 2103.8
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  55484
##           Std. Err.:  276884
## Warning while fitting theta: iteration limit reached
##
##  2 x log-likelihood:  -2051.817
```

```
library(pscl)
vuong(modeloTotal_u_p_reducido,modeloTotal_u_bn)
```

```
## Vuong Non-Nested Hypothesis Test-Statistic:
## (test-statistic is asymptotically distributed N(0,1) under the
##  null that the models are indistinguishible)
## -------------------------------------------------------------
##               Vuong z-statistic             H_A    p-value
## Raw                  -0.3406407 model2 > model1    0.36669
## AIC-corrected         5.9585734 model1 > model2 1.2722e-09
## BIC-corrected        20.6335184 model1 > model2 < 2.22e-16
```

Concluimos que el modelo de Poisson y el Binomial Negativo son equivalentes (porque el p-valor es muy pequeño tanto en el estadístico AIC como el BIC), por eso hemos descartado este estudio para nuestro modelo.

7.2.5. Aplicación del modelo de Poisson Bivariante

Procedemos a aplicar el modelo de Poisson Bivariante a través de la librería bivpois, que actualmente no se encuentra disponible en el software R pero se adjunta el script con las funciones en el anexo. Para ello hacemos uso de la función lm.bp, cuyos argumentos más importantes son:

$$lm.bp(l1, l2, l1l2 = NULL, l3 = 1, data, common.intercept = FALSE, zeroL3 = FALSE)$$

donde:

1. l1: fórmula de la forma $x \sim X_1 + ... + X_p$ para los parámetros del $log\lambda_1$.
2. l2: fórmula de la forma $y \sim X_1 + ... + X_p$ para los parámetros del $log\lambda_2$.
3. l1l2: fórmula de la forma $\sim X_1 + ... + X_p$ para los parámetros comunes del $log\lambda_1$ y $log\lambda_2$. Si la variable explicativa se encuentra también en l1 y/o l2 entonces el modelo ajusta la interacción entre los parámetros. Se pueden usar aquí términos especiales de la forma $c(X_1, X_2)$. Estos términos implican parámetros comunes de λ_1 y λ_2 para distintas variables.
4. l3: fórmula de la forma $\sim X_1 + ... + X_p$ para los parámetros del $log\lambda_3$.
5. data: Base de datos (data frame) que contiene las variables en el modelo.
6. common.intercept: función lógica que específica si se debe usar un término independiente común entre λ_1 y $lambda_2$. Por defecto su valor es FALSE.
7. zeroL3: argumento lógico que controla si λ_3 debería ser igual a cero (y por lo tanto se ajustaría el modelo de Poisson doble).

En nuestro caso consideraremos x=fallecidos_int_milhabr, y=fallecidos_urb_milhabr. Nuestros datos vienen dados del siguiente data frame:

```
cc=cbind(cbind(datos_f$year,datos_f[,7:10]),datos_f[,15:17])
head(cc)
```

```
##   datos_f$year precipitacion_media temperatura_media horas_sol_media
## 1         2002               44.375            11.808         169.133
## 2         2002               33.150            14.842         261.133
## 3         2002               20.933            18.342         265.158
## 4         2002               17.058            18.583         268.508
## 5         2002               28.358            11.258         226.983
## 6         2002               26.525            16.917         250.375
##   km_carretera fallecidos_int_milhabr fallecidos_urb_milhabr
## 1      1463.33                     14                     3
## 2      3710.45                     17                     2
## 3      2704.29                     10                     1
## 4      2368.19                     19                     2
## 5      2545.57                     28                     2
```

```
## 6      4882.83                    11                        3
##   numero_vehiculos_milhabr
## 1                    60381
## 2                    56728
## 3                    66915
## 4                    65967
## 5                    57290
## 6                    50716
```

A continuación se muestran 3 modelos diferentes: uno de Poisson doble y dos de Poisson Bivariante (uno con λ_3 constante).

```
source('bivpois.R')
```

```
ModeloBivariante1<-lm.bp(fallecidos_int_milhabr~.,
                         fallecidos_urb_milhabr~.,data=cc)
ModeloBivariante2<-lm.bp(fallecidos_int_milhabr~.,
                         fallecidos_urb_milhabr~.,l3=~.,data=cc)
ModeloBivariante3<-lm.bp(fallecidos_int_milhabr~.,
                         fallecidos_urb_milhabr~.,data=cc,zeroL3 = TRUE)
```

De cada modelo podemos sacar objetos como el AIC, BIC, coefficients, parameters o loglikelihood. Notemos que el mejor modelo será el que tenga menor AIC o BIC.

En el cuadro (7.1) se recogen los 3 modelos de Poisson Bivariante estudiados.

Cuadro 7.1: Modelos de Poisson Bivariante

Modelo	Detalles	Parámetros	AIC	BIC
Poisson doble	$\lambda_3 = 0$	14	6144.866	6219.80
Poisson Bivariante	λ_3 constante	15	6145.411	6225.697
Poisson Bivariante		21	6129.997	6242.398

Teoricamente, el modelo de Poisson doble debería de tener los mismos coeficientes que los modelos de Poisson de fallecidos_int_milhabr y fallecidos_urb_milhabr por separado. A continuación se muestra dicha afirmación.

```
ModeloBivariante3$coefficients
```

```
##              (l1):(Intercept)                (l1):datos_f$year
##                   2.833905e+02                    -1.405777e-01
##           (l1):horas_sol_media                (l1):km_carretera
##                   4.891743e-03                     8.104556e-05
## (l1):numero_vehiculos_milhabr      (l1):precipitacion_media
##                   2.543470e-05                    -1.918890e-03
##         (l1):temperatura_media                (l2):(Intercept)
##                  -1.333294e-01                     1.569376e+02
##              (l2):datos_f$year           (l2):horas_sol_media
##                  -7.800347e-02                     3.515869e-04
##              (l2):km_carretera (l2):numero_vehiculos_milhabr
##                   9.535800e-06                     1.308414e-05
```

```
##      (l2):precipitacion_media        (l2):temperatura_media
##                  -3.557485e-03                  -4.532604e-02
```

```
modeloTotal_i_p22<-glm(fallecidos_int_milhabr~year+
                         numero_vehiculos_milhabr+precipitacion_media+
                       temperatura_media+horas_sol_media+km_carretera,
                       family="poisson",data=datos_f)
```

```
modeloTotal_i_p22$coefficients
```

```
##              (Intercept)                      year numero_vehiculos_milhabr
##             2.833905e+02             -1.405777e-01              2.543470e-05
##      precipitacion_media         temperatura_media           horas_sol_media
##             -1.918890e-03             -1.333294e-01              4.891743e-03
##              km_carretera
##              8.104556e-05
```

```
modeloTotal_u_p22<-glm(fallecidos_urb_milhabr~year+
                         numero_vehiculos_milhabr+precipitacion_media+
                       temperatura_media+horas_sol_media+km_carretera,
                       family="poisson",data=datos_f)
```

```
modeloTotal_u_p22$coefficients
```

```
##              (Intercept)                      year numero_vehiculos_milhabr
##             1.569376e+02             -7.800347e-02              1.308414e-05
##      precipitacion_media         temperatura_media           horas_sol_media
##             -3.557485e-03             -4.532604e-02              3.515869e-04
##              km_carretera
##              9.535800e-06
```

7.2.6. Comparación y selección de modelos

7.2.6.1. Modelos univariantes

De los dos modelos estudiados para cada vía (poisson y binomial negativa), se analizó en la aplicación del modelo binomial negativo cada uno de los dos modelos en cada vía con el fin de ver cuál era el más adecuado para aplicarlo a nuestro estudio. Recogemos aquí los cálculos.

7.2.6.1.1. Vías interurbanas

```
tt=2*(logLik(modeloTotal_i_bn)-logLik(modeloTotal_i_p))
```

```
pchisq(tt,df=1,lower.tail=FALSE)
```

```
## 'log Lik.' 0.008846886 (df=26)
```

Se obtuvo que es más apropiado utilizar para nuestro estudio el *modelo Binomial Negativo* que el modelo de Poisson con el mismo número de variables.

```r
summary(modeloTotal_i_bn)
```

```
##
## Call:
## glm.nb(formula = fallecidos_int_milhabr ~ year + numero_vehiculos_milhabr +
##     precipitacion_media + temperatura_media + horas_sol_media +
##     km_carretera + CCAA, data = datos_f, init.theta = 87.47198285,
##     link = log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.8288  -0.5985  -0.0789   0.4125   4.1309
##
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)                2.849e+02  8.437e+00  33.766  < 2e-16 ***
## year                      -1.412e-01  4.285e-03 -32.947  < 2e-16 ***
## numero_vehiculos_milhabr   2.645e-05  2.665e-06   9.924  < 2e-16 ***
## precipitacion_media       -1.826e-03  1.264e-03  -1.445  0.14835
## temperatura_media         -1.249e-01  1.706e-02  -7.321 2.47e-13 ***
## horas_sol_media            4.464e-03  9.276e-04   4.812 1.49e-06 ***
## km_carretera              -3.272e-05  1.797e-05  -1.821  0.06865 .
## CCAAAragón                 2.028e-01  8.055e-02   2.517  0.01182 *
## CCAAAsturias              -3.285e-02  1.656e-01  -0.198  0.84276
## CCAABaleares              -4.211e-01  1.280e-01  -3.290  0.00100 **
## CCAACanarias              -9.556e-02  1.403e-01  -0.681  0.49582
## CCAACantabria             -7.604e-03  1.820e-01  -0.042  0.96667
## CCAACataluña               8.012e-02  7.139e-02   1.122  0.26178
## CCAACeuta                 -3.515e+01  2.475e+06   0.000  0.99999
## CCAACLeón                  5.080e-03  1.065e-01   0.048  0.96197
## CCAACLMancha               2.010e-01  7.216e-02   2.785  0.00536 **
## CCAACValenciana           -7.815e-02  7.672e-02  -1.019  0.30836
## CCAAExtremadura            1.089e-01  8.513e-02   1.280  0.20063
## CCAAGalicia                2.131e-01  1.133e-01   1.882  0.05986 .
## CCAAMadrid                -1.486e+00  1.900e-01  -7.823 5.14e-15 ***
## CCAAMelilla               -3.512e+01  2.517e+06   0.000  0.99999
## CCAAMurcia                 9.972e-03  1.203e-01   0.083  0.93392
## CCAANavarra               -1.116e-01  1.364e-01  -0.819  0.41303
## CCAAPVasco                -3.451e-01  1.499e-01  -2.303  0.02130 *
## CCAARioja                  2.217e-01  1.177e-01   1.883  0.05975 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(87.472) family taken to be 1)
##
##     Null deviance: 3426.70  on 779  degrees of freedom
## Residual deviance:  571.68  on 755  degrees of freedom
## AIC: 3503.6
```

```
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  87.5
##           Std. Err.:  37.2
##
##  2 x log-likelihood:  -3451.636
```

7.2.6.1.2. Vías urbanas

```
library(pscl)
vuong(modeloTotal_u_p_reducido,modeloTotal_u_bn)
```

```
## Vuong Non-Nested Hypothesis Test-Statistic:
## (test-statistic is asymptotically distributed N(0,1) under the
##  null that the models are indistinguishible)
## -------------------------------------------------------------
##               Vuong z-statistic             H_A    p-value
## Raw                  -0.3406407 model2 > model1    0.36669
## AIC-corrected         5.9585734 model1 > model2 1.2722e-09
## BIC-corrected        20.6335184 model1 > model2 < 2.22e-16
```

Se obtuvo que el modelo de Poisson y el Binomial Negativo eran equivalentes. Nos quedamos con el *modelo de Poisson* para el estudio de los fallecidos en vías urbanas.

```
summary(modeloTotal_u_p_reducido)
```

```
##
## Call:
## glm(formula = fallecidos_urb_milhabr ~ year + numero_vehiculos_milhabr +
##     temperatura_media + km_carretera + CCAA, family = "poisson",
##     data = datos_f)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.26563  -0.41525  -0.01642   0.32266   2.58170
##
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)                1.525e+02  1.696e+01   8.993  < 2e-16 ***
## year                      -7.699e-02  8.632e-03  -8.920  < 2e-16 ***
## numero_vehiculos_milhabr   1.214e-05  6.060e-06   2.003 0.045190 *
## temperatura_media          6.375e-02  3.688e-02   1.729 0.083842 .
## km_carretera               1.124e-04  4.007e-05   2.804 0.005040 **
## CCAAAragón                 3.952e-01  1.858e-01   2.128 0.033362 *
## CCAAAsturias               3.405e-01  3.003e-01   1.134 0.256804
## CCAABaleares               3.667e-01  2.427e-01   1.511 0.130779
## CCAACanarias              -2.531e-01  2.418e-01  -1.047 0.295301
## CCAACantabria              2.534e-01  2.918e-01   0.868 0.385327
```

```
## CCAACataluña              6.138e-01  1.404e-01   4.371 1.23e-05 ***
## CCAACeuta                 3.052e-01  2.692e-01   1.134 0.256909
## CCAACLeón                 9.017e-01  2.517e-01   3.582 0.000341 ***
## CCAACLMancha              1.523e-01  1.763e-01   0.864 0.387826
## CCAACValenciana           1.819e-01  1.547e-01   1.176 0.239637
## CCAAExtremadura          -2.365e-02  2.002e-01  -0.118 0.905953
## CCAAGalicia               3.018e-01  2.139e-01   1.411 0.158296
## CCAAMadrid                2.182e-01  2.562e-01   0.852 0.394442
## CCAAMelilla               7.959e-01  2.295e-01   3.468 0.000525 ***
## CCAAMurcia               -1.856e-02  2.479e-01  -0.075 0.940319
## CCAANavarra               1.882e-01  2.972e-01   0.633 0.526501
## CCAAPVasco                4.434e-01  2.754e-01   1.610 0.107376
## CCAARioja                 1.039e+00  2.501e-01   4.155 3.25e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 587.86  on 779  degrees of freedom
## Residual deviance: 397.81  on 757  degrees of freedom
## AIC: 2098
##
## Number of Fisher Scoring iterations: 5
```

7.2.6.2. Modelos bivariantes

Según el cuadro (7.1), se comprueba que tampoco varían demasiado los coeficientes AIC o BIC como para ver qué modelo es mejor, pero sí podemos ver que el modelo Bivariante es más complejo ya que nos introduce un mayor número de parámetros. Por lo tanto, seleccionamos para nuestro estudio bivariante el modelo siguiente:

```
ModeloBivariante1
```

```
##
## Call:
## lm.bp(l1 = fallecidos_int_milhabr ~ ., l2 = fallecidos_urb_milhabr ~
##     ., l1l2 = NULL, l3 = ~1, data = cc, common.intercept = FALSE,
##     zeroL3 = FALSE, maxit = 300, pres = 1e-08, verbose = FALSE)
##
## Coefficients:
##              (l1):(Intercept)              (l1):datos_f$year
##                     2.914e+02                     -1.446e-01
##          (l1):horas_sol_media              (l1):km_carretera
##                     5.017e-03                      8.239e-05
## (l1):numero_vehiculos_milhabr       (l1):precipitacion_media
##                     2.621e-05                     -1.939e-03
##        (l1):temperatura_media               (l2):(Intercept)
##                    -1.364e-01                      1.829e+02
##             (l2):datos_f$year           (l2):horas_sol_media
```

```
##                    -9.102e-02                     3.652e-04
##             (l2):km_carretera  (l2):numero_vehiculos_milhabr
##                     6.610e-06                     1.515e-05
##      (l2):precipitacion_media        (l2):temperatura_media
##                    -4.107e-03                    -5.115e-02
##              (l3):(Intercept)
##                    -1.784e+00
```

7.3. Conclusión

7.3.1. Vías interurbanas

A la vista de los resultados, un aumento de las variables year y temperatura_media provocan un decrecimiento en el número esperado de fallecidos en vías interurbanas por cien mil habitantes, hecho que se podría explicar gracias a que con el paso de los años las personas se van concienciando del peligro de la carretera, hay más avances tecnológicos y de seguridad en los coches, el aumento de las campañas y de controles realizadas por los cuerpos de seguridad y las restricciones, cada vez más bajas, de velocidad en las vías interurbanas. Sin embargo, aunque es del orden e-04, un aumento de las variables numero_vehiculos_milhabr y horas_sol_media provocan un incremento en el número esperado de fallecidos en vías interurbanas por cien mil habitantes, hecho que podría explicarse debido a que al haber más vehículos circulando y al haber más horas media de sol, las personas tienen más tiempo para conducir durante el día y es más probable que ocurra un accidente, y por tanto, fallecidos en dichas vías. Además, podemos destacar comunidades como la de Madrid o Baleares con menos fallecidos y Aragón o Castilla La Mancha con más fallecidos (con respecto a Andalucía).

7.3.2. Vías urbanas

A la vista de los resultados, un aumento de la variable year provoca un decrecimiento en el número esperado de fallecidos en vías urbanas por cien mil habitantes, hecho que se podría explicar gracias a que con el paso de los años las personas se van concienciando del peligro de la carretera, hay más avances tecnológicos en los coches y de seguridad, el aumento de las campañas realizadas por los cuerpos de seguridad y las restricciones, cada vez más bajas, de velocidad en el interior de las ciudades o en zonas de alta concentración de personas. Sin embargo, aunque el coeficiente es del orden e-04, un aumento de las variables numero_vehiculos_milhabr y km_carretera provocan un incremento en el número esperado de fallecidos en vías urbanas por cien mil habitantes, hecho lógico ya que si hay más cantidad de vehículos o mayor zona donde se pueda conducir será más probable que se produzcan accidentes, y por consiguiente, fallecidos en dichas vías.

Es de destacar un dato bastante relevante y es que Andalucía tiene un número esperado de fallecidos en vías urbana por cien mil habitantes menor que prácticamente todas las comunidades autónomas, excepto en Extremadura, Canarias y Murcia, pero éstas no son significativas.

7.3.3. Poisson Bivariante

A la vista de los resultados, prácticamente todos los coeficientes para el modelo fallecidos_int_milhabr son mayores que los del modelo fallecidos_urb_milhabr, esto quiere decir que un aumento de las variables cuyos coeficientes son positivos provocan un mayor incremento en el número esperado de fallecidos en vías interurbanas por cien mil habitantes que en vías urbanas, mientras que un aumento de las variables cuyos coeficientes son negativos provocan un mayor decrecimiento en el número esperado de fallecidos en vías urbanas por cien mil habitantes que en vías urbanas. El tercer modelo nos indica la relación entre los fallecidos en vías interurbanas y los fallecidos en vías urbanas, el cual es constante.

Apéndice A

Apéndice: Código del trabajo en R

```
library(readxl)
library(dplyr)
library(lmtest)
library(MASS)
library(sandwich)
library(AER)
library(car)
library(pscl)

#Lectura de la base de datos

load(file="datos.RData")

#Estudio previo

summary(datos_f)

mean(datos_f$fallecidos_interurbana)
var(datos_f$fallecidos_interurbana)

mean(datos_f$fallecidos_urbana)
var(datos_f$fallecidos_urbana)

q=c(0)
w=c(0)
e=c(0)
l=c(0)

for(i in (1:52))
{
q=c(q,mean(datos_f[datos_f$provincia==provincias[i],]$fallecidos_interurbana))
w=c(w,mean(datos_f[datos_f$provincia==provincias[i],]$fallecidos_urbana))
e=c(e,mean(datos_f[datos_f$provincia==provincias[i],]$numero_vehiculos))
l=c(l,mean(datos_f[datos_f$provincia==provincias[i],]$poblacion))
```

```
}

q1=c(0)
w1=c(0)

for(i in (1:52))
{
  q1=c(q1,sum(datos_f[datos_f$provincia==provincias[i],]$fallecidos_interurbana))
  w1=c(w1,sum(datos_f[datos_f$provincia==provincias[i],]$fallecidos_urbana))
}

fallecidos_interurbana_total=cbind(provincias,q1[2:53])
fallecidos_interurbana_media=cbind(provincias,q[2:53])
fallecidos_urbana_total=cbind(provincias,w1[2:53])
fallecidos_urbana_media=cbind(provincias,w[2:53])
numero_vehiculos_medio=cbind(provincias,e[2:53])
poblacion_media=cbind(provincias,l[2:53])

heatmap(cor(datos_f[,-c(1,11)]),scale="none")

#Estudio descriptivo

attach(datos_f)

## interurbana

pairs(fallecidos_interurbana~year+numero_vehiculos+poblacion+precipitacion_media+
        temperatura_media+horas_sol_media+km_carretera,panel=panel.smooth)

library(ggplot2)
ggplot( datos_f[datos_f$provincia=="Sevilla",],
aes(x=year, y=fallecidos_interurbana)) +
geom_point(size=1)+ geom_line()
ggplot( datos_f[datos_f$provincia=="Sevilla",],
aes(x=poblacion, y=fallecidos_interurbana)) +
geom_point(size=1)+geom_line()
ggplot( datos_f[datos_f$provincia=="Sevilla",],
aes(x=numero_vehiculos, y=fallecidos_interurbana)) +
geom_point(size=1)+ geom_line()

library(lattice)
xyplot(fallecidos_interurbana~year|CCAA,col=1)
xyplot(fallecidos_interurbana~poblacion|year,col=1)

## urbana

pairs(fallecidos_urbana~year+numero_vehiculos+poblacion+precipitacion_media+
    temperatura_media+horas_sol_media+km_carretera,panel=panel.smooth)
```

```
ggplot( datos_f[datos_f$provincia=="Sevilla",], aes(x=year, y=fallecidos_urbana)) +
geom_point(size=1)+ geom_line()
ggplot( datos_f[datos_f$provincia=="Sevilla",],
aes(x=poblacion, y=fallecidos_urbana)) +
geom_point(size=1)+ geom_line()
ggplot( datos_f[datos_f$provincia=="Sevilla",],
aes(x=numero_vehiculos, y=fallecidos_urbana)) +
geom_point(size=1)+ geom_line()

xyplot(fallecidos_urbana~year|CCAA,col=1)
#vemos como en algunas comunidades como catalunya destaca barcelona pero
#las demas provincias se mantienen con un numero mas bajo de fallecidos por año.
xyplot(fallecidos_urbana~poblacion|year,col=1)
#vemos como las cuidades con menor poblacion tienen un menor numero
#de fallecidos urbanos y por eso se concentran por debajo de los 35-40,
#mientras tanto las que tienen mas poblacion como barcelona y madrid
#tienen muchos fallecidos siempre mas de 50 y llegando hasta los 150.

