Máster en Ciencias Actuariales y Financieras

TRABAJO FINAL DE MÁSTER

Título: Métodos de Regularización *Lasso*, *Ridge* y *Elastic Net*: Una aplicación a los seguros de no vida

Autoría: José Ignacio von Lücken Giménez

Tutoría: Salvador Torra Porras

Curso académico: 2019-2021

UNIVERSITAT DE BARCELONA
Facultat d'Economia i Empresa

Màster de Ciències Actuarials i Financeres

Facultad de Economía y Empresa

Universidad de Barcelona

Trabajo Final de Máster

Máster en Ciencias Actuariales y Financieras

Métodos de Regularización Lasso, Ridge y Elastic Net: Una aplicación a los seguros de no vida

Autoría: José Ignacio von Lücken Giménez

Tutoría: Salvador Torra Porras

El contenido de este documento es de exclusiva responsabilidad del autor, quien declara que no ha incurrido en plagio y que la totalidad de referencias a otros autores han sido expresadas en el texto.

Agradecimientos

A Dios, por brindarme la fortaleza necesaria para culminar esta etapa y llegar a la meta.

A mis padres, por el apoyo incondicional en estos años de estudio, por ser siempre mi inspiración y mis ejemplos a seguir.

A mi hermana, por darme en todo momento sus palabras de apoyo y su amor.

A mi tutor Salvador Torra Porras, por su guía, acompañamiento, apoyo y predisposición durante el desarrollo del trabajo.

A mis profesores, por todo el conocimiento que me brindaron a lo largo de estos años de estudio.

A Natalia, por su soporte en todo este tiempo y su paciencia constante.

A mis compañeros, por las experiencias compartidas.

Métodos de Regularización *Lasso, Ridge* y *Elastic Net*: Una aplicación a los seguros de no vida

Máster en Ciencias Actuariales y Financieras
(2019-2021)

José Ignacio von Lücken Giménez
Tutor: Salvador Torra
Universidad de Barcelona

Resumen

La finalidad de este trabajo es el estudio de técnicas de regularización, tales como *Lasso*, *Ridge* y *Elastic Net* y su efecto sobre los seguros de no vida, en particular sobre el número de siniestros reclamados en pólizas de seguro de vehículos. Para lograr el objetivo, se presentan los contenidos teóricos necesarios para la comprensión de las técnicas y su posterior desarrollo en un caso práctico a fin de reflejar su utilidad.

Regularización, *Lasso*, *Ridge*, *Elastic Net*, Seguros

Abstract

The purpose of this work is the study of regularization techniques, such as *Lasso*, *Ridge* and *Elastic Net* and their effect on non-life insurance, particularly on the number of claims in motor insurance policies. To attain the objective, the theoretical contents necessary for understanding the techniques and a subsequent development in a practical case are presented in order to reflect their usefulness.

Regularization, *Lasso*, *Ridge*, *Elastic Net*, Insurance

ÍNDICE

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

El sector de seguros de automóviles representa el 28 % de las primas del ramo de no vida en España, siendo el sector más importante del mismo (ICEA, 2021).

Las compañías de seguros, para poder tarifar los productos de cobertura no vida, necesitan estimar la cuantía de los siniestros reclamados, así como la frecuencia de los mismos. El presente trabajo focaliza el esfuerzo en el estudio de una de las variables anteriores, la frecuencia de los siniestros, a partir de la tipología econométrica denominada *count data*.

A lo largo de los años, se han desarrollado diferentes metodologías para el análisis de la siniestralidad: Cameron y Trivedi (2013) mencionan que los modelos GLM ha sido un punto clave en lo que respecta al *count data*, siendo una primera aproximación en el contexto de seguros de no vida a través de Poisson ya que, tal y como indican Mihaela y Danut (2015), la estimación de la frecuencia de siniestros tiene a priori una estructura Poisson. Sin embargo, la distribución de Poisson no recoge bien la sobredispersión que puede existir en la estructura de datos, lo cual se puede manejar utilizando otras metodologías (Ismail y Jemain, 2007).

Con el avance de la tecnología, cada vez se ha hecho más accesible el trabajo con grandes bases de datos y metodologías que antes eran más complicadas aplicarlas. Noll et al. (2020) presentan diversos métodos de aprendizaje supervisado para el análisis del número de reclamaciones por siniestros generados en seguros de automóviles (árboles de decisión, *boosting* y redes neuronales). En el presente trabajo se ha optado por tomar un enfoque que analice tres técnicas de regularización utilizadas ampliamente en *machine learning*: *Lasso, Ridge* y *Elastic Net*. Estas técnicas adicionan una penalización extra a la estimación de los parámetros, a fin de buscar un equilibrio entre la simplicidad y la precisión (Navarro, 2019).

Devriendt et al. (2017) y Garrido et al. (2016) han trabajado con las técnicas mencionadas, por lo que ha motivado a realizar una comparación entre el comportamiento de los modelos utilizando Poisson y agregando las regularizaciones estudiadas. De forma complementaria, se ha estudiado el comportamiento de los coeficientes incorporando el inflado de ceros

(tanto con y sin la regularización *Lasso*), motivado por los trabajos realizados por Boucher et al. (2007) y Yip y Yau (2005) donde mencionan que la gran cantidad de ceros que se presentan en las bases de datos de siniestros de las compañías propicia considerar modelos inflados de cero.

La estructura del trabajo es la siguiente: en primer lugar se realiza una descripción de las bases teóricas sobre las técnicas de *machine learning* que se utilizarán, así como una breve mención a los modelos GLM, las normas que utilizan los métodos de regularización y la técnica *cross validation*.

Posteriormente, se desarrolla un caso práctico a fin de aplicar las técnicas presentadas y realizar una comparativa que nos permita observar las ventajas de las nuevas técnicas especificadas. Dicho estudio se realiza utilizando el lenguaje R ya que, como se mencionará posteriormente, presenta gran cantidad de paquetes para la implementación de *Lasso*, *Ridge* y *Elastic Net*.

Finalmente se exponen las conclusiones obtenidas y futuras líneas de investigación posibles incentivadas gracias a la realización del presente trabajo.

Capítulo 2

Marco Teórico

A continuación se presentarán los principios teóricos necesarios para la comprensión de la metodología a utilizar. En primer lugar, la definición matemática de las normas ℓ_p debido a su implicancia directa en los métodos de regularización *Lasso*, *Ridge* y *Elastic Net*. En segundo lugar, se realiza una breve mención a los modelos GLM Poisson y al modelo Poisson inflado de ceros debido a que son los modelos base que se utilizarán para la aplicación de las metodologías estudiadas.

Posteriormente se introduce la justificación del uso de los métodos especificados y el desarrollo teórico de los mismos, así como la descripción de la validación cruzada y *cyclic coordinate descent* (técnicas utilizadas para la determinación de los modelos).

Por último se exponen los criterios de comparación que se utilizarán en el capítulo 3 cuando se desarrolle el caso práctico estudiado.

2.1 Normas ℓ_p

Una norma permite estudiar la distancia existente entre dos puntos (o en otras palabras la longitud del vector); esto permitirá cuantificar la separación entre los mismos. La Teoría de la Medida, parte de las matemáticas, estudia dichas normas a fin de establecer diferentes aspectos importantes para el uso de las mismas.

Una norma es una función vectorial $v : V \to \mathbb{R}$, donde V es un espacio vectorial y v satisface las siguientes condiciones:

- $x \neq 0 \Rightarrow v(x) > 0$
- $x = 0 \Rightarrow v(x) = 0$
- $v(\alpha x) = |\alpha| v(x),\ \forall \alpha \in \mathbb{R}$
- $v(x + y) \leq v(x) + x(y),\ \forall x,\ y \in V$

Se estudiará el caso particular de las normas ℓ_p, donde $1 \leq p$. A fin de establecer una notación más sencilla se considera que dicha norma viene establecida por $\|x\|_p$, siendo

$$\|x\|_p = \begin{cases} \left(\sum_{i=1}^{N} |x_i|^p \right)^{1/p}, & \text{si } p < \infty \\ max\left\{ |x_i| \, : i \in \mathbb{N}, \, k \leq N \right\}, & \text{si } p = \infty \end{cases} \tag{2.1}$$

En particular, se utilizarán en las normas ℓ_1 y ℓ_2. Estas definiciones tendrán importancia en la sección 2.3 ya que son la base de las técnicas de regularización estudiadas.

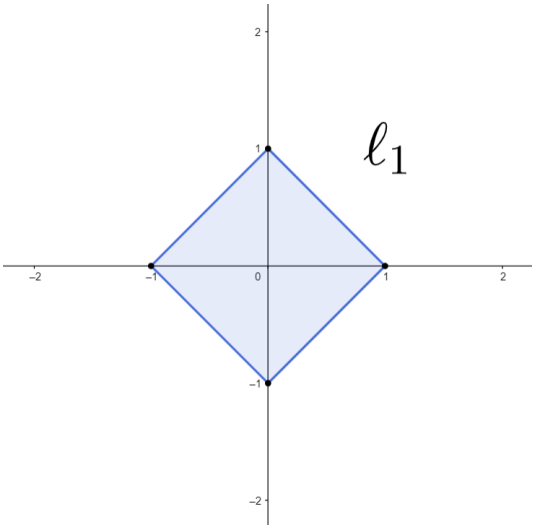
2.1.1 Norma ℓ_1

La norma ℓ_1 se define de la siguiente manera:

$$\|x\|_1 = \sum_{i=1}^{p} |x_i| \tag{2.2}$$

A modo de ejemplo, si $\|x\|_1 \leq 1$ en $\mathbb{R}^2$ se tiene el conjunto $\{x \in \mathbb{R}^2 \, : \|x\|_1 \leq 1\}$, que es un rombo con los vértices en los ejes de coordenadas, tal y como se observa en la siguiente figura:

Figura 2.1: Puntos del plano $\mathbb{R}^2$ tal que $\|x\|_1 \leq 1$.



Fuente: Hastie et al. (2015). Elaboración propia.

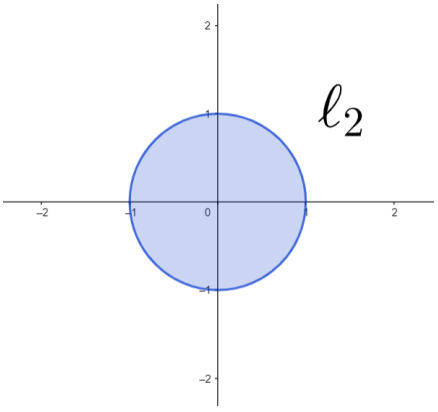
2.1.2 Norma ℓ_2

La norma ℓ_2 se encuentra definida por:

$$\|x\|_2 = \sqrt{\left(\sum_{i=1}^{p}(x_i)^2\right)} \tag{2.3}$$

De manera análoga, si se grafican los puntos que cumplen la condición $\|x\|_2 \leq 1$ se obtiene un círculo centrado en el punto $(0, 0)$.

Figura 2.2: Puntos del plano $\mathbb{R}^2$ tal que $\|x\|_2 \leq 1$.



Fuente: Hastie et al. (2015). Elaboración propia.

2.2 Modelos Lineales Generalizados

En la presente sección se realiza una breve descripción de los modelos lineales generalizados o GLM (*Generalized Linear Model*). Se trata de una clase de modelos que permiten cuantificar la relación funcional entre una variable respuesta y unas variables explicativas, donde la relación entre ambas es la siguiente:

$$g(y) = \beta X + \varepsilon \tag{2.4}$$

y

$$g(\mu_i) = \eta_i = \beta X_i \tag{2.5}$$

Siendo $g(\cdot)$ la función de enlace o *link*, μ_i el valor esperado de la i-ésima variable respuesta y η_i la componente i-ésima del predictor lineal; así se abre un gran abanico de posibilidades para encontrar la relación existente entre las variables explicativas y la variable respuesta.

En lo que refiere a la frecuencia de siniestralidad, la distribución de Poisson es una de las más utilizadas ya que permite realizar un recuento; siendo $g(\mu_i) = ln(\mu_i)$ (Ayuso et al., 2020). Cuando se trabaja con datos que tienen una gran cantidad de ceros existe una alternativa muy valida denominada Poisson inflado de ceros ya que, tal y como su nombre lo indica, tiene en consideración variables dependientes donde un amplio número de respuestas son iguales a cero.

2.2.1 Poisson

Entrando en mayor detalle en el estudio de GLM utilizando Poisson, se menciona que la función de probabilidad está determinada por:

$$P(Y_i = y_i) = \frac{exp\left(-\lambda_i\right) {\lambda_i}^{y_i}}{y_i!} \tag{2.6}$$

Y, reemplazando λ_i se obtiene:

$$P(Y_i = y_i) = \frac{exp\left(-exp\left(\sum_{j=1}^{p} x_{ij}\beta_j\right)\right) exp\left(\sum_{j=1}^{p} x_{ij}\beta_j\right)^{y_i}}{y_i!} \tag{2.7}$$

Ayuso et al. (2020) menciona que un inconveniente de Poisson es que se trata de una distribución con equidispersión y de esta forma algunas variables que se incorporan en realidad no son relevantes. Por este motivo, existen alternativas distribucionales que son interesantes considerarlas.

2.2.2 Poisson inflado de ceros

El modelo Poisson inflado de ceros fue desarrollado por Lambert (1992) y se puede entender como una mixtura de dos distribuciones, en el sentido de que, siendo y_i la variable respuesta, será igual a cero con una probabilidad igual a p_i y se distribuirá según una Poisson con media λ_i con probabilidad $1 - p_i$; de esta forma se tiene:

$$\begin{cases} P[y_i = 0] = p_i + (1 - p_i)e^{-\lambda_i} \\ P[y_i = k] = (1 - p_i)\dfrac{e^{-\lambda_i}\lambda_i^k}{k!}, \qquad con\ k = 1, 2, 3, ... \end{cases} \tag{2.8}$$

Se puede observar que cuando $P[y_i = 0]$, se está adicionando $p_i\left(1 - e^{-\lambda_i}\right)$ a la distribución de Poisson. De este modo, cuando se analice el caso práctico en el capítulo 3, se verá que se tiene tanto una componente de recuento como una componente que corresponde al inflado de ceros. Además, vale la pena mencionar que en la ecuación 2.8, a medida que p_i tiende a cero, el modelo tiende a ser una distribución de Poisson. A partir de aquí, se

mencionará a este modelo con las siglas ZIP para facilitar la lectura. Dicha abreviación proviene del nombre del modelo en inglés: *Zero-Inflated Poisson*.

2.3 Métodos de regularización

El presente apartado se centra en lo que corresponde a los métodos de regularización especificados, es decir, *Lasso, Ridge* y *Elastic Net*.

2.3.1 Antecedentes y justificativo de su uso

Se parte del modelo de regresión lineal típico, para luego profundizar en los modelos GLM. El modelo de regresión lineal típico se encuentra determinado a través de la siguiente fórmula:

$$y_i = \beta_0 + \sum_{j=1}^{p} x_{ij}\beta_j + e_i \tag{2.9}$$

donde β_0 y β_j son parámetros desconocidos, y_i es el valor objetivo, x_{ij} corresponden a las variables explicativas y e_i es el término del error. Una de las maneras de resolver la ecuación 2.9 es a través del método de mínimos cuadrados, el cual consiste en la búsqueda de los parámetros a estimar a través de la minimización de la siguiente función objetivo:

$$\min_{\beta_0, \beta} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 \tag{2.10}$$

Generalmente, en la ecuación 2.10 la mayoría de los parámetros estimados son distintos a cero. Al utilizar el método de mínimos cuadrados habitualmente se obtiene poco sesgo pero una varianza grande. Uno de los métodos aplicables para hacer frente a esto es considerar ciertos coeficientes iguales a cero o muy pequeños: de esta manera, se introduce un poco de sesgo pero se reduce la varianza y esto puede mejorar la precisión de predicción de forma global, tal y como menciona Ramirez (2016). Otro de los inconvenientes que se puede presentar es la interpretación de las variables predictoras en el sentido de que, si se estiman una gran cantidad de las mismas, se presenta un modelo complejo y su interpretabilidad se vuelve difícil. La simplificación de esto se puede conseguir a través de los métodos de regularización.

Debido a lo mencionado en el párrafo anterior, una opción es utilizar la norma ℓ_1 (ecuación 2.2), la cual permite regularizar el proceso de estimación. Una de las grandes ventajas de la utilización de esta norma es que permite una mayor penalización, con lo que se consigue

que una mayor cantidad de parámetros sean iguales a cero. Todo esto implica imponer una contracción (o *shrinkage* en inglés) adicional.

2.3.2 Aplicación de *Lasso*

El método *Lasso* (conocido así por su nombre en inglés *Least Absolute Shrinkage and Selection Operator*) ha sido desarrollado por Tibshirani (1996) y consiste en una combinación del método de mínimos cuadrados y una restricción del tipo ℓ_1, por lo que el problema se traduce a:

$$\min_{\beta_0,\,\beta} \frac{1}{N} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 \; sujeto\ a \|\beta\|_1 \leq t \tag{2.11}$$

donde t es un parámetro estimado y N es el tamaño de la muestra. Para trabajar de manera más sencilla, se considera $\mathbf{y}$ el vector de respuesta, y $\mathbf{X}$ la matriz de variables explicativas, por lo que lo anterior se podría escribir de la siguiente manera:

$$\min_{\beta_0,\,\beta} \frac{1}{N} \|\mathbf{y} - \beta_0\mathbf{1} - \mathbf{X}\beta_j\|_2^2 \; sujeto\ a\ \|\beta\|_1 \leq t \tag{2.12}$$

o también:

$$\min_{\beta_0,\,\beta} \frac{1}{N} \|\mathbf{y} - \beta_0\mathbf{1} - \mathbf{X}\beta_j\|_2^2 + \lambda\|\beta\|_1 \tag{2.13}$$

siendo λ el parámetro de contracción.

Tal y como menciona Hastie et al. (2015), el parámetro t presenta una gran influencia en el modelo, ya que al escoger un valor más grande habrá una mayor libertad y el modelo se adaptará mejor a los datos de entrenamiento. Por otra parte, con valores más pequeños se obtiene un modelo más disperso e interpretable. En otras palabras, a mayor valor se tiende a obtener una sobrestimación, mientras que si el valor de t es muy pequeño se corre el riesgo de no obtener la señal de salida correcta. Para la estimación t se suele utilizar la técnica conocida como *cross validation* o validación cruzada.

2.3.3 Validación Cruzada

La validación cruzada consiste en la división de la base de datos en K subgrupos distintos. Primeramente se separa uno de los subconjuntos y, con el resto de los subconjuntos, se ajusta el modelo para diferentes valores de t. Con cada uno de estos modelos estimados

se calcula el error cuadrático medio para el subconjunto que se ha apartado inicialmente. Posterior a esto, se repite el mismo proceso para cada uno de los subconjuntos y para cada valor de t se calcula el promedio de los K errores cuadráticos medios obtenidos, obteniéndose una curva de error.

En la sección 2.3.2 se mencionó que el parámetro t es posible estimar mediante *cross validation*, pero también es posible la estimación del parámetro λ de la ecuación 2.13 a través de la misma metodología. Este será el enfoque que se tendrá en cuenta en la aplicación en la sección 3.2.4.

2.3.4 *Cyclic Coordinate Descent*

Cyclic Coordinate Descent o descenso cíclico coordinado se trata de un un algoritmo iterativo donde se escoge una sola coordenada (de forma aleatoria) para actualizar y el resto queda fijo; luego se minimiza de forma univariante basándose en la coordenada escogida. Hastie et al. (2015) menciona que tiene una aplicación muy importante cuando se trabaja con *Lasso* debido a que aprovecha que varios coeficientes serán iguales a cero, por lo que no se moverán de ahí; es por esto que la mayor parte de las aplicaciones con *Lasso* utilizan este algoritmo. En general, no se busca tan solo un valor de λ, sino que interesa conocer las soluciones al problema de optimización sobre todo el rango de λ analizado. Con el fin de poder realizar esto, el algoritmo inicia con un valor de λ suficientemente grande como para que la solución óptima corresponda al vector de ceros. Posterior a esto, el algoritmo disminuye el valor de λ poco a poco y aparecen los diversos coeficientes (Hastie et al., 2015); esto se puede observar en el gráfico 3.7 de la aplicación práctica, dicho método es conocido como *pathwise coordinate descent*.

2.3.5 *Lasso* en GLM

Bühlmann y van de Geer (2011) indican que la aplicación del método *Lasso* en GLM consiste en la penalización del *log-likelihood* negativo con la norma ℓ_1:

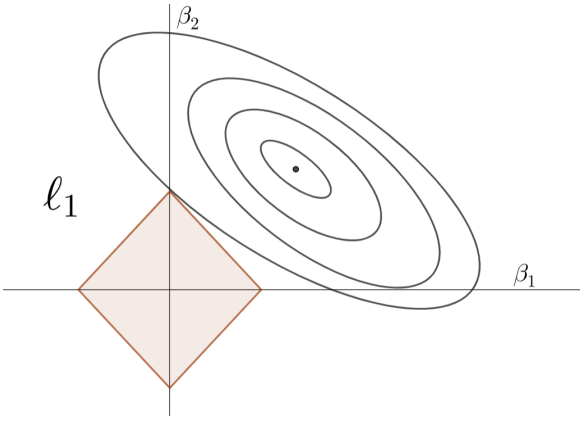
$$-\sum_{i=1}^{N} ln(p_{\mu,\beta}(Y_i|X_i)) \tag{2.14}$$

siendo μ el intercepto. Además, dicha expresión puede ser reescrita utilizando la función de pérdida ρ:

$$\frac{1}{N}\sum_{i=1}^{N} \rho_{\mu,\beta}\left(X_i, Y_i\right) \tag{2.15}$$

Por otra parte, si se grafican las líneas de contorno de la suma de cuadrados residual, siendo $\hat{\beta}$ el estimador de mínimos cuadrados, se tiene:

Figura 2.3: $\hat{\beta}$ con ℓ_1.



Fuente: Hastie et al. (2015). Elaboración propia.

Se puede observar que en los vértices, el coeficiente correspondiente al otro eje se vuelve cero. Cuando se trabaja con n coeficientes esto se traduce a un gráfico de más dimensiones, por lo que es imposible realizar el gráfico.

El objetivo de la aplicación de *Lasso* es que el modelo sea disperso (o *shrinkage*); de esta forma se reduciría la sobreestimación del modelo.

2.4 *Ridge* y *Elastic net*

Si bien el método *Lasso* presenta una ventaja a la hora de la penalización con la norma ℓ_1, también se han desarrollado otras técnicas de penalización con el fin de buscar mejorar la predicción para ciertos casos. Dichas variantes serán presentadas en este apartado, con el fin de tener las bases teóricas necesarias para el desarrollo de las mismas en el caso práctico.