#Estudio de los modelos

# INTERURBANA

#Modelo con todas las variables para fallecidos_interurbana con glm

modeloTotal_i_p1<-glm(fallecidos_interurbana~year+numero_vehiculos+
poblacion+precipitacion_media+temperatura_media+horas_sol_media+
km_carretera+CCAA,family="poisson",data=datos_f)

summary(modeloTotal_i_p1)
# coefconfi<-cbind(Estimate=coef(modeloTotal_i_p1), confint(modeloTotal_i_p1))
# exp(coefconfi)

#colinealidad
vif(modeloTotal_i_p1)

#Como poblacion y numero de vehiculos estan muy correladas, vamos a crear
#las siguientes nuevas variables para asi"ocultar" la variable poblacion

datos_f$fallecidos_int_milhab<-
100000*datos_f$fallecidos_interurbana/datos_f$poblacion
datos_f$fallecidos_urb_milhab<-
100000*datos_f$fallecidos_urbana/datos_f$poblacion
datos_f$numero_vehiculos_milhab<-
100000*datos_f$numero_vehiculos/datos_f$poblacion

# #Modelo con todas las variables para fallecidos_int_milhab con glm
#
```

```
# modeloTotal_i2<-glm(fallecidos_int_milhab~year+numero_vehiculos_milhab+
# precipitacion_media+temperatura_media+horas_sol_media+km_carretera+
# CCAA,data=datos_f)
#
# summary(modeloTotal_i2)
# # modeloTotal_i2$residuals
# # coefconfi2<-cbind(Estimate=coef(modeloTotal_i2), confint(modeloTotal_i2))
# # exp(coefconfi2)
# # coeftest(modeloTotal_i2,vcov=sandwich)
#
# #colinealidad
# vif(modeloTotal_i2) #Ya esta solucionado el problema que teniamos antes
# de la correlacion entre las variables
#
# #Reduccion del modelo
# modeloTotal_i2_reducido=step(modeloTotal_i2)

#Modelo con todas las variables para fallecidos_int_milhabr con poisson

datos_f$fallecidos_int_milhabr<-round(datos_f$fallecidos_int_milhab)
datos_f$fallecidos_urb_milhabr<-round(datos_f$fallecidos_urb_milhab)
datos_f$numero_vehiculos_milhabr<-round(datos_f$numero_vehiculos_milhab)

save(datos_f,file="datos2.RData")

modeloTotal_i_p<-glm(fallecidos_int_milhabr~year+numero_vehiculos_milhabr+
precipitacion_media+temperatura_media+horas_sol_media+km_carretera+CCAA,
family="poisson",data=datos_f)

modeloTotal_i_p22<-glm(fallecidos_int_milhabr~year+numero_vehiculos_milhabr+
precipitacion_media+temperatura_media+horas_sol_media+km_carretera,
family="poisson",data=datos_f)

summary(modeloTotal_i_p)
# modeloTotal_i_p$residuals
# coefconfip<-cbind(Estimate=coef(modeloTotal_i_p), confint(modeloTotal_i_p))
# exp(coefconfip)
# coeftest(modeloTotal_i_p,vcov=sandwich)

#Graficas
plot(modeloTotal_i_p,which=c(1,4))

#colinealidad
vif(modeloTotal_i_p)

#Reduccion del modelo
modeloTotal_i_p_reducido=step(modeloTotal_i_p)
```

```
#dispersion
dispersiontest(modeloTotal_i_p,trafo=1)
#Como el pvalor es casi 1, se acepta la hipotesis de que la media y la varianza
#sean iguales y por tanto aceptaríamos el estudio con poisson

#Modelo con todas las variables para fallecidos_int_milhabr con binomial negativa

modeloTotal_i_bn<-glm.nb(fallecidos_int_milhabr~year+numero_vehiculos_milhabr+
precipitacion_media+temperatura_media+horas_sol_media+km_carretera+
CCAA,data=datos_f)

summary(modeloTotal_i_bn)
# modeloTotal_i_bn$residuals
# coefconfibn<-cbind(Estimate=coef(modeloTotal_i_bn))
# coefconfibn

#colinealidad
vif(modeloTotal_i_bn)

#Reduccion de modelo
modeloTotal_i_bn_reducido=step(modeloTotal_i_bn)

#comparacion poisson-bn (se tienen que hacer con dos modelos con las misma
#variables)
tt=2*(logLik(modeloTotal_i_bn)-logLik(modeloTotal_i_p))
pchisq(tt,df=1,lower.tail=FALSE) #el hecho de que salga un numero proximo
#a cero nos dice que es mejor usar el modelo binomial negativo que el de poisson

#Como no tenemos exceso de ceros, los dos siguientes modelos no van
#a poderse realizar

#Modelo con todas las variables para fallecidos_int_milhabr con hurdle

modeloTotal_i_h<-hurdle(fallecidos_int_milhabr~numero_vehiculos_milhabr+
precipitacion_media+temperatura_media+horas_sol_media+km_carretera+year+
CCAA,data=datos_f)

#Modelo con todas las variables para fallecidos_int_milhabr con cero inflado

modeloTotal_i_z<-zeroinfl(formula=fallecidos_int_milhabr~numero_vehiculos_milhabr+
precipitacion_media+temperatura_media+horas_sol_media+km_carretera+CCAA|year,
data=datos_f)

# URBANA

#Modelo con todas las variables para fallecidos_urb_milhab con glm

modeloTotal_u<-glm(fallecidos_urb_milhab~year+numero_vehiculos_milhab+
```

```
precipitacion_media+temperatura_media+horas_sol_media+km_carretera+CCAA,
data=datos_f)

summary(modeloTotal_u)
# modeloTotal_u$residuals
# coefconfu<-cbind(Estimate=coef(modeloTotal_u), confint(modeloTotal_u))
# exp(coefconfu)

#colinealidad
vif(modeloTotal_u)

#Reduccion del modelo
modeloTotal_u_reducido=step(modeloTotal_u)

modeloTotal_u_reducido<-glm(fallecidos_urb_milhab~year+numero_vehiculos_milhab+
                     temperatura_media+km_carretera+CCAA,data=datos_f)

summary(modeloTotal_u_reducido)
# modeloTotal_u_reducido$residuals
# coefconfured<-cbind(Estimate=coef(modeloTotal_u_reducido),
# confint(modeloTotal_u_reducido))
# exp(coefconfured)

#Modelo con todas las variables para fallecidos_urb_milhabr con poisson

modeloTotal_u_p<-glm(fallecidos_urb_milhabr~year+numero_vehiculos_milhabr+
precipitacion_media+temperatura_media+horas_sol_media+km_carretera+CCAA,
family="poisson",data=datos_f)

modeloTotal_u_p22<-glm(fallecidos_urb_milhabr~year+numero_vehiculos_milhabr+
precipitacion_media+temperatura_media+horas_sol_media+km_carretera,
family="poisson",data=datos_f)

summary(modeloTotal_u_p)
# modeloTotal_u_p$residuals
# coefconfup<-cbind(Estimate=coef(modeloTotal_u_p),
# confint(modeloTotal_u_p))
# exp(coefconfup)
# coeftest(modeloTotal_u_p, vcov = sandwich)

#Graficas
plot(modeloTotal_u_p,which=c(1,4))

#colinealidad
vif(modeloTotal_u_p)

#Modelo reducido
modeloTotal_u_p_reducido=step(modeloTotal_u_p)
```

```
modeloTotal_u_p_reducido<-glm(fallecidos_urb_milhabr~year+
numero_vehiculos_milhabr+temperatura_media+km_carretera+CCAA,
family="poisson",data=datos_f)

summary(modeloTotal_u_p_reducido)
# modeloTotal_u_p_reducido$residuals
# coefconfup<-cbind(Estimate=coef(modeloTotal_u_p_reducido),
# confint(modeloTotal_u_p_reducido))
# exp(coefconfupred)
# coeftest(modeloTotal_u_p_reducido, vcov = sandwich)

#colinealidad
vif(modeloTotal_u_p_reducido)

#Graficas
plot(modeloTotal_u_p_reducido,which=c(1,4))


#dispersion
dispersiontest(modeloTotal_u_p_reducido,trafo=1)
#Como el pvalor es 1, se acepta la hipotesis de que la media
#y la varianza sean iguales por lo que se acepta el modelo de
#Poisson para el estudio de nuestro modelo

#Modelo con todas las variables para fallecidos_urb_milhabr con
#binomial negativa

modeloTotal_u_bn<-glm.nb(fallecidos_urb_milhabr~year+
numero_vehiculos_milhabr+precipitacion_media+temperatura_media+
horas_sol_media+km_carretera+CCAA,data=datos_f)

#comparacion poisson-bn
vuong(modeloTotal_u_p_reducido,modeloTotal_u_bn)

#Como no tenemos exceso de ceros, los dos siguientes modelos
#no van a poderse realizar

#Modelo con todas las variables para fallecidos_urb_milhabr con hurdle

modeloTotal_u_h<-hurdle(fallecidos_urb_milhabr~year+numero_vehiculos_milhabr+
precipitacion_media+temperatura_media+horas_sol_media+km_carretera+CCAA,
data=datos_f)

#Modelo con todas las variables para fallecidos_urb_milhabr con cero inflado

modeloTotal_u_z<-zeroinfl(fallecidos_urb_milhabr~year+numero_vehiculos_milhabr+
precipitacion_media+temperatura_media+horas_sol_media+km_carretera+CCAA,
```

```
data=datos_f,dist = "negbin")

#Modelo de Regresion Bivariante X=fallecidos_interurbana Y=fallecidos_urbana

cc=cbind(cbind(datos_f$year,datos_f[,7:10]),datos_f[,15:17])

ModeloBivariante1<-lm.bp(fallecidos_int_milhabr~.,fallecidos_urb_milhabr~.,
data=cc) #Poisson Bivariante con lambda3=cte
ModeloBivarianteBic1<-ModeloBivariante1$BIC
ModeloBivariante1
ModeloBivarianteAic1<-ModeloBivariante1$AIC
ModeloBivariante1$parameters

ModeloBivariante2<-lm.bp(fallecidos_int_milhabr~.,fallecidos_urb_milhabr~.,
l3=~.,data=cc) #Poisson Bivariante
ModeloBivarianteBic2<-ModeloBivariante2$BIC
ModeloBivariante2
ModeloBivarianteAic2<-ModeloBivariante2$AIC
ModeloBivariante2$parameters

ModeloBivariante3<-lm.bp(fallecidos_int_milhabr~.,fallecidos_urb_milhabr~.,
data=cc,zeroL3 = TRUE) #Poisson doble
ModeloBivarianteBic3<-ModeloBivariante3$BIC
ModeloBivariante3
ModeloBivarianteAic3<-ModeloBivariante3$AIC
ModeloBivariante3$parameters

#Comparacion de modelos univariantes

## Vías interurbanas (mismo número de variables)
tt=2*(logLik(modeloTotal_i_bn)-logLik(modeloTotal_i_p))
pchisq(tt,df=1,lower.tail=FALSE)

## Vías urbanas
vuong(modeloTotal_u_p_reducido,modeloTotal_u_bn)
```

Apéndice B

Apéndice: Paquete bivpois

Aquí se encuentra recogido el código con todas las funciones del paquete bivpois que actualmente no está disponible dentro del software R.

```
"lm.bp" <-
  function( l1, l2, l1l2=NULL, l3=~1, data, common.intercept=FALSE, zeroL3=FALSE,
  maxit=300, pres=1e-8, verbose=getOption('verbose') )
    #
    #
  {
    options(warn=-1)
    #
    # definition of function call
    templist<-list( l1=l1, l2=l2, l1l2=l1l2, l3=l3, data=substitute(data),
    common.intercept=common.intercept, zeroL3=zeroL3, maxit=maxit, pres=pres,
    verbose=verbose)
    tempcall<-as.call( c(expression(lm.bp), templist))
    rm(templist)
    #
    #
    # ------------------------------------------------------------------------------
    # Karlis and Ntzoufras (2003, 2004, 2005)
    # EM algorithms for Bivariate Poisson Models
    # ------------------------------------------------------------------------------
    #
    # l1            : formula for the first  linear predictor (of lambda1)
    # l2            : formula for the second linear predictor (of lambda2)
    # l1l2          : formula for common variables on both lambda1 and lambda2
    # l3            : formula for the third first linear predictor/covariance
    parameter (lambda3)
    # common.intercept: logical argument defining whether common intercept should
                        be used for lamdba1,lambda2
    #
    # data     : data.frame which contains data {required arguement}
    # zeroL3   : Logical argument controlling whether lambda3 is zero
                      (DblPoisson) or not
```

```
# maxit    : maximum number of iterations
# pres     : precision of the relative likelihood difference after
                which EM stops
# verbose    : Logical argument controlling whether beta parameters will we
#                printed while EM runs. Default value is taken
                 options()$verbose value.
# ------------------------------------------------------------------------------
#
#
#
# set common or noncommon intercept
if (common.intercept){ formula1.terms<-'1'  }
else {formula1.terms<-'internal.data1$noncommon'    }
#
#
namex<-as.character(l1[2])
namey<-as.character(l2[2])
x<-data[,names(data)==namex]
y<-data[,names(data)==namey]
#
# Data length
n<-length(x)
lengthpvec<-1
#
#
#
# initial values
s<-rep(0,n)
like<-1:n*0
zero<- ( x==0 )|( y==0 )
if   (zeroL3) { lambda3<-rep(0,n) }
else          { lambda3<-rep( max(0.1, cov(x,y,use='complete.obs')), n) }
#
#
# form dataframes used
# data1 includes modelling on lambda1 and lambda2
# data2 includes modelling on lambda3
# internal.data1 and internal.data2 are data frames used for
  additional internal variables
#
internal.data1<-data.frame( y1y2=c( x, y) )
internal.data2<-data.frame( y3 = rep(0, n ) )
#
p<-length(as.data.frame(data))
data1<-rbind(data, data)
names(data1)<-names(data)
#
# removing x and y
```

```
data1<-data1[ , names(data1)!=namex]
data1<-data1[ , names(data1)!=namey]
#
#
# define full model
if (as.character(l1[3])=='.') { l1<-formula( paste( as.character(l1[2]),
paste( names(data1),'',collapse='+',sep='' ), sep='~')  )   }
if (as.character(l2[3])=='.') { l2<-formula( paste( as.character(l2[2]),
paste( names(data1),'',collapse='+',sep='' ), sep='~')  )   }
if (as.character(l3[2])=='.') { l3<-formula( paste( '',
paste( names(data1),'',collapse='+',sep='' ) , sep='~') )   }
#
# define the formula used for covariance term
formula2<-formula(paste('internal.data2$y3~',as.character(l3[2]),sep=''))
#
internal.data1$noncommon<- as.factor(c(1:n*0,1:n*0+1))
contrasts(internal.data1$noncommon)<-contr.treatment(2, base=1)
internal.data1$indct1<-c(1:n*0+1,1:n*0  )
internal.data1$indct2<-c(1:n*0  ,1:n*0+1)
#
#
if (!zeroL3){
  data2<-data1[1:n,]
  names(data2)<-names(data1)
}
#
#####
#
# add the common terms
#
if ( !is.null(l1l2) ) {
  formula1.terms<-paste( formula1.terms, as.character(l1l2[2]),sep='+')
}
#
# add the special common terms (if any)
#
#
#
# in this section we identify non-common parameters
# if a variable X is common in all formulas the we use term x*noncommon
to include x+x:noncommon terms
# otherwise use I(internal.data1$indct1*x) to add sepererate parameter
on lambda1
#
templ1<- labels(terms(l1))
#
# run this only if there are terms in l1 formula
if (length( templ1 )>0){
```

```
  for ( k1 in 1:length( templ1 ) ){
    if (!is.null(l1l2)){checkvar1<-sum(labels(terms(l1l2))==templ1[k1])==1}
    else{ checkvar1<-FALSE }
    checkvar2<-sum(labels(terms(l2))==templ1[k1] )==1
    if (checkvar1&checkvar2) {formula1.terms<-paste(formula1.terms, paste('intern
    else{
      formula1.terms<-paste(formula1.terms, paste('+I(internal.data1$indct1*',
      templ1[k1],sep=''), sep='')
      formula1.terms<-paste(formula1.terms, ')',sep='')
    }
  }
}
#
# if a variable X is not common st
# otherwise use I(internal.data1$indct1*x) to add sepererate parameter
on lambda1
#
templ2<- labels(terms(l2))
#
# run this only if there are terms in l1 formula
if (length( templ2 )>0){
  for ( k1 in 1:length( templ2 ) ){
    if ( !is.null(l1l2) )
    {checkvar1<-(sum(labels(terms(l1l2))==templ2[k1] )+
    sum(labels(terms(l1))==templ2[k1] ))!=2     }
    else{ checkvar1<-TRUE }
    if ( checkvar1 ) {
      formula1.terms<-paste(formula1.terms,
      paste('+I(internal.data1$indct2*',templ2[k1],sep=''), sep='')
      formula1.terms<-paste(formula1.terms, ')',sep='')
    }
  }
}
#
rm(templ1)
rm(templ2)
rm(Checkvar1)
rm(Checkvar2)
#
#
#
#
#
#
# This bit creates labels for special terms of type c(x1,x2) used in l1l2
#
#
formula1<-formula(paste('internal.data1$y1y2~',formula1.terms,sep=''))
```

```
tmpform1<-as.character(formula1[3])
newformula<-formula1
while( regexpr('c\\(',tmpform1) != -1)
{
  temppos1<-regexpr('c\\(',tmpform1)[1]
  tempfor <-substring( tmpform1, first = temppos1+ 2 )
  temppos2<-regexpr('\\)' , tempfor)[1]
  tempvar <-substring( tempfor , first = 1,  last = temppos2-1 )
  temppos3<-regexpr(', ' , tempvar)[1]
  tempname1<-substring(tempfor    , first = 1,  last = temppos3-1 )
  tempname2<-substring(tempfor    , first = temppos3+2, last=temppos2-1)
  tempname2<-sub( '\\)','', tempname2  )
  tempvar1<-data[, names(data)==tempname1]
  tempvar2<-data[, names(data)==tempname2]
  data1$newvar1<-c(tempvar1, tempvar2)
  #
  if( is.factor(tempvar1)& is.factor(tempvar2) ){
    data1$newvar1<-as.factor(data1$newvar1)
    if (all(levels(tempvar1)==levels(tempvar2))){
      attributes(data1$newvar1)<-attributes(tempvar1)}
  }
  tempvar<-sub( ', ' , '..',  tempvar )
  names(data1)[names(data1)=='newvar1']<-tempvar
  newformula<-sub( 'c\\(','', tmpform1  )
  newformula<-sub( '\\)','', newformula  )
  newformula<-sub( ', ' , '..', newformula  )
  tmpform1<-newformula
  formula1<-formula(paste('internal.data1$y1y2~',newformula,sep=''))
}
#####
rm(temppos1)
rm(temppos2)
rm(temppos3)
rm(tmpform1)
rm(tempfor)
rm(tempvar)
rm(tempvar1)
rm(tempvar2)
rm(tempname1)
rm(tempname2)
#
#
# Initial values for lambda
#
lambda<-glm(formula1,family=poisson, data=data1)$fitted
#
lambda1<-lambda[1:n]
lambda2<-lambda[(n+1):(2*n)]
```

```
#
difllike<-100.0
loglike0<-1000.0
i<-0
#
# fitting the Double Poisson Model
if (zeroL3) {
  #
  # fit the double Poisson model
  y0<-c(x,y)
  m<-glm( formula1, family=poisson, data=data1 )
  p3<-length(m$coef)
  beta<-m$coef
  # ------------------------------------------------
  # creating names for parameters
  #
  names(beta)<-newnamesbeta( beta )
  #
  #     end of name creations (l1, l2, l2-l1, blank)
  #  ------------------------------------------------
  betaparameters<-splitbeta( beta )
  #
  lambda<-fitted(m)
  lambda1<-lambda[1:n]
  lambda2<-lambda[(n+1):(2*n)]
  like<-dpois(x, lambda1) * dpois( y, lambda2 )
  loglike<-sum(log(like))
  #
  # calculation of BIC and AIC for bivpoisson model
  noparams<- m$rank
  AIC<- -2*loglike + noparams * 2
  BIC<- -2*loglike + noparams * log(2*n)
  #
  #
  # Calculation of BIC, AIC of Poisson saturated model
  x.mean<-x
  x.mean[x==0]<-1e-12
  y.mean<-y
  y.mean[y==0]<-1e-12
  AIC.sat <-  sum(log( dpois( x , x.mean ) ) + log( dpois( y , y.mean ) ))
  BIC.sat <-  -2 * AIC.sat + (2*n)* log(2*n)
  AIC.sat <-  -2 * AIC.sat + (2*n)* 2
  #
  #
  AICtotal<-c(AIC.sat, AIC);
  BICtotal<-c(BIC.sat, BIC );
  names(AICtotal)<-c('Saturated', 'DblPois')
  names(BICtotal)<-c('Saturated', 'DblPois')
```

```
  #
  # putting all betas in one vector
  allbeta<-c(betaparameters$beta1,betaparameters$beta2)
  names(allbeta)<-c( paste( '(l1):', names(betaparameters$beta1),
  sep='' ),paste('(l2):', names(betaparameters$beta2), sep='' ) )

  result<-list(coefficients=allbeta, fitted.values=data.frame(x=m$fitted[1:n],
  y=m$fitted[(n+1):(2*n)]),
              residuals=data.frame(x=x-m$fitted[1:n],y=y-m$fitted[(n+1):(2*n)]),
              beta1=betaparameters$beta1, beta2=betaparameters$beta2,
              lambda1=m$fitted[1:n], lambda2=m$fitted[(n+1):(2*n)], lambda3=0,
              loglikelihood=loglike, iterations=1, parameters=noparams,
              AIC=AICtotal, BIC=BICtotal, call=tempcall)
}
else {
  loglike<-rep(0,maxit)
  while ( (difllike>pres) && (i <= maxit) ) {
    i<-i+1
    #####   E step  ######
    for (j in 1:n) {
      if (zero[j]) {
        s[j]<-0.0;
        like[j]<- log(dpois(x[j], lambda1[j]))+log(dpois(y[j],lambda2[j]))-
        lambda3[j];
      }
      else {
        lbp1<-pbivpois(x[j]-1, y[j]-1,lambda=c(lambda1[j],lambda2[j],
        lambda3[j]),log=TRUE);
        lbp2<-pbivpois(x[j]  , y[j]  ,lambda=c(lambda1[j],lambda2[j],
        lambda3[j]),log=TRUE);
        #
        s[j]<-exp(log(lambda3[j])+lbp1-lbp2);
        like[j]<-lbp2;
      }
    }
    ##### end of E step  ######
    x1<-x-s
    x2<-y-s

    x1[ (x1<0)&(x1>-1.0e-8)]<-0.00
    x2[ (x2<0)&(x2>-1.0e-8)]<-0.00

    loglike[i]<-sum(like)
    difllike<-abs( (loglike0-loglike[i])/loglike0 )
    loglike0<-loglike[i]
    #
    #
    #####   M step  ######
```

```
  #
  #   fit model on lambda3
  internal.data2$y3<-s
  m0<-glm( formula2, family=poisson, data=data2 )
  beta3<-m0$coef
  lambda3<-m0$fitted
  #
  #   fit model on lambda1 & lambda2
  internal.data1$y1y2<-c(x1,x2)

  m<-glm( formula1, family=poisson, data=data1 )
  p3<-length(m$coef)
  beta<-m$coef
  #   creating names for parameters
  names(beta)<-newnamesbeta( beta )
  #
  #

  lambda<-fitted(m)
  lambda1<-lambda[1:n]
  lambda2<-lambda[(n+1):(2*n)]
  #####   end of M step  ######
  #
  #   detailed or compressed printing during the EM iterations
  if (verbose) {
    printvector<-c( i, beta, beta3,loglike[i], difllike )
    names(printvector)<-c( 'iter', names(beta), paste('(l3):',
    names(beta3),sep=''), 'loglike', 'Rel.Dif.loglike')}
  else {
    printvector<-c( i, loglike[i], difllike )
    names(printvector)<-c( 'iter', 'loglike', 'Rel.Dif.loglike')}
  #
  lengthpvec<-length(printvector)
  print.default( printvector, digits=4 )
}
#
# calculation of BIC and AIC for bivpoisson model
noparams<- m$rank + m0$rank
AIC<- -2*loglike[i] + noparams * 2
BIC<- -2*loglike[i] + noparams * log(2*n)
#
#
# Calculation of BIC, AIC of Poisson saturated model
x.mean<-x
x.mean[x==0]<-1e-12
y.mean<-y
y.mean[y==0]<-1e-12
AIC.sat <-  sum(log( dpois( x , x.mean ) ) + log( dpois(y,y.mean)))
```