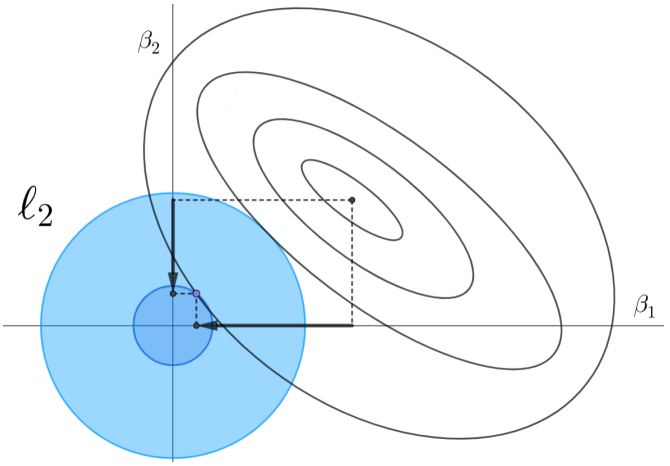
2.4.1 Método *Ridge*

Es un método que precede a *Lasso*, propuesto por Hoerl y Kennard (1970). Se trata de una minimización similar al método de mínimos cuadrados, pero los coeficientes son estimados de la siguiente manera:

$$\min_{\beta_0,\,\beta} \frac{1}{N}\|\mathbf{y} - \beta_0\mathbf{1} - \mathbf{X}\beta_j\|_2^2 + \lambda|\beta|^2 \tag{2.16}$$

Se puede observar que la diferencia con *Lasso* se encuentra en el segundo término ya que es aquí donde cambia la norma utilizada. Si se grafican las líneas de contorno de la suma de cuadrados residual, siendo $\hat{\beta}$ el estimador de mínimos cuadrados, se tiene:

Figura 2.4: $\hat{\beta}$ con ℓ_2.



Fuente: Elaboración propia.

En este caso, se puede observar el comportamiento de β_1 y β_2 cuando cambia el contorno de la suma de cuadrados residual con la estimación de los parámetros a través de *Ridge*. En este caso en particular, se aprecia que el coeficiente β_1 decrementa mucho más rápido que el coeficiente β_2. Los coeficientes tienden a cero, pero nunca llegan a este valor (Meulman et al., 2019). En este caso, la metodología busca reducir el peso que tienen las variables menos significativas. Esta diferencia con *Lasso* se verá en el caso práctico.

2.4.2 *Elastic Net*

Zou y Hastie (2005) hacen mención a que si bien *Lasso* y *Ridge* aportan herramientas interesantes a la hora de la determinación de los coeficientes en un modelo, una mixtura de ambos proporcionaría ventajas a considerar: Uno de los inconvenientes que presenta *Lasso* es que no trabaja bien cuando existen variables altamente correlacionadas: en el caso de que se deba elegir por la importancia, *Lasso* solamente escogería una de las variables correlacionadas. En lo que respecta a *Ridge* presenta el inconveniente de no producir un modelo parsimonioso ya que conserva todas las variables predictoras. Es por esto que desarrollaron *Elastic Net*, la cual combina las normas $\|\beta\|_1$ y $\|\beta\|_2$ de tal forma a resolver

$$\min_{\beta_0,\,\beta} \frac{1}{2}\sum_{i=1}^{N}\left(y_i - \beta_0 - \beta^T x_i\right)^2 + \lambda\left[\frac{1}{2}(1-\alpha)\|\beta\|_2^2 + \alpha\|\beta\|_1\right] \tag{2.17}$$

siendo $\alpha \in (0,1)$ un parámetro que determina la influencia de cada una de las normas. En otras palabras, *Elastic Net* es una combinación del método *Lasso* y *Ridge*, donde cada una de estas tiene un peso α. Meulman et al. (2019) hace mención a que "*con Elastic Net se obtienen modelos dispersos gracias al uso de Lasso y fomentando la agrupación de variables debido al uso de Ridge*".

2.5 Medidas de error

Para poder realizar una comparación entre los diferentes modelos que se presentan, se deben establecer ciertas medidas de error que permitan seleccionar el mejor modelo. Además, estas medidas permitirán escoger la λ óptima al trabajar con *Lasso*, *Ridge* y *Elastic Net*.

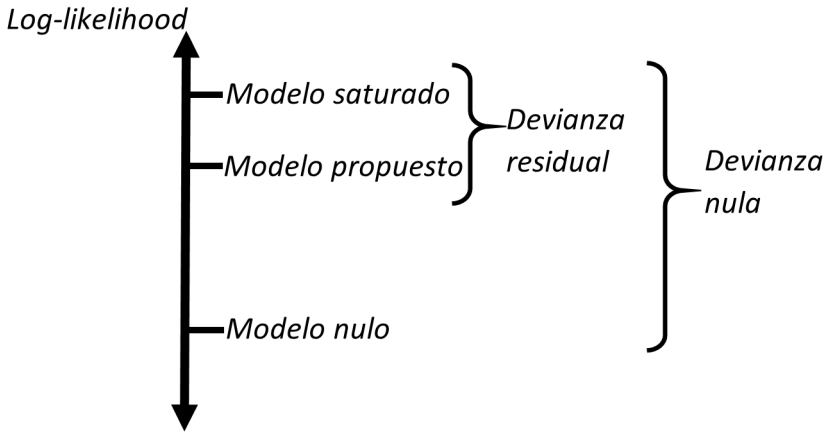
2.5.1 Devianza

La devianza es una medida de bondad del ajuste ampliamente utilizada. Tal y como mencionan de Jong y Heller (2008), la devianza juega un papel muy importante a la hora de medir la bondad del ajuste, en particular en los modelos de dispersión exponencial y GLM. Por una parte, la devianza residual se define como una medida de distancia entre el modelo saturado (modelo con igual cantidad de parámetros y observaciones) y el ajustado:

$$\Delta = 2\left(\check{\ell} - \hat{\ell}\right) \tag{2.18}$$

donde la medida de distancia utilizada es el *log-likelihood*. Por otra parte, para el cálculo de la devianza nula, se debe aplicar 2.18 entre el modelo saturado y el modelo nulo. Se espera que $\hat{\ell}$ este cercano a $\check{\ell}$, por lo que un valor alto de la devianza residual implica un mal ajuste del modelo. Esto se sintetiza en el siguiente gráfico.

Figura 2.5: Devianza



Fuente: Elaboración propia.

En particular, como se trabajará con la Poisson, se menciona que la devianza está determinada por la ecuación 2.19 (Agresti, 2015):

$$D(Y, \hat{\lambda}) = 2 \sum_i \left[y_i log \left(\frac{y_i}{\hat{\lambda}_i} \right) - y_i + \hat{\lambda}_i \right] \tag{2.19}$$

2.5.2 Criterio de Información de Akaike y Bayesiano

Introducido por Akaike (1973), el criterio de información de Akaike (o también conocido como AIC) consiste en un método utilizado de forma extensa para poder realizar una comparación entre modelos; esto es debido a que calcula el logaritmo de la máxima verosimilitud y además incorpora una penalización según la cantidad de parámetros que se utiliza. Viene determinada por:

$$AIC = -2\, log\ell(\hat{\theta}) + 2p \tag{2.20}$$

siendo $\hat{\theta}$ la máxima verosimilitud estimada del modelo y p el número de parámetros.

Por otra parte, el criterio de información Bayesiano fue desarrollado por Schwarz (1978) y difiere con AIC introduciendo el efecto del tamaño de la muestra, es decir:

$$BIC = -2\, log\ell(\hat{\theta}) + p\, log(n) \tag{2.21}$$

Comparando ambos criterios, vale la pena destacar lo mecionado por Brewer et al. (2016), quienes resaltan que AIC busca minimizar el error cuadrático medio mientras BIC tiende a seleccionar el modelo "real" a medida que el tamaño de la muestra crece. Sin embargo, existe otra versión de AIC, conocido como Akaike corregido, el cual introduce el efecto del tamaño de la muestra n. El mismo viene determinado por:

$$AIC_c = AIC + \frac{2p(p+1)}{n-p-1} \tag{2.22}$$

Burnham y Anderson (2003) recomiendan utilizar dicho criterio en vez del AIC estándar, por lo que también se utilizará este para realizar la comparación de los modelos.

Capítulo 3

Caso práctico

Una vez presentado en el capítulo 2 la base teórica necesaria, se procederá a presentar una aplicación en el ámbito de los seguros de no vida. El lenguaje de programación utilizado para esto es R, ya que se trata de una herramienta de gran ayuda por la existencia de paquetes con funciones ya desarrolladas que permiten el cálculo de cada uno de los elementos necesarios para el análisis. En cuanto a los paquetes utilizados resaltan *glmnet* y *mpath*, los cuales permiten la implementación de los métodos estudiados (se entrará en mayor profundidad en la sección 3.2.3).

Respecto a la variable que se desea modelar, será el número de siniestros reportados utilizando, como variables explicativas, las variables a introducir en el siguiente apartado.

3.1 Análisis de datos

La base de datos utilizada corresponde a seguros automovilísticos dentro de un año, la cual contiene 205.432 pólizas. Dicha base de datos se ha obtenido del paquete CASdatasets creado por Dutang y Charpentier (2020) como complemento del libro *Computational Actuarial Science with R* de Charpentier (2015). Las variables a utilizar dentro de la base de datos se presentan en la tabla 3.1.

En particular, la variable *Power* tiene como elementos $\{d, e, f, g, h, i, j, k, l, m, n, o\}$, *Brand* se encuentra categorizada por {"Fiat", "Japanese (except Nissan) or Korean", "Mercedes, Chrysler or BMW", "Opel, General Motors or Ford", "Renault, Nissan or Citroen", "Volkswagen, Audi, Skoda or Seat", "other"}. Además, la variable *Region* tiene como factores $\{A, B, C, D, E, F, G, H, I, J\}$. Así mismo, la variable *Exposure* se utilizará como *offset*[1] debido a que corresponde al periodo de exposición de la poliza en años, tal y como mencionan los autores del paquete utilizado Dutang y Charpentier (2020) y el autor del libro de referencia Charpentier (2015).

[1] Término de ajuste debido a la exposición variable de cada uno de los individuos (de Jong y Heller, 2008).

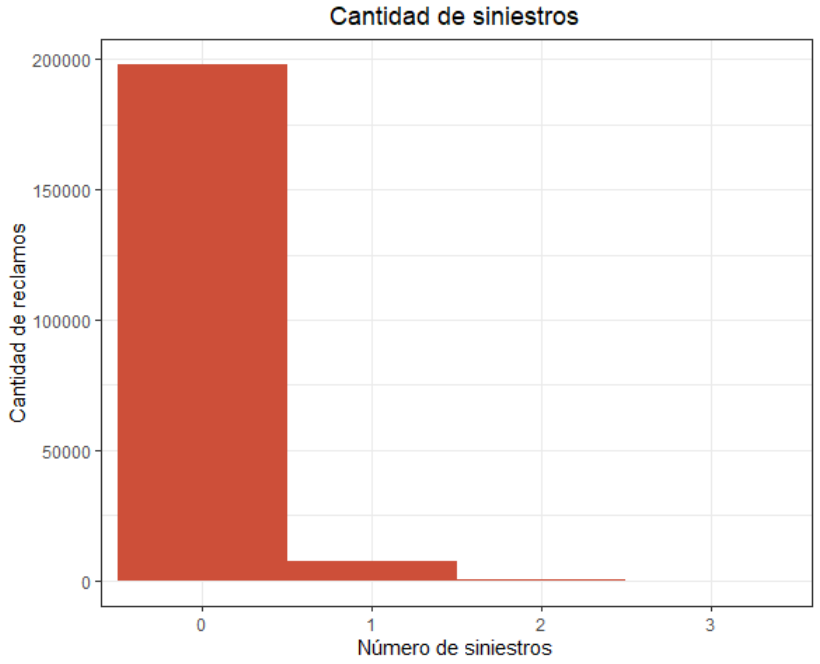
Tabla 3.1: Variables utilizadas

Variable	Tipo	Intervalo	Descripción
ClaimNb	numérica	{0, 1, 2, 3}	Cantidad de siniestros declarados
Exposure	numérica	(0, 1)	Exposición
Power	categórica	12 categorías distintas	Potencia del vehículo
CarAge	numérica	(0, 25)	Antigüedad del vehículo
DriverAge	numérica	(18, 99)	Edad del conductor
Brand	categórica	7 categorías distintas	Marca del vehículo según como se encuentra categorizado
Gas	categórica	{Diesel, Regular}	Tipo de combustible
Region	categórica	10 categorías distintas	Región de la póliza

Fuente: Elaboración propia.

A continuación se presenta la distribución de siniestros dentro de la cartera analizada. Se observa que, tal y como se esperaba, se presenta una gran cantidad de casos donde no se han presentado reclamaciones (es decir, se tienen 0 siniestros), por lo que es reducida la cantidad de pólizas con reclamaciones realizadas.

Figura 3.1: Distribución de cantidad de siniestros

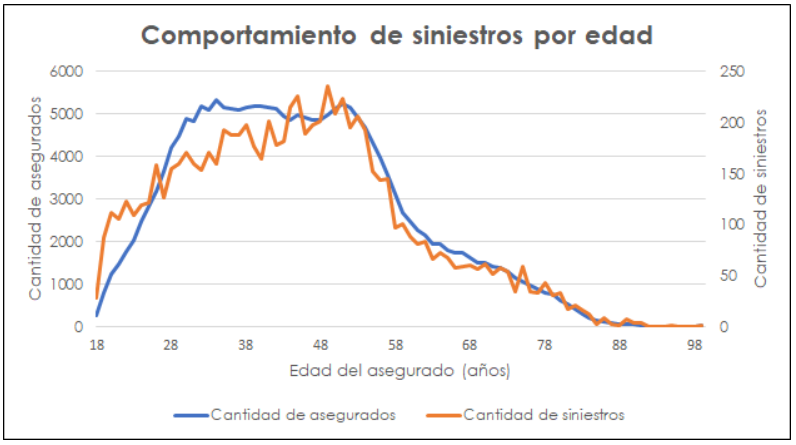


Fuente: Elaboración propia.

En cuanto a lo que respecta a la distribución de siniestros según la edad de los conductores, mediante el gráfico 3.2 se observa una mayor concentración entre 30 y 50 años, mientras que a medida que se incrementa la edad disminuyen los siniestros, de la misma manera que la cantidad de asegurados (según se observa en el mismo gráfico). Además, la cantidad de siniestros por edad sigue el comportamiento de la cantidad de asegurados. Otro aspecto

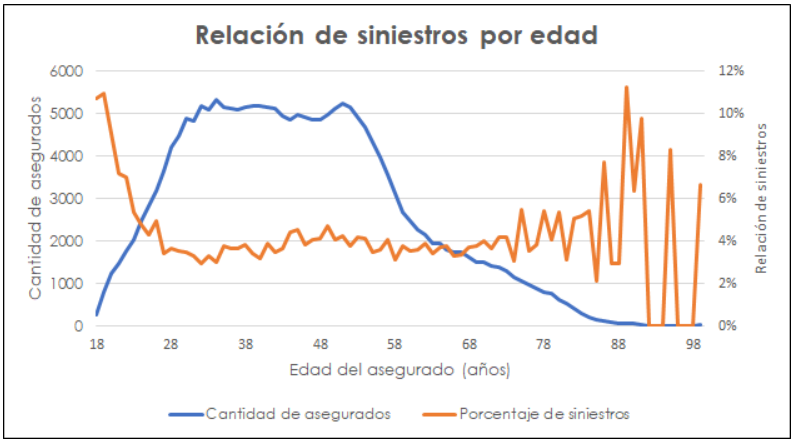
importante es que al calcular el porcentaje de siniestros por edad (Figura 3.3) se observa que entre 25 y 80 años se mantiene relativamente constante (alrededor de 4 % y 5 %), mientras que en edades mayores se presenta una variación muy grande (esto debido a la poca cantidad de asegurados en dicha franja).

Figura 3.2: Distribución de cantidad de siniestros según la edad del conductor



Fuente: Elaboración propia.

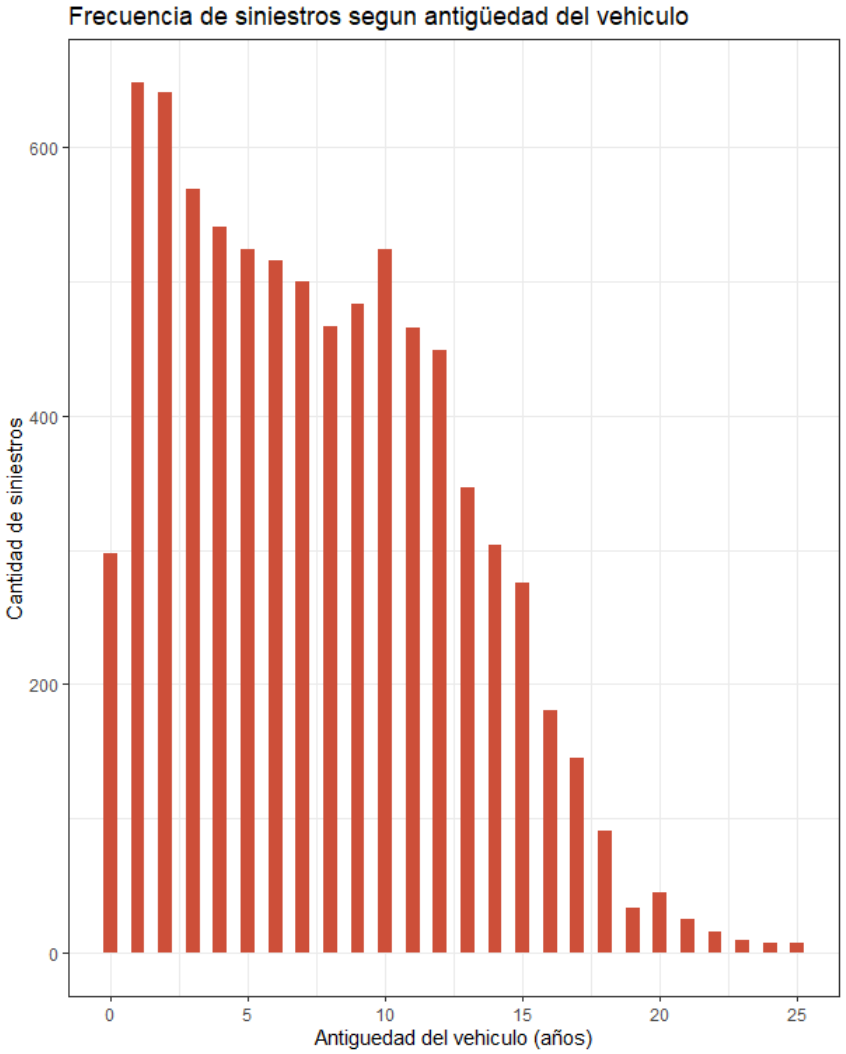
Figura 3.3: Porcentaje de siniestros según edad



Fuente: Elaboración propia.

Según la antigüedad del vehículo (Figura 3.4), se puede observar que en general es bastante uniforme entre 1 y 11 años, teniendo un máximo en 1 y 2 años; en cambio la cantidad de siniestros incurridos a partir de los 12 años disminuye considerablemente.

Figura 3.4: Distribución de siniestros según antigüedad del vehículo



Fuente: Elaboración propia.

3.1.1 Comparación de variables

Un factor que se debe tener presente a la hora de trabajar con la modelización es la colinealidad. Tal y como mencionan Dormann et al. (2012), la colinealidad se refiere a la existencia de una relación entre variables predictoras y esto puede conllevar a problemas de estimación debido al aumento de la varianza y selección incorrecta de los predictores. A fin de poder analizar esto, se realizarán tres análisis según los tipos de variables:

- Dos variables categóricas: V de Cramer.
- Dos variables numéricas: correlación de Pearson, Spearman y Kendall.
- Una variable categórica y una numérica: Se busca estimar la variable numérica a

través de variables *dummy* (correspondientes a la variable categórica) y se observa el porcentaje explicado por estas variables *dummy*.

En la tabla 3.2 se pueden observar los valores de la V de Cramer, los cuales miden la intensidad de relación entre variables (Shishkina et al., 2018). A fin de poder utilizar este análisis se generan variables *dummy* de las variables categóricas analizadas. Este test tiene un rango entre 0 y 1 (donde 0 indica que no existe asociación entre las dos variables). Se puede observar que se presentan siempre valores por debajo de 0,35, por lo que en todos los casos se considera que las variables son independientes (con ciertas consideraciones). Inicialmente se había considerado realizar el test de Chi Cuadrado y determinar la dependencia según el p-valor obtenido, sin embargo el gran tamaño de la base de datos genera problemas con dicho test, tal y como menciona Lin et al. (2013).

Tabla 3.2: Test V de Cramer para las variables categóricas

	Region	Gas	Brand	Power
Power	0,0485	0,3363	0,1945	
Brand	0,1887	0,0987		
Gas	0.1028			

Fuente: Elaboración propia.

En lo que respecta a las variables numéricas, se ha optado por trabajar con tres distintos coeficientes de correlación debido a que cada uno de ellos tiene diferentes ventajas: Si bien el coeficiente de correlación de Pearson es el más utilizado en general, las hipótesis que conlleva son muy restrictivas ya que considera la normalidad de las variables analizadas, tal y como puntualizan Hauke y Kossowski (2011). Los mismos autores mencionan que Spearman es no paramétrico (libre de distribución), por lo que no presenta este inconveniente. Además, el coeficiente de correlación τ de Kendall también es no paramétrico y también es muy utilizado. Los resultados se pueden observar en las tablas 3.3, 3.4 y 3.5, donde se deduce que no existe una fuerte correlación entre las variables analizadas.

Tabla 3.3: Matriz de correlación de Pearson

	Exposure	CarAge	DriverAge
Exposure	1,000	0,141	0,194
CarAge	0,141	1,000	-0,057
DriverAge	0,194	-0,057	1,000

Fuente: Elaboración propia.

Tabla 3.4: Matriz de correlación de Spearman

	Exposure	CarAge	DriverAge
Exposure	1,000	0,168	0,190
CarAge	0,168	1,000	-0,070
DriverAge	0,190	-0,070	1,000

Fuente: Elaboración propia.

Tabla 3.5: Matriz de correlación τ de Kendall

	Exposure	CarAge	DriverAge
Exposure	1,000	0,121	0,133
CarAge	0,121	1,000	-0,048
DriverAge	0,133	-0,048	1,000

Fuente: Elaboración propia.

En cuanto al análisis de las variables categóricas frente a las numéricas, presentado en la tabla 3.6, se tienen porcentajes explicados de la variable numérica a través de la variable categórica por debajo de 18 % (un valor muy bajo para poder concluir que existe una colinealidad entre ambas).

Tabla 3.6: Matriz de relación entre variables categóricas y numéricas

	Power	Brand	Gas	Region
Exposure	0,0051	0,0511	0,0010	0,0694
CarAge	0,0106	0,1893	0,0178	0,0748
DriverAge	0,0015	0,0127	0,0057	0,0095

Fuente: Elaboración propia.

Debido al análisis efectuado, se realizarán los modelados considerando todas las variables mencionadas en la tabla 3.1.

3.2 Modelización

Para la modelización se han utilizado diferentes paquetes de R, tal y como se mencionó anteriormente. Se ha optado por presentar tres modelos distintos para el GLM-Poisson normal y los que correponden a la adición de la penalización *Lasso*, *Ridge* y *Elastic Net*: La primera de ellas considerando como variables explicativas todas aquellas mencionadas en la tabla 3.1, la segunda considerando las mismas variables y añadiendo la posible interacción entre las variables *Power* y *Brand* y finalmente la tercera consiste en que tan solo las variables numéricas se encuentren en el modelo de forma independiente y además se añaden las posibles interacciones de las variables categóricas. Importante mencionar que la variable *Exposure* se considera como el *offset* en cada uno de los modelos.