```
      BIC.sat <-  -2 * AIC.sat + (2*n)* log(2*n)
      AIC.sat <-  -2 * AIC.sat + (2*n)* 2
      #
      #
      AICtotal<-c(AIC.sat, AIC);
      BICtotal<-c(BIC.sat, BIC );
      names(AICtotal)<-c('Saturated', 'BivPois')
      names(BICtotal)<-c('Saturated', 'BivPois')
      #
      # spliting parameter vector
      betaparameters<-splitbeta( beta )
      #
      # putting all betas in one vector
      allbeta<-c(betaparameters$beta1,betaparameters$beta2, beta3)
      names(allbeta)<-c( paste( '(l1):',names(betaparameters$beta1),
      sep=''),paste('(l2):', names(betaparameters$beta2), sep='' ),
      paste('(l3):', names(beta3), sep='' ) )
      #
      #  Calculation of output
      result<-list(coefficients=allbeta, fitted.values=data.frame(x=m$fitted[1:n]+
      lambda3,y=m$fitted[(n+1):(2*n)]+lambda3),
                   residuals=data.frame(x=x-m$fitted[1:n]-lambda3,
                   y=y-m$fitted[(n+1):(2*n)]-lambda3),
                   beta1=betaparameters$beta1, beta2=betaparameters$beta2,
                   beta3=beta3, lambda1=m$fitted[1:n], lambda2=m$fitted[(n+1):(2*n)],
                   lambda3=lambda3, loglikelihood=loglike[1:i], parameters=noparams,
                   AIC=AICtotal, BIC=BICtotal,iterations=i, call=tempcall )
      #
      #
    } # end of elseif
    options(warn=0)
    #
    class(result)<-c('lm.bp', 'lm')

    result
    #
    #
  }


"bivpois.table" <-
  function(x, y, lambda = c(1, 1, 1))
  {
    # ------------------------------------------------------------------------
    # Karlis and Ntzoufras (2003, 2004)
    # EM algorithms for Bivariate Poisson Models
    # ------------------------------------------------------------------------
    # x     : 1st count variable
```

```
    # y     : 2nd count variable
    # lambda: parameters of the bivariate poisson distribution.
    # ------------------------------------------------------------------------------

    j<-0
    n <- length(x)
    maxy <- c(max(x), max(y))   #Set initial values for parameters
    lambda1 <- lambda[1]
    lambda2 <- lambda[2]
    lambda3 <- lambda[3]
    if((x == 0) | (y == 0)) {
      prob <- matrix(NA, nrow = maxy[1] + 1, ncol = maxy[2]+1, byrow = T)
      prob[maxy[1] + 1, maxy[2] + 1] <- exp( - lambda3) *
        dpois(x[j], lambda1[j]) * dpois(y[j], lambda2[j])
    }
    else {
      prob <- matrix(NA, nrow = maxy[1] + 1, ncol = maxy[2]+1, byrow = T)
      k <- 1
      m <- 1
      prob[k, m] <- exp( - lambda1 - lambda2 - lambda3)
      for(i in 2:(maxy[1] + 1)) {
        prob[i, 1] <- (prob[i - 1, 1] * lambda1)/(i - 1)
      }
      for(j in 2:(maxy[2] + 1)) {
        prob[1, j] <- (prob[1, j - 1] * lambda2)/(j - 1)
      }
      for(j in 2:(maxy[2] + 1)) {
        for(i in 2:(maxy[1] + 1)) {
          prob[i, j] <- (lambda1 * prob[i - 1, j] +
                           lambda3 * prob[i - 1, j - 1])/(i - 1)
        }
      }
    }
    result <- prob
    result
  }

"lm.dibp" <-
  function
( l1, l2, l1l2=NULL, l3=~1, data, common.intercept=FALSE, zeroL3=FALSE,
distribution='discrete', jmax=2,maxit=300, pres=1e-8,
verbose=getOption('verbose') )
  {
  options(warn=-1)
  #
  # definition of function call
  templist<-list( l1=l1, l2=l2, l1l2=l1l2, l3=l3, data=substitute(data),
  common.intercept=common.intercept, zeroL3=zeroL3,
```

```
distribution=distribution,jmax=jmax, maxit=maxit, pres=pres,
verbose=verbose)
tempcall<-as.call( c(expression(lm.dibp), templist))
rm(templist)
#
#
#
# ------------------------------------------------------------------------
# Karlis and Ntzoufras (2003, 2004, 2005)
# EM algorithms for Bivariate Poisson Models
# ------------------------------------------------------------------------
#
# PARAMETERS COMMON WITH lm.bp
# l1              : formula for the first  linear predictor (of lambda1)
# l2              : formula for the second linear predictor (of lambda2)
# l1l2            : formula for common variables on both lambda1 and lambda2
# l3              : formula for the third first linear predictor/covariance
                parameter (lambda3)
# common.intercept: logical argument defining whether common intercept
                should be used for lamdba1,lambda2
#
# data     : data.frame which contains data {required arguement}
# zeroL3   : Logical argument controlling whether lambda3 is zero
             (DblPoisson) or not
# maxit    : maximum number of iterations
# pres     : precision of the relative likelihood difference after
             which EM stops
# verbose  : Logical argument controlling whether beta parameters will we
#                printed while EM runs. Default value is taken
                options()$verbose value.
#
# PARAMETERS ADDITIONAL TO lm.bp
# distribution : Selection of diagonal inflation distribution.
#                Three choices are available:
#                ='discrete'  : Discrete, jmax is the number of diagonal
                   elements [0,1,...,]
#                ='poisson'   : Poisson with mean theta.
#                ='geometrics': Geometric with success probability theta.
#                Default is DISCRETE(2). theta[1] and theta[2] stand for
                theta_1, theta_2
#                                       while theta_0=1-theta[1]-theta[2].
# jmax         : Used only for DISCRETE diagonal distribution
                (distribution='discrete').
#                Indicates the number of parameters of the DISCRETE
                distribution.
# ------------------------------------------------------------------------
#
#
```

```
#
# set common or noncommon intercept
if (common.intercept){ formula1.terms<-'1'    }
else {formula1.terms<-'internal.data1$noncommon'  }
#
#
namex<-as.character(l1[2])
namey<-as.character(l2[2])
x<-data[,names(data)==namex]
y<-data[,names(data)==namey]
#
#
# Data length
n<-length(x)
lengthprintvec<-1
#
#
#
# definition of diagonal inflated distribution
maxy<-max(c(x,y))
#
#  changing distribution to codes 1,2,3
dist<-distribution
if      ( charmatch( dist, 'poisson'  , nomatch=0) ==1 ) {distribution<-2}
else if ( charmatch( dist, 'geometric', nomatch=0) ==1 ) {distribution<-3}
else if ( charmatch( dist, 'discrete' , nomatch=0) ==1 ) {distribution<-1}
if ( distribution==1 ){
  dilabel<-paste('Inflation Distribution: Discrete with J=',jmax)
  if (jmax==0) {theta<-0}
  else         { theta<-1:jmax*0+1/(jmax+1) }
  di.f<-function (x, theta){
    JMAX<-length(theta)
    if      (x>JMAX) { res<-0 }
    else if (x==0)   { res<-1-sum(theta) }
    else             { res<-theta[x] }
    res
  }
}
else if ( distribution==2 ){
  dilabel<-'Inflation Distribution: Poisson'
  theta<-1.0;
  di.f<-function (x, theta){
    if (theta>0) { res<-dpois( x, theta ) }
    else {
      if (x==0) { res<-1}
      else {res<-1e-12}
    }
  }
```

```
}
else if ( distribution==3 ){
  dilabel<-'Inflation Distribution: Geometric'
  theta<-0.5;
  di.f<-function (x, theta){
    if (theta>0) {
      if(theta==1) {theta<-0.9999999}
      res<-dgeom( x, theta ) }
    else if (theta==1){
      if (x==0) { res<-1}
      else {res<-1e-12}
    }
    else {res<-1e-12}
  }
}
else {
  stop(paste(distribution, 'Not known distribution.', sep=': '))
}
# ------
# setting up data frames, vectors and data
#
# form dataframes used
# data1 includes modelling on lambda1 and lambda2
# data2 includes modelling on lambda3
# internal.data1 and internal.data2 are data frames used for additional
internal variables
#
internal.data1<-data.frame( y1y2=c( x, y) )
internal.data2<-data.frame( y3 = rep(0, n ) )
#
p<-length(as.data.frame(data))
data1<-rbind(data, data)
names(data1)<-names(data)
#
# removing x and y
data1<-data1[ , names(data1)!=namex]
data1<-data1[ , names(data1)!=namey]
#
#
#
# define full model
if (as.character(l1[3])=='.') {l1<-formula( paste(as.character(l1[2]),
paste( names(data1),'',collapse='+',sep='' ), sep='~')    ) }
if (as.character(l2[3])=='.') {l2<-formula( paste(as.character(l2[2]),
paste( names(data1),'',collapse='+',sep='' ), sep='~')  ) }
if (as.character(l3[2])=='.') {l3<-formula( paste( '',
paste( names(data1),'',collapse='+',sep='' ) , sep='~') ) }
#
```

```
# define the formula used for covariance term
formula2<-formula(paste('internal.data2$y3~',as.character(l3[2]),sep=''))
#
internal.data1$noncommon<- as.factor(c(1:n*0,1:n*0+1))
contrasts(internal.data1$noncommon)<-contr.treatment(2, base=1)
internal.data1$indct1<-c(1:n*0+1,1:n*0  )
internal.data1$indct2<-c(1:n*0  ,1:n*0+1)
#
#
if (!zeroL3){
  data2<-data1[1:n,]
  names(data2)<-names(data1)
}
#####
#
# add the common terms
#
if ( !is.null(l1l2) ) {
  formula1.terms<-paste( formula1.terms, as.character(l1l2[2]),sep='+')
}
#
# add the special common terms (if any)
#
#
#
# in this section we identify non-common parameters
# if a variable X is common in all formulas the we use term x*noncommon
to include x+x:noncommon terms
# otherwise use I(internal.data1$indct1*x) to add sepererate parameter
on lambda1
#
templ1<- labels(terms(l1))
#
# run this only if there are terms in l1 formula
if (length( templ1 )>0){
  for ( k1 in 1:length( templ1 ) ){
    if (!is.null(l1l2)){checkvar1<-sum(labels(terms(l1l2))==templ1[k1])==1}
    else{ checkvar1<-FALSE }
    checkvar2<-sum(labels(terms(l2))==templ1[k1] )==1
    if (checkvar1&checkvar2) {formula1.terms<-paste(formula1.terms,
    paste('internal.data1$noncommon*',templ1[k1],sep=''), sep='+')}
    else{
      formula1.terms<-paste(formula1.terms,
      paste('+I(internal.data1$indct1*',templ1[k1],sep=''), sep='')
      formula1.terms<-paste(formula1.terms, ')',sep='')
    }
  }
}
```

```
#
# if a variable X is not common st
# otherwise use I(internal.data1$indct1*x) to add sepererate
parameter on lambda1
#
templ2<- labels(terms(l2))
#
# run this only if there are terms in l1 formula
if (length( templ2 )>0){
  for ( k1 in 1:length( templ2 ) ){
    if ( !is.null(l1l2) )
    {checkvar1<-(sum(labels(terms(l1l2))==templ2[k1] )+
    sum(labels(terms(l1))==templ2[k1] ))!=2       }
    else{ checkvar1<-TRUE }
    if ( checkvar1 ) {
      formula1.terms<-paste(formula1.terms,
      paste('+I(internal.data1$indct2*',templ2[k1],sep=''), sep='')
      formula1.terms<-paste(formula1.terms, ')',sep='')
    }
  }
}
#
rm(templ1)
rm(templ2)
rm(Checkvar1)
rm(Checkvar2)
#
#
#
#
#
#
# This bit creates labels for special terms of type c(x1,x2) used in l1l2
#
#
formula1<-formula(paste('internal.data1$y1y2~',formula1.terms,sep=''))
tmpform1<-as.character(formula1[3])
newformula<-formula1
while( regexpr('c\\(',tmpform1) != -1)
{
  temppos1<-regexpr('c\\(',tmpform1)[1]
  tempfor <-substring( tmpform1, first = temppos1+ 2 )
  temppos2<-regexpr('\\)' , tempfor)[1]
  tempvar <-substring( tempfor , first = 1,  last = temppos2-1 )
  temppos3<-regexpr(', ' , tempvar)[1]
  tempname1<-substring(tempfor    , first = 1,  last = temppos3-1 )
  tempname2<-substring(tempfor    , first = temppos3+2, last=temppos2-1)
  tempname2<-sub( '\\)','', tempname2  )
```

```
  tempvar1<-data[, names(data)==tempname1]
  tempvar2<-data[, names(data)==tempname2]
  data1$newvar1<-c(tempvar1, tempvar2)
  #
  if( is.factor(tempvar1)& is.factor(tempvar2) ){
    data1$newvar1<-as.factor(data1$newvar1)
    if (all(levels(tempvar1)==levels(tempvar2))){
      attributes(data1$newvar1)<-attributes(tempvar1)}
  }
  tempvar<-sub( ', ' , '..',  tempvar )
  names(data1)[names(data1)=='newvar1']<-tempvar
  newformula<-sub( 'c\\(','', tmpform1  )
  newformula<-sub( '\\)','', newformula  )
  newformula<-sub( ', ' , '..', newformula  )
  tmpform1<-newformula
  formula1<-formula(paste('internal.data1$y1y2~',newformula,sep=''))
}
#####
rm(temppos1)
rm(temppos2)
rm(temppos3)
rm(tmpform1)
rm(tempfor)
rm(tempvar)
rm(tempvar1)
rm(tempvar2)
rm(tempname1)
rm(tempname2)
# ------
# initial values for parameters
prob<-0.20
s<-rep(0,n)
vi<-1:n*0
v1<-1-c(vi,vi)
like<-1:n*0
zero<- ( x==0 )|( y==0 )
if   (zeroL3) { lambda3<-rep(0,n) }
else { lambda3<-rep( max(0.1, cov(x,y,use='complete.obs')), n) }
#
#
#
#
# Initial values for lambda

internal.data1$v1<-1-c(vi,vi);

lambda<-glm( formula1, family=poisson, data=data1,
weights=internal.data1$v1, maxit=100)$fitted
```

```
#
lambda1<-lambda[1:n]
lambda2<-lambda[(n+1):(2*n)]
#
difllike<-100.0
loglike0<-1000.0
i<-0
ii<-0

if (zeroL3) {
  #
  # fit double poisson diagonal inflated model
  loglike<-rep(0, maxit)
  lambda3<-1:n*0
  while ( (difllike>pres) && (i <= maxit) ) {
    i<-i+1
    #####   E step  ######
    for (j in 1:n) {
      if (zero[j]) {
        s[j]<-0;
        #         calculation of log-likelihood
        if (x[j]==y[j]) {
          density.di<-di.f( 0.0, theta )
          like[j]<-log( (1-prob)*exp(-lambda1[j]-
          lambda2[j])+prob*density.di );
          vi[j]<-prob*density.di*exp(-like[j]) }
        else{
          like[j]<-log(1-prob)+log(dpois(x[j],lambda1[j]))+
          log(dpois(y[j],lambda2[j]));
          vi[j]<-0.0 ;}
      }
      else {
        if (x[j]==y[j]) {
          density.di<-di.f( x[j],theta );
          like[j]<-log( (1-prob)*dpois( x[j],lambda1[j] )*
          dpois( y[j],lambda2[j] ) + prob*density.di );
          vi[j]  <- prob*density.di*exp( -like[j] ) }
        else  {
          vi[j]<-0.0;
          like[j]<-log(1-prob)+log( dpois(x[j],lambda1[j])*
          dpois(y[j],lambda2[j]) )}
      }
    }
    #### end of E-step #########
    x1<-x;
    x2<-y;
    loglike[i]<-sum( like ) ;
    difllike<-abs( (loglike0-loglike[i])/loglike0 )
```

```
loglike0<-loglike[i]
#
#
############ M-step #############
# estimate mixing proportion
prob<-sum(vi)/n
#
# maximization of each theta parameter
if ( distribution == 1 ) {
  # calculation of theta_j, j=1,...,jmax ; theta_0=1-sum(theta)
  if (jmax==0) { theta<-0 }
  else {
    for (ii in 1:jmax) {
      temp<-as.numeric(( (x==ii) & (y==ii) ));
      theta[ii]<-sum(temp*vi)/sum(vi)
    }
  }
}
else if (distribution==2){
  # calculation of theta for poisson diagonal inflation
  theta<- sum(vi*x)/sum(vi) }
else if (distribution==3){
  # calculation of theta for geometric diagonal inflation
  theta<- sum(vi)/( sum(vi*x)+sum(vi) ) }
#
# fit model on lambda1 & lambda2
#
internal.data1$v1<- 1-c(vi,vi);
internal.data1$v1[ (internal.data1$v1<0)&
(internal.data1$v1>-1.0e-10) ]<-0.0
#
x1[(x1<0)&(x1>-1.0e-10)]<-0.0
x2[(x2<0)&(x2>-1.0e-10)]<-0.0
internal.data1$y1y2<-c(x1,x2)
m<-glm( formula1, family=poisson, data=data1,
weights=internal.data1$v1 , maxit=100)

p3<-length(m$coef)
beta<-m$coef
# ----------------------------------------------------
# creating names for parameters
names(beta)<-newnamesbeta( beta )
#
#     end of name creations (l1, l2, l2-l1, blank)
#  ------------------------------------------------
betaparameters<-splitbeta( beta )
#
lambda<-fitted(m)
```

```
  lambda1<-lambda[1:n]
  lambda2<-lambda[(n+1):(2*n)]
  #
  #####   end of M step  ######
  #
  #  printing also beta
  if (verbose) {
    printvec<- c( i,beta,100.0*prob, theta,
    loglike[i], difllike );
    names(printvec)<-c( 'iter', names(beta),'Mix.p(%)',
    paste( 'theta', 1:length(theta),sep='' ), 'loglike',
    'Rel.Dif.loglike')
  }
  #  limited print out
  else {
    printvec<- c( i, 100.0*prob, theta,  loglike[i], difllike );
    names(printvec)<-c( 'iter','Mix.p(%)', paste( 'theta',
    1:length(theta),sep='' ), 'loglike', 'Rel.Dif.loglike')
  }
  lengthprintvec<-length(printvec)
  print.default( printvec, digits=4 )
}

#
#   calculation of BIC and AIC for double poisson model
if ( (distribution==1)&&(jmax==0) ){noparams<- m$rank +1}
else                               {noparams<- m$rank + length( theta ) +1}
AIC<- -2*loglike[i] + noparams * 2
BIC<- -2*loglike[i] + noparams * log(2*n)
#
#
#   Calculation of BIC, AIC of Poisson saturated model
x.mean<-x
x.mean[x==0]<-1e-12
y.mean<-y
y.mean[y==0]<-1e-12
AIC.sat <-  sum(log( dpois( x , x.mean ) ) + log( dpois( y , y.mean ) ))
BIC.sat <-  -2 * AIC.sat + (2*n)* log(2*n)
AIC.sat <-  -2 * AIC.sat + (2*n)* 2
#
#
AICtotal<-c(AIC.sat, AIC);
BICtotal<-c(BIC.sat, BIC );
names(AICtotal)<-c('Saturated', 'DblPois')
names(BICtotal)<-c('Saturated', 'DblPois')
#
allbeta<-c(betaparameters$beta1,betaparameters$beta2)
names(allbeta)<-c( paste( '(l1):', names(betaparameters$beta1),
```

```
  sep='' ),paste('(l2):', names(betaparameters$beta2), sep='' ) )
  allparameters<-c(allbeta, prob, theta)
  if (distribution==1){ names(allparameters)<-c( names(allbeta),
  'p', paste('theta', 1:length(theta),sep='') ) }
  else {names(allparameters)<-c( names(allbeta), 'p', 'theta') }
  #
  #   calculation of fitted values
  #  ----------------------------
  fittedval1<-(1-prob)*m$fitted[1:n]
  fittedval2<-(1-prob)*m$fitted[(n+1):(2*n)]
  #
  meandiag<-0
  if ((distribution==1)&&(jmax>0)) { meandiag<-sum( theta[1:jmax]*1:jmax ) }
  else if (distribution==2) { meandiag<-theta }
  else if (distribution==3) { meandiag<- (1-theta)/theta }
  #
  fittedval1[x==y]<-prob*meandiag + fittedval1[x==y]
  fittedval2[x==y]<-prob*meandiag + fittedval2[x==y]
  #
  result<-list(coefficients=allparameters,
               fitted.values=data.frame(x=fittedval1,y=fittedval2),
               residuals=data.frame(x=x-fittedval1,y=y-fittedval2),
               beta1=betaparameters$beta1, beta2=betaparameters$beta2,
               p=prob, theta=theta, diagonal.distribution=dilabel,
               lambda1=m$fitted[1:n], lambda2=m$fitted[(n+1):(2*n)],
               loglikelihood=loglike[1:i], parameters=noparams, AIC=AICtotal,
               BIC=BICtotal,iterations=i , call=tempcall)
  #
  #
  # end of diagonal inflated double poisson model
}
else {
  loglike<-rep(0,maxit)
  while ( (difllike>pres) && (i <= maxit) ) {
    i<-i+1
    #####   E step  ######
    for (j in 1:n) {
      if (zero[j]) {
        s[j]<-0;
        #         calculation of log-likelihood
        if (x[j]==y[j]) {
          density.di<-di.f( 0.0, theta )
          like[j]<- log( (1-prob)*exp(-lambda1[j]-lambda2[j]-
          lambda3[j])+prob*density.di );
          vi[j]<-prob*density.di*exp(-like[j]) }
        else{
          like[j]<-log(1-prob)-lambda3[j] +log(dpois(x[j],lambda1[j]))
          +log(dpois(y[j],lambda2[j]));
```

```
      vi[j]<-0.0 ;}
  }
  else {
    lbp1<-pbivpois(x[j]-1, y[j]-1, lambda=c(lambda1[j],lambda2[j],
    lambda3[j]),  log=TRUE );
    lbp2<-pbivpois(x[j]  , y[j]  , lambda=c(lambda1[j],lambda2[j],
    lambda3[j]), log=TRUE );
    s[j]<-exp( log(lambda3[j]) + lbp1 - lbp2 );
    #         like[j]<-lbp2;
    if (x[j]==y[j]) {
      density.di<-di.f( x[j],theta );
      like[j]<-log( (1-prob)*exp(lbp2) + prob*density.di );
      vi[j]  <- prob*density.di*exp( -like[j] ) }
    else  {
      vi[j]<-0.0;
      like[j]<-log(1-prob)+lbp2 }
  }
}
#### end of E-step #########
x1<-x-s;
x2<-y-s;
loglike[i]<-sum( like ) ;
difllike<-abs( (loglike0-loglike[i])/loglike0 )
loglike0<-loglike[i]
#
#
############ M-step #############
# estimate mixing proportion
prob<-sum(vi)/n
#
# maximization of each theta parameter
if ( distribution == 1 ) {
  # calculation of theta_j, j=1,...,jmax ; theta_0=1-sum(theta)
  #       cat (c('1:discrete, jmax=', jmax), '\n')
  if (jmax==0){ theta<-0}
  else{
    for (ii in 1:jmax) {
      temp<-as.numeric(( (x==ii) & (y==ii) ));
      theta[ii]<-sum(temp*vi)/sum(vi)
      #               print( c(ii, sum(temp), sum(vi), sum(temp*vi) ) )
    }
    #             cat( c('2:discrete, jmax=', jmax), '\n')
  }
}
else if (distribution==2){
  # calculation of theta for poisson diagonal inflation
  theta<- sum(vi*x)/sum(vi) }
else if (distribution==3){
```

```
  #   else {
  # calculation of theta for geometric diagonal inflation
  theta<- sum(vi)/( sum(vi*x)+sum(vi) ) }
#     fit model on lambda3
internal.data2$v1<- 1-vi;
internal.data2$v1[ (internal.data2$v1<0)&(internal.data2$v1>-1.0e-10) ]<-0.0
#
internal.data2$y3<-s;
m0<-glm( formula2, family=poisson,data=data2,weights=internal.data2$v1,
maxit=100)
beta3<-m0$coef
lambda3<-m0$fitted
#
# fit model on lambda1 & lambda2
internal.data1$v1<- 1-c(vi,vi);
internal.data1$v1[ (internal.data1$v1<0)&(internal.data1$v1>-1.0e-10) ]<-0.0
#
x1[(x1<0)&(x1>-1.0e-10)]<-0.0
x2[(x2<0)&(x2>-1.0e-10)]<-0.0
internal.data1$y1y2<-c(x1,x2)
m<-glm( formula1, family=poisson,data=data1,weights=internal.data1$v1,
maxit=100)
p3<-length(m$coef)
beta<-m$coef
#  ----
# creating names for parameters
names(beta)<-newnamesbeta( beta )
#  ----
lambda<-fitted(m)
lambda1<-lambda[1:n]
lambda2<-lambda[(n+1):(2*n)]
#
#####   end of M step  ######
#
# print all parameters including beta
if (verbose) {
  printvec<- c( i,beta,beta3,100.0*prob, theta,  loglike[i], difllike );
  names(printvec)<-c( 'iter', names(beta),paste('l3_',names(beta3),sep=''),
  'Mix.p(%)', paste( 'theta', 1:length(theta),sep='' ),
  'loglike', 'Rel.Dif.loglike')
}
#
#  limited print out
else {
  printvec<- c( i, 100.0*prob, theta,  loglike[i], difllike );
  names(printvec)<-c( 'iter', 'Mix.p(%)', paste( 'theta', 1:length(theta),
  sep='' ), 'loglike', 'Rel.Dif.loglike')
}
```