Adicional a esto, se ha realizado el modelado a través de ZIP y ZIP-*Lasso*. No se han podido realizar los modelos ZIP-*Ridge* ni ZIP-*Elastic Net* debido a la exigencia del paquete *mpath* en cuanto al procesamiento del ordenador.

3.2.1 GLM-Poisson

Para el modelado GLM-Poisson se ha utilizado la función *glm* del paquete *Stats*. Dicha función ofrece una gran cantidad de parámetros a escoger con el fin de encontrar el modelo buscado. En la tabla 3.7 se presentan tanto los valores estimados como la significatividad de cada uno de ellos. Se encuentran resaltados los coeficientes con un nivel de confianza de por lo menos 5 %. Es importante mencionar que las variables categóricas se trabajan como factores; es por esto que aparecen cada de uno de los posibles valores de dichas variables.

Tabla 3.7: GLM-Poisson-1

Coeficientes	Valor estimado	valor z	$Pr(>\|z\|)$)
(Intercept)	**-1,91161**	**-20,348**	**< 2e-16**
Powere	0,08076	1,767	0,077222
Powerf	**0,11283**	**2,534**	**0,011287**
Powerg	0,07937	1,793	0,072968
Powerh	**0,12559**	**1,988**	**0,046834**
Poweri	0,12733	1,769	0,076952
Powerj	0,13859	1,906	0,056603
Powerk	0,15569	1,630	0,103148
Powerl	0,06755	0,478	0,632670
Powerm	**0,41671**	**2,411**	**0,015893**
Powern	0,05778	0,228	0,819992
Powero	0,08483	0,355	0,722662
CarAge	**-0,01295**	**-4,753**	**0,000002**
DriverAge	**-0,01080**	**-11,579**	**< 2e-16**
BrandJapanese (except Nissan) or Korean	**-0,25833**	**-3,513**	**0,000444**
BrandMercedes, Chrysler or BMW	0,05646	0,647	0,517547
BrandOpel, General Motors or Ford	0,08914	1,224	0,220850
Brandother	-0,02362	-0,232	0,816217
BrandRenault, Nissan or Citroen	-0,05726	-0,896	0,370438
BrandVolkswagen, Audi, Skoda or Seat	0,04531	0,606	0,544785
GasRegular	-0,03603	-1,290	0,197023
RegionB	**-0,20751**	**-2,276**	**0,022819**
RegionC	**-0,22189**	**-3,639**	**0,000274**
RegionD	**-0,28242**	**-5,376**	**7,60E-08**
RegionE	**-0,27203**	**-2,153**	**0,031283**
RegionF	0,06951	1,196	0,231516
RegionG	0,00481	0,038	0,969295
RegionH	0,02160	0,306	0,759334
RegionI	**-0,13442**	**-2,133**	**0,032901**
RegionJ	-0,12686	-1,691	0,090822

Fuente: Elaboración propia.

Por otra parte, en lo que respecta al segundo modelo y tercer modelo, las tablas se encuentran en el anexo A. En cuanto al segundo modelo se puede ver que las posibles interacciones no son consideradas para la estimación ya que se observa que ninguna es significativa. Además, se observa que dejan de ser significativas *Powerf* y *Powerh* (Tabla A.1). Analizando el tercer modelo presentado (en el cual se plantea la posibilidad de que tan solo las variables numéricas se encuentren en el modelo de forma independiente y se agreguen las posibles interacciones de las variables categóricas) se observa que *Powerm* deja de ser significativa, pero aparecen las interacciones de algunos factores de dicha variable con el factor *other* de la variable *Brand* (Tabla A.2).

Tal y como se mencionó en la sección 2.5, para poder comparar los modelos propuestos se utilizan la devianza residual y criterios como AIC_c y BIC. En la tabla 3.8 se presentan los valores obtenidos para los tres modelos GLM-Poisson presentados. Se observa que en cuanto a los criterios AIC_c y BIC el más adecuado es el GLM-Poisson sin considerar las interacciones (es decir el primer modelo presentado). Esto es algo esperado ya que una gran parte de las interacciones no son significativamente distintas a cero. Sin embargo, teniendo en cuenta la devianza residual, se observa que es ligeramente mejor el modelo GLM-Poisson que considera algunas interacciones entre las variables categóricas (el tercer modelo), siendo la diferencia ínfima, de aproximadamente $0,7\,\%$ entre el mejor y el peor.

Tabla 3.8: Comparación GLM-Poisson

Modelo	AIC_c	BIC	Devianza residual
GLM-Poisson-1	51117,75	51774,45	39343
GLM-Poisson-2	51152,62	53253,97	39246
GLM-Poisson-3	51300,28	57297,26	39037

Fuente: Elaboración propia.

3.2.2 Poisson Inflado de Ceros

En la sección 3.1, correspondiente al análisis de las variables, se observó que la base de datos presenta un exceso de ceros; por esta razón se procede a modelizar el número de siniestros a través de ZIP. El paquete de R utilizado para esto es *pscl*, desarrollado y mantenido por Jackman (2020), cuya funcionalidad es mostrada en *Regression Models for Count Data in R* (Zeileis et al., 2008). Se presentan dos modelos propuestos: El primero de ellos es uno simple donde todos los ceros tienen la misma probabilidad de pertenecer al componente cero (Jackman, 2020), mientras que en el segundo se analiza si cada una de las variables es un componente de parte del inflado de ceros. En las tablas 3.9 y 3.10 se observan los valores estimados para coeficientes y se encuentran resaltados aquellos que son significativos a un nivel mínimo del $5\,\%$.

Se pueden apreciar diferencias notables: En el modelo de recuento estimado se observa que mientras que en el primer modelo propuesto utilizando ZIP se tiene en consideración algunos factores de la variable correspondiente a la potencia del vehículo y otras que corresponden a la región, al considerar la posibilidad de que todos los coeficientes tengan

influencia en el inflado de ceros deja de ser significativo la potencia del vehículo y tan solo es considerado una región en particular. A su vez, en la parte correspondiente al inflado de ceros el intercepto deja de ser significativo en el segundo modelo propuesto y la antigüedad del vehículo y la región H pasan a serlo.

Tabla 3.9: ZIP-1

Modelo de recuento

Coeficientes	Valor estimado	valor z	$Pr(>\|z\|)$
(Intercept)	**-1,10013**	**-9,838**	**< 2e-16**
Powere	0,08103	1,715	0,086407
Powerf	**0,11262**	**2,447**	**0,014423**
Powerg	0,07862	1,720	0,085431
Powerh	0,12318	1,885	0,059406
Poweri	0,12857	1,727	0,084171
Powerj	0,13357	1,778	0,075461
Powerk	0,15002	1,519	0,128856
Powerl	0,06503	0,446	0,655437
Powerm	**0,39930**	**2,209**	**0,027207**
Powern	0,05346	0,204	0,838307
Powero	0,09749	0,394	0,693314
CarAge	**-0,01244**	**-4,411**	**0,000010**
DriverAge	**-0,01095**	**-11,415**	**< 2e-16**
BrandJapanese (except Nissan) or Korean	**-0,25891**	**-3,404**	**0,000664**
BrandMercedes, Chrysler or BMW	0,05839	0,646	0,518501
BrandOpel, General Motors or Ford	0,09011	1,192	0,233403
Brandother	-0,02541	-0,241	0,809388
BrandRenault, Nissan or Citroen	-0,05741	-0,866	0,386603
BrandVolkswagen, Audi, Skoda or Seat	0,04609	0,593	0,553038
GasRegular	-0,03667	-1,268	0,204754
RegionB	**-0,20689**	**-2,191**	**0,028430**
RegionC	**-0,22437**	**-3,548**	**0,000387**
RegionD	**-0,28248**	**-5,186**	**2,15E-07**
RegionE	**-0,27598**	**-2,125**	**0,033589**
RegionF	0,07228	1,199	0,230456
RegionG	0,00486	0,037	0,970143
RegionH	0,01773	0,242	0,808487
RegionI	**-0,13539**	**-2,072**	**0,038284**
RegionJ	-0,12705	-1,633	0,102525

Modelo inflado de ceros

Coeficientes	Valor estimado	valor z	$Pr(>\|z\|)$
(Intercept)	**0,20412**	**2,067**	**0,0388**

Fuente: Elaboración propia.

Tabla 3.10: ZIP-2

Modelo de recuento

Coeficientes	Valor estimado	valor z	$Pr(>\|z\|)$
(Intercept)	**-1,58636**	**-7,074**	**1,50E-12**
Powere	-0,08629	-0,801	0,423200
Powerf	0,00733	0,071	0,943800
Powerg	-0,07544	-0,742	0,458000
Powerh	-0,07191	-0,510	0,610000
Poweri	0,03081	0,189	0,850300
Powerj	-0,10575	-0,612	0,540800
Powerk	-0,20361	-1,045	0,296200
Powerl	0,33467	1,021	0,307300
Powerm	-0,00660	-0,020	0,984400
Powern	0,27388	0,477	0,633700
Powero	0,66137	1,147	0,251400
CarAge	**0,04862**	**7,450**	**9,35E-14**
DriverAge	**-0,00830**	**-4,319**	**0,000016**
BrandJapanese (except Nissan) or Korean	**-0,39051**	**-2,258**	**0,023900**
BrandMercedes, Chrysler or BMW	-0,16161	-0,837	0,402600
BrandOpel, General Motors or Ford	0,09641	0,570	0,568900
Brandother	-0,01384	-0,058	0,953500
BrandRenault, Nissan or Citroen	-0,02460	-0,164	0,869700
BrandVolkswagen, Audi, Skoda or Seat	0,10285	0,591	0,554600
GasRegular	0,01223	0,188	0,851200
RegionB	-0,27975	-1,302	0,192900
RegionC	-0,22792	-1,633	0,102500
RegionD	-0,14687	-1,260	0,207700
RegionE	**-0,56166**	**-2,293**	**0,021800**
RegionF	0,10192	0,825	0,409600
RegionG	-0,06119	-0,236	0,813800
RegionH	-0,25801	-1,791	0,073300
RegionI	-0,11373	-0,825	0,409400
RegionJ	0,02297	0,137	0,890700

Modelo inflado de ceros

Coeficientes	Valor estimado	valor z	$Pr(>\|z\|)$
(Intercept)	-0,03756	-0,092	0,926900
Powere	-0,29747	-1,559	0,118900
Powerf	-0,15888	-0,891	0,372700
Powerg	-0,26113	-1,494	0,135200
Powerh	-0,36337	-1,379	0,168000
Poweri	-0,10583	-0,379	0,704800
Powerj	-0,40079	-1,158	0,246900
Powerk	-0,76175	-1,748	0,080400
Powerl	0,65629	1,114	0,265400
Powerm	-0,88881	-1,054	0,291700
Powern	0,52459	0,609	0,542400
Powero	1,15522	1,281	0,200200
CarAge	**0,11015**	**9,492**	**<2e-16**
DriverAge	-0,00172	-0,521	0,602100
BrandJapanese (except Nissan) or Korean	-0,26487	-0,674	0,500000
BrandMercedes, Chrysler or BMW	-0,44272	-1,182	0,237200
BrandOpel, General Motors or Ford	0,00870	0,029	0,976600
Brandother	0,01239	0,030	0,976000

BrandRenault, Nissan or Citroen	0,03729	0,143	0,886500
BrandVolkswagen, Audi, Skoda or Seat	0,09649	0,317	0,751400
GasRegular	0,07579	0,634	0,525900
RegionB	-0,29681	-0,694	0,488000
RegionC	-0,17980	-0,679	0,496900
RegionD	0,09903	0,451	0,651900
RegionE	-0,60568	-0,968	0,332800
RegionF	0,07584	0,305	0,760700
RegionG	-0,23273	-0,459	0,646500
RegionH	**-0,67293**	**-2,006**	**0,044900**
RegionI	-0,06982	-0,271	0,786700
RegionJ	0,14456	0,494	0,621300

Fuente: Elaboración propia.

En lo que respecta a los criterios utilizados para la selección de los modelos, es importante mencionar que la devianza (residual) es una medida definida para los modelos lineales generalizados, no siendo posible su aplicación a los modelos ZIP, ya que no lo son estrictamente debido a la mixtura que se realiza para la adición del inflado de ceros; sin embargo, se han desarrollado medidas similares para el análisis de los modelos ZIP, tales como el desarrollado por Martin y Hall (2016). A pesar de esto, a fin de unificar el mismo criterio y no considerar otros tipos de test, no se ha calculado esta medida. En la tabla 3.11 se puede apreciar que, comparando con los resultados obtenidos hasta ahora con Poisson, se tienen valores inferiores en los criterios de información y esto es un resultado para inferir que la aproximación con los modelos ZIP es mejor (en particular, con el segundo generado).

Tabla 3.11: Comparación ZIP

Modelo	AIC_c	BIC
ZIP-1	50958,2	51266,5
ZIP-2	50369,8	50966,5

Fuente: Elaboración propia.

3.2.3 Paquetes *glmnet* y *mpath*

Se han utilizado estos dos paquetes de R, desarrollados por Friedman et al. (2010) y Wang (2021) respectivamente, debido a la disponibilidad de una gran cantidad de funciones eficientes para la aplicación de las regularizaciones estudiadas.

En relación al paquete *glmnet*, existen dos funciones principales, las cuales son *glmnet* y *cv.glmnet*. La primera de ellas permite el ajuste del GLM a través de la máxima verosimilitud penalizada y permite al usuario introducir diversos parámetros, tales como $0 \leq \alpha \leq 1$ para la selección entre *Lasso*, *Ridge* y *Elastic Net*, la familia de la distribución (en nuestro caso en particular Poisson) y el número de valores de λ. La función *cv.glmnet* permite al usuario la estimación utilizando validación cruzada; acepta los mismos parámetros

anteriores y también la selección de cantidad de subgrupos utilizados para la validación cruzada, donde para el caso práctico se ha determinado trabajar con 5 subgrupos.

Por lo que refiere a *mpath*, se trata de un paquete que necesita mayores requerimientos computacionales, lo cual se debe tener presente a la hora de trabajar con el mismo. Debido a esto, si bien tiene funciones para la estimación de otros modelos, solamente se empleará para el cálculo de Poisson inflado de ceros con penalización. Con esta finalidad se utiliza la función *zipath* que ajusta un modelo inflado de ceros con penalización, siendo los parámetros más importantes la fórmula que debe seguir, la familia y la cantidad de λ. Tal y como ocurre con el paquete *glmnet*, aquí se dispone de una función para la estimación a través de validación cruzada. Dicha función es *cv.zipath*, la cual tiene los mismos parámetros principales.

3.2.4 GLM-Poisson con *Lasso*, *Ridge* y *Elastic Net*

En el presente apartado se estudian los modelos considerando las regularizaciones *Lasso*, *Ridge* y la combinación de ambas (*Elastic Net*). Se ha optado por considerar un punto intermedio entre ambas, es decir, $\alpha = 0,5$ debido a que las variaciones al cambiar los valores de α para este caso en particular tienden a ser muy similares (lo cual se puede observar ejecutando el código disponible en el anexo C). Tal y como se ha hecho en el apartado 3.2.1, se han probado diferentes combinaciones de las variables (considerando o no la interacción entre ellas), a fin de buscar cual es el modelo más adecuado para la base de datos que se analiza. Además se estimó el valor óptimo de λ según el método especificado en la sección 2.3.3 y se ha dividido la base de datos en cinco subconjuntos de forma aleatoria sobre los cuales se realizan los cálculos correspondientes.

Antes de presentar los resultados, se considera que es importante hacer una mención en lo que respecta a los intervalos de confianza cuando se trabaja con estas regularizaciones. Goeman et al. (2018), autores del paquete *penalize* que permite la estimación utilizando *Lasso* y *Ridge*, indican lo siguiente:

> *Es una cuestión muy natural preguntar por los errores estándar de los coeficientes de regresión u otras cantidades estimadas. En principio, estos errores estándar pueden calcularse fácilmente, por ejemplo, utilizando bootstrap. Sin embargo, este paquete no los proporciona deliberadamente. La razón es que los errores estándar no son muy significativos para las estimaciones fuertemente sesgadas, como las que surgen de los métodos de estimación penalizada. La estimación penalizada es un procedimiento que reduce la varianza de los estimadores introduciendo un sesgo sustancial. El sesgo de cada estimador es, por tanto, un componente importante de su error cuadrático medio, mientras que su varianza puede contribuir sólo en una pequeña parte.*

Tibshirani (2011) indica que uno de los retos en cuanto a *Lasso* es el desarrollo de herramientas utilizadas en estadística, poniendo especial énfasis en el error estándar y p-valores.

Sin embargo, una posible solución es trabajar con *Lasso* Bayesiano, el cual provee un intervalo estimado que permite la elección de las variables (Park y Casella, 2008).

Otra acotación importante es que utilizando *cross validation* se obtienen diferentes λ, siendo las dos más interesantes a estudiar λ_{min} y λ_{1se}. La primera de ellas consiste en el valor de λ que minimiza la medida de error utilizada para el cálculo (en este caso en particular la devianza residual), mientras que la segunda es la mayor λ que se encuentra a "un error estándar" de la medida de error mínima calculada. Se consideran ambas ya que si se puede permitir no ser tan estricto a la hora de acotar el error se consigue una reducción de variables mucho mayor.

En la tabla 3.12 se presentan los valores estimados sin considerar la interacción posible existente entre las variables categóricas para los tres métodos estudiados (además se recuerda que no se ofrecen los resultados de la contrastación de los parámetros). En la misma, el "." indica que dicho predictor no se considera, pero se ha optado por dejar en la tabla a modo de poder visualizar la diferencia entre *Lasso* y *Ridge*. Se puede ver que los resultados son bastante similares entre *Lasso* y *Elastic Net*: las pocas diferencias que se encuentran son en los valores de los parámetros estimados, pero estos son muy similares. Se puede apreciar que si se considera λ_{min} se eliminan 7 variables, mientras que si se tiene en cuenta λ_{1se} desaparecen todos los coeficientes y solo tiene peso el intercepto. En cuanto a *Ridge* se puede ver que no se descarta ninguna variable (algo que se esperaba, tal y como se mencionó en la sección 2.4.1), sin embargo en todas ellas se tiene en común que la influencia es muy pequeña cuando se considera λ_{1se}.

Tabla 3.12: Coeficientes del primer modelo según *Lasso*, *Elastic Net* y *Ridge*

Coeficientes	Lasso		Ridge		Elastic Net	
	λ_{min}	λ_{1se}	λ_{min}	λ_{1se}	λ_{min}	λ_{1se}
(Intercept)	-1,99935	-2,65353	-2,02469	-2,65353	-2,00822	-2,65353
Powere	.	.	0,05322	1,36E-39	.	.
Powerf	0,01947	.	0,08089	1,42E-39	0,01930	.
Powerg	.	.	0,04798	-1,12E-39	.	.
Powerh	0,00884	.	0,08783	1,59E-39	0,00852	.
Poweri	0,00815	.	0,09355	1,47E-39	0,00804	.
Powerj	0,01136	.	0,09675	1,81E-39	0,01069	.
Powerk	0,01431	.	0,11574	2,85E-39	0,01389	.
Powerl	.	.	0,02845	7,20E-40	.	.
Powerm	0,25765	.	0,37204	1,82E-38	0,25639	.
Powern	.	.	0,02782	-1,17E-40	.	.
Powero	.	.	0,04827	8,25E-40	.	.
CarAge	-0,01088	.	-0,01150	-4,83E-40	-0,01072	.
DriverAge	-0,01032	.	-0,01003	-4,70E-40	-0,01023	.
BrandJapanese (except Nissan) or Korean	-0,17013	.	-0,18905	-7,36E-40	-0,16454	.
BrandMercedes, Chrysler or BMW	0,06922	.	0,08844	5,37E-39	0,07091	.
BrandOpel, General Motors or Ford	0,09383	.	0,11623	7,23E-39	0,09558	.
Brandother	.	.	0,01043	1,30E-39	.	.
BrandRenault, Nissan or Citroen	-0,03104	.	-0,02913	-5,37E-39	-0,02911	.
BrandVolkswagen, Audi, Skoda or Seat	0,05364	.	0,07686	6,74E-39	0,05584	.
GasRegular	-0,03928	.	-0,04160	-3,67E-39	-0,03934	.
RegionB	-0,05580	.	-0,13231	-1,77E-39	-0,05368	.
RegionC	-0,09252	.	-0,14577	-2,35E-39	-0,09009	.

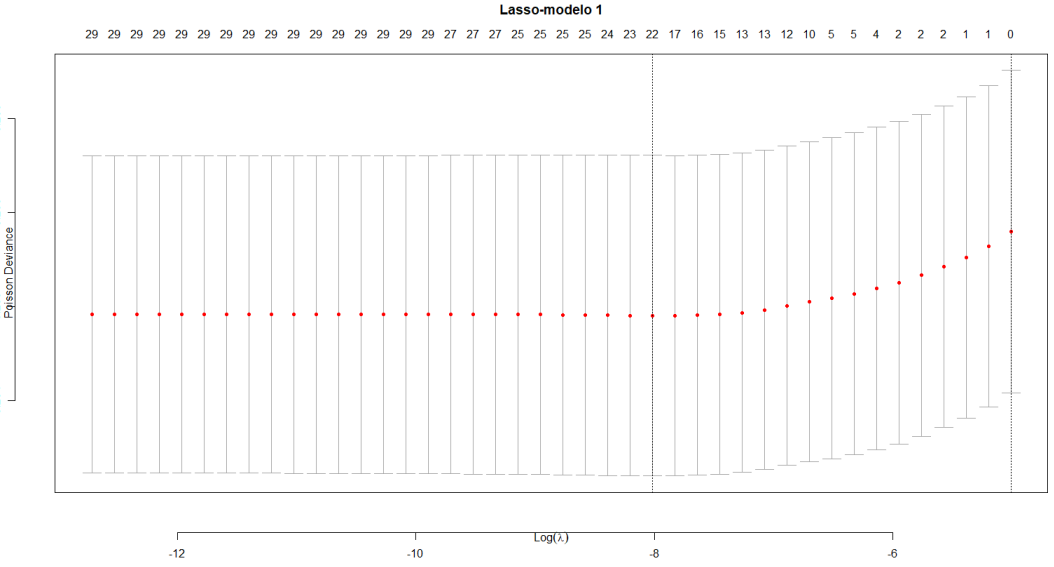
Coeficientes	Lasso		Ridge		Elastic Net	
	λ_{min}	λ_{1se}	λ_{min}	λ_{1se}	λ_{min}	λ_{1se}
RegionD	-0,16619	.	-0,20609	-8,57E-39	-0,16356	.
RegionE	-0,07582	.	-0,18421	-2,27E-39	-0,07318	.
RegionF	0,13914	.	0,11937	8,37E-39	0,13918	.
RegionG	0,02427	.	0,06081	6,06E-39	0,02538	.
RegionH	0,08328	.	0,07922	7,89E-39	0,08435	.
RegionI	-0,00669	.	-0,06425	1,65E-39	-0,00484	.
RegionJ	.	.	-0,05792	1,51E-39	.	.

Fuente: Elaboración propia.

Las tablas correspondientes a los otros dos modelos se presentan en el Anexo B, debido a su extensa longitud. Sin embargo, de estas se puede resaltar que en el segundo modelo *Lasso* elimina 55 de 96 coeficientes, asi como *Elastic Net*. Por otra parte, *Ridge* no elimina ninguno de los coeficientes y tienen un peso considerable. Además, en lo que respecta al tercer modelo presentado *Lasso* y *Elastic Net* eliminan 212 coeficientes de 275.

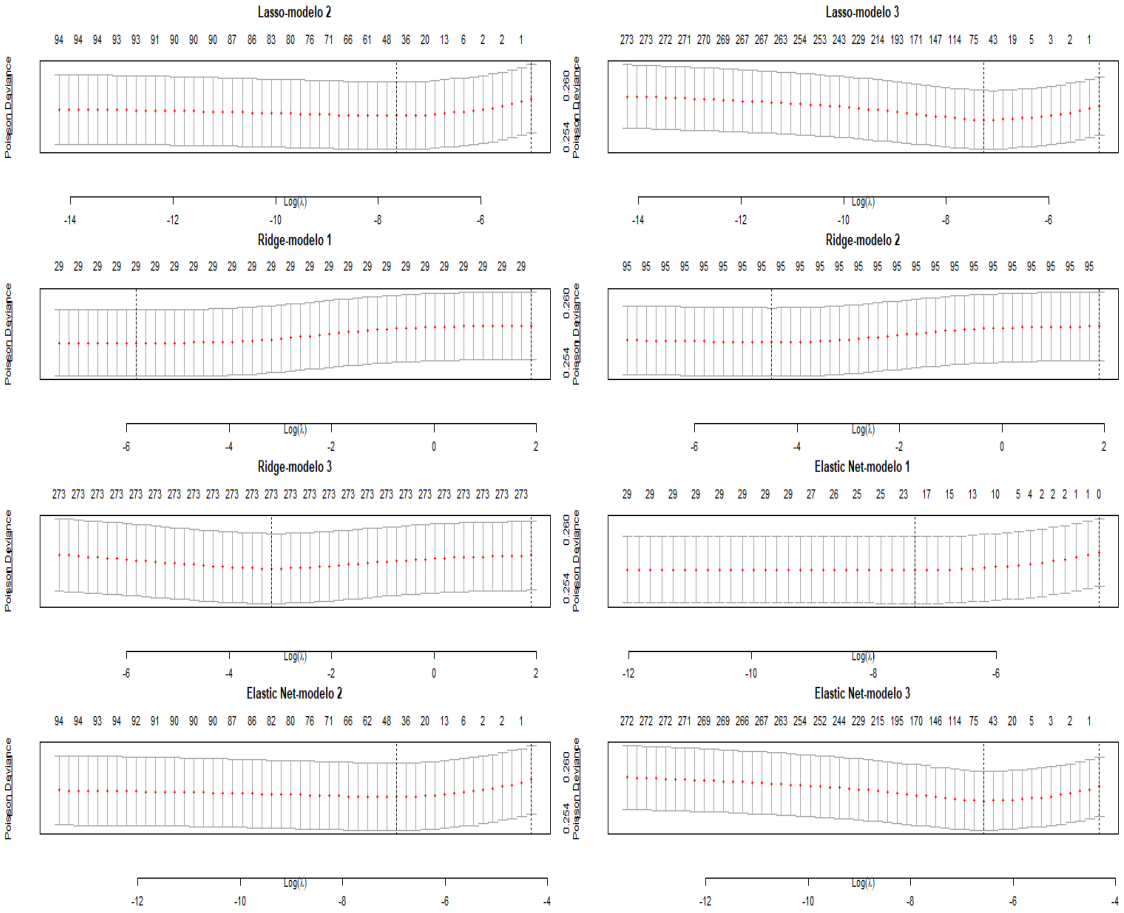
En las figuras 3.5 y 3.6 se observa el comportamiento de la devianza a partir del cambio del logaritmo de λ. Dichos gráficos permiten observar la curva generada por la aplicación de *cross validation*. Se puede apreciar que el logaritmo de λ de los *Lasso* y *Elastic Net* se encuentran entre -8 y -6, mientras que los de *Ridge* se encuentran entre -6 y -4.

Figura 3.5: Devianza vs $log(\lambda)$ del primer modelo *Lasso*



Fuente: Elaboración propia.

Figura 3.6: Devianza vs $log(\lambda)$, modelos Poisson con *Lasso*, *Ridge* y *Elastic Net*



Fuente: Elaboración propia.

En cuanto a las λ_{min} y λ_{1se} calculadas (tabla 3.13), se observa que el método *Ridge* presenta un valor mayor para ambas λ, donde se destaca como λ_{1se} es significativamente mayor. La importancia de esto recae en que cuanto mayor sea el valor de λ mayor será la penalización aplicada a las variables, tal y como se ha observado en las fórmulas 2.13, 2.16 y 2.17.

Tabla 3.13: λ_{min} y λ_{1se} estimadas con los modelos estudiados

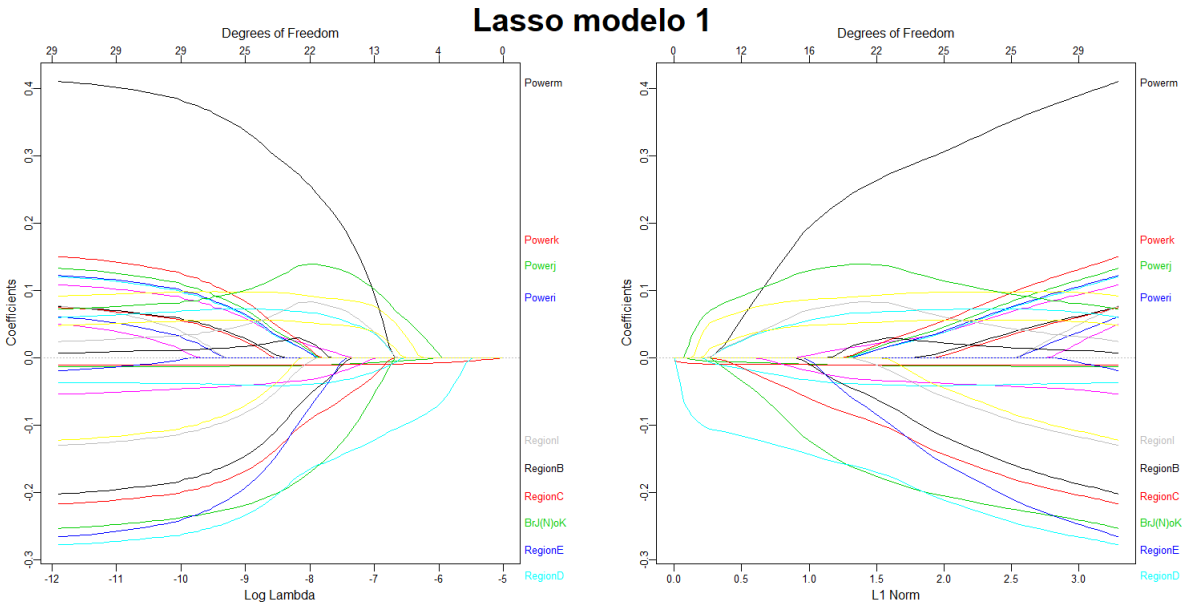
Modelo	λ_{min}	λ_{1se}
Lasso: modelo 1	0,00033	0,00667
Lasso: modelo 2	0,00048	0,00667
Lasso: modelo 3	0,00070	0,00667
Ridge: modelo 1	0,00300	6,66603
Ridge: modelo 2	0,01118	6,66603
Ridge: modelo 3	0,04167	6,66603
Elastic Net: modelo 1	0,00066	0,01333
Elastic Net: modelo 2	0,00096	0,01333
Elastic Net: modelo 3	0,00140	0,01333

Fuente: Elaboración propia.

Para una mejor visualización del comportamiento de los métodos presentados se muestran los gráficos realizados con la función *plot_glmnet* del paquete *Plotmo* de Milborrow (2020), de los tres modelos analizados de cada uno de los métodos (Figuras 3.7 y 3.8).

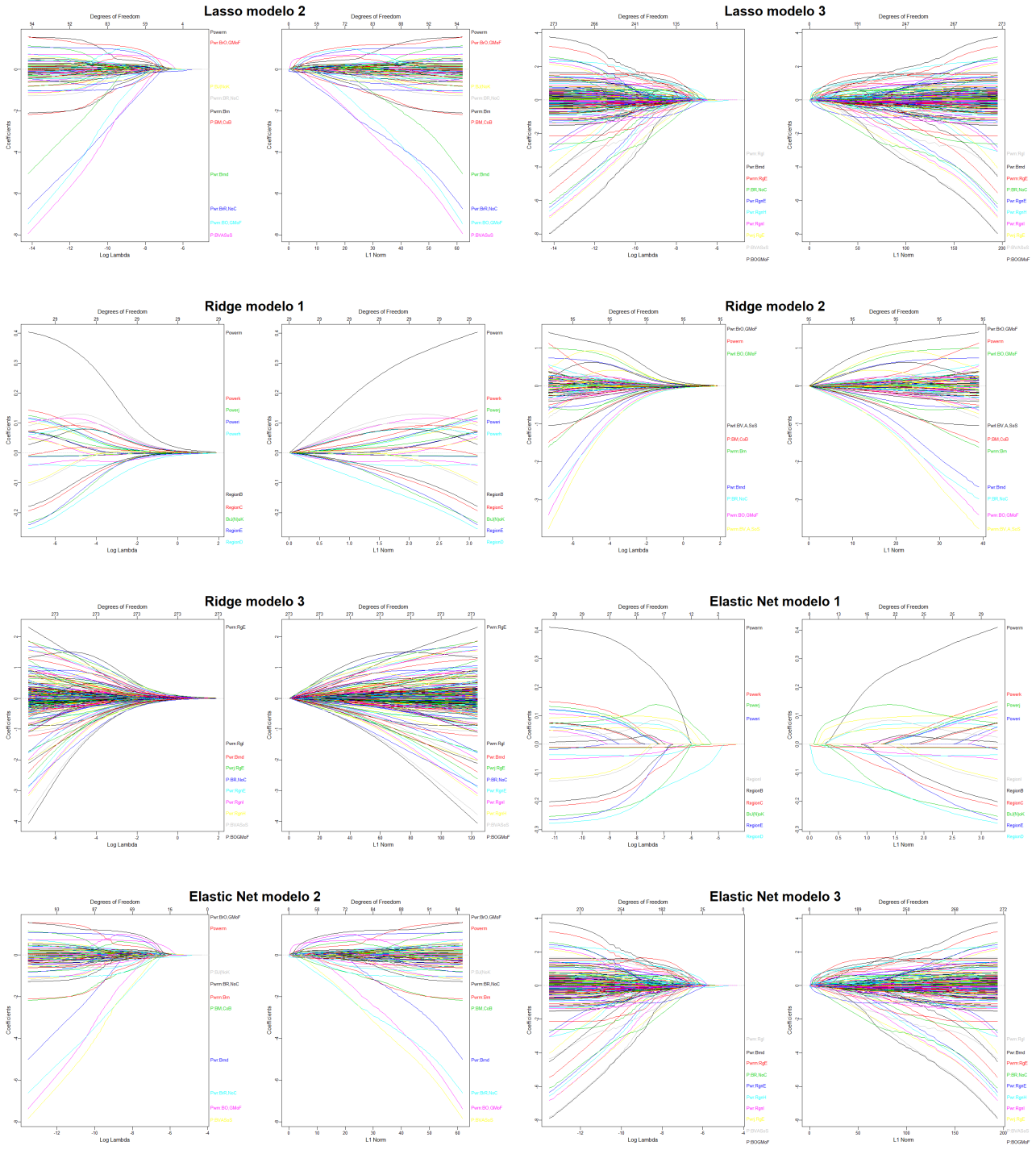
En dichos gráficos se encuentran representados tanto los valores que van adquiriendo los coeficientes a medida que varía el valor de λ, como el recorrido de los coeficientes al variar la norma de penalización correspondiente. Se puede notar el efecto de penalización que tienen los métodos estudiados: a medida que incrementa el valor de λ disminuye la cantidad de coeficientes distintos a cero y, en el caso de *Ridge*, cercanos a cero. En la parte superior de los gráficos está indicado la cantidad de grados de libertad del modelo, según el valor que tome el logaritmo de λ o la norma.

Figura 3.7: Comportamiento de los coeficientes del primer modelo *Lasso* según la norma ℓ_1 y según $log(\lambda)$



Fuente: Elaboración propia.

Figura 3.8: Comportamiento de los coeficientes según la norma ℓ_1 y según $log(\lambda)$, modelos Poisson con *Lasso*, *Ridge* y *Elastic Net*



Fuente: Elaboración propia.

Como se mencionó anteriormente, es importante el cálculo de la devianza residual, el AIC_c y BIC para la comparación con los otros modelos presentados. En este caso, el tercer modelo planteado con *Ridge* es que el presenta una devianza menor, seguido por el segundo modelo de *Ridge* y por el tercer modelo *Lasso*. Además, vale la pena resaltar que la diferencia existente entre las devianzas nuevamente es ínfima: la variación entre el máximo y el mínimo es apenas del $0,3\,\%$. Sin embargo, si se tienen en cuenta los criterios de información escogidos, el modelo adecuado es el primero de *Elastic Net*, seguido por el primer modelo *Lasso* presentado. Aquí si existe una diferencia sustancial, siendo $1,4\,\%$ para AIC_c y $6,8\,\%$ para BIC.

Tabla 3.14: Devianza residual, AIC_c y BIC de GLM con *Lasso*, *Ridge* y *Elastic Net*

Modelo	devianza residual	AIC_c	BIC
Lasso: modelo 1	39340,92	50971,89	51223,62
Lasso: modelo 2	39302,87	51008,87	51479,49
Lasso: modelo 3	39278,32	51138,01	51750,90
Ridge: modelo 1	39326,99	51002,46	51330,80
Ridge: modelo 2	39270,11	51440,28	52490,90
Ridge: modelo 3	39221,91	51496,91	54494,91
Elastic Net: modelo 1	39341,68	50762,09	51002,88
Elastic Net: modelo 2	39304,23	51099,49	51570,11
Elastic Net: modelo 3	39280,47	51138,15	51772,92

Fuente: Elaboración propia.

3.2.5 ZIP-*Lasso*

A continuación se procede a plantear la modelización siguiendo una distribución ZIP con *Lasso*. Se ha realizado solo la estimación de los coeficientes de forma independiente (es decir, sin considerar la interacción entre ellos) debido a los recursos computacionales necesarios, siendo la misma razón por la cual no se ha analizado ZIP con *Ridge* ni con *Elastic Net*. La figura 3.9 muestra la estimación del valor de λ óptimo a través de validación cruzada. Por otra parte, la función *zipath* solo retorna los valores de los coeficientes correspondientes a λ_{min}, los cuales se encuentran indicados en la tabla 3.15. Se observa que el efecto de la aplicación de *Lasso* aparece en el recuento de ceros ya que únicamente los coeficientes significativos son intercepto y la antigüedad del vehículo. Se considera que es importante volver a mencionar que no se ha calculado el error estándar debido a lo mencionado por Tibshirani (2011) y Goeman et al. (2018) en cuanto al mismo.

Tabla 3.15: Coeficientes del modelo ZIP-*Lasso*

Modelo de recuento

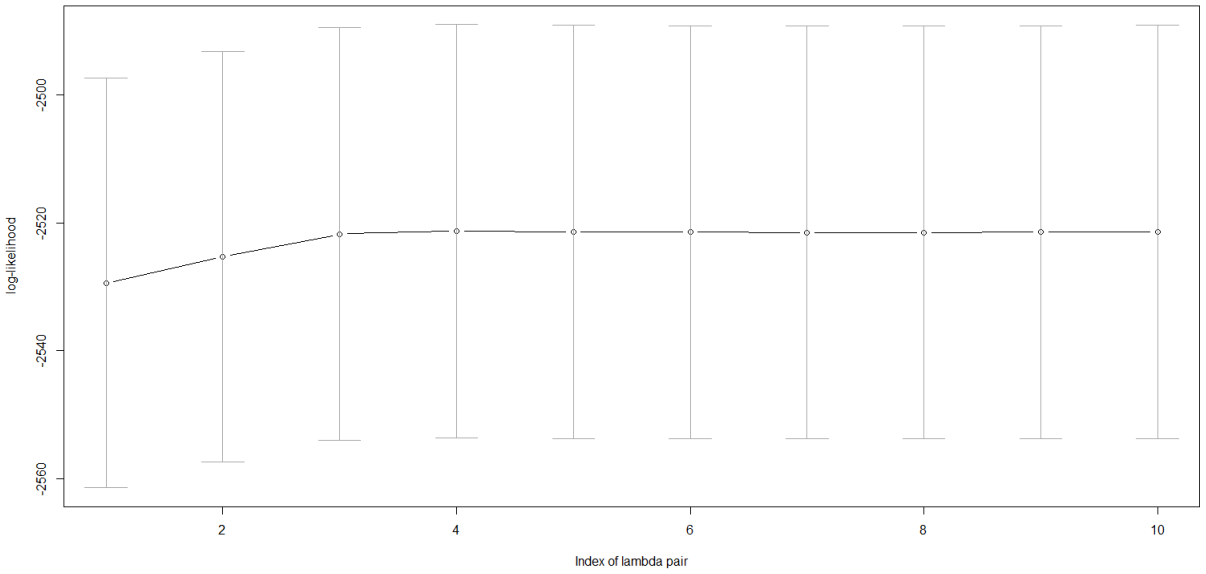
Variable	Valor estimado
(Intercept)	-1,28092
Powere	0,08297
Powerf	0,11485
Powerg	0,07302
Powerh	0,11269
Poweri	0,10728
Powerj	0,12143
Powerk	0,12022
Powerl	0,05723
Powerm	0,36833
Powern	0,00861
Powero	0,12034
CarAge	-0,00549
DriverAge	-0,00762
BrandJapanese (except Nissan) or Korean	-0,37104
BrandMercedes, Chrysler or BMW	0,03202
BrandOpel, General Motors or Ford	0,09365
Brandother	-0,02621
BrandRenault, Nissan or Citroen	-0,04649
BrandVolkswagen, Audi, Skoda or Seat	0,04616
GasRegular	-0,02295
RegionB	-0,12305
RegionC	-0,12371
RegionD	-0,19892
RegionE	-0,34672
RegionF	0,06699
RegionG	0,05324
RegionH	-0,00427
RegionI	-0,07388
RegionJ	-0,05885

Modelo inflado de ceros

Variable	Valor estimado
(Intercept)	0,60145
Powere	.
Powerf	.
Powerg	.
Powerh	.
Poweri	.
Powerj	.
Powerk	.
Powerl	.
Powerm	.
Powern	.
Powero	.
CarAge	0,00914
DriverAge	.
BrandJapanese (except Nissan) or Korean	.
BrandMercedes, Chrysler or BMW	.
BrandOpel, General Motors or Ford	.
Brandother	.
BrandRenault, Nissan or Citroen	.
BrandVolkswagen, Audi, Skoda or Seat	.
GasRegular	.
RegionB	.
RegionC	.
RegionD	.
RegionE	.
RegionF	.
RegionG	.
RegionH	.
RegionI	.
RegionJ	.

Fuente: Elaboración propia.

Figura 3.9: Estimación a través de validación cruzada de λ óptima



Fuente: Elaboración propia.

La tabla 3.16 expone las medidas utilizadas para comparar los modelos. Es importante recordar que, tal y como se mencionó en la sección 3.2.2 correspondiente al modelo ZIP, no se determina la desvianza residual por lo ya comentado anteriormente. Se puede observar que, comparando con los valores correspondientes a GLM con penalización *Lasso*, *Ridge* y *Elastic Net*, se han obtenido mejores resultados. Sin embargo, comparando con los modelados ZIP sin penalización se tiene que para AIC_c es mejor ZIP-*Lasso* pero en cuanto a BIC es mejor el segundo modelo presentado con ZIP.

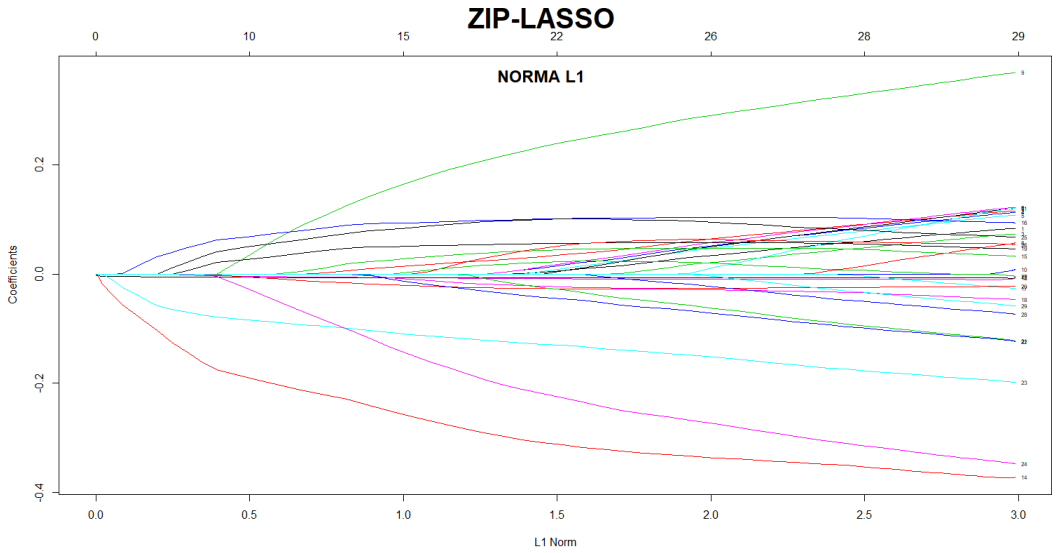
Tabla 3.16: AIC_c y BIC del modelo ZIP-*Lasso*

Modelo	AIC_c	BIC
ZIP-*Lasso*	50421,96	50740,21

Fuente: Elaboración propia.

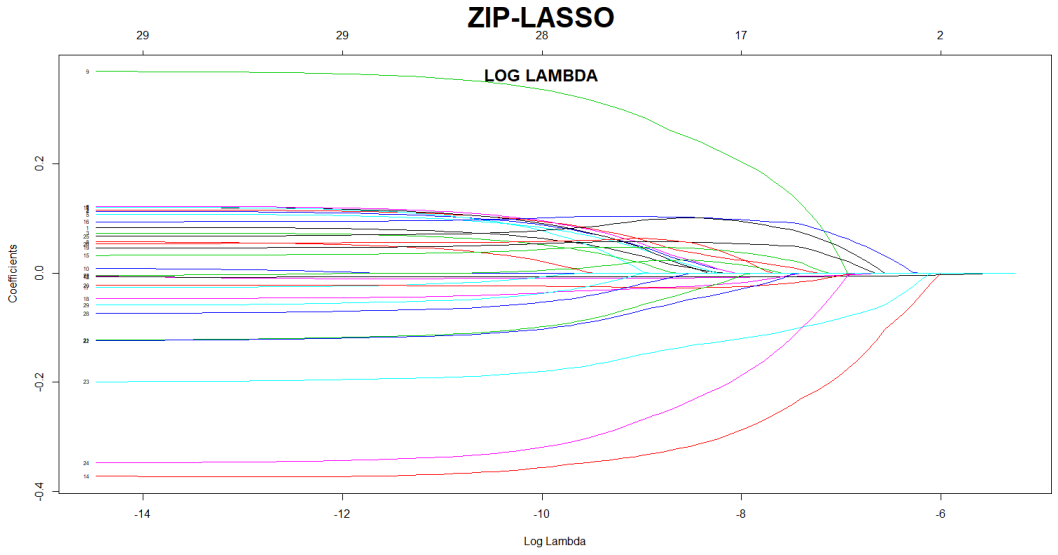
En lo que respecta al comportamiento de los coeficientes según el valor de λ y del valor de la penalización (es decir, el valor de la norma ℓ_1), se puede ver en el gráfico 3.10 que se tiene un comportamiento más controlado que en el caso Poisson-*Lasso* (Figura 3.7). Lo mismo se puede observar comparando la figura 3.11 y 3.7.