```
  #
  lengthprintvec<-length(printvec)
  print.default( printvec, digits=4 )
}
#
#   calculation of BIC and AIC for bivpoisson model
if ( (distribution==1)&&(jmax==0) ){noparams<- m$rank + m0$rank + 1}
else              {noparams<- m$rank + m0$rank + length( theta ) +1}
AIC<- -2*loglike[i] + noparams * 2
BIC<- -2*loglike[i] + noparams * log(2*n)
#
#
#   Calculation of BIC, AIC of Poisson saturated model
x.mean<-x
x.mean[x==0]<-1e-12
y.mean<-y
y.mean[y==0]<-1e-12
AIC.sat <-  sum(log(dpois(x,x.mean))+log(dpois(y,y.mean)))
BIC.sat <-  -2 * AIC.sat + (2*n)* log(2*n)
AIC.sat <-  -2 * AIC.sat + (2*n)* 2
#
#
AICtotal<-c(AIC.sat, AIC);
BICtotal<-c(BIC.sat, BIC );
names(AICtotal)<-c('Saturated', 'BivPois')
names(BICtotal)<-c('Saturated', 'BivPois')
#  -------
#
#   spliting parameter vector
betaparameters<-splitbeta( beta )
#
#   putting all betas in one vector
allbeta<-c(betaparameters$beta1,betaparameters$beta2, beta3)
names(allbeta)<-c( paste( '(l1):', names(betaparameters$beta1),
sep='' ),paste('(l2):', names(betaparameters$beta2), sep='' ),
paste('(l3):', names(beta3), sep='' ) )
allparameters<-c(allbeta, prob, theta)
if (distribution==1){ names(allparameters)<-c( names(allbeta),
'p', paste('theta', 1:length(theta),sep='') ) }
else {names(allparameters)<-c( names(allbeta), 'p', 'theta') }
#
#   calculation of fitted values
#  ---------------------------
fittedval1<-(1-prob)*(m$fitted[1:n]         + lambda3)
fittedval2<-(1-prob)*(m$fitted[(n+1):(2*n)] + lambda3)
#
meandiag<-0
if ((distribution==1)&&(jmax>0)) { meandiag<-sum( theta[1:jmax]*1:jmax ) }
```

```
    else if (distribution==2) { meandiag<-theta }
    else if (distribution==3) { meandiag<- (1-theta)/theta }
    #
    fittedval1[x==y]<-prob*meandiag + fittedval1[x==y]
    fittedval2[x==y]<-prob*meandiag + fittedval2[x==y]
    #
    #
    #   saving output
    result<-list(coefficients=allparameters, fitted.values=
    data.frame(x=fittedval1,y=fittedval2), residuals=
    data.frame(x=x-fittedval1,y=y-fittedval2),
                beta1=betaparameters$beta1, beta2=betaparameters$beta2,
                beta3=beta3,  p=prob, theta=theta, diagonal.distribution=dilabel,
                lambda1=m$fitted[1:n], lambda2=m$fitted[(n+1):(2*n)],
                lambda3=lambda3, loglikelihood=loglike[1:i],
                parameters=noparams, AIC=AICtotal, BIC=BICtotal,
                iterations=i , call=tempcall)
    #
  } # end of elseif
  #
  options(warn=0)
  class(result)<-c('lm.dibp', 'lm')
  #
  result
  #
  #
}

"newnamesbeta" <-
  function( bvec ) {
    #
    # ------------------------------------------------------------------------
    # Karlis and Ntzoufras (2004)
    # EM algorithms for Bivariate Poisson Models
    # ------------------------------------------------------------------------
    #   Internal function for renaming parameters according to their
    interpretation
    #
    #   (c) June 2004 I. Ntzoufras, Chios, Greece
    #   (c) Revised on May 2005 by I. Ntzoufras, Athens, Greece
    # ------------------------------------------------------------------------
    #
    #   (l1): is parameter rererring to the log(lamdba1)
    #   (l2): is parameter rererring to the log(lamdba2)
    #   (l2-l1): this parameter is added to the current parameter of l1
    #   no label: the parameter is common for both log(lambda1) and log(lambda2)
    # ------------------------------------------------------------------------
```

```
    names(bvec)<-sub('\\)','',names(bvec))      #remove right parenthesis

    names(bvec)<-sub('\\(Intercept','(Intercept)',names(bvec))
    # replace "(Intercept" with "(Intercept)"
    names(bvec)[pmatch('internal.data1$noncommon2',
    names(bvec))]<-'(l2-l1):(Intercept)'
    # replace 'internal.data1$noncommon2' with 'l2-l1' for intercept
    names(bvec)<-sub('internal.data1\\$noncommon2:','(l2-l1):',names(bvec))
    # the same for the rest of parameters
    names(bvec)<-sub('internal.data1\\$noncommon0:','(l1):',names(bvec))
    # replace 'internal.data1\\$noncommon0:' by '(l1)'
    names(bvec)<-sub('internal.data1\\$noncommon1:','(l2):',names(bvec))
    # replace 'internal.data1\\$noncommon1:' by '(l2)'

    names(bvec)<-sub(':internal.data1\\$noncommon2','(l2-l1):',names(bvec))
    # same as above with ":" in front of expressions
    names(bvec)<-sub(':internal.data1\\$noncommon0','(l1):',names(bvec))
    names(bvec)<-sub(':internal.data1\\$noncommon1','(l2):',names(bvec))

    names(bvec)<-sub('I\\(internal.data1\\$indct1 \\* ','(l1):',names(bvec))
    # replace 'I(internal.data1$indct1 * ' with '(l1):'
    names(bvec)<-sub('I\\(internal.data1\\$indct2 \\* ','(l2):',names(bvec))
    # replace 'I(internal.data1$indct2 * ' with '(l2):'

    names(bvec)
  }


"pbivpois" <-
  function(x, y=NULL, lambda = c(1, 1, 1), log=FALSE) {
    # ------------------------------------------------------------------------
    # Karlis and Ntzoufras (2003, 2004)
    # EM algorithms for Bivariate Poisson Models
    # ------------------------------------------------------------------------
    # x      : matrix or vector of length n
    # y      : vector of length n. If x is matrix then it is not used
    # lambda : parameters of the bivariate poisson distribution
    # log    : argument controlling the calculation of the log-probability
    #          or the probability function.
    # ------------------------------------------------------------------------
    #
    if ( is.matrix(x) ) {
      var1<-x[,1]
      var2<-x[,2]
    }
    else if (is.vector(x)&is.vector(y)){
      if (length(x)==length(y)){
        var1<-x
```

```
      var2<-y
    }
    else{
      stop('lengths of x and y are not equal')
    }
  }
  else{
    stop('x is not a matrix or x and y are not vectors')
  }
  n <- length(var1)
  logbp<-vector(length=n)
  #
  for (k in 1:n){
    x0<-var1[k]
    y0<-var2[k]
    xymin<-min( x0,y0 )
    lambdaratio<-lambda[3]/(lambda[1]*lambda[2])
    #
    i<-0:xymin
    sums<- -lgamma(var1[k]-i+1)-lgamma(i+1)-lgamma(var2[k]-i+1)+i*
    log(lambdaratio)
    maxsums <- max(sums)
    sums<- sums - maxsums
    logsummation<- log( sum(exp(sums)) ) + maxsums
    logbp[k]<- -sum(lambda) + var1[k] * log( lambda[1] ) + var2[k] *
    log( lambda[2] ) + logsummation
  }
  if (log) { result<-    logbp }
  else     { result<-exp(logbp)  }
  result
  #   end of function bivpois
}

"simple.bp" <-
  function(x, y, ini3=1.0, maxit=300, pres=1e-8)
  {
    #
    # ------------------------------------------------------------------------
    # Karlis and Ntzoufras (2003, 2004)
    # (last revision 25/8/2005)
    # Athens University of Economics and Business
    #
    # EM algorithms for Bivariate Poisson Models
    # ------------------------------------------------------------------------
    #
    # x     : matrix or vector of length n
    # y     : vector of length n. If x is matrix then it is not used
    # ini3  : initial value for lambda3
```

```
# maxit : maximum number of iterations
# pres  : precision of the relative likelihood difference after which
          EM stops
# ------------------------------------------------------------------------
# Data length
#
#
if ( is.matrix(x) ) {
  var1<-x[,1]
  var2<-x[,2]
}
else if (is.vector(x)&is.vector(y)){
  if (length(x)==length(y)){
    var1<-x
    var2<-y
  }
  else{
    stop('lengths of x and y are not equal')
  }
}
else{
  stop('x is not a matrix or x and y are not vectors')
}

#
#
#
n<-length(var1)
#
# initial values
s<-rep(0,n)
like<-1:n*0
zero<- ( var1==0 )|( var2==0 )
#
#
#
# Initial values for lambda

lambda3<- ini3
lambda1<- max( 0.1, mean(var1)-lambda3 )
lambda2<- max( 0.1, mean(var2)-lambda3 )
#
#
difllike<-1000.0
loglike0<-1000.0
i<-0
loglike<-rep(0,maxit)
while ( (difllike>pres) && (i <= maxit) ) {
```

```
  i<-i+1
  #####   E step  ######
  for (j in 1:n) {
    if (zero[j]) {
      s[j]<-0;
      like[j]<- log(dpois(var1[j], lambda1)) + log(dpois(var2[j],
      lambda2))-lambda3;
    }
    else {
      lbp1<- pbivpois( var1[j]-1, var2[j]-1,
      lambda=c(lambda1,lambda2,lambda3), log=TRUE );
      lbp2<- pbivpois( var1[j]  , var2[j]  ,
      lambda=c(lambda1,lambda2,lambda3) , log=TRUE );

      s[j]<-exp( log(lambda3) + lbp1 - lbp2 );
      like[j]<-lbp2;
    }
  }
  ##### end of E step  ######
  x1<-var1-s
  x2<-var2-s
  loglike[i]<-sum(like)
  difllike<-abs( (loglike0-loglike[i])/loglike0 )
  loglike0<-loglike[i]
  #
  #
  #####   M step  ######
  #
  #     fit model on lambda3
  lambda1<-mean(x1)
  lambda2<-mean(x2)
  lambda3<-mean(s)
  #####   end of M step  ######
  printpars<-c(i,lambda1, lambda2, lambda3, loglike[i] )
  names(printpars)<-c('Iter.', 'lambda1', 'lambda2', 'lambda3','loglike' )
  print( round(printpars ,3 ) )
  cat( 'Relative Difference in Loglike:', difllike, '\n' )
}
#
#   calculation of BIC and AIC of Bivariate Poisson model
noparams<- 3
AIC<- -2*loglike[i] + noparams * 2
BIC<- -2*loglike[i] + noparams * log(2*n)
#
#
#   Calculation of BIC, AIC of Poisson saturated model
x.mean<-var1
x.mean[var1==0]<-1e-12
```

```
    y.mean<-var2
    y.mean[var2==0]<-1e-12
    AIC.sat <-  sum(log(dpois(var1,x.mean)) + log( dpois(var2,y.mean)))
    BIC.sat <-  -2 * AIC.sat + (2*n)* log(2*n)
    AIC.sat <-  -2 * AIC.sat + (2*n)* 2
    #
    #
    #   Calculation of BIC, AIC of simple Poisson model
    x.mean<-mean(var1)
    y.mean<-mean(var2)
    AIC.pois <-  sum(log( dpois(var1,x.mean)) + log(dpois(var2,y.mean)))
    BIC.pois <-  -2 * AIC.pois + 2* log(2*n)
    AIC.pois <-  -2 * AIC.pois + 2* 2

    AICtotal<-c(AIC.sat, AIC.pois, AIC)
    BICtotal<-c(BIC.sat, BIC.pois, BIC )

    names(AICtotal)<- c( 'Saturated', 'DblPois', 'BivPois' )
    names(BICtotal)<- c( 'Saturated', 'DblPois', 'BivPois' )
    #
    # Calculation of fitted values
    result<-list(lambda=c(lambda1, lambda2, lambda3),loglikelihood=loglike[1:i],
                 parameters=noparams, AIC=AICtotal, BIC=BICtotal ,iterations=i )
    #
    result
    #
    #
  }

"splitbeta" <-
  function( bvec ){
    #
    # ---------------------------------------------------------------------
    # Karlis and Ntzoufras (2004)
    # EM algorithms for Bivariate Poisson Models
    # ---------------------------------------------------------------------
    #   Internal function for spliting beta parameters according to their
    interpretation
    #
    #   (c) June 2004 I. Ntzoufras, Chios, Greece
    # ---------------------------------------------------------------------
    #
    p3<-length(bvec)

    indx1<-grep( '\\(l1\\):', names(bvec) ) # identify parameters for lambda1
    indx2<-grep( '\\(l2\\):', names(bvec) ) # identify parameters for lambda2
    indx3<-grep( '\\(l2-l1\\):', names(bvec) ) # identify difference
    parameters for lambda2
```

```
    #
    #   create temporary labels to identify common parameters
    tempnames<-sub( '\\(l2-l1)\\:', 'k', names(bvec)  )
    tempnames<-sub( '\\(l2)\\:', 'k', tempnames  )
    tempnames<-sub( '\\(l1)\\:', 'k', tempnames  )

    indx4<-tempnames%in%names(bvec) # common parameters are identified as TRUE
    #
    beta1<-c(bvec[indx4],bvec[indx1])
    beta2<-c(bvec[indx4],bvec[indx3],bvec[indx2])
    indexbeta2<-c( rep(0,sum(indx4)), rep(1,length(indx3)), rep(2,length(indx2)) )

    names(beta1)<-sub('\\(l1\\):','',names(beta1))
    names(beta2)<-sub('\\(l2\\):','',names(beta2))
    names(beta2)<-sub('\\(l2-l1\\):','',names(beta2))

    beta1<-beta1[order(names(beta1))]
    indexbeta2<-indexbeta2[order(names(beta2))]
    beta2<-beta2[order(names(beta2))]
    ii<-1:length(beta2)
    ii<-ii[indexbeta2==0]
    for ( i in ii ) {
      #     beta2[i]<-sum( beta2[ grep( names(beta2)[i], names(beta2) ) ] )
      beta2[i]<-sum( beta2[ names(beta2)[i]==names(beta2) ] )
    }
    beta2<-beta2[indexbeta2%in%c(0,2)]

    btemp<-list(beta1=beta1,beta2=beta2)
    btemp
  }


# ------------------------------------------
# COMMANDS FOR FITTING CONSTANT MODEL FOR EX1
# SECTION 4.1.1
# ------------------------------------------
#
# # library(bivpois)       # load bivpois library
# data(ex1.sim)          # load data of example 1
# # ---------------------------------------------------------------------------
#
# xtemp<-readline(prompt = "Press Enter to Continue")
#
# # Simple Bivariate Poisson Model
# ex1.simple<-simple.bp( ex1.sim$x, ex1.sim$y ) # fit simple model of
# section 4.1.1
#
# names(ex1.simple)      # monitor output variables
```

```
# ex1.simple$lambda      # view lambda1
# ex1.simple$BIC         # view BIC
# ex1.simple             # view all results of the model
# xtemp<-readline(prompt = "Press Enter to Continue")
# #
# # plot of loglikelihood vs. iterations
# win.graph()
# plot( 1:ex1.simple$iterations, ex1.simple$loglikelihood, xlab='Iterations',
# ylab='Log-likelihood', type='l' )
#
# # ------------------------------------
# # COMMANDS FOR FITTING MODELS FOR EX1
# # SECTION 4.1.2
# # ------------------------------------
# ex1.m2<-lm.bp(x~1 , y~1 , data=ex1.sim, zeroL3=TRUE)
# Model 2: DblPoisson(l1, l2)
# ex1.m3<-lm.bp(x~1 , y~1 , data=ex1.sim)
# Model 3: BivPoisson(l1, l2, l3)
# ex1.m4<-lm.bp(x~. , y~. , data=ex1.sim, zeroL3=TRUE)
# Model 4: DblPoisson (l1=Full, l2=Full)
# ex1.m5<-lm.bp(x~. , y~. , data=ex1.sim)
# Model 5: BivPoisson(l1=full, l2=full, l3=constant)
# ex1.m6<-lm.bp(x~z1 , y~z1+z5 , l1l2=~z3, data=ex1.sim, zeroL3=TRUE)
# Model 6: DblPois(l1,l2)
# ex1.m7<-lm.bp(x~z1 , y~z1+z5 , l1l2=~z3, data=ex1.sim)
# Model 7: BivPois(l1,l2,l3=constant)
# ex1.m8<-lm.bp(x~. , y~. , l3=~., data=ex1.sim)
# Model 8: BivPoisson(l1=full, l2=full, l3=full)
# ex1.m9<-lm.bp(x~. , y~. , l3=~.-z5, data=ex1.sim)
# Model 9: BivPoisson(l1=full, l2=full, l3=z1+z2+z3+z4)
# ex1.m10<-lm.bp(x~z1 , y~z1+z5 , l1l2=~z3, l3=~., data=ex1.sim)
# Model 10: BivPoisson(l1, l2, l3=full)
# ex1.m11<-lm.bp(x~z1 , y~z1+z5 , l1l2=~z3, l3=~.-z5, data=ex1.sim)
# Model 11: BivPoisson(l1, l2, l3= z1+z2+z3+z4)
# ex1.m11$coef  # monitor all beta parameters of model 11
# #
# ex1.m11$beta1 # monitor all beta parameters of lambda1 of model 11
# ex1.m11$beta2 # monitor all beta parameters of lambda2 of model 11
# ex1.m11$beta3 # monitor all beta parameters of lambda3 of model 11
```

Apéndice C

Apéndice: Depuración de los datos

Los datos que provienen de la base de datos de la DGT (referentes al número de fallecidos en vías urbanas e interurbanas por ciudad en cada año) fueron modificados con el mismo nombre de la provincia para poder hacer el estudio. La variable km_carretera ha sido creada a partir de los datos tomados del Ministerio de Fomento, además ha sido considerada constante a lo largo de los años para cada provincia. Las variables creadas a partir de los datos tomados del INE (precipitación, temperatura y horas de sol media) han sido consideradas como media de los datos tomados por cada mes en cada provincia. Además, se considera como constante la variación de los km de carretera en cada provincia por años. Por último, una vez añadidas las variables CCAA (comunidades autónomas) y poblacion (población de la provincia por año), se ha creado una tabla con todos los datos, pasando las variables a variables de tipo numérica, y se concluye generando la tabla (datos_f) con la que trabajamos que se encuentra en el archivo datos.RData.

```
library(readxl)
library(dplyr)
library(lmtest)
library(MASS)
library(sandwich)
library(AER)
library(car)
library(pscl)

t2016<-read_xlsx("TG16.xlsx")
t2016a<-t2016[-c(1,3),c(1,4,9)]
t2016b<-t2016a[-c(1),]
t2016b$year<-2016

t2016pv<-read_xlsx("PV16.xlsx")
t2016PV<-t2016pv[-c(1,3),]
t2016bb=cbind(t2016b,t2016PV[,c(9)])
t2016b=cbind(t2016bb,t2016PV[,c(10)])
t2016b=t2016b[-c(53),]


#datost<-rbind(t2002b,t2016b)
```

```
colnames(t2016b)<-c("provincia","fallecidos_interurbana","fallecidos_urbana",
"año","numero_vehiculos","poblacion")
nombresactuales<-unique(t2016b$provincia)
table(t2016b$provincia)
t2016b$numero_vehiculos[1]<-"207219"
t2016b$poblacion[1]<-324126



t2002<-read_xls("TG02.xls")
t2002a<-t2002[-c(1,3),c(1,10,15)]
t2002b<-t2002a[-c(1),]
t2002b$year<-2002

t2002pv<-read_xls("PV02.xls")
t2002PV<-t2002pv[-c(1,3),]
t2002bb=cbind(t2002b,t2002PV[,c(8)])
t2002b=cbind(t2002bb,t2002PV[,c(13)])
t2002b=t2002b[-c(53),]

nombresantiguos<-unique(t2002b$provincia)
colnames(t2002b)<-c("provincia","fallecidos_interurbana","fallecidos_urbana",
"año","numero_vehiculos","poblacion")

if(t2002b$numero_vehiculos[1]=="TOTAL") t2002b$numero_vehiculos[1]<-"174235"
t2002b$poblacion[1]<-288558

for(i in (1:length(t2002b$provincia)))
{cat(t2002b$provincia[i],"\n")
  if(t2002b$provincia[i]=="ÁLAVA") t2002b$provincia[i]<-"Araba/Álava"
  if(t2002b$provincia[i]=="ALBACETE") t2002b$provincia[i]<-"Albacete"
  if(t2002b$provincia[i]=="ALICANTE" ) t2002b$provincia[i]<-"Alicante/Alacant"
  if(t2002b$provincia[i]=="ALMERÍA" ) t2002b$provincia[i]<-"Almería"
  if(t2002b$provincia[i]=="ÁVILA" ) t2002b$provincia[i]<-"Ávila"
  if(t2002b$provincia[i]=="BADAJOZ" ) t2002b$provincia[i]<-"Badajoz"
  if(t2002b$provincia[i]=="ILLES BALEARS" ) t2002b$provincia[i]<-"Balears (Illes)"
  if(t2002b$provincia[i]=="BARCELONA" ) t2002b$provincia[i]<-"Barcelona"
  if(t2002b$provincia[i]=="BURGOS" ) t2002b$provincia[i]<-"Burgos"
  if(t2002b$provincia[i]=="CÁCERES" ) t2002b$provincia[i]<-"Cáceres"
  if(t2002b$provincia[i]=="CÁDIZ" ) t2002b$provincia[i]<-"Cádiz"
  if(t2002b$provincia[i]=="CASTELLÓN" ) t2002b$provincia[i]<-"Castellón/Castelló"
  if(t2002b$provincia[i]=="CIUDAD REAL" ) t2002b$provincia[i]<-"Ciudad Real"
  if(t2002b$provincia[i]=="CÓRDOBA" ) t2002b$provincia[i]<-"Córdoba"
  if(t2002b$provincia[i]=="A CORUÑA" ) t2002b$provincia[i]<-"Coruña (A)"
  if(t2002b$provincia[i]=="CUENCA" ) t2002b$provincia[i]<-"Cuenca"
  if(t2002b$provincia[i]=="GIRONA" ) t2002b$provincia[i]<-"Girona"
  if(t2002b$provincia[i]=="GRANADA" ) t2002b$provincia[i]<-"Granada"
  if(t2002b$provincia[i]=="GUADALAJARA" ) t2002b$provincia[i]<-"Guadalajara"
```