Figura 3.10: Comportamiento de los coeficientes según la norma ℓ_1, modelo ZIP-*Lasso*



Fuente: Elaboración propia.

Figura 3.11: Comportamiento de los coeficientes según $log(\lambda)$, modelo ZIP-*Lasso*



Fuente: Elaboración propia.

3.3 Análisis de resultados

Una vez examinados todos los modelos presentados, se realizará un estudio de los resultados que se han obtenido. En cuanto a los coeficientes, se observa que el intercepto es considerado en todos los modelos, así como la antigüedad del vehículo y la edad del conductor. Estos resultados eran esperados debido a que numerosos trabajos mencionan a dichas variables como importantes en el modelado de la frecuencia de siniestros, tal y

como se menciona en Charpentier (2015). Las variables correspondientes a la potencia del motor y la región en la cual el asegurado conduce presentan un comportamiento variable: si bien en algunos de los modelos tienen influencia, en otros tienen un peso menor. Además, no todos los factores de estas variables son incluidos en los modelos (siendo la región la que en general incluye a una mayor cantidad de factores). Las marcas de los vehículos y el tipo de combustible parecen no tener mucha influencia salvo en contados casos.

Se ha observado el comportamiento esperado en cuanto a los métodos de regularización estudiados: mientras que *Lasso* y *Elastic Net* realizan una selección de predictores y producen modelos más simples e interpretables seleccionando un subconjunto de ellos, *Ridge* aproxima los predictores a cero por lo que se tiene un modelo poco parsimonioso.

En la tabla 3.17 se presentan los valores obtenidos de los criterios seleccionados para realizar la comparación de los modelos. La devianza residual es el criterio que menos variación presenta: tan solo $0,78\,\%$ entre el valor máximo (correspondiente al primer modelo GLM-Poisson presentado) y el el valor mínimo (el tercer modelo GLM-Poisson). Por otra parte, el criterio de información de Akaike corregido tiene una variación de $2,24\,\%$ entre el máximo y mínimo y para el BIC $12,92\,\%$. Para estos dos criterios se han obtenido los mejores resultados con los modelos ZIP (tanto considerando como no considerando *Lasso*). Esto es un indicador, tal y como se ha mencionado a lo largo del trabajo, que se trata de una base de datos con un exceso de ceros. Seguido de estos, se tienen los modelos presentados utilizando *Elastic Net*, *Lasso* y *Ridge*, esto propiciado por la estimación de los parámetros. Por otra parte, se puede ver que el tercer modelo presentado utilizando GLM-Poisson presenta la menor devianza residual, seguido por *Ridge*.

Tabla 3.17: Criterios de comparación entre modelos

Modelo	AIC_c	BIC	Devianza residual
GLM-Poisson-1	51117,75	51774,45	39343,00
GLM-Poisson-2	51152,62	53253,97	39246,00
GLM-Poisson-3	51300,28	57297,26	39037,00
ZIP-1	50958,2	51266,5	-
ZIP-2	50369,8	50966,5	-
Lasso: modelo 1	50971,89	51223,62	39340,92
Lasso: modelo 2	51008,87	51479,49	39302,87
Lasso: modelo 3	51138,01	51750,9	39278,32
Ridge: modelo 1	51002,46	51330,8	39326,99
Ridge: modelo 2	51440,28	52490,9	39270,11
Ridge: modelo 3	51496,91	54494,91	39221,91
Elastic Net: modelo 1	50762,09	51002,88	39341,68
Elastic Net: modelo 2	51099,49	51570,11	39304,23
Elastic Net: modelo 3	51138,15	51772,92	39280,47
ZIP-Lasso	50421,96	50740,21	-

Fuente: Elaboración propia.

Capítulo 4

Conclusiones y futuras líneas de investigación

4.1 Conclusiones

El presente trabajo pretende ser un aporte inicial en cuanto al análisis en el campo actuarial (específicamente en lo que se refiere al sector de seguros de automóviles) en la aplicación de técnicas no utilizadas en general como lo son las regularizaciones a través de *Lasso*, *Ridge* y *Elastic Net*. Tras la realización del proyecto, a continuación se extraen las principales conclusiones obtenidas gracias al desarrollo del mismo.

En primer lugar, se puede mencionar que la aplicación de las técnicas estudiadas permiten una perspectiva diferente en cuanto al modelado de la variable analizada en términos de los predictores. Se ha visto que cada uno de ellos aporta diferentes enfoques, con sus virtudes y defectos. En el capítulo 2, correspondiente a la parte teórica, se ha realizado una descripción de los puntos más importantes a fin de poder comprender las técnicas utilizadas. En cambio, en el capítulo 3 se presenta una aplicación de la teoría en la determinación de modelos para la estimación de la cantidad de reclamaciones en seguros de automóviles.

En segundo lugar, se ha constatado que, tal y como era esperado, la frecuencia de los siniestros en seguros de automóviles presenta un exceso de ceros y esto debe ser considerado a la hora de estimar los coeficientes de las variables predictoras en aras de obtener el modelo más adecuado.

En tercer lugar, se puede mencionar que existen ventajas con la aplicación de las técnicas de regularización en cuanto a los criterios analizados comparándolos con los modelos correspondiente sin la regularización, en especial en el modelo ZIP-*Lasso*.

Gracias a los resultados obtenidos podemos afirmar la importancia de ciertas variables predictoras a la hora de realizar la modelización de la frecuencia de siniestros reclamados.

4.2 Futuras líneas de investigación

El trabajo realizado motiva diversas líneas de investigación futuras, dentro de las cuales se pueden encontrar:

- Binomial Negativa
- *Group Lasso*
- *Fused Lasso*
- *Bayesian Lasso*

En lo que respecta a la binomial negativa, no se ha trabajado con ella por dos motivos: En primer lugar se ha escogido enfocarse en los modelos que involucran a Poisson. En segundo lugar, debido a las exigencias computacionales que se tienen al trabajar con el paquete *mpath*, es necesario utilizar servidores *online* para ejecutar el programa. Una de las opciones sería trabajar con AWS de Amazon o Azure de Microsoft. La ventaja del trabajo con la binomial negativa recae en que capta mejor que Poisson la sobredispersión.

Por lo demás, *Group Lasso* tiene en cuenta la estructura implícita de agrupación de los datos, por lo que permite estimar que ciertas covariables agrupadas tengan valor cero (o no) al mismo tiempo (Hastie et al., 2015). Sería interesante aplicar dicha metodología al campo actuarial y ver sus implicancias en la determinación de los modelos.

Otro enfoque posible sería la aplicación de *Fused Lasso*, el cual consiste en la incorporación no solo de la norma ℓ_1 sino también añade un término de diferencias. Así, se incorpora una penalización que sanciona grandes diferencias en la estructura.

Tal y como se mencionó en la sección 3.2.4 existe la posiblidad de incorporar el error estándar e intervalos de confianza para ayudar a la selección del modelo (Park y Casella, 2008) por lo que sería interesante realizar un análisis con este enfoque que permita la comparación con los modelos analizados donde no se ha aplicado la regularización.

Bibliografía

Agresti, A. (2015). *Foundations of Linear and Generalized Linear Models*. John Wiley & Sons Inc, New Jersey (Estados Unidos), 1º edición.

Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. *Proceedings of the 2nd international symposium on information theory*, 2:267–281.

Ayuso, M., Alemany, R., y Bolancé, C. (2020). *Modelos Lineales Generalizados (tarificación a priori). Asignatura Modelos Estadísticos Aplicados.*

Boucher, J.-P., Denuit, M., y Guillén, M. (2007). Risk classification for claim counts. *North American Actuarial Journal*, 11(4):110–131.

Brewer, M., Butler, A., y Cooksley, S. (2016). The relative performance of AIC, AICc and BIC in the presence of unobserved heterogeneity. *Methods in Ecology and Evolution*, 7:679–692.

Bühlmann, P. y van de Geer, S. (2011). *Statistics for High-Dimensional Data: Methods, Theory and Applications*. Springer, Heidelberg(Alemania), 1º edición.

Burnham, K. y Anderson, D. (2003). *Model selection and multimodel inference: a practical information-theoretic approach*. Springer, New York (Estados Unidos), 2º edición.

Cameron, A. y Trivedi, P. (2013). *Regression Analysis of Count Data*. Cambridge Univ. Press, Cambridge (Inglaterra), 2º edición.

Charpentier, A. (2015). *Computational actuarial science with R*. CRC Press/Taylor and Francis Group, Florida (Estados Unidos), 1º edición.

de Jong, P. y Heller, G. (2008). *Generalized Linear Models for Insurance Data*. Cambridge University Press, Cambridge (Inglaterra), 1º edición.

Devriendt, S., Antonio, K., Frees, E., y Verbelen, R. (2017). Sparse modeling of risk factors in insurance analytics. Londres (Inglaterra). *L2 Perspectives in actuarial science, insurance and risk theory.*

Dormann, C., Elith, J., Bacher, S., Buchmann, C., Carl, G., Carré, G., García Márquez, J., Gruber, B., Lafourcade, B., Leitão, P., Münkemüller, T., Mcclean, C., Osborne, P., Reineking, B., Schröder, B., Skidmore, A., Zurell, D., y Lautenback, S. (2012). Collinearity: a review of methods to deal with it and a simulation study evaluating their performance. *Ecography*, 36:01–20.

Dutang, C. y Charpentier, A. (2020). *CASDatasets: Insurance datasets*. R package version 1.0-11.

Friedman, J., Hastie, T., y Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22.

Garrido, J., Genest, C., y Schulz, J. (2016). Generalized linear models for dependent frequency and severity of insurance claims. *Insurance: Mathematics and Economics*, 70:205–215.

Goeman, J., Meijer, R., y Chaturvedi, N. (2018). *Penalized: L1 (Lasso and Fused Lasso) and L2 (Ridge) Penalized Estimation in GLMs and in the Cox Model*. R package version 0.9-51.

Hastie, T., Tibshirani, R., y Wainwright, M. (2015). *Statistical Learning with Sparsity: The Lasso and Generalizations*. Chapmanr & Hall, Londres (Inglaterra), 1º edición.

Hauke, J. y Kossowski, T. (2011). Comparison of values of Pearson's and Spearman's correlation coefficients on the same sets of data. *Quaestiones Geographicae*, 30(2):87–93.

Hoerl, A. y Kennard, R. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12:55–67.

ICEA (2021). Primas de seguros. `https://www.icea.es/es-es/informacion-seguro/almacen-datos/primas-crecimientos`. Recuperado el 10 de mayo del 2021.

Ismail, N. y Jemain, A. (2007). Handling overdispersion with negative binomial and generalized poisson regression models. *Casualty Actuarial Society Forum*, pp. 103–158.

Jackman, S. (2020). *pscl: Classes and Methods for R Developed in the Political Science Computational Laboratory*. R package version 1.5.5.

Lambert, D. (1992). Zero-inflated poisson regression, with an application to defects in manufacturing. *Technometrics*, 34:1–14.

Lin, M., Lucas, H., y Shmueli, G. (2013). Research commentary - too big to fail: Large samples and the p-value problem. *Inf. Syst. Res.*, 24:906–917.

Martin, J. y Hall, D. B. (2016). R2 measures for zero-inflated regression models for count data with excess zeros. *Journal of Statistical Computation and Simulation*, 86(18):3777–3790.

Meulman, J. J., van der Kooij, A. J., y Duisters, K. L. W. (2019). ROS regression: Integrating regularization with optimal scaling regression. *Statistical Science*, 34(3):361 – 390.

Mihaela, D. y Danut, J. (2015). Modeling the frequency of auto insurance claims by means of poisson and negative binomial models. *Analele Stiintifice ale Universitatii Al I Cuza din Iasi - Sectiunea Stiinte Economice*, 62:151–168.

Milborrow, S. (2020). *Plotmo: Plot a Model's Residuals, Response, and Partial Dependence Plots*. R package version 3.6.0.

Navarro, J. R. (2019). *Técnicas de Regularización en el Aprendizaje Estadístico*. Tesis de máster, UNED.

Noll, A., Salzmann, R., y Wuthrich, M. (2020). Case study: French motor third-party liability claims. *SSRN*, pp. 01–41.

Park, T. y Casella, G. (2008). The bayesian Lasso. *Journal of the American Statistical Association*, 103(482):681–686.

Ramirez, D. (2016). Limitations of the least squares estimators; a teaching perspective. *Athens Institute for Educations and Research*, 2074:01–17.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461 – 464.

Shishkina, T., Farmus, L., y Cribbie, R. (2018). Testing for a lack of relationship among categorical variables. *The Quantitative Methods for Psychology*, 14:1–27.

Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288.

Tibshirani, R. (2011). The Lasso: some novel algorithms and applications. San Fransisco (Estados Unidos). *American Statistical Association*.

Wang, Z. (2021). *mpath: Regularized Linear Models*. R package version 0.4-2.19.

Yip, K. y Yau, K. (2005). On modeling claim frequency data in general insurance with extra zeros. *Insurance: Mathematics and Economics*, 36(2):153–163.

Zeileis, A., Kleiber, C., y Jackman, S. (2008). Regression models for count data in R. *Journal of Statistical Software*, 27(8):01–25.

Zou, H. y Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B*, 67(2):301–320.

Anexo A

Coeficientes de GLM-Poisson

Tabla A.1: GLM-Poisson-2

Coeficientes	Valor estimado	valor z	$Pr(>\|z\|)$
(Intercept)	**-1,97500**	**-14,250**	**< 2e-16**
Powere	0,14090	0,714	0,475316
Powerf	0,15290	0,853	0,393576
Powerg	0,18430	0,984	0,325129
Powerh	0,33840	1,591	0,111664
Poweri	-0,28000	-0,705	0,480731
Powerj	0,60300	1,520	0,128525
Powerk	-0,00490	-0,005	0,996120
Powerl	-11,41000	-0,015	0,988237
Powerm	**1,62900**	**2,760**	**0,005774**
Powern	1,28000	1,271	0,203901
Powero	-11,21000	-0,030	0,976073
CarAge	**-0,01330**	**-4,839**	**0,000001**
DriverAge	**-0,01081**	**-11,569**	**< 2e-16**
BrandJapanese (except Nissan) or Korean	**-0,30030**	**-2,014**	**0,044062**
BrandMercedes, Chrysler or BMW	0,07562	0,147	0,883269
BrandOpel, General Motors or Ford	0,05262	0,326	0,744661
Brandother	0,24730	0,974	0,330294
BrandRenault, Nissan or Citroen	0,07826	0,610	0,541603
BrandVolkswagen, Audi, Skoda or Seat	-0,05452	-0,279	0,780055
GasRegular	-0,03856	-1,326	0,184913
RegionB	**-0,20320**	**-2,228**	**0,025910**
RegionC	**-0,22190**	**-3,636**	**0,000277**
RegionD	**-0,28390**	**-5,399**	**6,68E-08**
RegionE	**-0,27680**	**-2,189**	**0,028604**
RegionF	0,07389	1,270	0,204159
RegionG	0,01419	0,113	0,909640
RegionH	0,02874	0,407	0,683983
RegionI	**-0,13540**	**-2,147**	**0,031809**
RegionJ	-0,12500	-1,665	0,095933
Powere:BrandJapanese (except Nissan) or Korean	-0,20520	-0,840	0,400953
Powerf:BrandJapanese (except Nissan) or Korean	0,15540	0,718	0,472600
Powerg:BrandJapanese (except Nissan) or Korean	-0,06142	-0,279	0,780470
Powerh:BrandJapanese (except Nissan) or Korean	-0,07345	-0,292	0,770607
Poweri:BrandJapanese (except Nissan) or Korean	0,61680	1,410	0,158555
Powerj:BrandJapanese (except Nissan) or Korean	-0,15480	-0,370	0,711139
Powerk:BrandJapanese (except Nissan) or Korean	0,31830	0,311	0,756075
Powerl:BrandJapanese (except Nissan) or Korean	11,78000	0,015	0,987856
Powerm:BrandJapanese (except Nissan) or Korean	-1,19800	-1,296	0,195123
Powern:BrandJapanese (except Nissan) or Korean	-1,35200	-0,950	0,341960
Powero:BrandJapanese (except Nissan) or Korean	11,64000	0,031	0,975167
Powere:BrandMercedes, Chrysler or BMW	-0,37660	-0,513	0,607975
Powerf:BrandMercedes, Chrysler or BMW	0,63780	1,102	0,270527
Powerg:BrandMercedes, Chrysler or BMW	0,10170	0,188	0,851243

Coeficientes	Valor estimado	valor z	$Pr(>\|z\|))$
Powerh:BrandMercedes, Chrysler or BMW	-0,01479	-0,026	0,979025
Poweri:BrandMercedes, Chrysler or BMW	0,13900	0,207	0,835873
Powerj:BrandMercedes, Chrysler or BMW	-0,75770	-1,132	0,257519
Powerk:BrandMercedes, Chrysler or BMW	0,18920	0,166	0,868517
Powerl:BrandMercedes, Chrysler or BMW	11,25000	0,015	0,988397
Powerm:BrandMercedes, Chrysler or BMW	-1,16000	-1,450	0,147114
Powern:BrandMercedes, Chrysler or BMW	-2,36500	-1,780	0,075126
Powero:BrandMercedes, Chrysler or BMW	10,41000	0,028	0,977776
Powere:BrandOpel, General Motors or Ford	0,12610	0,541	0,588658
Powerf:BrandOpel, General Motors or Ford	-0,03181	-0,138	0,889892
Powerg:BrandOpel, General Motors or Ford	0,00825	0,035	0,971809
Powerh:BrandOpel, General Motors or Ford	-0,00771	-0,025	0,979658
Poweri:BrandOpel, General Motors or Ford	0,54990	1,143	0,253168
Powerj:BrandOpel, General Motors or Ford	-0,46000	-1,023	0,306307
Powerk:BrandOpel, General Motors or Ford	-0,50950	-0,451	0,652023
Powerl:BrandOpel, General Motors or Ford	12,46000	0,016	0,987152
Powerm:BrandOpel, General Motors or Ford	-0,87250	-1,117	0,263934
Powern:BrandOpel, General Motors or Ford	-11,97000	-0,070	0,944133
Powero:BrandOpel, General Motors or Ford	12,45000	0,033	0,973437
Powere:Brandother	-0,51650	-1,188	0,235013
Powerf:Brandother	-0,15110	-0,455	0,649297
Powerg:Brandother	-0,35670	-0,971	0,331641
Powerh:Brandother	-0,67840	-1,749	0,080238
Poweri:Brandother	0,02501	0,048	0,961617
Powerj:Brandother	-0,53320	-1,055	0,291312
Powerk:Brandother	0,21330	0,198	0,843392
Powerl:Brandother	11,21000	0,014	0,988441
Powerm:Brandother	-2,19500	-1,856	0,063445
Powern:Brandother	-0,36380	-0,308	0,758312
Powero:Brandother	0,33920	0,001	0,999533
Powere:BrandRenault, Nissan or Citroen	-0,15280	-0,742	0,458025
Powerf:BrandRenault, Nissan or Citroen	-0,10650	-0,572	0,567180
Powerg:BrandRenault, Nissan or Citroen	-0,20590	-1,054	0,291947
Powerh:BrandRenault, Nissan or Citroen	-0,29490	-1,275	0,202453
Poweri:BrandRenault, Nissan or Citroen	0,45490	1,112	0,266104
Powerj:BrandRenault, Nissan or Citroen	-0,68570	-1,629	0,103292
Powerk:BrandRenault, Nissan or Citroen	0,09422	0,092	0,926320
Powerl:BrandRenault, Nissan or Citroen	11,62000	0,015	0,988020
Powerm:BrandRenault, Nissan or Citroen	-1,35400	-1,638	0,101525
Powern:BrandRenault, Nissan or Citroen	-0,58310	-0,528	0,597155
Powero:BrandRenault, Nissan or Citroen	0,20300	0,001	0,999582
Powere:BrandVolkswagen, Audi, Skoda or Seat	0,22280	0,857	0,391298
Powerf:BrandVolkswagen, Audi, Skoda or Seat	0,04227	0,170	0,865125
Powerg:BrandVolkswagen, Audi, Skoda or Seat	0,23000	0,878	0,380131
Powerh:BrandVolkswagen, Audi, Skoda or Seat	-0,49890	-1,397	0,162506
Poweri:BrandVolkswagen, Audi, Skoda or Seat	0,48950	1,031	0,302342
Powerj:BrandVolkswagen, Audi, Skoda or Seat	-0,35110	-0,712	0,476540
Powerk:BrandVolkswagen, Audi, Skoda or Seat	0,50230	0,462	0,643948
Powerl:BrandVolkswagen, Audi, Skoda or Seat	10,38000	0,013	0,989297
Powerm:BrandVolkswagen, Audi, Skoda or Seat	-12,38000	-0,083	0,933848
Powern:BrandVolkswagen, Audi, Skoda or Seat	-0,97160	-0,856	0,392038
Powero:BrandVolkswagen, Audi, Skoda or Seat	11,44000	0,031	0,975596

Fuente: Elaboración propia.

Tabla A.2: GLM-Poisson-3

Coeficientes	Valor estimado	valor z	$\Pr(Z > \|z\|)$
(Intercept)	**-2,17300**	**-6,443000**	**1,17E-10**
CarAge	**-0,01373**	**-4,959000**	**0,000001**
DriverAge	**-0,01064**	**-11,332000**	**< 2e-16**
Powere	0,30920	1,130000	0,258490
Powerf	0,18120	0,663000	0,507020
Powerg	0,17770	0,660000	0,509560
Powerh	0,60900	1,829000	0,067400
Poweri	-0,49170	-0,995000	0,319730
Powerj	0,61130	1,241000	0,214430
Powerk	-0,56850	-0,516000	0,606190

Coeficientes	Valor estimado	valor z	$Pr(Z > \|z\|)$
Powerl	-12,55000	-0,006000	0,995240
Powerm	1,36200	1,201000	0,229630
Powern	-23,38000	-0,070000	0,944070
Powero	-11,55000	-0,012000	0,990810
RegionB	-0,67520	-1,105000	0,269300
RegionC	0,09421	0,277000	0,782000
RegionD	-0,08099	-0,273000	0,784530
RegionE	0,87740	1,464000	0,143310
RegionF	0,49390	1,515000	0,129720
RegionG	-0,18030	-0,256000	0,797660
RegionH	0,56580	1,466000	0,142750
RegionI	0,10420	0,301000	0,763620
RegionJ	-0,24190	-0,557000	0,577440
BrandJapanese (except Nissan) or Korean	-0,19790	-0,593000	0,553340
BrandMercedes, Chrysler or BMW	-0,03783	-0,060000	0,952530
BrandOpel, General Motors or Ford	0,01211	0,035000	0,972320
Brandother	0,73100	1,317000	0,187880
BrandRenault, Nissan or Citroen	0,28640	0,918000	0,358680
BrandVolkswagen, Audi, Skoda or Seat	0,13270	0,353000	0,723900
GasRegular	-0,05886	-0,295000	0,768320
Powere:RegionB	-0,05971	-0,177000	0,859410
Powerf:RegionB	0,23470	0,740000	0,459450
Powerg:RegionB	0,06679	0,207000	0,835900
Powerh:RegionB	-0,04520	-0,102000	0,918930
Poweri:RegionB	-0,16130	-0,265000	0,791180
Powerj:RegionB	0,29940	0,588000	0,556730
Powerk:RegionB	-0,04773	-0,064000	0,948610
Powerl:RegionB	1,11500	1,131000	0,257910
Powerm:RegionB	1,45300	1,271000	0,203840
Powern:RegionB	0,74160	0,001000	0,998930
Powero:RegionB	-10,84000	-0,015000	0,988240
Powere:RegionC	-0,01918	-0,089000	0,929420
Powerf:RegionC	-0,00876	-0,041000	0,967540
Powerg:RegionC	-0,02807	-0,130000	0,896850
Powerh:RegionC	-0,35960	-1,212000	0,225560
Poweri:RegionC	0,38750	1,147000	0,251370
Powerj:RegionC	0,15690	0,440000	0,659690
Powerk:RegionC	0,79760	1,795000	0,072640
Powerl:RegionC	0,97150	1,343000	0,179420
Powerm:RegionC	0,83680	0,971000	0,331430
Powern:RegionC	12,80000	0,046000	0,963540
Powero:RegionC	-0,58710	-0,495000	0,620460
Powere:RegionD	-0,07669	-0,415000	0,678110
Powerf:RegionD	-0,04926	-0,270000	0,787320
Powerg:RegionD	0,01010	0,055000	0,955960
Powerh:RegionD	-0,29610	-1,198000	0,230930
Poweri:RegionD	0,00818	0,028000	0,977890
Powerj:RegionD	-0,21520	-0,712000	0,476650
Powerk:RegionD	0,11190	0,268000	0,788380
Powerl:RegionD	-0,86590	-1,132000	0,257540
Powerm:RegionD	0,12180	0,144000	0,885680
Powern:RegionD	12,47000	0,045000	0,964490
Powero:RegionD	-1,25100	-1,315000	0,188430
Powere:RegionE	-0,72610	-1,462000	0,143630
Powerf:RegionE	-0,42550	-0,963000	0,335580
Powerg:RegionE	-0,41110	-0,920000	0,357510
Powerh:RegionE	-1,18500	-1,680000	0,092890
Poweri:RegionE	0,48560	0,795000	0,426740
Powerj:RegionE	-12,29000	-0,083000	0,933900
Powerk:RegionE	0,53640	0,815000	0,414870
Powerl:RegionE	1,60300	1,815000	0,069520
Powerm:RegionE	-11,45000	-0,038000	0,969720
Powern:RegionE	14,09000	0,050000	0,959850
Powero:RegionE	-12,17000	-0,036000	0,971490
Powere:RegionF	-0,24640	-1,213000	0,225300
Powerf:RegionF	-0,16080	-0,808000	0,419090
Powerg:RegionF	-0,33340	-1,671000	0,094650
Powerh:RegionF	-0,36460	-1,397000	0,162360
Poweri:RegionF	0,03951	0,127000	0,898790

Coeficientes	Valor estimado	valor z	$Pr(Z > \|z\|)$
Powerj:RegionF	-0,11370	-0,376000	0,706840
Powerk:RegionF	-0,09090	-0,217000	0,828210
Powerl:RegionF	0,29230	0,447000	0,654740
Powerm:RegionF	-0,41890	-0,480000	0,631020
Powern:RegionF	12,38000	0,044000	0,964740
Powero:RegionF	-0,73910	-1,077000	0,281280
Powere:RegionG	0,22490	0,515000	0,606360
Powerf:RegionG	-0,22830	-0,517000	0,604810
Powerg:RegionG	-0,25050	-0,571000	0,568120
Powerh:RegionG	-0,18080	-0,313000	0,754420
Poweri:RegionG	-1,10000	-1,004000	0,315310
Powerj:RegionG	-0,37650	-0,454000	0,649750
Powerk:RegionG	-0,17180	-0,153000	0,878740
Powerl:RegionG	1,31800	1,285000	0,198940
Powerm:RegionG	1,18300	0,786000	0,431900
Powern:RegionG	0,68830	0,001000	0,999350
Powere:RegionH	-0,37590	-1,513000	0,130260
Powerf:RegionH	-0,01880	-0,079000	0,936720
Powerg:RegionH	-0,16230	-0,663000	0,507370
Powerh:RegionH	-0,31150	-0,874000	0,382170
Poweri:RegionH	0,00068	0,002000	0,998660
Powerj:RegionH	0,35570	0,928000	0,353290
Powerk:RegionH	0,41650	0,807000	0,419560
Powerl:RegionH	1,07500	1,433000	0,151850
Powerm:RegionH	0,64840	0,495000	0,620810
Powern:RegionH	13,44000	0,048000	0,961720
Powero:RegionH	-13,04000	-0,026000	0,978910
Powere:RegionI	-0,07485	-0,333000	0,739190
Powerf:RegionI	-0,01110	-0,051000	0,959690
Powerg:RegionI	-0,06047	-0,277000	0,782070
Powerh:RegionI	-0,45280	-1,465000	0,143030
Poweri:RegionI	-0,04593	-0,128000	0,898320
Powerj:RegionI	0,11140	0,320000	0,749100
Powerk:RegionI	0,20450	0,409000	0,682730
Powerl:RegionI	0,34380	0,425000	0,670510
Powerm:RegionI	1,33300	1,491000	0,135920
Powern:RegionI	0,08465	0,000000	0,999810
Powero:RegionI	-12,80000	-0,043000	0,965490
Powere:RegionJ	-0,40150	-1,496000	0,134620
Powerf:RegionJ	-0,32340	-1,242000	0,214190
Powerg:RegionJ	-0,20560	-0,798000	0,425080
Powerh:RegionJ	-0,27250	-0,792000	0,428310
Poweri:RegionJ	-0,06823	-0,165000	0,868920
Powerj:RegionJ	0,08547	0,216000	0,828810
Powerk:RegionJ	0,52410	1,002000	0,316550
Powerl:RegionJ	0,72390	0,893000	0,371750
Powerm:RegionJ	0,18640	0,179000	0,857600
Powern:RegionJ	11,59000	0,041000	0,966970
Powero:RegionJ	1,13900	1,184000	0,236300
Powere:BrandJapanese (except Nissan) or Korean	-0,14730	-0,574000	0,565970
Powerf:BrandJapanese (except Nissan) or Korean	0,21180	0,862000	0,388510
Powerg:BrandJapanese (except Nissan) or Korean	0,09281	0,389000	0,697310
Powerh:BrandJapanese (except Nissan) or Korean	-0,08839	-0,308000	0,758420
Poweri:BrandJapanese (except Nissan) or Korean	0,51560	1,146000	0,251990
Powerj:BrandJapanese (except Nissan) or Korean	-0,14920	-0,349000	0,726890
Powerk:BrandJapanese (except Nissan) or Korean	0,72630	0,693000	0,488210
Powerl:BrandJapanese (except Nissan) or Korean	12,51000	0,006000	0,995250
Powerm:BrandJapanese (except Nissan) or Korean	-1,40700	-1,470000	0,141670
Powern:BrandJapanese (except Nissan) or Korean	-0,37990	-0,246000	0,805990
Powero:BrandJapanese (except Nissan) or Korean	12,77000	0,013000	0,989840
Powere:BrandMercedes, Chrysler or BMW	-0,43270	-0,583000	0,559940
Powerf:BrandMercedes, Chrysler or BMW	0,69580	1,170000	0,241860
Powerg:BrandMercedes, Chrysler or BMW	0,19500	0,350000	0,726220
Powerh:BrandMercedes, Chrysler or BMW	0,07324	0,126000	0,899880
Poweri:BrandMercedes, Chrysler or BMW	0,22740	0,332000	0,739890
Powerj:BrandMercedes, Chrysler or BMW	-0,65890	-0,967000	0,333360
Powerk:BrandMercedes, Chrysler or BMW	0,53170	0,458000	0,647100
Powerl:BrandMercedes, Chrysler or BMW	12,14000	0,006000	0,995400
Powerm:BrandMercedes, Chrysler or BMW	-1,31200	-1,524000	0,127590

Coeficientes	Valor estimado	valor z	$Pr(Z > \|z\|)$
Powern:BrandMercedes, Chrysler or BMW	-1,98500	-1,475000	0,140310
Powero:BrandMercedes, Chrysler or BMW	11,63000	0,012000	0,990750
Powere:BrandOpel, General Motors or Ford	0,12610	0,521000	0,602390
Powerf:BrandOpel, General Motors or Ford	-0,00668	-0,026000	0,978940
Powerg:BrandOpel, General Motors or Ford	-0,00937	-0,038000	0,969440
Powerh:BrandOpel, General Motors or Ford	-0,06587	-0,202000	0,840030
Poweri:BrandOpel, General Motors or Ford	0,55850	1,143000	0,253000
Powerj:BrandOpel, General Motors or Ford	-0,45820	-1,002000	0,316330
Powerk:BrandOpel, General Motors or Ford	-0,26140	-0,230000	0,818430
Powerl:BrandOpel, General Motors or Ford	13,27000	0,006000	0,994970
Powerm:BrandOpel, General Motors or Ford	-1,10300	-1,264000	0,206240
Powern:BrandOpel, General Motors or Ford	-14,30000	-0,031000	0,975320
Powero:BrandOpel, General Motors or Ford	13,86000	0,014000	0,988970
Powere:Brandother	**-1,24900**	**-2,597000**	**0,009400**
Powerf:Brandother	-0,57440	-1,532000	0,125580
Powerg:Brandother	-0,75350	-1,894000	0,058280
Powerh:Brandother	**-1,29600**	**-2,925000**	**0,003450**
Poweri:Brandother	-0,21520	-0,405000	0,685280
Powerj:Brandother	-1,01100	-1,863000	0,062520
Powerk:Brandother	0,15250	0,138000	0,890040
Powerl:Brandother	11,36000	0,005000	0,995690
Powerm:Brandother	**-2,66800**	**-2,151000**	**0,031440**
Powern:Brandother	-0,53720	-0,442000	0,658580
Powero:Brandother	-0,53650	0,000000	0,999730
Powere:BrandRenault, Nissan or Citroen	-0,17120	-0,806000	0,420420
Powerf:BrandRenault, Nissan or Citroen	-0,08969	-0,418000	0,675600
Powerg:BrandRenault, Nissan or Citroen	-0,21590	-1,039000	0,298860
Powerh:BrandRenault, Nissan or Citroen	-0,30330	-1,175000	0,239940
Poweri:BrandRenault, Nissan or Citroen	0,38690	0,919000	0,358140
Powerj:BrandRenault, Nissan or Citroen	-0,65490	-1,537000	0,124240
Powerk:BrandRenault, Nissan or Citroen	0,29440	0,286000	0,774510
Powerl:BrandRenault, Nissan or Citroen	12,41000	0,006000	0,995290
Powerm:BrandRenault, Nissan or Citroen	-1,56000	-1,815000	0,069560
Powern:BrandRenault, Nissan or Citroen	-0,67260	-0,603000	0,546540
Powero:BrandRenault, Nissan or Citroen	-0,38330	0,000000	0,999700
Powere:BrandVolkswagen, Audi, Skoda or Seat	0,20140	0,740000	0,459230
Powerf:BrandVolkswagen, Audi, Skoda or Seat	0,07757	0,282000	0,777870
Powerg:BrandVolkswagen, Audi, Skoda or Seat	0,27290	0,978000	0,327950
Powerh:BrandVolkswagen, Audi, Skoda or Seat	-0,49280	-1,301000	0,193290
Poweri:BrandVolkswagen, Audi, Skoda or Seat	0,48160	0,983000	0,325370
Powerj:BrandVolkswagen, Audi, Skoda or Seat	-0,31560	-0,627000	0,530950
Powerk:BrandVolkswagen, Audi, Skoda or Seat	0,70060	0,639000	0,522830
Powerl:BrandVolkswagen, Audi, Skoda or Seat	11,14000	0,005000	0,995770
Powerm:BrandVolkswagen, Audi, Skoda or Seat	-14,37000	-0,035000	0,971770
Powern:BrandVolkswagen, Audi, Skoda or Seat	-1,42000	-1,181000	0,237460
Powero:BrandVolkswagen, Audi, Skoda or Seat	12,93000	0,013000	0,989710
Powere:GasRegular	-0,09130	-0,841000	0,400370
Powerf:GasRegular	-0,00473	-0,044000	0,964800
Powerg:GasRegular	0,12610	1,189000	0,234490
Powerh:GasRegular	0,11130	0,751000	0,452780
Poweri:GasRegular	0,25680	1,309000	0,190560
Powerj:GasRegular	-0,00771	-0,046000	0,963550
Powerk:GasRegular	0,14410	0,644000	0,519460
Powerl:GasRegular	0,14510	0,417000	0,676850
Powerm:GasRegular	0,22610	0,478000	0,632580
Powern:GasRegular	12,44000	0,069000	0,945140
Powero:GasRegular	-0,29360	-0,481000	0,630750
RegionB:BrandJapanese (except Nissan) or Korean	0,02229	0,036000	0,971340
RegionC:BrandJapanese (except Nissan) or Korean	-0,21290	-0,612000	0,540380
RegionD:BrandJapanese (except Nissan) or Korean	0,17040	0,571000	0,568170
RegionE:BrandJapanese (except Nissan) or Korean	-0,93390	-1,534000	0,125140
RegionF:BrandJapanese (except Nissan) or Korean	-0,34280	-1,111000	0,266500
RegionG:BrandJapanese (except Nissan) or Korean	-0,18640	-0,265000	0,791200
RegionH:BrandJapanese (except Nissan) or Korean	**-0,80950**	**-2,152000**	**0,031360**
RegionI:BrandJapanese (except Nissan) or Korean	-0,13990	-0,409000	0,682880
RegionJ:BrandJapanese (except Nissan) or Korean	0,17970	0,402000	0,688010
RegionB:BrandMercedes, Chrysler or BMW	0,17010	0,234000	0,815270
RegionC:BrandMercedes, Chrysler or BMW	-0,05106	-0,123000	0,901760
RegionD:BrandMercedes, Chrysler or BMW	0,15050	0,405000	0,685620

Coeficientes	Valor estimado	valor z	$\Pr(Z > \|z\|)$
RegionE:BrandMercedes, Chrysler or BMW	-0,81530	-1,025000	0,305580
RegionF:BrandMercedes, Chrysler or BMW	0,14260	0,345000	0,730330
RegionG:BrandMercedes, Chrysler or BMW	0,41910	0,416000	0,677690
RegionH:BrandMercedes, Chrysler or BMW	-0,59090	-1,114000	0,265110
RegionI:BrandMercedes, Chrysler or BMW	-0,12690	-0,285000	0,775590
RegionJ:BrandMercedes, Chrysler or BMW	0,73020	1,414000	0,157330
RegionB:BrandOpel, General Motors or Ford	0,60260	0,971000	0,331680
RegionC:BrandOpel, General Motors or Ford	-0,27070	-0,794000	0,427270
RegionD:BrandOpel, General Motors or Ford	-0,04524	-0,152000	0,879570
RegionE:BrandOpel, General Motors or Ford	-0,90550	-1,263000	0,206660
RegionF:BrandOpel, General Motors or Ford	-0,04321	-0,129000	0,897740
RegionG:BrandOpel, General Motors or Ford	-0,15680	-0,197000	0,843450
RegionH:BrandOpel, General Motors or Ford	-0,23230	-0,586000	0,557700
RegionI:BrandOpel, General Motors or Ford	0,05810	0,167000	0,867290
RegionJ:BrandOpel, General Motors or Ford	0,66790	1,523000	0,127870
RegionB:Brandother	0,74600	0,945000	0,344870
RegionC:Brandother	0,22220	0,433000	0,665110
RegionD:Brandother	0,19990	0,427000	0,669240
RegionE:Brandother	-0,17510	-0,209000	0,834600
RegionF:Brandother	0,22430	0,441000	0,659130
RegionG:Brandother	1,60700	1,780000	0,075060
RegionH:Brandother	0,47320	0,817000	0,414140
RegionI:Brandother	0,15440	0,284000	0,776360
RegionJ:Brandother	0,63950	0,981000	0,326700
RegionB:BrandRenault, Nissan or Citroen	0,32450	0,571000	0,568000
RegionC:BrandRenault, Nissan or Citroen	-0,39450	-1,290000	0,197160
RegionD:BrandRenault, Nissan or Citroen	-0,25150	-0,935000	0,349690
RegionE:BrandRenault, Nissan or Citroen	-0,62200	-1,128000	0,259480
RegionF:BrandRenault, Nissan or Citroen	-0,28750	-0,959000	0,337760
RegionG:BrandRenault, Nissan or Citroen	0,38190	0,582000	0,560870
RegionH:BrandRenault, Nissan or Citroen	-0,37280	-1,058000	0,290180
RegionI:BrandRenault, Nissan or Citroen	-0,23420	-0,751000	0,452730
RegionJ:BrandRenault, Nissan or Citroen	0,30880	0,771000	0,440450
RegionB:BrandVolkswagen, Audi, Skoda or Seat	0,17730	0,271000	0,786210
RegionC:BrandVolkswagen, Audi, Skoda or Seat	-0,33330	-0,945000	0,344560
RegionD:BrandVolkswagen, Audi, Skoda or Seat	-0,18970	-0,608000	0,543070
RegionE:BrandVolkswagen, Audi, Skoda or Seat	-0,42800	-0,648000	0,517050
RegionF:BrandVolkswagen, Audi, Skoda or Seat	-0,49630	-1,403000	0,160720
RegionG:BrandVolkswagen, Audi, Skoda or Seat	0,03671	0,048000	0,962040
RegionH:BrandVolkswagen, Audi, Skoda or Seat	-0,18860	-0,471000	0,637600
RegionI:BrandVolkswagen, Audi, Skoda or Seat	-0,12370	-0,339000	0,734890
RegionJ:BrandVolkswagen, Audi, Skoda or Seat	0,19330	0,412000	0,680190
RegionB:GasRegular	0,15870	0,806000	0,420350
RegionC:GasRegular	-0,05328	-0,403000	0,686770
RegionD:GasRegular	0,02188	0,192000	0,847510
RegionE:GasRegular	-0,42760	-1,436000	0,150950
RegionF:GasRegular	0,02435	0,194000	0,845840
RegionG:GasRegular	0,14730	0,532000	0,594690
RegionH:GasRegular	-0,15770	-1,017000	0,309120
RegionI:GasRegular	-0,08566	-0,625000	0,532200
RegionJ:GasRegular	-0,09208	-0,558000	0,576920
BrandJapanese (except Nissan) or Korean:GasRegular	0,12060	0,700000	0,484170
BrandMercedes, Chrysler or BMW:GasRegular	-0,15340	-0,740000	0,459580
BrandOpel, General Motors or Ford:GasRegular	0,10370	0,602000	0,546850
Brandother:GasRegular	**-0,76630**	**-2,980000**	**0,002880**
BrandRenault, Nissan or Citroen:GasRegular	0,01077	0,069000	0,944770
BrandVolkswagen, Audi, Skoda or Seat:GasRegular	-0,07151	-0,403	0,687070

Fuente: Elaboración propia.

Anexo B

Coeficientes de *Lasso*, *Ridge* y *Elastic Net*

Tabla B.1: Coeficientes del segundo modelo según *Lasso*, *Elastic Net* y *Ridge*

Coeficientes	Lasso		Ridge		Elastic Net	
	λ_{min}	λ_{1se}	λ_{min}	λ_{1se}	λ_{min}	λ_{1se}
(Intercept)	-2,06014	-2,65353	-2,15999	-2,65E+00	-2,06824	-2,65353
Powere	.	.	0,02353	1,36E-39	.	.
Powerf	0,02171	.	0,03854	1,42E-39	0,02109	.
Powerg	.	.	0,02156	-1,12E-39	.	.
Powerh	.	.	0,05768	1,59E-39	.	.
Poweri	.	.	0,02068	1,47E-39	.	.
Powerj	.	.	0,05668	1,81E-39	.	.
Powerk	0,00713	.	0,04321	2,85E-39	0,00587	.
Powerl	.	.	0,00696	7,20E-40	.	.
Powerm	0,28210	.	0,29576	1,82E-38	0,27865	.
Powern	.	.	0,00574	-1,17E-40	.	.
Powero	.	.	-0,02182	8,25E-40	.	.
CarAge	-0,00985	.	-0,00890	-4,83E-40	-0,00965	.
DriverAge	-0,01014	.	-0,00840	-4,70E-40	-0,01001	.
BrandJapanese (except Nissan) or Korean	-0,10735	.	-0,11115	-7,36E-40	-0,10271	.
BrandMercedes, Chrysler or BMW	.	.	0,04589	5,37E-39	.	.
BrandOpel, General Motors or Ford	0,06536	.	0,05068	7,23E-39	0,06551	.
Brandother	.	.	0,04198	1,30E-39	.	.
BrandRenault, Nissan or Citroen	.	.	-0,00445	-5,37E-39	.	.
BrandVolkswagen, Audi, Skoda or Seat	.	.	0,02960	6,74E-39	.	.
GasRegular	-0,02830	.	-0,04176	-3,67E-39	-0,02861	.
RegionB	-0,02258	.	-0,07491	-1,77E-39	-0,02112	.
RegionC	-0,07174	.	-0,08915	-2,35E-39	-0,06971	.
RegionD	-0,15151	.	-0,14631	-8,57E-39	-0,14918	.
RegionE	-0,02782	.	-0,11625	-2,27E-39	-0,02616	.
RegionF	0,13564	.	0,13047	8,37E-39	0,13474	.
RegionG	.	.	0,08369	6,06E-39	.	.
RegionH	0,07750	.	0,10421	7,89E-39	0,07790	.
RegionI	.	.	-0,01961	1,65E-39	.	.
RegionJ	.	.	-0,01457	1,51E-39	.	.
Powere:BrandJapanese (except Nissan) or Korean	-0,14790	.	-0,18421	-6,20E-39	-0,14719	.
Powerf:BrandJapanese (except Nissan) or Korean	.	.	0,04889	2,24E-39	.	.
Powerg:BrandJapanese (except Nissan) or Korean	-0,01218	.	-0,06529	-1,87E-39	-0,01441	.
Powerh:BrandJapanese (except Nissan) or Korean	.	.	0,00572	1,59E-39	.	.
Poweri:BrandJapanese (except Nissan) or Korean	.	.	0,07426	1,30E-39	.	.
Powerj:BrandJapanese (except Nissan) or Korean	0,10337	.	0,13777	6,20E-39	0,09887	.
Powerk:BrandJapanese (except Nissan) or Korean	.	.	0,05134	2,71E-39	.	.
Powerl:BrandJapanese (except Nissan) or Korean	.	.	0,11898	4,75E-39	.	.
Powerm:BrandJapanese (except Nissan) or Korean	.	.	-0,05516	3,63E-39	.	.
Powern:BrandJapanese (except Nissan) or Korean	.	.	-0,23809	-1,59E-38	.	.