```
  if(t2002b$provincia[i]=="GUIPÚZCOA" ) t2002b$provincia[i]<-"Gipuzkoa"
  if(t2002b$provincia[i]=="HUELVA" ) t2002b$provincia[i]<-"Huelva"
  if(t2002b$provincia[i]=="HUESCA" ) t2002b$provincia[i]<-"Huesca"
  if(t2002b$provincia[i]=="JAÉN" ) t2002b$provincia[i]<-"Jaén"
  if(t2002b$provincia[i]=="LEÓN" ) t2002b$provincia[i]<-"León"
  if(t2002b$provincia[i]=="LLEIDA" ) t2002b$provincia[i]<-"Lleida"
  if(t2002b$provincia[i]=="LA RIOJA" ) t2002b$provincia[i]<-"Rioja (La)"
  if(t2002b$provincia[i]=="LUGO" ) t2002b$provincia[i]<-"Lugo"
  if(t2002b$provincia[i]=="MADRID" ) t2002b$provincia[i]<-"Madrid"
  if(t2002b$provincia[i]=="MÁLAGA" ) t2002b$provincia[i]<-"Málaga"
  if(t2002b$provincia[i]=="MURCIA" ) t2002b$provincia[i]<-"Murcia"
  if(t2002b$provincia[i]=="NAVARRA" ) t2002b$provincia[i]<-"Navarra"
  if(t2002b$provincia[i]=="OURENSE" ) t2002b$provincia[i]<-"Ourense"
  if(t2002b$provincia[i]=="ASTURIAS" ) t2002b$provincia[i]<-"Asturias"
  if(t2002b$provincia[i]=="PALENCIA" ) t2002b$provincia[i]<-"Palencia"
  if(t2002b$provincia[i]=="LAS PALMAS" ) t2002b$provincia[i]<-"Palmas (Las)"
  if(t2002b$provincia[i]=="PONTEVEDRA" ) t2002b$provincia[i]<-"Pontevedra"
  if(t2002b$provincia[i]=="SALAMANCA" ) t2002b$provincia[i]<-"Salamanca"
  if(t2002b$provincia[i]=="STA. C. TENERIFE" )
  t2002b$provincia[i]<-"Santa Cruz de Tenerife"
  if(t2002b$provincia[i]=="CANTABRIA" ) t2002b$provincia[i]<-"Cantabria"
  if(t2002b$provincia[i]=="SEGOVIA" ) t2002b$provincia[i]<-"Segovia"
  if(t2002b$provincia[i]=="SEVILLA" ) t2002b$provincia[i]<-"Sevilla"
  if(t2002b$provincia[i]=="SORIA" ) t2002b$provincia[i]<-"Soria"
  if(t2002b$provincia[i]=="TARRAGONA" ) t2002b$provincia[i]<-"Tarragona"
  if(t2002b$provincia[i]=="TERUEL" ) t2002b$provincia[i]<-"Teruel"
  if(t2002b$provincia[i]=="TOLEDO" ) t2002b$provincia[i]<-"Toledo"
  if(t2002b$provincia[i]=="VALENCIA" ) t2002b$provincia[i]<-"Valencia/València"
  if(t2002b$provincia[i]=="VALLADOLID" ) t2002b$provincia[i]<-"Valladolid"
  if(t2002b$provincia[i]=="VIZCAYA" ) t2002b$provincia[i]<-"Bizkaia"
  if(t2002b$provincia[i]=="ZAMORA" ) t2002b$provincia[i]<-"Zamora"
  if(t2002b$provincia[i]=="ZARAGOZA" ) t2002b$provincia[i]<-"Zaragoza"
  if(t2002b$provincia[i]=="CEUTA" ) t2002b$provincia[i]<-"Ceuta"
  if(t2002b$provincia[i]=="MELILLA" ) t2002b$provincia[i]<-"Melilla"
}

t2003<-read_xls("TG03.xls")
t2003a<-t2003[-c(1,3),c(1,10,15)]
t2003b<-t2003a[-c(1),]
t2003b$year<-2003

t2003pv<-read_xls("PV03.xls")
t2003PV<-t2003pv[-c(1,3),]
t2003bb=cbind(t2003b,t2003PV[,c(8)])
t2003b=cbind(t2003bb,t2003PV[,c(13)])
t2003b=t2003b[-c(53),]

nombresantiguos<-unique(t2003b$provincia)
```

```
colnames(t2003b)<-c("provincia","fallecidos_interurbana","fallecidos_urbana",
"año","numero_vehiculos","poblacion")

if(t2003b$numero_vehiculos[1]=="TOTAL") t2003b$numero_vehiculos[1]<-"163905"
t2003b$poblacion[1]<-292166

for(i in (1:length(t2003b$provincia)))
{cat(t2003b$provincia[i],"\n")
  if(t2003b$provincia[i]=="ÁLAVA") t2003b$provincia[i]<-"Araba/Álava"
  if(t2003b$provincia[i]=="ALBACETE") t2003b$provincia[i]<-"Albacete"
  if(t2003b$provincia[i]=="ALICANTE" ) t2003b$provincia[i]<-"Alicante/Alacant"
  if(t2003b$provincia[i]=="ALMERÍA" ) t2003b$provincia[i]<-"Almería"
  if(t2003b$provincia[i]=="ÁVILA" ) t2003b$provincia[i]<-"Ávila"
  if(t2003b$provincia[i]=="BADAJOZ" ) t2003b$provincia[i]<-"Badajoz"
  if(t2003b$provincia[i]=="ILLES BALEARS" ) t2003b$provincia[i]<-"Balears (Illes)"
  if(t2003b$provincia[i]=="BARCELONA" ) t2003b$provincia[i]<-"Barcelona"
  if(t2003b$provincia[i]=="BURGOS" ) t2003b$provincia[i]<-"Burgos"
  if(t2003b$provincia[i]=="CÁCERES" ) t2003b$provincia[i]<-"Cáceres"
  if(t2003b$provincia[i]=="CÁDIZ" ) t2003b$provincia[i]<-"Cádiz"
  if(t2003b$provincia[i]=="CASTELLÓN" ) t2003b$provincia[i]<-"Castellón/Castelló"
  if(t2003b$provincia[i]=="CIUDAD REAL" ) t2003b$provincia[i]<-"Ciudad Real"
  if(t2003b$provincia[i]=="CÓRDOBA" ) t2003b$provincia[i]<-"Córdoba"
  if(t2003b$provincia[i]=="A CORUÑA" ) t2003b$provincia[i]<-"Coruña (A)"
  if(t2003b$provincia[i]=="CUENCA" ) t2003b$provincia[i]<-"Cuenca"
  if(t2003b$provincia[i]=="GIRONA" ) t2003b$provincia[i]<-"Girona"
  if(t2003b$provincia[i]=="GRANADA" ) t2003b$provincia[i]<-"Granada"
  if(t2003b$provincia[i]=="GUADALAJARA" ) t2003b$provincia[i]<-"Guadalajara"
  if(t2003b$provincia[i]=="GUIPÚZCOA" ) t2003b$provincia[i]<-"Gipuzkoa"
  if(t2003b$provincia[i]=="HUELVA" ) t2003b$provincia[i]<-"Huelva"
  if(t2003b$provincia[i]=="HUESCA" ) t2003b$provincia[i]<-"Huesca"
  if(t2003b$provincia[i]=="JAÉN" ) t2003b$provincia[i]<-"Jaén"
  if(t2003b$provincia[i]=="LEÓN" ) t2003b$provincia[i]<-"León"
  if(t2003b$provincia[i]=="LLEIDA" ) t2003b$provincia[i]<-"Lleida"
  if(t2003b$provincia[i]=="LA RIOJA" ) t2003b$provincia[i]<-"Rioja (La)"
  if(t2003b$provincia[i]=="LUGO" ) t2003b$provincia[i]<-"Lugo"
  if(t2003b$provincia[i]=="MADRID" ) t2003b$provincia[i]<-"Madrid"
  if(t2003b$provincia[i]=="MÁLAGA" ) t2003b$provincia[i]<-"Málaga"
  if(t2003b$provincia[i]=="MURCIA" ) t2003b$provincia[i]<-"Murcia"
  if(t2003b$provincia[i]=="NAVARRA" ) t2003b$provincia[i]<-"Navarra"
  if(t2003b$provincia[i]=="OURENSE" ) t2003b$provincia[i]<-"Ourense"
  if(t2003b$provincia[i]=="ASTURIAS" ) t2003b$provincia[i]<-"Asturias"
  if(t2003b$provincia[i]=="PALENCIA" ) t2003b$provincia[i]<-"Palencia"
  if(t2003b$provincia[i]=="LAS PALMAS" ) t2003b$provincia[i]<-"Palmas (Las)"
  if(t2003b$provincia[i]=="PONTEVEDRA" ) t2003b$provincia[i]<-"Pontevedra"
  if(t2003b$provincia[i]=="SALAMANCA" ) t2003b$provincia[i]<-"Salamanca"
  if(t2003b$provincia[i]=="STA. C. TENERIFE" )
  t2003b$provincia[i]<-"Santa Cruz de Tenerife"
  if(t2003b$provincia[i]=="CANTABRIA" ) t2003b$provincia[i]<-"Cantabria"
```

```
  if(t2003b$provincia[i]=="SEGOVIA" ) t2003b$provincia[i]<-"Segovia"
  if(t2003b$provincia[i]=="SEVILLA" ) t2003b$provincia[i]<-"Sevilla"
  if(t2003b$provincia[i]=="SORIA" ) t2003b$provincia[i]<-"Soria"
  if(t2003b$provincia[i]=="TARRAGONA" ) t2003b$provincia[i]<-"Tarragona"
  if(t2003b$provincia[i]=="TERUEL" ) t2003b$provincia[i]<-"Teruel"
  if(t2003b$provincia[i]=="TOLEDO" ) t2003b$provincia[i]<-"Toledo"
  if(t2003b$provincia[i]=="VALENCIA" ) t2003b$provincia[i]<-"Valencia/València"
  if(t2003b$provincia[i]=="VALLADOLID" ) t2003b$provincia[i]<-"Valladolid"
  if(t2003b$provincia[i]=="VIZCAYA" ) t2003b$provincia[i]<-"Bizkaia"
  if(t2003b$provincia[i]=="ZAMORA" ) t2003b$provincia[i]<-"Zamora"
  if(t2003b$provincia[i]=="ZARAGOZA" ) t2003b$provincia[i]<-"Zaragoza"
  if(t2003b$provincia[i]=="CEUTA" ) t2003b$provincia[i]<-"Ceuta"
  if(t2003b$provincia[i]=="MELILLA" ) t2003b$provincia[i]<-"Melilla"
}

t2004<-read_xls("TG04.xls")
t2004a<-t2004[-c(1,3),c(1,10,15)]
t2004b<-t2004a[-c(1),]
t2004b$year<-2004

t2004pv<-read_xlsx("PV04.xlsx")
t2004PV<-t2004pv[-c(1,3),]
t2004bb=cbind(t2004b,t2004PV[,c(8)])
t2004b=cbind(t2004bb,t2004PV[,c(13)])
t2004b=t2004b[-c(53),]

nombresantiguos<-unique(t2004b$provincia)
colnames(t2004b)<-c("provincia","fallecidos_interurbana","fallecidos_urbana",
"año","numero_vehiculos","poblacion")

if(t2004b$numero_vehiculos[1]=="TOTAL") t2004b$numero_vehiculos[1]<-"170333"
t2004b$poblacion[1]<-295699

for(i in (1:length(t2004b$provincia)))
{cat(t2004b$provincia[i],"\n")
  if(t2004b$provincia[i]=="ÁLAVA") t2004b$provincia[i]<-"Araba/Álava"
  if(t2004b$provincia[i]=="ALBACETE") t2004b$provincia[i]<-"Albacete"
  if(t2004b$provincia[i]=="ALICANTE" ) t2004b$provincia[i]<-"Alicante/Alacant"
  if(t2004b$provincia[i]=="ALMERÍA" ) t2004b$provincia[i]<-"Almería"
  if(t2004b$provincia[i]=="ÁVILA" ) t2004b$provincia[i]<-"Ávila"
  if(t2004b$provincia[i]=="BADAJOZ" ) t2004b$provincia[i]<-"Badajoz"
  if(t2004b$provincia[i]=="ILLES BALEARS" ) t2004b$provincia[i]<-"Balears (Illes)"
  if(t2004b$provincia[i]=="BARCELONA" ) t2004b$provincia[i]<-"Barcelona"
  if(t2004b$provincia[i]=="BURGOS" ) t2004b$provincia[i]<-"Burgos"
  if(t2004b$provincia[i]=="CÁCERES" ) t2004b$provincia[i]<-"Cáceres"
  if(t2004b$provincia[i]=="CÁDIZ" ) t2004b$provincia[i]<-"Cádiz"
  if(t2004b$provincia[i]=="CASTELLÓN" ) t2004b$provincia[i]<-"Castellón/Castelló"
  if(t2004b$provincia[i]=="CIUDAD REAL" ) t2004b$provincia[i]<-"Ciudad Real"
```

```
  if(t2004b$provincia[i]=="CÓRDOBA" ) t2004b$provincia[i]<-"Córdoba"
  if(t2004b$provincia[i]=="A CORUÑA" ) t2004b$provincia[i]<-"Coruña (A)"
  if(t2004b$provincia[i]=="CUENCA" ) t2004b$provincia[i]<-"Cuenca"
  if(t2004b$provincia[i]=="GIRONA" ) t2004b$provincia[i]<-"Girona"
  if(t2004b$provincia[i]=="GRANADA" ) t2004b$provincia[i]<-"Granada"
  if(t2004b$provincia[i]=="GUADALAJARA" ) t2004b$provincia[i]<-"Guadalajara"
  if(t2004b$provincia[i]=="GUIPÚZCOA" ) t2004b$provincia[i]<-"Gipuzkoa"
  if(t2004b$provincia[i]=="HUELVA" ) t2004b$provincia[i]<-"Huelva"
  if(t2004b$provincia[i]=="HUESCA" ) t2004b$provincia[i]<-"Huesca"
  if(t2004b$provincia[i]=="JAÉN" ) t2004b$provincia[i]<-"Jaén"
  if(t2004b$provincia[i]=="LEÓN" ) t2004b$provincia[i]<-"León"
  if(t2004b$provincia[i]=="LLEIDA" ) t2004b$provincia[i]<-"Lleida"
  if(t2004b$provincia[i]=="LA RIOJA" ) t2004b$provincia[i]<-"Rioja (La)"
  if(t2004b$provincia[i]=="LUGO" ) t2004b$provincia[i]<-"Lugo"
  if(t2004b$provincia[i]=="MADRID" ) t2004b$provincia[i]<-"Madrid"
  if(t2004b$provincia[i]=="MÁLAGA" ) t2004b$provincia[i]<-"Málaga"
  if(t2004b$provincia[i]=="MURCIA" ) t2004b$provincia[i]<-"Murcia"
  if(t2004b$provincia[i]=="NAVARRA" ) t2004b$provincia[i]<-"Navarra"
  if(t2004b$provincia[i]=="OURENSE" ) t2004b$provincia[i]<-"Ourense"
  if(t2004b$provincia[i]=="ASTURIAS" ) t2004b$provincia[i]<-"Asturias"
  if(t2004b$provincia[i]=="PALENCIA" ) t2004b$provincia[i]<-"Palencia"
  if(t2004b$provincia[i]=="LAS PALMAS" ) t2004b$provincia[i]<-"Palmas (Las)"
  if(t2004b$provincia[i]=="PONTEVEDRA" ) t2004b$provincia[i]<-"Pontevedra"
  if(t2004b$provincia[i]=="SALAMANCA" ) t2004b$provincia[i]<-"Salamanca"
  if(t2004b$provincia[i]=="STA. C. TENERIFE" )
  t2004b$provincia[i]<-"Santa Cruz de Tenerife"
  if(t2004b$provincia[i]=="CANTABRIA" ) t2004b$provincia[i]<-"Cantabria"
  if(t2004b$provincia[i]=="SEGOVIA" ) t2004b$provincia[i]<-"Segovia"
  if(t2004b$provincia[i]=="SEVILLA" ) t2004b$provincia[i]<-"Sevilla"
  if(t2004b$provincia[i]=="SORIA" ) t2004b$provincia[i]<-"Soria"
  if(t2004b$provincia[i]=="TARRAGONA" ) t2004b$provincia[i]<-"Tarragona"
  if(t2004b$provincia[i]=="TERUEL" ) t2004b$provincia[i]<-"Teruel"
  if(t2004b$provincia[i]=="TOLEDO" ) t2004b$provincia[i]<-"Toledo"
  if(t2004b$provincia[i]=="VALENCIA" ) t2004b$provincia[i]<-"Valencia/València"
  if(t2004b$provincia[i]=="VALLADOLID" ) t2004b$provincia[i]<-"Valladolid"
  if(t2004b$provincia[i]=="VIZCAYA" ) t2004b$provincia[i]<-"Bizkaia"
  if(t2004b$provincia[i]=="ZAMORA" ) t2004b$provincia[i]<-"Zamora"
  if(t2004b$provincia[i]=="ZARAGOZA" ) t2004b$provincia[i]<-"Zaragoza"
  if(t2004b$provincia[i]=="CEUTA" ) t2004b$provincia[i]<-"Ceuta"
  if(t2004b$provincia[i]=="MELILLA" ) t2004b$provincia[i]<-"Melilla"
}

t2005<-read_xls("TG05.xls")
t2005a<-t2005[-c(1,3),c(1,10,15)]
t2005b<-t2005a[-c(1),]
t2005b$year<-2005

t2005pv<-read_xlsx("PV05.xlsx")
```

```
t2005PV<-t2005pv[-c(1,3),]
t2005bb=cbind(t2005b,t2005PV[,c(8)])
t2005b=cbind(t2005bb,t2005PV[,c(13)])
t2005b=t2005b[-c(53),]

nombresantiguos<-unique(t2005b$provincia)
colnames(t2005b)<-c("provincia","fallecidos_interurbana","fallecidos_urbana",
"año","numero_vehiculos","poblacion")

t2005b$numero_vehiculos[1]<-"176204"
t2005b$poblacion[1]<-297367

for(i in (1:length(t2005b$provincia)))
{cat(t2005b$provincia[i],"\n")
  if(t2005b$provincia[i]=="ÁLAVA") t2005b$provincia[i]<-"Araba/Álava"
  if(t2005b$provincia[i]=="ALBACETE") t2005b$provincia[i]<-"Albacete"
  if(t2005b$provincia[i]=="ALICANTE" ) t2005b$provincia[i]<-"Alicante/Alacant"
  if(t2005b$provincia[i]=="ALMERÍA" ) t2005b$provincia[i]<-"Almería"
  if(t2005b$provincia[i]=="ÁVILA" ) t2005b$provincia[i]<-"Ávila"
  if(t2005b$provincia[i]=="BADAJOZ" ) t2005b$provincia[i]<-"Badajoz"
  if(t2005b$provincia[i]=="ILLES BALEARS" ) t2005b$provincia[i]<-"Balears (Illes)"
  if(t2005b$provincia[i]=="BARCELONA" ) t2005b$provincia[i]<-"Barcelona"
  if(t2005b$provincia[i]=="BURGOS" ) t2005b$provincia[i]<-"Burgos"
  if(t2005b$provincia[i]=="CÁCERES" ) t2005b$provincia[i]<-"Cáceres"
  if(t2005b$provincia[i]=="CÁDIZ" ) t2005b$provincia[i]<-"Cádiz"
  if(t2005b$provincia[i]=="CASTELLÓN" ) t2005b$provincia[i]<-"Castellón/Castelló"
  if(t2005b$provincia[i]=="CIUDAD REAL" ) t2005b$provincia[i]<-"Ciudad Real"
  if(t2005b$provincia[i]=="CÓRDOBA" ) t2005b$provincia[i]<-"Córdoba"
  if(t2005b$provincia[i]=="A CORUÑA" ) t2005b$provincia[i]<-"Coruña (A)"
  if(t2005b$provincia[i]=="CUENCA" ) t2005b$provincia[i]<-"Cuenca"
  if(t2005b$provincia[i]=="GIRONA" ) t2005b$provincia[i]<-"Girona"
  if(t2005b$provincia[i]=="GRANADA" ) t2005b$provincia[i]<-"Granada"
  if(t2005b$provincia[i]=="GUADALAJARA" ) t2005b$provincia[i]<-"Guadalajara"
  if(t2005b$provincia[i]=="GUIPÚZCOA" ) t2005b$provincia[i]<-"Gipuzkoa"
  if(t2005b$provincia[i]=="HUELVA" ) t2005b$provincia[i]<-"Huelva"
  if(t2005b$provincia[i]=="HUESCA" ) t2005b$provincia[i]<-"Huesca"
  if(t2005b$provincia[i]=="JAÉN" ) t2005b$provincia[i]<-"Jaén"
  if(t2005b$provincia[i]=="LEÓN" ) t2005b$provincia[i]<-"León"
  if(t2005b$provincia[i]=="LLEIDA" ) t2005b$provincia[i]<-"Lleida"
  if(t2005b$provincia[i]=="LA RIOJA" ) t2005b$provincia[i]<-"Rioja (La)"
  if(t2005b$provincia[i]=="LUGO" ) t2005b$provincia[i]<-"Lugo"
  if(t2005b$provincia[i]=="MADRID" ) t2005b$provincia[i]<-"Madrid"
  if(t2005b$provincia[i]=="MÁLAGA" ) t2005b$provincia[i]<-"Málaga"
  if(t2005b$provincia[i]=="MURCIA" ) t2005b$provincia[i]<-"Murcia"
  if(t2005b$provincia[i]=="NAVARRA" ) t2005b$provincia[i]<-"Navarra"
  if(t2005b$provincia[i]=="OURENSE" ) t2005b$provincia[i]<-"Ourense"
  if(t2005b$provincia[i]=="ASTURIAS" ) t2005b$provincia[i]<-"Asturias"
  if(t2005b$provincia[i]=="PALENCIA" ) t2005b$provincia[i]<-"Palencia"
```

```
  if(t2005b$provincia[i]=="LAS PALMAS" ) t2005b$provincia[i]<-"Palmas (Las)"
  if(t2005b$provincia[i]=="PONTEVEDRA" ) t2005b$provincia[i]<-"Pontevedra"
  if(t2005b$provincia[i]=="SALAMANCA" ) t2005b$provincia[i]<-"Salamanca"
  if(t2005b$provincia[i]=="STA. C. TENERIFE" )
  t2005b$provincia[i]<-"Santa Cruz de Tenerife"
  if(t2005b$provincia[i]=="CANTABRIA" ) t2005b$provincia[i]<-"Cantabria"
  if(t2005b$provincia[i]=="SEGOVIA" ) t2005b$provincia[i]<-"Segovia"
  if(t2005b$provincia[i]=="SEVILLA" ) t2005b$provincia[i]<-"Sevilla"
  if(t2005b$provincia[i]=="SORIA" ) t2005b$provincia[i]<-"Soria"
  if(t2005b$provincia[i]=="TARRAGONA" ) t2005b$provincia[i]<-"Tarragona"
  if(t2005b$provincia[i]=="TERUEL" ) t2005b$provincia[i]<-"Teruel"
  if(t2005b$provincia[i]=="TOLEDO" ) t2005b$provincia[i]<-"Toledo"
  if(t2005b$provincia[i]=="VALENCIA" ) t2005b$provincia[i]<-"Valencia/València"
  if(t2005b$provincia[i]=="VALLADOLID" ) t2005b$provincia[i]<-"Valladolid"
  if(t2005b$provincia[i]=="VIZCAYA" ) t2005b$provincia[i]<-"Bizkaia"
  if(t2005b$provincia[i]=="ZAMORA" ) t2005b$provincia[i]<-"Zamora"
  if(t2005b$provincia[i]=="ZARAGOZA" ) t2005b$provincia[i]<-"Zaragoza"
  if(t2005b$provincia[i]=="CEUTA" ) t2005b$provincia[i]<-"Ceuta"
  if(t2005b$provincia[i]=="MELILLA" ) t2005b$provincia[i]<-"Melilla"
}

t2006<-read_xls("TG06.xls")
t2006a<-t2006[-c(1,3),c(1,10,15)]
t2006b<-t2006a[-c(1),]
t2006b$year<-2006

t2006pv<-read_xlsx("PV06.xlsx")
t2006PV<-t2006pv[-c(1,3),]
t2006bb=cbind(t2006b,t2006PV[,c(8)])
t2006b=cbind(t2006bb,t2006PV[,c(13)])
t2006b=t2006b[-c(53),]

nombresantiguos<-unique(t2006b$provincia)
colnames(t2006b)<-c("provincia","fallecidos_interurbana","fallecidos_urbana",
"año","numero_vehiculos","poblacion")

t2006b$numero_vehiculos[1]<-"181038"
t2006b$poblacion[1]<-299921

for(i in (1:length(t2006b$provincia)))
{cat(t2006b$provincia[i],"\n")
  if(t2006b$provincia[i]=="ÁLAVA") t2006b$provincia[i]<-"Araba/Álava"
  if(t2006b$provincia[i]=="ALBACETE") t2006b$provincia[i]<-"Albacete"
  if(t2006b$provincia[i]=="ALICANTE" ) t2006b$provincia[i]<-"Alicante/Alacant"
  if(t2006b$provincia[i]=="ALMERÍA" ) t2006b$provincia[i]<-"Almería"
  if(t2006b$provincia[i]=="ÁVILA" ) t2006b$provincia[i]<-"Ávila"
  if(t2006b$provincia[i]=="BADAJOZ" ) t2006b$provincia[i]<-"Badajoz"
  if(t2006b$provincia[i]=="ILLES BALEARS" ) t2006b$provincia[i]<-"Balears (Illes)"
```