Coeficientes	Lasso		Ridge		Elastic Net	
	λ_{min}	λ_{1se}	λ_{min}	λ_{1se}	λ_{min}	λ_{1se}
Powero:BrandJapanese (except Nissan) or Korean	.	.	0,18951	6,76E-39	.	.
Powere:BrandMercedes, Chrysler or BMW	.	.	-0,19051	-4,31E-39	.	.
Powerf:BrandMercedes, Chrysler or BMW	0,59257	.	0,62252	4,14E-38	0,58862	.
Powerg:BrandMercedes, Chrysler or BMW	0,19006	.	0,18433	1,11E-38	0,18789	.
Powerh:BrandMercedes, Chrysler or BMW	0,16833	.	0,17483	1,09E-38	0,16582	.
Poweri:BrandMercedes, Chrysler or BMW	.	.	-0,15770	-7,68E-39	.	.
Powerj:BrandMercedes, Chrysler or BMW	.	.	-0,19210	-8,14E-39	.	.
Powerk:BrandMercedes, Chrysler or BMW	.	.	0,08765	6,14E-39	.	.
Powerl:BrandMercedes, Chrysler or BMW	.	.	-0,15001	-6,76E-39	.	.
Powerm:BrandMercedes, Chrysler or BMW	.	.	0,11692	2,00E-38	.	.
Powern:BrandMercedes, Chrysler or BMW	-0,26798	.	-0,61928	-2,38E-38	-0,26353	.
Powero:BrandMercedes, Chrysler or BMW	.	.	-0,50483	-2,10E-38	.	.
Powere:BrandOpel, General Motors or Ford	0,11850	.	0,15538	1,18E-38	0,11748	.
Powerf:BrandOpel, General Motors or Ford	.	.	0,02470	4,04E-39	.	.
Powerg:BrandOpel, General Motors or Ford	0,01177	.	0,09209	6,70E-39	0,01177	.
Powerh:BrandOpel, General Motors or Ford	0,06589	.	0,17940	1,33E-38	0,06535	.
Poweri:BrandOpel, General Motors or Ford	.	.	0,15872	1,01E-38	.	.
Powerj:BrandOpel, General Motors or Ford	.	.	0,01314	3,30E-39	.	.
Powerk:BrandOpel, General Motors or Ford	-0,11705	.	-0,40951	-1,43E-38	-0,11378	.
Powerl:BrandOpel, General Motors or Ford	0,44965	.	0,83411	4,86E-38	0,44445	.
Powerm:BrandOpel, General Motors or Ford	0,06748	.	0,39547	5,74E-38	0,07134	.
Powern:BrandOpel, General Motors or Ford	.	.	-1,15694	-3,79E-38	.	.
Powero:BrandOpel, General Motors or Ford	0,68673	.	1,10203	9,40E-38	0,68277	.
Powere:Brandother	.	.	-0,19712	-8,49E-39	.	.
Powerf:Brandother	.	.	0,08223	6,19E-39	.	.
Powerg:Brandother	.	.	-0,04268	-1,84E-39	.	.
Powerh:Brandother	.	.	-0,19961	-7,91E-39	.	.
Poweri:Brandother	.	.	-0,10815	-5,76E-39	.	.
Powerj:Brandother	0,02665	.	0,12786	9,77E-39	0,02625	.
Powerk:Brandother	0,07379	.	0,26053	1,80E-38	0,07450	.
Powerl:Brandother	.	.	-0,05138	-2,35E-39	.	.
Powerm:Brandother	.	.	-0,51838	-1,25E-38	.	.
Powern:Brandother	0,57898	.	0,91106	4,93E-38	0,57272	.
Powero:Brandother	.	.	-0,92101	-2,99E-38	.	.
Powere:BrandRenault, Nissan or Citroen	.	.	-0,01750	-3,17E-39	.	.
Powerf:BrandRenault, Nissan or Citroen	.	.	0,01251	-6,57E-40	.	.
Powerg:BrandRenault, Nissan or Citroen	-0,01307	.	-0,03274	-5,45E-39	-0,01354	.
Powerh:BrandRenault, Nissan or Citroen	.	.	-0,01014	-1,29E-39	.	.
Poweri:BrandRenault, Nissan or Citroen	0,07409	.	0,13382	4,75E-39	0,07311	.
Powerj:BrandRenault, Nissan or Citroen	.	.	-0,10368	-6,05E-39	.	.
Powerk:BrandRenault, Nissan or Citroen	.	.	0,04049	5,25E-40	.	.
Powerl:BrandRenault, Nissan or Citroen	.	.	0,18851	9,55E-39	.	.
Powerm:BrandRenault, Nissan or Citroen	.	.	-0,01505	6,76E-39	.	.
Powern:BrandRenault, Nissan or Citroen	0,31324	.	0,60715	3,43E-38	0,31024	.
Powero:BrandRenault, Nissan or Citroen	-0,54628	.	-1,22201	-5,33E-38	-0,53791	.
Powere:BrandVolkswagen, Audi, Skoda or Seat	0,16522	.	0,16784	1,26E-38	0,16422	.
Powerf:BrandVolkswagen, Audi, Skoda or Seat	.	.	0,01533	3,83E-39	.	.
Powerg:BrandVolkswagen, Audi, Skoda or Seat	0,18878	.	0,20894	1,33E-38	0,18745	.
Powerh:BrandVolkswagen, Audi, Skoda or Seat	-0,05732	.	-0,26141	-1,01E-38	-0,05627	.
Poweri:BrandVolkswagen, Audi, Skoda or Seat	.	.	0,03620	2,74E-39	.	.
Powerj:BrandVolkswagen, Audi, Skoda or Seat	.	.	0,04501	5,52E-39	.	.
Powerk:BrandVolkswagen, Audi, Skoda or Seat	.	.	0,26225	1,63E-38	.	.
Powerl:BrandVolkswagen, Audi, Skoda or Seat	-0,13845	.	-0,66001	-2,35E-38	-0,13535	.
Powerm:BrandVolkswagen, Audi, Skoda or Seat	-0,27532	.	-1,29225	-4,13E-38	-0,26860	.
Powern:BrandVolkswagen, Audi, Skoda or Seat	.	.	0,14809	1,04E-38	.	.
Powero:BrandVolkswagen, Audi, Skoda or Seat	.	.	0,11809	9,60E-39	.	.

Fuente: Elaboración propia.

Tabla B.2: Coeficientes del tercer modelo según *Lasso*, *Elastic Net* y *Ridge*

Coeficientes	Lasso		Ridge		Elastic Net	
	λ_{min}	λ_{1se}	λ_{min}	λ_{1se}	λ_{min}	λ_{1se}
(Intercept)	-2,09228	-2,65353	-2,33586	-2,65E+00	-2,10212	-2,65353
CarAge	-0,00854	.	-0,00512	-4,83E-40	-0,00831	.
DriverAge	-0,00972	.	-0,00515	-4,70E-40	-0,00955	.
Powere	.	.	0,01778	1,36E-39	.	.

Coeficientes	Lasso		Ridge		Elastic Net	
	λ_{min}	λ_{1se}	λ_{min}	λ_{1se}	λ_{min}	λ_{1se}
Powerf	.	.	0,01907	1,42E-39	.	.
Powerg	.	.	0,00157	-1,12E-39	.	.
Powerh	.	.	0,01760	1,59E-39	.	.
Poweri	.	.	0,00384	1,47E-39	.	.
Powerj	.	.	0,01312	1,81E-39	.	.
Powerk	.	.	0,01387	2,85E-39	.	.
Powerl	.	.	-0,01098	7,20E-40	.	.
Powerm	0,06192	.	0,08030	1,82E-38	0,06276	.
Powern	.	.	-0,02895	-1,17E-40	.	.
Powero	.	.	0,00858	8,25E-40	.	.
RegionB	.	.	-0,03380	-1,77E-39	.	.
RegionC	.	.	-0,02185	-2,35E-39	.	.
RegionD	-0,05218	.	-0,05009	-8,57E-39	-0,05284	.
RegionE	.	.	-0,02568	-2,27E-39	.	.
RegionF	0,06887	.	0,04130	8,37E-39	0,06520	.
RegionG	.	.	0,00948	6,06E-39	.	.
RegionH	.	.	0,04008	7,89E-39	.	.
RegionI	.	.	-0,00092	1,65E-39	.	.
RegionJ	.	.	-0,00254	1,51E-39	.	.
BrandJapanese (except Nissan) or Korean	-0,05077	.	-0,03456	-7,36E-40	-0,04749	.
BrandMercedes, Chrysler or BMW	.	.	0,02279	5,37E-39	.	.
BrandOpel, General Motors or Ford	.	.	0,02758	7,23E-39	.	.
Brandother	.	.	0,02491	1,30E-39	.	.
BrandRenault, Nissan or Citroen	.	.	-0,01016	-5,37E-39	.	.
BrandVolkswagen, Audi, Skoda or Seat	.	.	0,03083	6,74E-39	.	.
GasRegular	-0,01075	.	-0,02005	-3,67E-39	-0,01195	.
Powere:RegionB	.	.	-0,04970	-4,04E-39	.	.
Powerf:RegionB	.	.	0,06678	7,14E-39	.	.
Powerg:RegionB	.	.	-0,02925	-3,35E-39	.	.
Powerh:RegionB	.	.	0,00170	1,05E-40	.	.
Poweri:RegionB	.	.	-0,15851	-1,28E-38	.	.
Powerj:RegionB	.	.	0,07226	5,59E-39	.	.
Powerk:RegionB	.	.	-0,20232	-1,59E-38	.	.
Powerl:RegionB	.	.	0,14385	5,49E-39	.	.
Powerm:RegionB	0,07538	.	0,67126	7,42E-38	0,07435	.
Powern:RegionB	.	.	-0,50210	-5,27E-38	.	.
Powero:RegionB	.	.	-0,26906	-2,50E-38	.	.
Powere:RegionC	.	.	0,00781	-4,42E-40	.	.
Powerf:RegionC	.	.	-0,01673	-2,21E-39	.	.
Powerg:RegionC	.	.	-0,02892	-4,36E-39	.	.
Powerh:RegionC	.	.	-0,06352	-5,05E-39	.	.
Poweri:RegionC	.	.	0,10453	8,47E-39	.	.
Powerj:RegionC	.	.	0,04666	2,80E-39	.	.
Powerk:RegionC	0,27340	.	0,30602	2,68E-38	0,26938	.
Powerl:RegionC	.	.	0,25144	1,94E-38	.	.
Powerm:RegionC	0,22848	.	0,35728	4,52E-38	0,22496	.
Powern:RegionC	.	.	0,18916	1,81E-38	.	.
Powero:RegionC	.	.	-0,09177	-1,24E-38	.	.
Powere:RegionD	.	.	-0,00553	-3,21E-39	.	.
Powerf:RegionD	.	.	-0,03006	-5,79E-39	.	.
Powerg:RegionD	.	.	0,00851	-3,55E-39	.	.
Powerh:RegionD	.	.	-0,02045	-3,38E-39	.	.
Poweri:RegionD	.	.	-0,04130	-5,40E-39	.	.
Powerj:RegionD	-0,02754	.	-0,09870	-1,04E-38	-0,02750	.
Powerk:RegionD	.	.	-0,06094	-6,97E-39	.	.
Powerl:RegionD	-0,43773	.	-0,36726	-2,84E-38	-0,42541	.
Powerm:RegionD	.	.	0,03219	4,87E-39	.	.
Powern:RegionD	.	.	0,05079	1,89E-39	.	.
Powero:RegionD	.	.	-0,25627	-2,32E-38	.	.
Powere:RegionE	.	.	-0,09959	-6,59E-39	.	.
Powerf:RegionE	.	.	-0,00735	-2,77E-40	.	.
Powerg:RegionE	.	.	-0,05288	-4,99E-39	.	.
Powerh:RegionE	.	.	-0,19320	-1,36E-38	.	.
Poweri:RegionE	.	.	0,23390	1,49E-38	.	.
Powerj:RegionE	-0,06427	.	-0,39336	-2,57E-38	-0,06198	.
Powerk:RegionE	.	.	0,31251	2,41E-38	.	.
Powerl:RegionE	0,59445	.	0,85328	6,95E-38	0,58723	.
Powerm:RegionE	.	.	-0,28336	-1,55E-38	.	.

Coeficientes	Lasso		Ridge		Elastic Net	
	λ_{min}	λ_{1se}	λ_{min}	λ_{1se}	λ_{min}	λ_{1se}
Powern:RegionE	.	.	0,57315	3,38E-38	.	.
Powero:RegionE	.	.	-0,41437	-2,96E-38	.	.
Powere:RegionF	.	.	0,01726	5,37E-39	.	.
Powerf:RegionF	0,01164	.	0,06572	1,16E-38	0,01348	.
Powerg:RegionF	.	.	-0,02369	2,77E-39	.	.
Powerh:RegionF	.	.	0,05439	8,32E-39	.	.
Poweri:RegionF	0,04740	.	0,10390	1,52E-38	0,04728	.
Powerj:RegionF	.	.	0,07301	1,05E-38	.	.
Powerk:RegionF	.	.	-0,02291	3,19E-39	.	.
Powerl:RegionF	.	.	0,05861	7,14E-39	.	.
Powerm:RegionF	.	.	-0,10384	4,49E-39	.	.
Powern:RegionF	.	.	0,06862	8,82E-39	.	.
Powero:RegionF	.	.	0,04452	9,38E-39	.	.
Powere:RegionG	0,09499	.	0,23115	2,38E-38	0,09408	.
Powerf:RegionG	.	.	-0,02542	4,54E-39	.	.
Powerg:RegionG	.	.	-0,03206	2,07E-39	.	.
Powerh:RegionG	.	.	0,08127	1,16E-38	.	.
Poweri:RegionG	.	.	-0,32544	-1,56E-38	.	.
Powerj:RegionG	.	.	-0,14384	-9,97E-39	.	.
Powerk:RegionG	.	.	-0,13396	-7,34E-39	.	.
Powerl:RegionG	.	.	0,52571	4,54E-38	.	.
Powerm:RegionG	0,15452	.	0,75852	1,14E-37	0,14779	.
Powern:RegionG	.	.	-0,51681	-4,07E-38	.	.
Powero:RegionG	.	.	.	.	.	.
Powere:RegionH	.	.	-0,01847	4,90E-39	.	.
Powerf:RegionH	0,16240	.	0,12503	1,65E-38	0,15962	.
Powerg:RegionH	.	.	-0,00429	2,36E-39	.	.
Powerh:RegionH	.	.	0,02306	4,90E-39	.	.
Poweri:RegionH	.	.	0,01443	6,20E-39	.	.
Powerj:RegionH	0,13366	.	0,23305	2,21E-38	0,13043	.
Powerk:RegionH	.	.	0,11498	1,15E-38	.	.
Powerl:RegionH	0,03641	.	0,36234	2,84E-38	0,03354	.
Powerm:RegionH	.	.	0,11834	1,52E-38	.	.
Powern:RegionH	.	.	0,66460	5,97E-38	.	.
Powero:RegionH	.	.	-0,46892	-2,81E-38	.	.
Powere:RegionI	.	.	0,03955	5,91E-39	.	.
Powerf:RegionI	.	.	0,03550	4,90E-39	.	.
Powerg:RegionI	.	.	-0,00037	4,10E-40	.	.
Powerh:RegionI	.	.	-0,06423	-3,53E-39	.	.
Poweri:RegionI	.	.	-0,05479	-3,71E-39	.	.
Powerj:RegionI	.	.	0,07253	6,02E-39	.	.
Powerk:RegionI	.	.	-0,00760	-2,88E-40	.	.
Powerl:RegionI	.	.	-0,02967	-3,78E-39	.	.
Powerm:RegionI	0,68542	.	0,77794	9,23E-38	0,67928	.
Powern:RegionI	.	.	-0,45795	-3,46E-38	.	.
Powero:RegionI	.	.	-0,44130	-2,95E-38	.	.
Powere:RegionJ	.	.	-0,05600	-1,41E-39	.	.
Powerf:RegionJ	.	.	-0,06045	-3,09E-39	.	.
Powerg:RegionJ	.	.	-0,00323	1,46E-39	.	.
Powerh:RegionJ	.	.	0,07133	8,30E-39	.	.
Poweri:RegionJ	.	.	-0,03450	-7,00E-40	.	.
Powerj:RegionJ	.	.	0,12910	1,29E-38	.	.
Powerk:RegionJ	.	.	0,21494	1,85E-38	.	.
Powerl:RegionJ	.	.	0,26711	2,27E-38	.	.
Powerm:RegionJ	.	.	0,11483	2,70E-38	.	.
Powern:RegionJ	.	.	-0,19122	-1,60E-38	.	.
Powero:RegionJ	0,50543	.	1,01950	8,73E-38	0,49530	.
Powere:BrandJapanese (except Nissan) or Korean	-0,07532	.	-0,09216	-6,20E-39	-0,07447	.
Powerf:BrandJapanese (except Nissan) or Korean	.	.	0,01600	2,24E-39	.	.
Powerg:BrandJapanese (except Nissan) or Korean	.	.	-0,02764	-1,87E-39	.	.
Powerh:BrandJapanese (except Nissan) or Korean	.	.	-0,00139	1,59E-39	.	.
Poweri:BrandJapanese (except Nissan) or Korean	.	.	-0,00018	1,30E-39	.	.
Powerj:BrandJapanese (except Nissan) or Korean	0,01337	.	0,05900	6,20E-39	0,01165	.
Powerk:BrandJapanese (except Nissan) or Korean	.	.	0,01775	2,71E-39	.	.
Powerl:BrandJapanese (except Nissan) or Korean	.	.	0,03398	4,75E-39	.	.
Powerm:BrandJapanese (except Nissan) or Korean	.	.	-0,03453	3,63E-39	.	.
Powern:BrandJapanese (except Nissan) or Korean	.	.	-0,16418	-1,59E-38	.	.
Powero:BrandJapanese (except Nissan) or Korean	.	.	0,09512	6,76E-39	.	.

Coeficientes	Lasso		Ridge		Elastic Net	
	λ_{min}	λ_{1se}	λ_{min}	λ_{1se}	λ_{min}	λ_{1se}
Powere:BrandMercedes, Chrysler or BMW	.	.	-0,12250	-4,31E-39	.	.
Powerf:BrandMercedes, Chrysler or BMW	0,47314	.	0,42263	4,14E-38	0,46789	.
Powerg:BrandMercedes, Chrysler or BMW	0,09847	.	0,10708	1,11E-38	0,09770	.
Powerh:BrandMercedes, Chrysler or BMW	0,05148	.	0,11324	1,09E-38	0,05065	.
Poweri:BrandMercedes, Chrysler or BMW	.	.	-0,12225	-7,68E-39	.	.
Powerj:BrandMercedes, Chrysler or BMW	.	.	-0,12647	-8,14E-39	.	.
Powerk:BrandMercedes, Chrysler or BMW	.	.	0,02951	6,14E-39	.	.
Powerl:BrandMercedes, Chrysler or BMW	.	.	-0,09335	-6,76E-39	.	.
Powerm:BrandMercedes, Chrysler or BMW	.	.	0,04460	2,00E-38	.	.
Powern:BrandMercedes, Chrysler or BMW	-0,08368	.	-0,32360	-2,38E-38	-0,08274	.
Powero:BrandMercedes, Chrysler or BMW	.	.	-0,28673	-2,10E-38	.	.
Powere:BrandOpel, General Motors or Ford	0,10489	.	0,09364	1,18E-38	0,10468	.
Powerf:BrandOpel, General Motors or Ford	.	.	0,00266	4,04E-39	.	.
Powerg:BrandOpel, General Motors or Ford	.	.	0,03519	6,70E-39	.	.
Powerh:BrandOpel, General Motors or Ford	.	.	0,10355	1,33E-38	.	.
Poweri:BrandOpel, General Motors or Ford	.	.	0,06893	1,01E-38	.	.
Powerj:BrandOpel, General Motors or Ford	.	.	-0,00562	3,30E-39	.	.
Powerk:BrandOpel, General Motors or Ford	.	.	-0,24880	-1,43E-38	.	.
Powerl:BrandOpel, General Motors or Ford	0,27059	.	0,58602	4,86E-38	0,26096	.
Powerm:BrandOpel, General Motors or Ford	0,00379	.	0,27550	5,74E-38	0,00517	.
Powern:BrandOpel, General Motors or Ford	.	.	-0,64962	-3,79E-38	.	.
Powero:BrandOpel, General Motors or Ford	0,34430	.	0,86609	9,40E-38	0,34053	.
Powere:Brandother	.	.	-0,13084	-8,49E-39	.	.
Powerf:Brandother	.	.	0,06905	6,19E-39	.	.
Powerg:Brandother	.	.	-0,01143	-1,84E-39	.	.
Powerh:Brandother	.	.	-0,11732	-7,91E-39	.	.
Poweri:Brandother	.	.	-0,06627	-5,76E-39	.	.
Powerj:Brandother	.	.	0,10762	9,77E-39	.	.
Powerk:Brandother	.	.	0,20863	1,80E-38	.	.
Powerl:Brandother	.	.	-0,07728	-2,35E-39	.	.
Powerm:Brandother	.	.	-0,32279	-1,25E-38	.	.
Powern:Brandother	0,23787	.	0,61013	4,93E-38	0,23114	.
Powero:Brandother	.	.	-0,38833	-2,99E-38	.	.
Powere:BrandRenault, Nissan or Citroen	.	.	-0,00528	-3,17E-39	.	.
Powerf:BrandRenault, Nissan or Citroen	.	.	0,01052	-6,57E-40	.	.
Powerg:BrandRenault, Nissan or Citroen	-0,00125	.	-0,02908	-5,45E-39	-0,00208	.
Powerh:BrandRenault, Nissan or Citroen	.	.	0,00239	-1,29E-39	.	.
Poweri:BrandRenault, Nissan or Citroen	0,01803	.	0,06905	4,75E-39	0,01748	.
Powerj:BrandRenault, Nissan or Citroen	.	.	-0,05488	-6,05E-39	.	.
Powerk:BrandRenault, Nissan or Citroen	.	.	0,00771	5,25E-40	.	.
Powerl:BrandRenault, Nissan or Citroen	.	.	0,14246	9,55E-39	.	.
Powerm:BrandRenault, Nissan or Citroen	.	.	-0,00093	6,76E-39	.	.
Powern:BrandRenault, Nissan or Citroen	0,07697	.	0,39472	3,43E-38	0,07501	.
Powero:BrandRenault, Nissan or Citroen	-0,23214	.	-0,62219	-5,33E-38	-0,22820	.
Powere:BrandVolkswagen, Audi, Skoda or Seat	0,09272	.	0,10506	1,26E-38	0,09325	.
Powerf:BrandVolkswagen, Audi, Skoda or Seat	.	.	0,00437	3,83E-39	.	.
Powerg:BrandVolkswagen, Audi, Skoda or Seat	0,09404	.	0,13075	1,33E-38	0,09462	.
Powerh:BrandVolkswagen, Audi, Skoda or Seat	-0,01457	.	-0,15351	-1,01E-38	-0,01285	.
Poweri:BrandVolkswagen, Audi, Skoda or Seat	.	.	-0,00021	2,74E-39	.	.
Powerj:BrandVolkswagen, Audi, Skoda or Seat	.	.	0,03175	5,52E-39	.	.
Powerk:BrandVolkswagen, Audi, Skoda or Seat	.	.	0,14374	1,63E-38	.	.
Powerl:BrandVolkswagen, Audi, Skoda or Seat	.	.	-0,35074	-2,35E-38	.	.
Powerm:BrandVolkswagen, Audi, Skoda or Seat	.	.	-0,62516	-4,13E-38	.	.
Powern:BrandVolkswagen, Audi, Skoda or Seat	.	.	0,00400	1,04E-38	.	.
Powero:BrandVolkswagen, Audi, Skoda or Seat	.	.	0,12974	9,60E-39	.	.
Powere:GasRegular	-0,03828	.	-0,04341	-4,97E-39	-0,03775	.
Powerf:GasRegular	.	.	-0,01544	-2,27E-39	.	.
Powerg:GasRegular	.	.	0,02873	-7,41E-40	.	.
Powerh:GasRegular	.	.	0,04184	3,18E-39	.	.
Poweri:GasRegular	.	.	0,04132	3,17E-39	.	.
Powerj:GasRegular	.	.	0,01656	2,57E-39	.	.
Powerk:GasRegular	.	.	0,00971	1,99E-39	.	.
Powerl:GasRegular	.	.	0,02798	2,44E-39	.	.
Powerm:GasRegular	.	.	0,07813	1,70E-38	.	.
Powern:GasRegular	.	.	0,06020	5,25E-39	.	.
Powero:GasRegular	.	.	-0,03218	-1,32E-39	.	.
RegionB:BrandJapanese (except Nissan) or Korean	.	.	-0,12886	-9,70E-39	.	.
RegionC:BrandJapanese (except Nissan) or Korean	.	.	0,00017	-5,38E-40	.	.