```
  if(t2006b$provincia[i]=="BARCELONA" ) t2006b$provincia[i]<-"Barcelona"
  if(t2006b$provincia[i]=="BURGOS" ) t2006b$provincia[i]<-"Burgos"
  if(t2006b$provincia[i]=="CÁCERES" ) t2006b$provincia[i]<-"Cáceres"
  if(t2006b$provincia[i]=="CÁDIZ" ) t2006b$provincia[i]<-"Cádiz"
  if(t2006b$provincia[i]=="CASTELLÓN" ) t2006b$provincia[i]<-"Castellón/Castelló"
  if(t2006b$provincia[i]=="CIUDAD REAL" ) t2006b$provincia[i]<-"Ciudad Real"
  if(t2006b$provincia[i]=="CÓRDOBA" ) t2006b$provincia[i]<-"Córdoba"
  if(t2006b$provincia[i]=="A CORUÑA" ) t2006b$provincia[i]<-"Coruña (A)"
  if(t2006b$provincia[i]=="CUENCA" ) t2006b$provincia[i]<-"Cuenca"
  if(t2006b$provincia[i]=="GIRONA" ) t2006b$provincia[i]<-"Girona"
  if(t2006b$provincia[i]=="GRANADA" ) t2006b$provincia[i]<-"Granada"
  if(t2006b$provincia[i]=="GUADALAJARA" ) t2006b$provincia[i]<-"Guadalajara"
  if(t2006b$provincia[i]=="GUIPÚZCOA" ) t2006b$provincia[i]<-"Gipuzkoa"
  if(t2006b$provincia[i]=="HUELVA" ) t2006b$provincia[i]<-"Huelva"
  if(t2006b$provincia[i]=="HUESCA" ) t2006b$provincia[i]<-"Huesca"
  if(t2006b$provincia[i]=="JAÉN" ) t2006b$provincia[i]<-"Jaén"
  if(t2006b$provincia[i]=="LEÓN" ) t2006b$provincia[i]<-"León"
  if(t2006b$provincia[i]=="LLEIDA" ) t2006b$provincia[i]<-"Lleida"
  if(t2006b$provincia[i]=="LA RIOJA" ) t2006b$provincia[i]<-"Rioja (La)"
  if(t2006b$provincia[i]=="LUGO" ) t2006b$provincia[i]<-"Lugo"
  if(t2006b$provincia[i]=="MADRID" ) t2006b$provincia[i]<-"Madrid"
  if(t2006b$provincia[i]=="MÁLAGA" ) t2006b$provincia[i]<-"Málaga"
  if(t2006b$provincia[i]=="MURCIA" ) t2006b$provincia[i]<-"Murcia"
  if(t2006b$provincia[i]=="NAVARRA" ) t2006b$provincia[i]<-"Navarra"
  if(t2006b$provincia[i]=="OURENSE" ) t2006b$provincia[i]<-"Ourense"
  if(t2006b$provincia[i]=="ASTURIAS" ) t2006b$provincia[i]<-"Asturias"
  if(t2006b$provincia[i]=="PALENCIA" ) t2006b$provincia[i]<-"Palencia"
  if(t2006b$provincia[i]=="LAS PALMAS" ) t2006b$provincia[i]<-"Palmas (Las)"
  if(t2006b$provincia[i]=="PONTEVEDRA" ) t2006b$provincia[i]<-"Pontevedra"
  if(t2006b$provincia[i]=="SALAMANCA" ) t2006b$provincia[i]<-"Salamanca"
  if(t2006b$provincia[i]=="STA. C. TENERIFE" )
  t2006b$provincia[i]<-"Santa Cruz de Tenerife"
  if(t2006b$provincia[i]=="CANTABRIA" ) t2006b$provincia[i]<-"Cantabria"
  if(t2006b$provincia[i]=="SEGOVIA" ) t2006b$provincia[i]<-"Segovia"
  if(t2006b$provincia[i]=="SEVILLA" ) t2006b$provincia[i]<-"Sevilla"
  if(t2006b$provincia[i]=="SORIA" ) t2006b$provincia[i]<-"Soria"
  if(t2006b$provincia[i]=="TARRAGONA" ) t2006b$provincia[i]<-"Tarragona"
  if(t2006b$provincia[i]=="TERUEL" ) t2006b$provincia[i]<-"Teruel"
  if(t2006b$provincia[i]=="TOLEDO" ) t2006b$provincia[i]<-"Toledo"
  if(t2006b$provincia[i]=="VALENCIA" ) t2006b$provincia[i]<-"Valencia/València"
  if(t2006b$provincia[i]=="VALLADOLID" ) t2006b$provincia[i]<-"Valladolid"
  if(t2006b$provincia[i]=="VIZCAYA" ) t2006b$provincia[i]<-"Bizkaia"
  if(t2006b$provincia[i]=="ZAMORA" ) t2006b$provincia[i]<-"Zamora"
  if(t2006b$provincia[i]=="ZARAGOZA" ) t2006b$provincia[i]<-"Zaragoza"
  if(t2006b$provincia[i]=="CEUTA" ) t2006b$provincia[i]<-"Ceuta"
  if(t2006b$provincia[i]=="MELILLA" ) t2006b$provincia[i]<-"Melilla"
}
```

```
t2007<-read_xls("TG07.xls")
t2007a<-t2007[-c(1,3),c(1,10,15)]
t2007b<-t2007a[-c(1),]
t2007b$year<-2007

t2007pv<-read_xls("PV07.xls")
t2007PV<-t2007pv[-c(1,3),]
t2007bb=cbind(t2007b,t2007PV[,c(8)])
t2007b=cbind(t2007bb,t2007PV[,c(13)])
t2007b=t2007b[-c(53),]

nombresantiguos<-unique(t2007b$provincia)
colnames(t2007b)<-c("provincia","fallecidos_interurbana","fallecidos_urbana",
"año","numero_vehiculos","poblacion")

if(t2007b$numero_vehiculos[1]=="TOTAL") t2007b$numero_vehiculos[1]<-"191501"
t2007b$poblacion[1]<-301548

for(i in (1:length(t2007b$provincia)))
{cat(t2007b$provincia[i],"\n")
  if(t2007b$provincia[i]=="ÁLAVA") t2007b$provincia[i]<-"Araba/Álava"
  if(t2007b$provincia[i]=="ALBACETE") t2007b$provincia[i]<-"Albacete"
  if(t2007b$provincia[i]=="ALICANTE" ) t2007b$provincia[i]<-"Alicante/Alacant"
  if(t2007b$provincia[i]=="ALMERÍA" ) t2007b$provincia[i]<-"Almería"
  if(t2007b$provincia[i]=="ÁVILA" ) t2007b$provincia[i]<-"Ávila"
  if(t2007b$provincia[i]=="BADAJOZ" ) t2007b$provincia[i]<-"Badajoz"
  if(t2007b$provincia[i]=="ILLES BALEARS" ) t2007b$provincia[i]<-"Balears (Illes)"
  if(t2007b$provincia[i]=="BARCELONA" ) t2007b$provincia[i]<-"Barcelona"
  if(t2007b$provincia[i]=="BURGOS" ) t2007b$provincia[i]<-"Burgos"
  if(t2007b$provincia[i]=="CÁCERES" ) t2007b$provincia[i]<-"Cáceres"
  if(t2007b$provincia[i]=="CÁDIZ" ) t2007b$provincia[i]<-"Cádiz"
  if(t2007b$provincia[i]=="CASTELLÓN" ) t2007b$provincia[i]<-"Castellón/Castelló"
  if(t2007b$provincia[i]=="CIUDAD REAL" ) t2007b$provincia[i]<-"Ciudad Real"
  if(t2007b$provincia[i]=="CÓRDOBA" ) t2007b$provincia[i]<-"Córdoba"
  if(t2007b$provincia[i]=="A CORUÑA" ) t2007b$provincia[i]<-"Coruña (A)"
  if(t2007b$provincia[i]=="CUENCA" ) t2007b$provincia[i]<-"Cuenca"
  if(t2007b$provincia[i]=="GIRONA" ) t2007b$provincia[i]<-"Girona"
  if(t2007b$provincia[i]=="GRANADA" ) t2007b$provincia[i]<-"Granada"
  if(t2007b$provincia[i]=="GUADALAJARA" ) t2007b$provincia[i]<-"Guadalajara"
  if(t2007b$provincia[i]=="GUIPÚZCOA" ) t2007b$provincia[i]<-"Gipuzkoa"
  if(t2007b$provincia[i]=="HUELVA" ) t2007b$provincia[i]<-"Huelva"
  if(t2007b$provincia[i]=="HUESCA" ) t2007b$provincia[i]<-"Huesca"
  if(t2007b$provincia[i]=="JAÉN" ) t2007b$provincia[i]<-"Jaén"
  if(t2007b$provincia[i]=="LEÓN" ) t2007b$provincia[i]<-"León"
  if(t2007b$provincia[i]=="LLEIDA" ) t2007b$provincia[i]<-"Lleida"
  if(t2007b$provincia[i]=="LA RIOJA" ) t2007b$provincia[i]<-"Rioja (La)"
  if(t2007b$provincia[i]=="LUGO" ) t2007b$provincia[i]<-"Lugo"
  if(t2007b$provincia[i]=="MADRID" ) t2007b$provincia[i]<-"Madrid"
```

```
  if(t2007b$provincia[i]=="MÁLAGA" ) t2007b$provincia[i]<-"Málaga"
  if(t2007b$provincia[i]=="MURCIA" ) t2007b$provincia[i]<-"Murcia"
  if(t2007b$provincia[i]=="NAVARRA" ) t2007b$provincia[i]<-"Navarra"
  if(t2007b$provincia[i]=="OURENSE" ) t2007b$provincia[i]<-"Ourense"
  if(t2007b$provincia[i]=="ASTURIAS" ) t2007b$provincia[i]<-"Asturias"
  if(t2007b$provincia[i]=="PALENCIA" ) t2007b$provincia[i]<-"Palencia"
  if(t2007b$provincia[i]=="LAS PALMAS" ) t2007b$provincia[i]<-"Palmas (Las)"
  if(t2007b$provincia[i]=="PONTEVEDRA" ) t2007b$provincia[i]<-"Pontevedra"
  if(t2007b$provincia[i]=="SALAMANCA" ) t2007b$provincia[i]<-"Salamanca"
  if(t2007b$provincia[i]=="STA. C. TENERIFE" )
  t2007b$provincia[i]<-"Santa Cruz de Tenerife"
  if(t2007b$provincia[i]=="CANTABRIA" ) t2007b$provincia[i]<-"Cantabria"
  if(t2007b$provincia[i]=="SEGOVIA" ) t2007b$provincia[i]<-"Segovia"
  if(t2007b$provincia[i]=="SEVILLA" ) t2007b$provincia[i]<-"Sevilla"
  if(t2007b$provincia[i]=="SORIA" ) t2007b$provincia[i]<-"Soria"
  if(t2007b$provincia[i]=="TARRAGONA" ) t2007b$provincia[i]<-"Tarragona"
  if(t2007b$provincia[i]=="TERUEL" ) t2007b$provincia[i]<-"Teruel"
  if(t2007b$provincia[i]=="TOLEDO" ) t2007b$provincia[i]<-"Toledo"
  if(t2007b$provincia[i]=="VALENCIA" ) t2007b$provincia[i]<-"Valencia/València"
  if(t2007b$provincia[i]=="VALLADOLID" ) t2007b$provincia[i]<-"Valladolid"
  if(t2007b$provincia[i]=="VIZCAYA" ) t2007b$provincia[i]<-"Bizkaia"
  if(t2007b$provincia[i]=="ZAMORA" ) t2007b$provincia[i]<-"Zamora"
  if(t2007b$provincia[i]=="ZARAGOZA" ) t2007b$provincia[i]<-"Zaragoza"
  if(t2007b$provincia[i]=="CEUTA" ) t2007b$provincia[i]<-"Ceuta"
  if(t2007b$provincia[i]=="MELILLA" ) t2007b$provincia[i]<-"Melilla"
}

t2008<-read_xls("TG08.xls")
t2008a<-t2008[-c(1,3),c(1,10,15)]
t2008b<-t2008a[-c(1),]
t2008b$year<-2008

t2008pv<-read_xls("PV08.xls")
t2008PV<-t2008pv[-c(1,3),]
t2008bb=cbind(t2008b,t2008PV[-c(54,55),c(8)])
t2008b=cbind(t2008bb,t2008PV[-c(54,55),c(13)])
t2008b=t2008b[-c(53),]

nombresantiguos<-unique(t2008b$provincia)
colnames(t2008b)<-c("provincia","fallecidos_interurbana","fallecidos_urbana",
"año","numero_vehiculos","poblacion")

if(t2008b$numero_vehiculos[1]=="TOTAL") t2008b$numero_vehiculos[1]<-"196087"
t2008b$poblacion[1]<-306527

for(i in (1:length(t2008b$provincia)))
{cat(t2008b$provincia[i],"\n")
  if(t2008b$provincia[i]=="ÁLAVA") t2008b$provincia[i]<-"Araba/Álava"
```

```
if(t2008b$provincia[i]=="ALBACETE") t2008b$provincia[i]<-"Albacete"
if(t2008b$provincia[i]=="ALICANTE" ) t2008b$provincia[i]<-"Alicante/Alacant"
if(t2008b$provincia[i]=="ALMERÍA" ) t2008b$provincia[i]<-"Almería"
if(t2008b$provincia[i]=="ÁVILA" ) t2008b$provincia[i]<-"Ávila"
if(t2008b$provincia[i]=="BADAJOZ" ) t2008b$provincia[i]<-"Badajoz"
if(t2008b$provincia[i]=="ILLES BALEARS" ) t2008b$provincia[i]<-"Balears (Illes)"
if(t2008b$provincia[i]=="BARCELONA" ) t2008b$provincia[i]<-"Barcelona"
if(t2008b$provincia[i]=="BURGOS" ) t2008b$provincia[i]<-"Burgos"
if(t2008b$provincia[i]=="CÁCERES" ) t2008b$provincia[i]<-"Cáceres"
if(t2008b$provincia[i]=="CÁDIZ" ) t2008b$provincia[i]<-"Cádiz"
if(t2008b$provincia[i]=="CASTELLÓN" ) t2008b$provincia[i]<-"Castellón/Castelló"
if(t2008b$provincia[i]=="CIUDAD REAL" ) t2008b$provincia[i]<-"Ciudad Real"
if(t2008b$provincia[i]=="CÓRDOBA" ) t2008b$provincia[i]<-"Córdoba"
if(t2008b$provincia[i]=="A CORUÑA" ) t2008b$provincia[i]<-"Coruña (A)"
if(t2008b$provincia[i]=="CUENCA" ) t2008b$provincia[i]<-"Cuenca"
if(t2008b$provincia[i]=="GIRONA" ) t2008b$provincia[i]<-"Girona"
if(t2008b$provincia[i]=="GRANADA" ) t2008b$provincia[i]<-"Granada"
if(t2008b$provincia[i]=="GUADALAJARA" ) t2008b$provincia[i]<-"Guadalajara"
if(t2008b$provincia[i]=="GUIPÚZCOA" ) t2008b$provincia[i]<-"Gipuzkoa"
if(t2008b$provincia[i]=="HUELVA" ) t2008b$provincia[i]<-"Huelva"
if(t2008b$provincia[i]=="HUESCA" ) t2008b$provincia[i]<-"Huesca"
if(t2008b$provincia[i]=="JAÉN" ) t2008b$provincia[i]<-"Jaén"
if(t2008b$provincia[i]=="LEÓN" ) t2008b$provincia[i]<-"León"
if(t2008b$provincia[i]=="LLEIDA" ) t2008b$provincia[i]<-"Lleida"
if(t2008b$provincia[i]=="LA RIOJA" ) t2008b$provincia[i]<-"Rioja (La)"
if(t2008b$provincia[i]=="LUGO" ) t2008b$provincia[i]<-"Lugo"
if(t2008b$provincia[i]=="MADRID" ) t2008b$provincia[i]<-"Madrid"
if(t2008b$provincia[i]=="MÁLAGA" ) t2008b$provincia[i]<-"Málaga"
if(t2008b$provincia[i]=="MURCIA" ) t2008b$provincia[i]<-"Murcia"
if(t2008b$provincia[i]=="NAVARRA" ) t2008b$provincia[i]<-"Navarra"
if(t2008b$provincia[i]=="OURENSE" ) t2008b$provincia[i]<-"Ourense"
if(t2008b$provincia[i]=="ASTURIAS" ) t2008b$provincia[i]<-"Asturias"
if(t2008b$provincia[i]=="PALENCIA" ) t2008b$provincia[i]<-"Palencia"
if(t2008b$provincia[i]=="LAS PALMAS" ) t2008b$provincia[i]<-"Palmas (Las)"
if(t2008b$provincia[i]=="PONTEVEDRA" ) t2008b$provincia[i]<-"Pontevedra"
if(t2008b$provincia[i]=="SALAMANCA" ) t2008b$provincia[i]<-"Salamanca"
if(t2008b$provincia[i]=="STA. C. TENERIFE" )
t2008b$provincia[i]<-"Santa Cruz de Tenerife"
if(t2008b$provincia[i]=="CANTABRIA" ) t2008b$provincia[i]<-"Cantabria"
if(t2008b$provincia[i]=="SEGOVIA" ) t2008b$provincia[i]<-"Segovia"
if(t2008b$provincia[i]=="SEVILLA" ) t2008b$provincia[i]<-"Sevilla"
if(t2008b$provincia[i]=="SORIA" ) t2008b$provincia[i]<-"Soria"
if(t2008b$provincia[i]=="TARRAGONA" ) t2008b$provincia[i]<-"Tarragona"
if(t2008b$provincia[i]=="TERUEL" ) t2008b$provincia[i]<-"Teruel"
if(t2008b$provincia[i]=="TOLEDO" ) t2008b$provincia[i]<-"Toledo"
if(t2008b$provincia[i]=="VALENCIA" ) t2008b$provincia[i]<-"Valencia/València"
if(t2008b$provincia[i]=="VALLADOLID" ) t2008b$provincia[i]<-"Valladolid"
if(t2008b$provincia[i]=="VIZCAYA" ) t2008b$provincia[i]<-"Bizkaia"
```

```
  if(t2008b$provincia[i]=="ZAMORA" ) t2008b$provincia[i]<-"Zamora"
  if(t2008b$provincia[i]=="ZARAGOZA" ) t2008b$provincia[i]<-"Zaragoza"
  if(t2008b$provincia[i]=="CEUTA" ) t2008b$provincia[i]<-"Ceuta"
  if(t2008b$provincia[i]=="MELILLA" ) t2008b$provincia[i]<-"Melilla"
}

t2009<-read_xls("TG09.xls")
t2009a<-t2009[-c(1,3),c(1,10,15)]
t2009b<-t2009a[-c(1),]
t2009b$year<-2009

t2009pv<-read_xlsx("PV09.xlsx")
t2009PV<-t2009pv[-c(1,3),]
t2009bb=cbind(t2009b,t2009PV[-c(54,55),c(9)])
t2009b=cbind(t2009bb,t2009PV[-c(54,55),c(14)])
t2009b=t2009b[-c(53),]

nombresantiguos<-unique(t2009b$provincia)
colnames(t2009b)<-c("provincia","fallecidos_interurbana","fallecidos_urbana",
"año","numero_vehiculos","poblacion")

t2009b$numero_vehiculos[1]<-"199308"
t2009b$poblacion[1]<-307656

for(i in (1:length(t2009b$provincia)))
{cat(t2009b$provincia[i],"\n")
  if(t2009b$provincia[i]=="ÁLAVA") t2009b$provincia[i]<-"Araba/Álava"
  if(t2009b$provincia[i]=="ALBACETE") t2009b$provincia[i]<-"Albacete"
  if(t2009b$provincia[i]=="ALICANTE" ) t2009b$provincia[i]<-"Alicante/Alacant"
  if(t2009b$provincia[i]=="ALMERÍA" ) t2009b$provincia[i]<-"Almería"
  if(t2009b$provincia[i]=="ÁVILA" ) t2009b$provincia[i]<-"Ávila"
  if(t2009b$provincia[i]=="BADAJOZ" ) t2009b$provincia[i]<-"Badajoz"
  if(t2009b$provincia[i]=="ILLES BALEARS" ) t2009b$provincia[i]<-"Balears (Illes)"
  if(t2009b$provincia[i]=="BARCELONA" ) t2009b$provincia[i]<-"Barcelona"
  if(t2009b$provincia[i]=="BURGOS" ) t2009b$provincia[i]<-"Burgos"
  if(t2009b$provincia[i]=="CÁCERES" ) t2009b$provincia[i]<-"Cáceres"
  if(t2009b$provincia[i]=="CÁDIZ" ) t2009b$provincia[i]<-"Cádiz"
  if(t2009b$provincia[i]=="CASTELLÓN" ) t2009b$provincia[i]<-"Castellón/Castelló"
  if(t2009b$provincia[i]=="CIUDAD REAL" ) t2009b$provincia[i]<-"Ciudad Real"
  if(t2009b$provincia[i]=="CÓRDOBA" ) t2009b$provincia[i]<-"Córdoba"
  if(t2009b$provincia[i]=="A CORUÑA" ) t2009b$provincia[i]<-"Coruña (A)"
  if(t2009b$provincia[i]=="CUENCA" ) t2009b$provincia[i]<-"Cuenca"
  if(t2009b$provincia[i]=="GIRONA" ) t2009b$provincia[i]<-"Girona"
  if(t2009b$provincia[i]=="GRANADA" ) t2009b$provincia[i]<-"Granada"
  if(t2009b$provincia[i]=="GUADALAJARA" ) t2009b$provincia[i]<-"Guadalajara"
  if(t2009b$provincia[i]=="GUIPÚZCOA" ) t2009b$provincia[i]<-"Gipuzkoa"
  if(t2009b$provincia[i]=="HUELVA" ) t2009b$provincia[i]<-"Huelva"
  if(t2009b$provincia[i]=="HUESCA" ) t2009b$provincia[i]<-"Huesca"
```

```
  if(t2009b$provincia[i]=="JAÉN" ) t2009b$provincia[i]<-"Jaén"
  if(t2009b$provincia[i]=="LEÓN" ) t2009b$provincia[i]<-"León"
  if(t2009b$provincia[i]=="LLEIDA" ) t2009b$provincia[i]<-"Lleida"
  if(t2009b$provincia[i]=="LA RIOJA" ) t2009b$provincia[i]<-"Rioja (La)"
  if(t2009b$provincia[i]=="LUGO" ) t2009b$provincia[i]<-"Lugo"
  if(t2009b$provincia[i]=="MADRID" ) t2009b$provincia[i]<-"Madrid"
  if(t2009b$provincia[i]=="MÁLAGA" ) t2009b$provincia[i]<-"Málaga"
  if(t2009b$provincia[i]=="MURCIA" ) t2009b$provincia[i]<-"Murcia"
  if(t2009b$provincia[i]=="NAVARRA" ) t2009b$provincia[i]<-"Navarra"
  if(t2009b$provincia[i]=="OURENSE" ) t2009b$provincia[i]<-"Ourense"
  if(t2009b$provincia[i]=="ASTURIAS" ) t2009b$provincia[i]<-"Asturias"
  if(t2009b$provincia[i]=="PALENCIA" ) t2009b$provincia[i]<-"Palencia"
  if(t2009b$provincia[i]=="LAS PALMAS" ) t2009b$provincia[i]<-"Palmas (Las)"
  if(t2009b$provincia[i]=="PONTEVEDRA" ) t2009b$provincia[i]<-"Pontevedra"
  if(t2009b$provincia[i]=="SALAMANCA" ) t2009b$provincia[i]<-"Salamanca"
  if(t2009b$provincia[i]=="STA. C. TENERIFE" )
  t2009b$provincia[i]<-"Santa Cruz de Tenerife"
  if(t2009b$provincia[i]=="CANTABRIA" ) t2009b$provincia[i]<-"Cantabria"
  if(t2009b$provincia[i]=="SEGOVIA" ) t2009b$provincia[i]<-"Segovia"
  if(t2009b$provincia[i]=="SEVILLA" ) t2009b$provincia[i]<-"Sevilla"
  if(t2009b$provincia[i]=="SORIA" ) t2009b$provincia[i]<-"Soria"
  if(t2009b$provincia[i]=="TARRAGONA" ) t2009b$provincia[i]<-"Tarragona"
  if(t2009b$provincia[i]=="TERUEL" ) t2009b$provincia[i]<-"Teruel"
  if(t2009b$provincia[i]=="TOLEDO" ) t2009b$provincia[i]<-"Toledo"
  if(t2009b$provincia[i]=="VALENCIA" ) t2009b$provincia[i]<-"Valencia/València"
  if(t2009b$provincia[i]=="VALLADOLID" ) t2009b$provincia[i]<-"Valladolid"
  if(t2009b$provincia[i]=="VIZCAYA" ) t2009b$provincia[i]<-"Bizkaia"
  if(t2009b$provincia[i]=="ZAMORA" ) t2009b$provincia[i]<-"Zamora"
  if(t2009b$provincia[i]=="ZARAGOZA" ) t2009b$provincia[i]<-"Zaragoza"
  if(t2009b$provincia[i]=="CEUTA" ) t2009b$provincia[i]<-"Ceuta"
  if(t2009b$provincia[i]=="MELILLA" ) t2009b$provincia[i]<-"Melilla"
}

t2010<-read_xls("TG10.xls")
t2010a<-t2010[-c(1,3),c(1,10,15)]
t2010b<-t2010a[-c(1),]
t2010b$year<-2010

t2010pv<-read_xlsx("PV10.xlsx")
t2010PV<-t2010pv[-c(1,3),]
t2010bb=cbind(t2010b,t2010PV[-c(54,55),c(9)])
t2010b=cbind(t2010bb,t2010PV[-c(54,55),c(14)])
t2010b=t2010b[-c(53),]

nombresantiguos<-unique(t2010b$provincia)
colnames(t2010b)<-c("provincia","fallecidos_interurbana","fallecidos_urbana",
"año","numero_vehiculos","poblacion")
```

```
t2010b$numero_vehiculos[1]<-"202542"
t2010b$poblacion[1]<-310562

for(i in (1:length(t2010b$provincia)))
{cat(t2010b$provincia[i],"\n")
  if(t2010b$provincia[i]=="ARABA/ÁLAVA") t2010b$provincia[i]<-"Araba/Álava"
  if(t2010b$provincia[i]=="ALBACETE") t2010b$provincia[i]<-"Albacete"
  if(t2010b$provincia[i]=="ALICANTE" ) t2010b$provincia[i]<-"Alicante/Alacant"
  if(t2010b$provincia[i]=="ALMERÍA" ) t2010b$provincia[i]<-"Almería"
  if(t2010b$provincia[i]=="ÁVILA" ) t2010b$provincia[i]<-"Ávila"
  if(t2010b$provincia[i]=="BADAJOZ" ) t2010b$provincia[i]<-"Badajoz"
  if(t2010b$provincia[i]=="ILLES BALEARS" ) t2010b$provincia[i]<-"Balears (Illes)"
  if(t2010b$provincia[i]=="BARCELONA" ) t2010b$provincia[i]<-"Barcelona"
  if(t2010b$provincia[i]=="BURGOS" ) t2010b$provincia[i]<-"Burgos"
  if(t2010b$provincia[i]=="CÁCERES" ) t2010b$provincia[i]<-"Cáceres"
  if(t2010b$provincia[i]=="CÁDIZ" ) t2010b$provincia[i]<-"Cádiz"
  if(t2010b$provincia[i]=="CASTELLÓN" ) t2010b$provincia[i]<-"Castellón/Castelló"
  if(t2010b$provincia[i]=="CIUDAD REAL" ) t2010b$provincia[i]<-"Ciudad Real"
  if(t2010b$provincia[i]=="CÓRDOBA" ) t2010b$provincia[i]<-"Córdoba"
  if(t2010b$provincia[i]=="A CORUÑA" ) t2010b$provincia[i]<-"Coruña (A)"
  if(t2010b$provincia[i]=="CUENCA" ) t2010b$provincia[i]<-"Cuenca"
  if(t2010b$provincia[i]=="GIRONA" ) t2010b$provincia[i]<-"Girona"
  if(t2010b$provincia[i]=="GRANADA" ) t2010b$provincia[i]<-"Granada"
  if(t2010b$provincia[i]=="GUADALAJARA" ) t2010b$provincia[i]<-"Guadalajara"
  if(t2010b$provincia[i]=="GIPUZKOA" ) t2010b$provincia[i]<-"Gipuzkoa"
  if(t2010b$provincia[i]=="HUELVA" ) t2010b$provincia[i]<-"Huelva"
  if(t2010b$provincia[i]=="HUESCA" ) t2010b$provincia[i]<-"Huesca"
  if(t2010b$provincia[i]=="JAÉN" ) t2010b$provincia[i]<-"Jaén"
  if(t2010b$provincia[i]=="LEÓN" ) t2010b$provincia[i]<-"León"
  if(t2010b$provincia[i]=="LLEIDA" ) t2010b$provincia[i]<-"Lleida"
  if(t2010b$provincia[i]=="LA RIOJA" ) t2010b$provincia[i]<-"Rioja (La)"
  if(t2010b$provincia[i]=="LUGO" ) t2010b$provincia[i]<-"Lugo"
  if(t2010b$provincia[i]=="MADRID" ) t2010b$provincia[i]<-"Madrid"
  if(t2010b$provincia[i]=="MÁLAGA" ) t2010b$provincia[i]<-"Málaga"
  if(t2010b$provincia[i]=="MURCIA" ) t2010b$provincia[i]<-"Murcia"
  if(t2010b$provincia[i]=="NAVARRA" ) t2010b$provincia[i]<-"Navarra"
  if(t2010b$provincia[i]=="OURENSE" ) t2010b$provincia[i]<-"Ourense"
  if(t2010b$provincia[i]=="ASTURIAS" ) t2010b$provincia[i]<-"Asturias"
  if(t2010b$provincia[i]=="PALENCIA" ) t2010b$provincia[i]<-"Palencia"
  if(t2010b$provincia[i]=="LAS PALMAS" ) t2010b$provincia[i]<-"Palmas (Las)"
  if(t2010b$provincia[i]=="PONTEVEDRA" ) t2010b$provincia[i]<-"Pontevedra"
  if(t2010b$provincia[i]=="SALAMANCA" ) t2010b$provincia[i]<-"Salamanca"
  if(t2010b$provincia[i]=="STA. C. TENERIFE" )
  t2010b$provincia[i]<-"Santa Cruz de Tenerife"
  if(t2010b$provincia[i]=="CANTABRIA" ) t2010b$provincia[i]<-"Cantabria"
  if(t2010b$provincia[i]=="SEGOVIA" ) t2010b$provincia[i]<-"Segovia"
  if(t2010b$provincia[i]=="SEVILLA" ) t2010b$provincia[i]<-"Sevilla"
  if(t2010b$provincia[i]=="SORIA" ) t2010b$provincia[i]<-"Soria"
```

```
  if(t2010b$provincia[i]=="TARRAGONA" ) t2010b$provincia[i]<-"Tarragona"
  if(t2010b$provincia[i]=="TERUEL" ) t2010b$provincia[i]<-"Teruel"
  if(t2010b$provincia[i]=="TOLEDO" ) t2010b$provincia[i]<-"Toledo"
  if(t2010b$provincia[i]=="VALENCIA" ) t2010b$provincia[i]<-"Valencia/València"
  if(t2010b$provincia[i]=="VALLADOLID" ) t2010b$provincia[i]<-"Valladolid"
  if(t2010b$provincia[i]=="BIZKAIA" ) t2010b$provincia[i]<-"Bizkaia"
  if(t2010b$provincia[i]=="ZAMORA" ) t2010b$provincia[i]<-"Zamora"
  if(t2010b$provincia[i]=="ZARAGOZA" ) t2010b$provincia[i]<-"Zaragoza"
  if(t2010b$provincia[i]=="CEUTA" ) t2010b$provincia[i]<-"Ceuta"
  if(t2010b$provincia[i]=="MELILLA" ) t2010b$provincia[i]<-"Melilla"
}

t2011<-read_xls("TG11.xls")
t2011a<-t2011[-c(1,3),c(1,10,15)]
t2011b<-t2011a[-c(1),]
t2011b$year<-2011

t2011pv<-read_xlsx("PV11.xlsx")
t2011PV<-t2011pv[-c(1,3),]
t2011bb=cbind(t2011b,t2011PV[-c(54,55),c(9)])
t2011b=cbind(t2011bb,t2011PV[-c(54,55),c(14)])
t2011b=t2011b[-c(53),]

nombresantiguos<-unique(t2011b$provincia)
colnames(t2011b)<-c("provincia","fallecidos_interurbana","fallecidos_urbana",
"año","numero_vehiculos","poblacion")

t2011b$numero_vehiculos[1]<-"203328"
t2011b$poblacion[1]<-312763

for(i in (1:length(t2011b$provincia)))
{cat(t2011b$provincia[i],"\n")
  if(t2011b$provincia[i]=="ARABA/ÁLAVA") t2011b$provincia[i]<-"Araba/Álava"
  if(t2011b$provincia[i]=="ALBACETE") t2011b$provincia[i]<-"Albacete"
  if(t2011b$provincia[i]=="ALICANTE" ) t2011b$provincia[i]<-"Alicante/Alacant"
  if(t2011b$provincia[i]=="ALMERÍA" ) t2011b$provincia[i]<-"Almería"
  if(t2011b$provincia[i]=="ÁVILA" ) t2011b$provincia[i]<-"Ávila"
  if(t2011b$provincia[i]=="BADAJOZ" ) t2011b$provincia[i]<-"Badajoz"
  if(t2011b$provincia[i]=="ILLES BALEARS" ) t2011b$provincia[i]<-"Balears (Illes)"
  if(t2011b$provincia[i]=="BARCELONA" ) t2011b$provincia[i]<-"Barcelona"
  if(t2011b$provincia[i]=="BURGOS" ) t2011b$provincia[i]<-"Burgos"
  if(t2011b$provincia[i]=="CÁCERES" ) t2011b$provincia[i]<-"Cáceres"
  if(t2011b$provincia[i]=="CÁDIZ" ) t2011b$provincia[i]<-"Cádiz"
  if(t2011b$provincia[i]=="CASTELLÓN" ) t2011b$provincia[i]<-"Castellón/Castelló"
  if(t2011b$provincia[i]=="CIUDAD REAL" ) t2011b$provincia[i]<-"Ciudad Real"
  if(t2011b$provincia[i]=="CÓRDOBA" ) t2011b$provincia[i]<-"Córdoba"
  if(t2011b$provincia[i]=="A CORUÑA" ) t2011b$provincia[i]<-"Coruña (A)"
  if(t2011b$provincia[i]=="CUENCA" ) t2011b$provincia[i]<-"Cuenca"
```

```
  if(t2011b$provincia[i]=="GIRONA" ) t2011b$provincia[i]<-"Girona"
  if(t2011b$provincia[i]=="GRANADA" ) t2011b$provincia[i]<-"Granada"
  if(t2011b$provincia[i]=="GUADALAJARA" ) t2011b$provincia[i]<-"Guadalajara"
  if(t2011b$provincia[i]=="GIPUZKOA" ) t2011b$provincia[i]<-"Gipuzkoa"
  if(t2011b$provincia[i]=="HUELVA" ) t2011b$provincia[i]<-"Huelva"
  if(t2011b$provincia[i]=="HUESCA" ) t2011b$provincia[i]<-"Huesca"
  if(t2011b$provincia[i]=="JAÉN" ) t2011b$provincia[i]<-"Jaén"
  if(t2011b$provincia[i]=="LEÓN" ) t2011b$provincia[i]<-"León"
  if(t2011b$provincia[i]=="LLEIDA" ) t2011b$provincia[i]<-"Lleida"
  if(t2011b$provincia[i]=="LA RIOJA" ) t2011b$provincia[i]<-"Rioja (La)"
  if(t2011b$provincia[i]=="LUGO" ) t2011b$provincia[i]<-"Lugo"
  if(t2011b$provincia[i]=="MADRID" ) t2011b$provincia[i]<-"Madrid"
  if(t2011b$provincia[i]=="MÁLAGA" ) t2011b$provincia[i]<-"Málaga"
  if(t2011b$provincia[i]=="MURCIA" ) t2011b$provincia[i]<-"Murcia"
  if(t2011b$provincia[i]=="NAVARRA" ) t2011b$provincia[i]<-"Navarra"
  if(t2011b$provincia[i]=="OURENSE" ) t2011b$provincia[i]<-"Ourense"
  if(t2011b$provincia[i]=="ASTURIAS" ) t2011b$provincia[i]<-"Asturias"
  if(t2011b$provincia[i]=="PALENCIA" ) t2011b$provincia[i]<-"Palencia"
  if(t2011b$provincia[i]=="LAS PALMAS" ) t2011b$provincia[i]<-"Palmas (Las)"
  if(t2011b$provincia[i]=="PONTEVEDRA" ) t2011b$provincia[i]<-"Pontevedra"
  if(t2011b$provincia[i]=="SALAMANCA" ) t2011b$provincia[i]<-"Salamanca"
  if(t2011b$provincia[i]=="STA. C. TENERIFE" )
  t2011b$provincia[i]<-"Santa Cruz de Tenerife"
  if(t2011b$provincia[i]=="CANTABRIA" ) t2011b$provincia[i]<-"Cantabria"
  if(t2011b$provincia[i]=="SEGOVIA" ) t2011b$provincia[i]<-"Segovia"
  if(t2011b$provincia[i]=="SEVILLA" ) t2011b$provincia[i]<-"Sevilla"
  if(t2011b$provincia[i]=="SORIA" ) t2011b$provincia[i]<-"Soria"
  if(t2011b$provincia[i]=="TARRAGONA" ) t2011b$provincia[i]<-"Tarragona"
  if(t2011b$provincia[i]=="TERUEL" ) t2011b$provincia[i]<-"Teruel"
  if(t2011b$provincia[i]=="TOLEDO" ) t2011b$provincia[i]<-"Toledo"
  if(t2011b$provincia[i]=="VALENCIA" ) t2011b$provincia[i]<-"Valencia/València"
  if(t2011b$provincia[i]=="VALLADOLID" ) t2011b$provincia[i]<-"Valladolid"
  if(t2011b$provincia[i]=="BIZKAIA" ) t2011b$provincia[i]<-"Bizkaia"
  if(t2011b$provincia[i]=="ZAMORA" ) t2011b$provincia[i]<-"Zamora"
  if(t2011b$provincia[i]=="ZARAGOZA" ) t2011b$provincia[i]<-"Zaragoza"
  if(t2011b$provincia[i]=="CEUTA" ) t2011b$provincia[i]<-"Ceuta"
  if(t2011b$provincia[i]=="MELILLA" ) t2011b$provincia[i]<-"Melilla"
}

t2012<-read_xls("TG12.xls")
t2012a<-t2012[-c(1,3),c(1,10,15)]
t2012b<-t2012a[-c(1),]
t2012b$year<-2012

t2012pv<-read_xlsx("PV12.xlsx")
t2012PV<-t2012pv[-c(1,3),]
t2012bb=cbind(t2012b,t2012PV[-c(54,55),c(9)])
t2012b=cbind(t2012bb,t2012PV[-c(54,55),c(14)])
```

```
t2012b=t2012b[-c(53),]

nombresantiguos<-unique(t2012b$provincia)
colnames(t2012b)<-c("provincia","fallecidos_interurbana","fallecidos_urbana",
"año","numero_vehiculos","poblacion")

t2012b$numero_vehiculos[1]<-"203390"
t2012b$poblacion[1]<-310085

for(i in (1:length(t2012b$provincia)))
{cat(t2012b$provincia[i],"\n")
  if(t2012b$provincia[i]=="ARABA/ÁLAVA") t2012b$provincia[i]<-"Araba/Álava"
  if(t2012b$provincia[i]=="ALBACETE") t2012b$provincia[i]<-"Albacete"
  if(t2012b$provincia[i]=="ALICANTE/ALACANT" ) t2012b$provincia[i]<-"Alicante/Alacant
  if(t2012b$provincia[i]=="ALMERÍA" ) t2012b$provincia[i]<-"Almería"
  if(t2012b$provincia[i]=="ÁVILA" ) t2012b$provincia[i]<-"Ávila"
  if(t2012b$provincia[i]=="BADAJOZ" ) t2012b$provincia[i]<-"Badajoz"
  if(t2012b$provincia[i]=="BALEARS, ILLES" ) t2012b$provincia[i]<-"Balears (Illes)"
  if(t2012b$provincia[i]=="BARCELONA" ) t2012b$provincia[i]<-"Barcelona"
  if(t2012b$provincia[i]=="BURGOS" ) t2012b$provincia[i]<-"Burgos"
  if(t2012b$provincia[i]=="CÁCERES" ) t2012b$provincia[i]<-"Cáceres"
  if(t2012b$provincia[i]=="CÁDIZ" ) t2012b$provincia[i]<-"Cádiz"
  if(t2012b$provincia[i]=="CASTELLÓN/CASTELLÓ" ) t2012b$provincia[i]<-"Castellón/Cast
  if(t2012b$provincia[i]=="CIUDAD REAL" ) t2012b$provincia[i]<-"Ciudad Real"
  if(t2012b$provincia[i]=="CÓRDOBA" ) t2012b$provincia[i]<-"Córdoba"
  if(t2012b$provincia[i]=="CORUÑA, A" ) t2012b$provincia[i]<-"Coruña (A)"
  if(t2012b$provincia[i]=="CUENCA" ) t2012b$provincia[i]<-"Cuenca"
  if(t2012b$provincia[i]=="GIRONA" ) t2012b$provincia[i]<-"Girona"
  if(t2012b$provincia[i]=="GRANADA" ) t2012b$provincia[i]<-"Granada"
  if(t2012b$provincia[i]=="GUADALAJARA" ) t2012b$provincia[i]<-"Guadalajara"
  if(t2012b$provincia[i]=="GIPUZKOA" ) t2012b$provincia[i]<-"Gipuzkoa"
  if(t2012b$provincia[i]=="HUELVA" ) t2012b$provincia[i]<-"Huelva"
  if(t2012b$provincia[i]=="HUESCA" ) t2012b$provincia[i]<-"Huesca"
  if(t2012b$provincia[i]=="JAÉN" ) t2012b$provincia[i]<-"Jaén"
  if(t2012b$provincia[i]=="LEÓN" ) t2012b$provincia[i]<-"León"
  if(t2012b$provincia[i]=="LLEIDA" ) t2012b$provincia[i]<-"Lleida"
  if(t2012b$provincia[i]=="RIOJA, LA" ) t2012b$provincia[i]<-"Rioja (La)"
  if(t2012b$provincia[i]=="LUGO" ) t2012b$provincia[i]<-"Lugo"
  if(t2012b$provincia[i]=="MADRID" ) t2012b$provincia[i]<-"Madrid"
  if(t2012b$provincia[i]=="MÁLAGA" ) t2012b$provincia[i]<-"Málaga"
  if(t2012b$provincia[i]=="MURCIA" ) t2012b$provincia[i]<-"Murcia"
  if(t2012b$provincia[i]=="NAVARRA" ) t2012b$provincia[i]<-"Navarra"
  if(t2012b$provincia[i]=="OURENSE" ) t2012b$provincia[i]<-"Ourense"
  if(t2012b$provincia[i]=="ASTURIAS" ) t2012b$provincia[i]<-"Asturias"
  if(t2012b$provincia[i]=="PALENCIA" ) t2012b$provincia[i]<-"Palencia"
  if(t2012b$provincia[i]=="PALMAS, LAS" ) t2012b$provincia[i]<-"Palmas (Las)"
  if(t2012b$provincia[i]=="PONTEVEDRA" ) t2012b$provincia[i]<-"Pontevedra"
  if(t2012b$provincia[i]=="SALAMANCA" ) t2012b$provincia[i]<-"Salamanca"
```

```
  if(t2012b$provincia[i]=="STA. C. TENERIFE" )
  t2012b$provincia[i]<-"Santa Cruz de Tenerife"
  if(t2012b$provincia[i]=="CANTABRIA" ) t2012b$provincia[i]<-"Cantabria"
  if(t2012b$provincia[i]=="SEGOVIA" ) t2012b$provincia[i]<-"Segovia"
  if(t2012b$provincia[i]=="SEVILLA" ) t2012b$provincia[i]<-"Sevilla"
  if(t2012b$provincia[i]=="SORIA" ) t2012b$provincia[i]<-"Soria"
  if(t2012b$provincia[i]=="TARRAGONA" ) t2012b$provincia[i]<-"Tarragona"
  if(t2012b$provincia[i]=="TERUEL" ) t2012b$provincia[i]<-"Teruel"
  if(t2012b$provincia[i]=="TOLEDO" ) t2012b$provincia[i]<-"Toledo"
  if(t2012b$provincia[i]=="VALENCIA/VALÈNCIA" ) t2012b$provincia[i]<-"Valencia/Valènc
  if(t2012b$provincia[i]=="VALLADOLID" ) t2012b$provincia[i]<-"Valladolid"
  if(t2012b$provincia[i]=="BIZKAIA" ) t2012b$provincia[i]<-"Bizkaia"
  if(t2012b$provincia[i]=="ZAMORA" ) t2012b$provincia[i]<-"Zamora"
  if(t2012b$provincia[i]=="ZARAGOZA" ) t2012b$provincia[i]<-"Zaragoza"
  if(t2012b$provincia[i]=="CEUTA" ) t2012b$provincia[i]<-"Ceuta"
  if(t2012b$provincia[i]=="MELILLA" ) t2012b$provincia[i]<-"Melilla"
}

t2013<-read_xls("TG13.xls")
t2013a<-t2013[-c(1,3),c(1,10,15)]
t2013aa<-t2013a[-c(1),]
t2013b<-t2013aa[-c(54),]
t2013b$year<-2013

t2013pv<-read_xlsx("PV13.xlsx")
t2013PV<-t2013pv[-c(1,3),]
t2013bb=cbind(t2013b,t2013PV[-c(54,55),c(9)])
t2013b=cbind(t2013bb,t2013PV[-c(54,55),c(14)])
t2013b=t2013b[-c(53),]

nombsesantiguos<-unique(t2013b$provincia)
colnames(t2013b)<-c("provincia","fallecidos_interurbana","fallecidos_urbana",
"año","numero_vehiculos","poblacion")

t2013b$numero_vehiculos[1]<-"201494"
t2013b$poblacion[1]<-319927

for(i in (1:length(t2013b$provincia)))
{cat(t2013b$provincia[i],"\n")
  if(t2013b$provincia[i]=="ARABA/ÁLAVA") t2013b$provincia[i]<-"Araba/Álava"
  if(t2013b$provincia[i]=="ALBACETE") t2013b$provincia[i]<-"Albacete"
  if(t2013b$provincia[i]=="ALICANTE/ALACANT" ) t2013b$provincia[i]<-"Alicante/Alacant
  if(t2013b$provincia[i]=="ALMERÍA" ) t2013b$provincia[i]<-"Almería"
  if(t2013b$provincia[i]=="ÁVILA" ) t2013b$provincia[i]<-"Ávila"
  if(t2013b$provincia[i]=="BADAJOZ" ) t2013b$provincia[i]<-"Badajoz"
  if(t2013b$provincia[i]=="BALEARS, ILLES" ) t2013b$provincia[i]<-"Balears (Illes)"
  if(t2013b$provincia[i]=="BARCELONA" ) t2013b$provincia[i]<-"Barcelona"
  if(t2013b$provincia[i]=="BURGOS" ) t2013b$provincia[i]<-"Burgos"
```

```
  if(t2013b$provincia[i]=="CÁCERES" ) t2013b$provincia[i]<-"Cáceres"
  if(t2013b$provincia[i]=="CÁDIZ" ) t2013b$provincia[i]<-"Cádiz"
  if(t2013b$provincia[i]=="CASTELLÓN/CASTELLÓ" ) t2013b$provincia[i]<-"Castellón/Cast
  if(t2013b$provincia[i]=="CIUDAD REAL" ) t2013b$provincia[i]<-"Ciudad Real"
  if(t2013b$provincia[i]=="CÓRDOBA" ) t2013b$provincia[i]<-"Córdoba"
  if(t2013b$provincia[i]=="CORUÑA, A" ) t2013b$provincia[i]<-"Coruña (A)"
  if(t2013b$provincia[i]=="CUENCA" ) t2013b$provincia[i]<-"Cuenca"
  if(t2013b$provincia[i]=="GIRONA" ) t2013b$provincia[i]<-"Girona"
  if(t2013b$provincia[i]=="GRANADA" ) t2013b$provincia[i]<-"Granada"
  if(t2013b$provincia[i]=="GUADALAJARA" ) t2013b$provincia[i]<-"Guadalajara"
  if(t2013b$provincia[i]=="GIPUZKOA" ) t2013b$provincia[i]<-"Gipuzkoa"
  if(t2013b$provincia[i]=="HUELVA" ) t2013b$provincia[i]<-"Huelva"
  if(t2013b$provincia[i]=="HUESCA" ) t2013b$provincia[i]<-"Huesca"
  if(t2013b$provincia[i]=="JAÉN" ) t2013b$provincia[i]<-"Jaén"
  if(t2013b$provincia[i]=="LEÓN" ) t2013b$provincia[i]<-"León"
  if(t2013b$provincia[i]=="LLEIDA" ) t2013b$provincia[i]<-"Lleida"
  if(t2013b$provincia[i]=="RIOJA, LA" ) t2013b$provincia[i]<-"Rioja (La)"
  if(t2013b$provincia[i]=="LUGO" ) t2013b$provincia[i]<-"Lugo"
  if(t2013b$provincia[i]=="MADRID" ) t2013b$provincia[i]<-"Madrid"
  if(t2013b$provincia[i]=="MÁLAGA" ) t2013b$provincia[i]<-"Málaga"
  if(t2013b$provincia[i]=="MURCIA" ) t2013b$provincia[i]<-"Murcia"
  if(t2013b$provincia[i]=="NAVARRA" ) t2013b$provincia[i]<-"Navarra"
  if(t2013b$provincia[i]=="OURENSE" ) t2013b$provincia[i]<-"Ourense"
  if(t2013b$provincia[i]=="ASTURIAS" ) t2013b$provincia[i]<-"Asturias"
  if(t2013b$provincia[i]=="PALENCIA" ) t2013b$provincia[i]<-"Palencia"
  if(t2013b$provincia[i]=="PALMAS, LAS" ) t2013b$provincia[i]<-"Palmas (Las)"
  if(t2013b$provincia[i]=="PONTEVEDRA" ) t2013b$provincia[i]<-"Pontevedra"
  if(t2013b$provincia[i]=="SALAMANCA" ) t2013b$provincia[i]<-"Salamanca"
  if(t2013b$provincia[i]=="STA. C. TENERIFE" )
  t2013b$provincia[i]<-"Santa Cruz de Tenerife"
  if(t2013b$provincia[i]=="CANTABRIA" ) t2013b$provincia[i]<-"Cantabria"
  if(t2013b$provincia[i]=="SEGOVIA" ) t2013b$provincia[i]<-"Segovia"
  if(t2013b$provincia[i]=="SEVILLA" ) t2013b$provincia[i]<-"Sevilla"
  if(t2013b$provincia[i]=="SORIA" ) t2013b$provincia[i]<-"Soria"
  if(t2013b$provincia[i]=="TARRAGONA" ) t2013b$provincia[i]<-"Tarragona"
  if(t2013b$provincia[i]=="TERUEL" ) t2013b$provincia[i]<-"Teruel"
  if(t2013b$provincia[i]=="TOLEDO" ) t2013b$provincia[i]<-"Toledo"
  if(t2013b$provincia[i]=="VALENCIA/VALÈNCIA" ) t2013b$provincia[i]<-"Valencia/Valènc
  if(t2013b$provincia[i]=="VALLADOLID" ) t2013b$provincia[i]<-"Valladolid"
  if(t2013b$provincia[i]=="BIZKAIA" ) t2013b$provincia[i]<-"Bizkaia"
  if(t2013b$provincia[i]=="ZAMORA" ) t2013b$provincia[i]<-"Zamora"
  if(t2013b$provincia[i]=="ZARAGOZA" ) t2013b$provincia[i]<-"Zaragoza"
  if(t2013b$provincia[i]=="CEUTA" ) t2013b$provincia[i]<-"Ceuta"
  if(t2013b$provincia[i]=="MELILLA" ) t2013b$provincia[i]<-"Melilla"
}

t2014<-read_xls("TG14.xls")
t2014a<-t2014[-c(1,3),c(1,11,17)]
```

```
t2014aa<-t2014a[-c(1),]
t2014b<-t2014aa[-c(54),]
t2014b$year<-2014

t2014pv<-read_xlsx("PV14.xlsx")
t2014PV<-t2014pv[-c(1,3),]
t2014bb=cbind(t2014b,t2014PV[,c(9)])
t2014b=cbind(t2014bb,t2014PV[,c(10)])
t2014b=t2014b[-c(53),]

nombsesantiguos<-unique(t2014b$provincia)
colnames(t2014b)<-c("provincia","fallecidos_interurbana","fallecidos_urbana",
"año","numero_vehiculos","poblacion")

t2014b$numero_vehiculos[1]<-"202179"
t2014b$poblacion[1]<-321932

for(i in (1:length(t2014b$provincia)))
{cat(t2014b$provincia[i],"\n")
  if(t2014b$provincia[i]=="ARABA/ÁLAVA") t2014b$provincia[i]<-"Araba/Álava"
  if(t2014b$provincia[i]=="ALBACETE") t2014b$provincia[i]<-"Albacete"
  if(t2014b$provincia[i]=="ALICANTE/ALACANT" ) t2014b$provincia[i]<-"Alicante/Alacant
  if(t2014b$provincia[i]=="ALMERÍA" ) t2014b$provincia[i]<-"Almería"
  if(t2014b$provincia[i]=="ÁVILA" ) t2014b$provincia[i]<-"Ávila"
  if(t2014b$provincia[i]=="BADAJOZ" ) t2014b$provincia[i]<-"Badajoz"
  if(t2014b$provincia[i]=="BALEARS, ILLES" ) t2014b$provincia[i]<-"Balears (Illes)"
  if(t2014b$provincia[i]=="BARCELONA" ) t2014b$provincia[i]<-"Barcelona"
  if(t2014b$provincia[i]=="BURGOS" ) t2014b$provincia[i]<-"Burgos"
  if(t2014b$provincia[i]=="CÁCERES" ) t2014b$provincia[i]<-"Cáceres"
  if(t2014b$provincia[i]=="CÁDIZ" ) t2014b$provincia[i]<-"Cádiz"
  if(t2014b$provincia[i]=="CASTELLÓN/CASTELLÓ" ) t2014b$provincia[i]<-"Castellón/Cast
  if(t2014b$provincia[i]=="CIUDAD REAL" ) t2014b$provincia[i]<-"Ciudad Real"
  if(t2014b$provincia[i]=="CÓRDOBA" ) t2014b$provincia[i]<-"Córdoba"
  if(t2014b$provincia[i]=="CORUÑA, A" ) t2014b$provincia[i]<-"Coruña (A)"
  if(t2014b$provincia[i]=="CUENCA" ) t2014b$provincia[i]<-"Cuenca"
  if(t2014b$provincia[i]=="GIRONA" ) t2014b$provincia[i]<-"Girona"
  if(t2014b$provincia[i]=="GRANADA" ) t2014b$provincia[i]<-"Granada"
  if(t2014b$provincia[i]=="GUADALAJARA" ) t2014b$provincia[i]<-"Guadalajara"
  if(t2014b$provincia[i]=="GIPUZKOA" ) t2014b$provincia[i]<-"Gipuzkoa"
  if(t2014b$provincia[i]=="HUELVA" ) t2014b$provincia[i]<-"Huelva"
  if(t2014b$provincia[i]=="HUESCA" ) t2014b$provincia[i]<-"Huesca"
  if(t2014b$provincia[i]=="JAÉN" ) t2014b$provincia[i]<-"Jaén"
  if(t2014b$provincia[i]=="LEÓN" ) t2014b$provincia[i]<-"León"
  if(t2014b$provincia[i]=="LLEIDA" ) t2014b$provincia[i]<-"Lleida"
  if(t2014b$provincia[i]=="RIOJA, LA" ) t2014b$provincia[i]<-"Rioja (La)"
  if(t2014b$provincia[i]=="LUGO" ) t2014b$provincia[i]<-"Lugo"
  if(t2014b$provincia[i]=="MADRID" ) t2014b$provincia[i]<-"Madrid"
  if(t2014b$provincia[i]=="MÁLAGA" ) t2014b$provincia[i]<-"Málaga"
```

```
  if(t2014b$provincia[i]=="MURCIA" ) t2014b$provincia[i]<-"Murcia"
  if(t2014b$provincia[i]=="NAVARRA" ) t2014b$provincia[i]<-"Navarra"
  if(t2014b$provincia[i]=="OURENSE" ) t2014b$provincia[i]<-"Ourense"
  if(t2014b$provincia[i]=="ASTURIAS" ) t2014b$provincia[i]<-"Asturias"
  if(t2014b$provincia[i]=="PALENCIA" ) t2014b$provincia[i]<-"Palencia"
  if(t2014b$provincia[i]=="PALMAS, LAS" ) t2014b$provincia[i]<-"Palmas (Las)"
  if(t2014b$provincia[i]=="PONTEVEDRA" ) t2014b$provincia[i]<-"Pontevedra"
  if(t2014b$provincia[i]=="SALAMANCA" ) t2014b$provincia[i]<-"Salamanca"
  if(t2014b$provincia[i]=="STA. C. TENERIFE" )
  t2014b$provincia[i]<-"Santa Cruz de Tenerife"
  if(t2014b$provincia[i]=="CANTABRIA" ) t2014b$provincia[i]<-"Cantabria"
  if(t2014b$provincia[i]=="SEGOVIA" ) t2014b$provincia[i]<-"Segovia"
  if(t2014b$provincia[i]=="SEVILLA" ) t2014b$provincia[i]<-"Sevilla"
  if(t2014b$provincia[i]=="SORIA" ) t2014b$provincia[i]<-"Soria"
  if(t2014b$provincia[i]=="TARRAGONA" ) t2014b$provincia[i]<-"Tarragona"
  if(t2014b$provincia[i]=="TERUEL" ) t2014b$provincia[i]<-"Teruel"
  if(t2014b$provincia[i]=="TOLEDO" ) t2014b$provincia[i]<-"Toledo"
  if(t2014b$provincia[i]=="VALENCIA/VALÈNCIA" ) t2014b$provincia[i]<-"Valencia/Valènc
  if(t2014b$provincia[i]=="VALLADOLID" ) t2014b$provincia[i]<-"Valladolid"
  if(t2014b$provincia[i]=="BIZKAIA" ) t2014b$provincia[i]<-"Bizkaia"
  if(t2014b$provincia[i]=="ZAMORA" ) t2014b$provincia[i]<-"Zamora"
  if(t2014b$provincia[i]=="ZARAGOZA" ) t2014b$provincia[i]<-"Zaragoza"
  if(t2014b$provincia[i]=="CEUTA" ) t2014b$provincia[i]<-"Ceuta"
  if(t2014b$provincia[i]=="MELILLA" ) t2014b$provincia[i]<-"Melilla"
}

t2015<-read_xlsx("TG15.xlsx")
t2015a<-t2015[-c(1,3),c(1,4,9)]
t2015b<-t2015a[-c(1),]
t2015b$year<-2015
t2015pv<-read_xlsx("PV15.xlsx")
t2015PV<-t2015pv[-c(1,3),]
t2015bb=cbind(t2015b,t2015PV[,c(9)])
t2015b=cbind(t2015bb,t2015PV[,c(10)])
t2015b=t2015b[-c(53),]

nombsesantiguos<-unique(t2015b$provincia)
colnames(t2015b)<-c("provincia","fallecidos_interurbana","fallecidos_urbana",
"año","numero_vehiculos","poblacion")

t2015b$numero_vehiculos[1]<-"203881"
t2015b$poblacion[1]<-323648

for(i in (1:length(t2015b$provincia)))
{cat(t2015b$provincia[i],"\n")
  if(t2015b$provincia[i]=="ARABA/ÁLAVA") t20105$provincia[i]<-"Araba/Álava"
  if(t2015b$provincia[i]=="ALBACETE") t2015b$provincia[i]<-"Albacete"
  if(t2015b$provincia[i]=="ALICANTE" ) t2015b$provincia[i]<-"Alicante/Alacant"
```

```
if(t2015b$provincia[i]=="ALMERÍA" ) t2015b$provincia[i]<-"Almería"
if(t2015b$provincia[i]=="ÁVILA" ) t2015b$provincia[i]<-"Ávila"
if(t2015b$provincia[i]=="BADAJOZ" ) t2015b$provincia[i]<-"Badajoz"
if(t2015b$provincia[i]=="ILLES BALEARS" ) t2015b$provincia[i]<-"Balears (Illes)"
if(t2015b$provincia[i]=="BARCELONA" ) t2015b$provincia[i]<-"Barcelona"
if(t2015b$provincia[i]=="BURGOS" ) t2015b$provincia[i]<-"Burgos"
if(t2015b$provincia[i]=="CÁCERES" ) t2015b$provincia[i]<-"Cáceres"
if(t2015b$provincia[i]=="CÁDIZ" ) t2015b$provincia[i]<-"Cádiz"
if(t2015b$provincia[i]=="CASTELLÓN" ) t2015b$provincia[i]<-"Castellón/Castelló"
if(t2015b$provincia[i]=="CIUDAD REAL" ) t2015b$provincia[i]<-"Ciudad Real"
if(t2015b$provincia[i]=="CÓRDOBA" ) t2015b$provincia[i]<-"Córdoba"
if(t2015b$provincia[i]=="A CORUÑA" ) t2015b$provincia[i]<-"Coruña (A)"
if(t2015b$provincia[i]=="CUENCA" ) t2015b$provincia[i]<-"Cuenca"
if(t2015b$provincia[i]=="GIRONA" ) t2015b$provincia[i]<-"Girona"
if(t2015b$provincia[i]=="GRANADA" ) t2015b$provincia[i]<-"Granada"
if(t2015b$provincia[i]=="GUADALAJARA" ) t2015b$provincia[i]<-"Guadalajara"
if(t2015b$provincia[i]=="GUIPÚZCOA" ) t2015b$provincia[i]<-"Gipuzkoa"
if(t2015b$provincia[i]=="HUELVA" ) t2015b$provincia[i]<-"Huelva"
if(t2015b$provincia[i]=="HUESCA" ) t2015b$provincia[i]<-"Huesca"
if(t2015b$provincia[i]=="JAÉN" ) t2015b$provincia[i]<-"Jaén"
if(t2015b$provincia[i]=="LEÓN" ) t2015b$provincia[i]<-"León"
if(t2015b$provincia[i]=="LLEIDA" ) t2015b$provincia[i]<-"Lleida"
if(t2015b$provincia[i]=="LA RIOJA" ) t2015b$provincia[i]<-"Rioja (La)"
if(t2015b$provincia[i]=="LUGO" ) t2015b$provincia[i]<-"Lugo"
if(t2015b$provincia[i]=="MADRID" ) t2015b$provincia[i]<-"Madrid"
if(t2015b$provincia[i]=="MÁLAGA" ) t2015b$provincia[i]<-"Málaga"
if(t2015b$provincia[i]=="MURCIA" ) t2015b$provincia[i]<-"Murcia"
if(t2015b$provincia[i]=="NAVARRA" ) t2015b$provincia[i]<-"Navarra"
if(t2015b$provincia[i]=="OURENSE" ) t2015b$provincia[i]<-"Ourense"
if(t2015b$provincia[i]=="ASTURIAS" ) t2015b$provincia[i]<-"Asturias"
if(t2015b$provincia[i]=="PALENCIA" ) t2015b$provincia[i]<-"Palencia"
if(t2015b$provincia[i]=="LAS PALMAS" ) t2015b$provincia[i]<-"Palmas (Las)"
if(t2015b$provincia[i]=="PONTEVEDRA" ) t2015b$provincia[i]<-"Pontevedra"
if(t2015b$provincia[i]=="SALAMANCA" ) t2015b$provincia[i]<-"Salamanca"
if(t2015b$provincia[i]=="STA. C. TENERIFE" )
t2015b$provincia[i]<-"Santa Cruz de Tenerife"
if(t2015b$provincia[i]=="CANTABRIA" ) t2015b$provincia[i]<-"Cantabria"
if(t2015b$provincia[i]=="SEGOVIA" ) t2015b$provincia[i]<-"Segovia"
if(t2015b$provincia[i]=="SEVILLA" ) t2015b$provincia[i]<-"Sevilla"
if(t2015b$provincia[i]=="SORIA" ) t2015b$provincia[i]<-"Soria"
if(t2015b$provincia[i]=="TARRAGONA" ) t2015b$provincia[i]<-"Tarragona"
if(t2015b$provincia[i]=="TERUEL" ) t2015b$provincia[i]<-"Teruel"
if(t2015b$provincia[i]=="TOLEDO" ) t2015b$provincia[i]<-"Toledo"
if(t2015b$provincia[i]=="VALENCIA" ) t2015b$provincia[i]<-"Valencia/València"
if(t2015b$provincia[i]=="VALLADOLID" ) t2015b$provincia[i]<-"Valladolid"
if(t2015b$provincia[i]=="VIZCAYA" ) t2015b$provincia[i]<-"Bizkaia"
if(t2015b$provincia[i]=="ZAMORA" ) t2015b$provincia[i]<-"Zamora"
if(t2015b$provincia[i]=="ZARAGOZA" ) t2015b$provincia[i]<-"Zaragoza"
```

```
  if(t2015b$provincia[i]=="CEUTA" ) t2015b$provincia[i]<-"Ceuta"
  if(t2015b$provincia[i]=="MELILLA" ) t2015b$provincia[i]<-"Melilla"
}

datos_f<-rbind(t2002b,t2003b,t2004b,t2005b,t2006b,t2007b,t2008b,t2009b,
               t2010b,t2011b,t2012b,t2013b,t2014b,t2015b,t2016b)

x<-read_xlsx("P_media.xlsx")
x<-rbind(c("Araba/Álava",44.375),x)
x<-x[,2]
x<-rbind(x,x,x,x,x,x,x,x,x,x,x,x,x,x,x)
datos_f=cbind(datos_f,x)

y<-read_xlsx("T_media.xlsx")
y<-rbind(c("Araba/Álava",11.808),y)
y<-y[,2]
y<-rbind(y,y,y,y,y,y,y,y,y,y,y,y,y,y,y)
datos_f=cbind(datos_f,y)

z<-read_xlsx("H_sol.xlsx")
z<-rbind(c("Araba/Álava",169.133),z)
z<-z[,2]
z<-rbind(z,z,z,z,z,z,z,z,z,z,z,z,z,z,z)
datos_f=cbind(datos_f,z)

h<-read_xlsx("km_carretera.xlsx")
h<-rbind(c("Araba/Álava",1463.33),h)
h<-h[,2]
h<-rbind(h,h,h,h,h,h,h,h,h,h,h,h,h,h,h)
datos_f=cbind(datos_f,h)

r<-read_xlsx("CCAA.xlsx")
r<-rbind(c("Araba/Álava","PVasco"),r)
r<-r[,2]
r<-rbind(r,r,r,r,r,r,r,r,r,r,r,r,r,r,r)
datos_f=cbind(datos_f,r)

colnames(datos_f)<-c("provincia","fallecidos_interurbana","fallecidos_urbana",
"year","numero_vehiculos","poblacion","precipitacion_media",
                     "temperatura_media","horas_sol_media","km_carretera","CCAA")

#Lectura de los datos

str(datos_f)

datos_f$fallecidos_interurbana<-as.numeric(datos_f$fallecidos_interurbana)
datos_f$fallecidos_urbana<-as.numeric(datos_f$fallecidos_urbana)
datos_f$numero_vehiculos<-as.numeric(datos_f$numero_vehiculos)
```

```
datos_f$poblacion<-round(as.numeric(datos_f$poblacion),digits = 0)
datos_f$precipitacion_media<-as.numeric(datos_f$precipitacion_media)
datos_f$temperatura_media<-as.numeric(datos_f$temperatura_media)
datos_f$horas_sol_media<-as.numeric(datos_f$horas_sol_media)
datos_f$km_carretera<-as.numeric(datos_f$km_carretera)

unique(datos_f$provincia)->provincias
table(datos_f$provincia)

# Tabla

save(datos_f,file = "datos.RData")
```

Bibliografía

[1] Anscombe, F.J. 1948. The transformation of poisson, binomial and negative-binomial data. *Biometrika*. 35, 3/4 (1948), 246–254.

[2] Atoche Calzada, P. 2017. Modelos de regresión con datos de conteo: Aplicación a competiciones deportivas. (2017).

[3] Calcaterra, E.M. 2017. *Una revisión de los modelos de conteo con excesos de ceros*. Disponible en http://www.iesta.edu.uy/wp-content/uploads/2018/01/pasantia_martinez_voucher.pdf.

[4] Dobson, A.J. and Barnett, A. 2008. *An introduction to generalized linear models*. Chapman; Hall/CRC.

[5] Hachuel, L.S., Boggio, G.S. and Harvey, G. 2010. Modelos alternativos para el análisis de datos de conteo con exceso de ceros. (2010).

[6] Hilbe, J.M. 2011. *Negative binomial regression*.

[7] Lawless, J.F. 1987. Negative binomial and mixed poisson regression. *Canadian Journal of Statistics*. 15, 3 (1987), 209–225.

[8] López-González, E. and Ruiz-Soler, M. 2011. Análisis de datos con el modelo lineal generalizado. una aplicación con r. *Revista española de pedagogía*. 248 (2011), 59–80.

[9] Luque-Calvo, P.L. 2017. *Escribir un trabajo fin de estudios con r markdown*. Disponible en http://destio.us.es/calvo.

[10] Madsen, H. and Thyregod, P. 2010. *Introduction to general and generalized linear models*. CRC Press.

[11] McCullagh, N., P. 1989. *Generalized linear models*. Disponible en http://www.utstat.toronto.edu/~brunner/oldclass/2201s11/readings/glmbook.pdf.

[12] Pino-Mejías, J.L. 2017. *Apuntes del máster data science & big data: Modelo de regresión de poisson y modelo de regresión con exceso de ceros*.

[13] Quinteiro, E.M.G. 2014. *Aplicación de modelos de regresión de poisson bivariados a los resultados de los partidos de la liga española de fútbol*.

[14] R Core Team 2013. *R: A language and environment for statistical computing*. R Foundation for Statistical Computing.

[15] William, J., Hilbe, J.M. and Hilbe 2012. *Generalized linear models and extensions*.

[16] Zeileis, A., Kleiber, C. and Jackman, S. 2008. Regression models for count data in r. *Journal of statistical software*. 27, 8 (2008), 1–25.