Coeficientes	Lasso		Ridge		Elastic Net	
	λ_{min}	λ_{1se}	λ_{min}	λ_{1se}	λ_{min}	λ_{1se}
RegionD:BrandJapanese (except Nissan) or Korean	.	.	0,07050	1,71E-39	.	.
RegionE:BrandJapanese (except Nissan) or Korean	-0,00678	.	-0,12809	-9,24E-39	-0,00862	.
RegionF:BrandJapanese (except Nissan) or Korean	.	.	-0,01870	1,96E-39	.	.
RegionG:BrandJapanese (except Nissan) or Korean	.	.	-0,13396	-7,35E-39	.	.
RegionH:BrandJapanese (except Nissan) or Korean	-0,14618	.	-0,13311	-6,38E-39	-0,14221	.
RegionI:BrandJapanese (except Nissan) or Korean	.	.	-0,03488	-2,81E-39	.	.
RegionJ:BrandJapanese (except Nissan) or Korean	.	.	-0,05655	-3,41E-39	.	.
RegionB:BrandMercedes, Chrysler or BMW	.	.	-0,12143	-6,41E-39	.	.
RegionC:BrandMercedes, Chrysler or BMW	.	.	0,05725	1,12E-38	.	.
RegionD:BrandMercedes, Chrysler or BMW	.	.	0,01798	1,16E-39	.	.
RegionE:BrandMercedes, Chrysler or BMW	.	.	-0,09582	-3,39E-39	.	.
RegionF:BrandMercedes, Chrysler or BMW	0,03972	.	0,13741	1,76E-38	0,04217	.
RegionG:BrandMercedes, Chrysler or BMW	.	.	0,06263	1,55E-38	.	.
RegionH:BrandMercedes, Chrysler or BMW	.	.	-0,05602	3,18E-39	.	.
RegionI:BrandMercedes, Chrysler or BMW	.	.	-0,04852	4,37E-40	.	.
RegionJ:BrandMercedes, Chrysler or BMW	0,02378	.	0,16195	1,92E-38	0,02423	.
RegionB:BrandOpel, General Motors or Ford	.	.	0,08136	1,07E-38	.	.
RegionC:BrandOpel, General Motors or Ford	.	.	-0,05170	-6,86E-40	.	.
RegionD:BrandOpel, General Motors or Ford	.	.	-0,01679	1,12E-39	.	.
RegionE:BrandOpel, General Motors or Ford	.	.	-0,12837	-7,08E-39	.	.
RegionF:BrandOpel, General Motors or Ford	0,16941	.	0,13710	2,17E-38	0,16982	.
RegionG:BrandOpel, General Motors or Ford	.	.	-0,15125	-2,66E-39	.	.
RegionH:BrandOpel, General Motors or Ford	.	.	0,06824	1,26E-38	.	.
RegionI:BrandOpel, General Motors or Ford	0,04762	.	0,07864	1,38E-38	0,04758	.
RegionJ:BrandOpel, General Motors or Ford	0,06153	.	0,13937	1,77E-38	0,06175	.
RegionB:Brandother	.	.	0,03183	1,59E-39	.	.
RegionC:Brandother	.	.	0,06803	3,19E-39	.	.
RegionD:Brandother	.	.	-0,04061	-7,46E-39	.	.
RegionE:Brandother	.	.	0,12037	1,01E-38	.	.
RegionF:Brandother	.	.	0,10831	1,26E-38	.	.
RegionG:Brandother	0,36257	.	0,64449	5,93E-38	0,35669	.
RegionH:Brandother	0,41270	.	0,43426	4,03E-38	0,40757	.
RegionI:Brandother	.	.	-0,03274	-2,96E-39	.	.
RegionJ:Brandother	.	.	0,00158	-8,81E-40	.	.
RegionB:BrandRenault, Nissan or Citroen	.	.	0,01637	2,62E-40	.	.
RegionC:BrandRenault, Nissan or Citroen	-0,08133	.	-0,04752	-6,45E-39	-0,07942	.
RegionD:BrandRenault, Nissan or Citroen	-0,11048	.	-0,06216	-1,08E-38	-0,10810	.
RegionE:BrandRenault, Nissan or Citroen	.	.	-0,00591	-1,95E-39	.	.
RegionF:BrandRenault, Nissan or Citroen	0,07682	.	0,08325	1,19E-38	0,07819	.
RegionG:BrandRenault, Nissan or Citroen	0,10584	.	0,16337	1,73E-38	0,10489	.
RegionH:BrandRenault, Nissan or Citroen	0,04677	.	0,08390	1,06E-38	0,04835	.
RegionI:BrandRenault, Nissan or Citroen	.	.	-0,01199	-6,32E-40	.	.
RegionJ:BrandRenault, Nissan or Citroen	.	.	0,01381	-2,40E-41	.	.
RegionB:BrandVolkswagen, Audi, Skoda or Seat	.	.	-0,05536	-6,99E-40	.	.
RegionC:BrandVolkswagen, Audi, Skoda or Seat	.	.	0,00246	4,67E-39	.	.
RegionD:BrandVolkswagen, Audi, Skoda or Seat	.	.	-0,00759	1,84E-39	.	.
RegionE:BrandVolkswagen, Audi, Skoda or Seat	.	.	0,15252	1,49E-38	.	.
RegionF:BrandVolkswagen, Audi, Skoda or Seat	.	.	-0,03202	6,03E-39	.	.
RegionG:BrandVolkswagen, Audi, Skoda or Seat	.	.	-0,01017	6,77E-39	.	.
RegionH:BrandVolkswagen, Audi, Skoda or Seat	0,18717	.	0,18990	2,41E-38	0,18705	.
RegionI:BrandVolkswagen, Audi, Skoda or Seat	.	.	0,05807	1,07E-38	.	.
RegionJ:BrandVolkswagen, Audi, Skoda or Seat	.	.	-0,04397	-3,23E-40	.	.
RegionB:GasRegular	.	.	0,00022	-2,02E-39	.	.
RegionC:GasRegular	-0,01810	.	-0,04777	-6,04E-39	-0,01826	.
RegionD:GasRegular	.	.	-0,03076	-8,63E-39	.	.
RegionE:GasRegular	-0,02778	.	-0,08834	-7,05E-39	-0,02575	.
RegionF:GasRegular	0,00273	.	0,03958	7,81E-39	0,00599	.
RegionG:GasRegular	.	.	0,02490	5,13E-39	.	.
RegionH:GasRegular	.	.	-0,03509	1,97E-39	.	.
RegionI:GasRegular	-0,00352	.	-0,04043	-2,44E-39	-0,00274	.
RegionJ:GasRegular	.	.	-0,03317	-1,98E-39	.	.
BrandJapanese (except Nissan) or Korean:GasRegul	.	.	-0,00891	-1,15E-41	.	.
BrandMercedes, Chrysler or BMW:GasRegular	.	.	-0,03347	1,95E-39	.	.
BrandOpel, General Motors or Ford:GasRegular	.	.	0,02925	6,11E-39	.	.
Brandother:GasRegular	-0,08976	.	-0,14103	-8,55E-39	-0,08667	.
BrandRenault, Nissan or Citroen:GasRegular	.	.	-0,01361	-6,11E-39	.	.
BrandVolkswagen, Audi, Skoda or Seat:GasRegular	.	.	-0,02435	2,03E-39	.	.

Fuente: Elaboración propia.

Anexo C

Código en R

```
#install.packages("caret")
library("caret")
#install.packages("visreg")
library("visreg")
#install.packages("MASS")
library("MASS")
#install.packages("lme4")
library("lme4")
#install.packages("glmnet")
library("glmnet")
#install.packages("ggplot2")
library("ggplot2")
#install.packages("mpath")
library("mpath")
#install.packages("AICcmodavg")
library("AICcmodavg")
#install.packages("zic")
library("zic")
#install.packages("pscl")
library("pscl")
#install.packages("dvmisc")
library("dvmisc")
#install.packages("plotmo")
library(plotmo) # for plot_glmnet
#install.packages("rcompanion")
library(rcompanion)
#install.packages("xts")
library("xts")
#install.packages("zoo")
library("zoo")
#install.packages("CASdatasets",repos="http://cas.uqam.ca/pub/R/",type="
    source")
library("CASdatasets")


ptn_total = Sys.time()
```

```
data("freMTPLfreq")

#Analisis de los valores de la Base de Datos
summary(freMTPLfreq)
summary(freMTPLfreq$ClaimNb)
summary(freMTPLfreq$Exposure)
summary(freMTPLfreq$Power)
summary(freMTPLfreq$CarAge)
summary(freMTPLfreq$DriverAge)
summary(freMTPLfreq$Brand)
summary(freMTPLfreq$Gas)
summary(freMTPLfreq$Region)

#Como en CarAge existen datos irreales (el maximo es=100), se corrige esto.
#La exposicion debe encontrarse entre 0 y 1. Tambien se corrige esto.
freMTPLfreq = subset(freMTPLfreq, Exposure<=1 & Exposure >= 0 & CarAge<=25)

#Se renombran las regiones
summary(freMTPLfreq$Region)
levels(freMTPLfreq$Region)[levels(freMTPLfreq$Region) == "Aquitaine"] = "A"
levels(freMTPLfreq$Region)[levels(freMTPLfreq$Region) == "Basse-Normandie"]
    = "B"
levels(freMTPLfreq$Region)[levels(freMTPLfreq$Region) == "Bretagne"] = "C"
levels(freMTPLfreq$Region)[levels(freMTPLfreq$Region) == "Centre"] = "D"
levels(freMTPLfreq$Region)[levels(freMTPLfreq$Region) == "Haute-Normandie"]
    = "E"
levels(freMTPLfreq$Region)[levels(freMTPLfreq$Region) == "Ile-de-France"] =
    "F"
levels(freMTPLfreq$Region)[levels(freMTPLfreq$Region) == "Limousin"] = "G"
levels(freMTPLfreq$Region)[levels(freMTPLfreq$Region) == "Nord-Pas-de-
    Calais"] = "H"
levels(freMTPLfreq$Region)[levels(freMTPLfreq$Region) == "Pays-de-la-Loire"
    ] = "I"
levels(freMTPLfreq$Region)[levels(freMTPLfreq$Region) == "Poitou-Charentes"
    ] = "J"
summary(freMTPLfreq$Region)

#se establece una semilla de modo a poder replicar
set.seed(1902)
#se crea una particion para el calculo de Lasso, Elastic Net y Ridge
folds = createDataPartition(freMTPLfreq$ClaimNb, 0.5)

dataCar = freMTPLfreq[folds[[1]], ]

set.seed(1979)
index = createDataPartition(dataCar$ClaimNb, times = 1, p = 0.75, list=
    FALSE)
train = dataCar[index,]
test  = dataCar[-index,]



#######################################################################
#                          ANALISIS PREVIO                            #
#######################################################################
```

```r
summary(dataCar)
summary(dataCar$ClaimNb)
summary(dataCar$Exposure)
summary(dataCar$Power)
summary(dataCar$CarAge)
summary(dataCar$DriverAge)
summary(dataCar$Brand)
summary(dataCar$Gas)
summary(dataCar$Region)
dim(dataCar)

#V de Cramer

tabla = ftable(as.factor(train$Power), train$Brand,
               dnn = c("Power", "Brand"))
cramerV(tabla)
tabla = ftable(as.factor(train$Power), train$Gas,
               dnn = c("Power", "Gas"))
cramerV(tabla)
tabla = ftable(as.factor(train$Power), train$Region,
               dnn = c("Power", "Region"))
cramerV(tabla)
tabla = ftable(as.factor(train$Brand), train$Gas,
               dnn = c("Brand", "Gas"))
cramerV(tabla)
tabla = ftable(as.factor(train$Brand), train$Region,
               dnn = c("Brand", "Region"))
cramerV(tabla)
tabla = ftable(as.factor(train$Gas), train$Region,
               dnn = c("Gas", "Region"))
cramerV(tabla)
#Pearson, Kendall y Spearmen de variables numericas
mat_continuous = data.frame(dataCar$Exposure,dataCar$CarAge,dataCar$
    DriverAge)
cor_pearson = cor(mat_continuous, method = c("pearson")); cor_pearson
cor_kendall = cor(mat_continuous, method = c("kendall")); cor_kendall
cor_spearman = cor(mat_continuous, method = c("spearman")); cor_spearman


#Variables categoricas vs numericas

boxplot(Exposure ~ Power, data = dataCar, ylab = "Exposure")
model.lm.1 <- lm(Exposure ~ Power, data = dataCar)
summary(model.lm.1)
boxplot(Exposure ~ Brand, data = dataCar, ylab = "Exposure")
model.lm.2 <- lm(Exposure ~ Brand, data = dataCar)
summary(model.lm.2)
boxplot(Exposure ~ Gas, data = dataCar, ylab = "Exposure")
model.lm.3 <- lm(Exposure ~ Gas, data = dataCar)
summary(model.lm.3)
boxplot(Exposure ~ Region, data = dataCar, ylab = "Exposure")
model.lm.4 <- lm(Exposure ~ Region, data = dataCar)
summary(model.lm.4)
boxplot(CarAge ~ Power, data = dataCar, ylab = "CarAge")
```

```
model.lm.5 <- lm(CarAge ~ Power, data = dataCar)
summary(model.lm.5)
boxplot(CarAge ~ Brand, data = dataCar, ylab = "CarAge")
model.lm.6 <- lm(CarAge ~ Brand, data = dataCar)
summary(model.lm.6)
boxplot(CarAge ~ Gas, data = dataCar, ylab = "CarAge")
model.lm.7 <- lm(CarAge ~ Gas, data = dataCar)
summary(model.lm.7)
boxplot(CarAge ~ Region, data = dataCar, ylab = "CarAge")
model.lm.8 <- lm(CarAge ~ Region, data = dataCar)
summary(model.lm.8)
boxplot(DriverAge ~ Power, data = dataCar, ylab = "DriverAge")
model.lm.9 <- lm(DriverAge ~ Power, data = dataCar)
summary(model.lm.9)
boxplot(DriverAge ~ Brand, data = dataCar, ylab = "DriverAge")
model.lm.10 <- lm(DriverAge ~ Brand, data = dataCar)
summary(model.lm.10)
boxplot(DriverAge ~ Gas, data = dataCar, ylab = "DriverAge")
model.lm.11 <- lm(DriverAge ~ Gas, data = dataCar)
summary(model.lm.11)
boxplot(DriverAge ~ Region, data = dataCar, ylab = "DriverAge")
model.lm.12 <- lm(DriverAge ~ Region, data = dataCar)
summary(model.lm.12)


##########
#graficos#
##########

theme_set(theme_bw())

#exposicion
ggplot(dataCar, aes(x=Exposure))+geom_histogram(color="darkblue", fill="
    lightblue")+
    labs(title="Exposicion",x="Exposicion", y = "Cantidad")+
    theme(plot.title = element_text(hjust = 0.5))

#ClaimNb
ggplot(dataCar, aes(x=factor(ClaimNb)))+
    geom_bar(stat="count", width=1, fill="tomato3")+
    labs(title="Cantidad de Siniestros",x="Numero de Siniestros", y = "
        Cantidad de reclamos")+
    theme(plot.title = element_text(hjust = 0.5))

#CarAge-siniestros
ggplot(dataCar, aes(x=CarAge, y=ClaimNb)) +
    geom_bar(stat="identity", width=.5, fill="tomato3") +
    labs(title="Frecuencia de siniestros segun antiguedad del vehiculo", x="
        Antiguedad del vehiculo", y="Cantidad de siniestros")

#DriverAge-siniestros
ggplot(dataCar, aes(x=DriverAge, y=ClaimNb)) +
    geom_bar(stat="identity", width=.5, fill="tomato3") +
    labs(title="Cantidad de siniestros segun edad del conductor", x="Edad
        del asegurado", y="Cantidad de siniestros")
```

```
#DriverAge-cantidad
ggplot(dataCar, aes(x=DriverAge))+geom_histogram(color="black", fill="
    tomato3")+
    labs(title="Distribucion de edad de asegurados",x="Edad del asegurado",
        y="Cantidad")

#relacion entre edad y siniestros
a=table(dataCar$DriverAge)
b=aggregate(ClaimNb ~ DriverAge, data = dataCar, FUN = sum)



#####################################################################
#                              POISSON                              #
#####################################################################

ptn=Sys.time()
Poisson_1 = glm(ClaimNb ~ Power + CarAge + DriverAge + Brand + Gas + Region
    +offset(log(Exposure)),data = train,family="poisson")
summary(Poisson_1)
ptn_Poi_1 = Sys.time() - ptn
ptn_Poi_1

ptn=Sys.time()
Poisson_2 = glm(ClaimNb ~ Power + CarAge + DriverAge + Brand + Gas + Region
    +offset(log(Exposure)) + Power*Brand,data = train,family="poisson")
summary(Poisson_2)
ptn_Poi_2 = Sys.time() - ptn
ptn_Poi_2

ptn=Sys.time()
Poisson_3 = glm(ClaimNb ~ CarAge + DriverAge + Power*Region + Power*Brand +
    Power*Gas + Region* Brand + Region* Gas + Brand*Gas+offset(log(
    Exposure)),data = train,family="poisson")
ptn_Poi_3 = Sys.time() - ptn
ptn_Poi_3

options(max.print=1000000)
summary(Poisson_3)
options(max.print=200)


#AIC, AICc y BIC
AICc_poisson = function(fit){

    loglik = logLik(fit)
    n    = attributes(loglik)$nobs
    p    = attributes(loglik)$df
    dev = -2*as.numeric(loglik)

    AIC_poi  = dev+2*p
    AICc_poi = AIC_poi + (2*p^2+2*p)/(n-p-1)
    BIC_poi  = dev+2*p*log(n)
```

```
    return(list('AIC' = AIC_poi,"AICc"=AICc_poi,'BIC' = BIC_poi))
}

medidas_Poi=cbind(AICc_poisson(Poisson_1),AICc_poisson(Poisson_2),AICc_
    poisson(Poisson_3))
medidas_Poi



#####################################################################
#                          LASSO POISSON                            #
#####################################################################


#####################################
#     LASSO SIN INTERACCION         #
#####################################

train.x_1 = model.matrix(ClaimNb ~ Power + CarAge + DriverAge + Brand + Gas
    + Region, data=train)
set.seed(2002)
folds = createFolds(train$ClaimNb, 5, list=FALSE)

set.seed(3)
ptn=Sys.time()
cv_lasso_1 = cv.glmnet(train.x_1, y = train$ClaimNb, offset = log(train$
    Exposure),family = "poisson",alpha = 1,nfolds = 5,foldid = folds,maxit
    =10^3,nlambda = 50)
ptn_1 = Sys.time() - ptn
ptn_1

optimal_lambda_1=cv_lasso_1$lambda.min; optimal_lambda_1
opt.lam_1 = c(cv_lasso_1$lambda.min, cv_lasso_1$lambda.1se)
coef(cv_lasso_1, s = opt.lam_1)

lasso_reg_1 = glmnet(train.x_1, y =train$ClaimNb, offset = log(train$
    Exposure), family = "poisson", alpha=1,lambda = optimal_lambda_1)
lasso_reg_1$dev.ratio
lasso_reg_1$nulldev

lasso_reg_1.5 = glmnet(train.x_1, y =train$ClaimNb, offset = log(train$
    Exposure), family = "poisson", alpha=1)

plot(cv_lasso_1)
plot(lasso_reg_1.5)

#####################################
#LASSO INTERACCION POWER Y BRAND#
#####################################

train.x_2 = model.matrix(ClaimNb ~ Power + CarAge + DriverAge + Brand + Gas
    + Region + Power*Brand, data=train)
set.seed(2002)
folds = createFolds(train$ClaimNb, 5, list=FALSE)
```

```
set.seed(3)
ptn=Sys.time()
cv_lasso_2 = cv.glmnet(train.x_2, y = train$ClaimNb, offset = log(train$
    Exposure),family = "poisson",alpha = 1,nfolds = 5,foldid = folds,maxit
    =10^3,nlambda = 50)
ptn_2 = Sys.time() - ptn
ptn_2

optimal_lambda_2=cv_lasso_2$lambda.min; optimal_lambda_2
opt.lam_2 = c(cv_lasso_2$lambda.min, cv_lasso_2$lambda.1se)
coef(cv_lasso_2, s = opt.lam_2)

lasso_reg_2 = glmnet(train.x_2, y =train$ClaimNb, offset = log(train$
    Exposure), family = "poisson", alpha=1,lambda = optimal_lambda_2)
lasso_reg_2$dev.ratio
lasso_reg_2$nulldev

lasso_reg_2.5 = glmnet(train.x_2, y =train$ClaimNb, offset = log(train$
    Exposure), family = "poisson", alpha=1)

plot(cv_lasso_2)
plot(lasso_reg_2.5)

####################################
#LASSO INTERACCION VARIABLES CAT#
####################################

train.x_3 = model.matrix(ClaimNb ~ CarAge + DriverAge + Power*Region +
    Power*Brand + Power*Gas +  Region* Brand + Region* Gas + Brand*Gas ,
    data=train)
set.seed(2002)
folds = createFolds(train$ClaimNb, 5, list=FALSE)

set.seed(3)
ptn_3=Sys.time()
cv_lasso_3 = cv.glmnet(train.x_3, y = train$ClaimNb, offset = log(train$
    Exposure),family = "poisson",alpha = 1,nfolds = 5,foldid = folds,maxit
    =10^3,nlambda = 50)
ptn_3 = Sys.time() - ptn
ptn_3

optimal_lambda_3=cv_lasso_3$lambda.min; optimal_lambda_3
opt.lam_3 = c(cv_lasso_3$lambda.min, cv_lasso_3$lambda.1se)
coef(cv_lasso_3, s = opt.lam_3)

lasso_reg_3 = glmnet(train.x_3, y =train$ClaimNb, offset = log(train$
    Exposure), family = "poisson", alpha=1,lambda = optimal_lambda_3)
lasso_reg_3$dev.ratio
lasso_reg_3$nulldev

lasso_reg_3.5 = glmnet(train.x_3, y =train$ClaimNb, offset = log(train$
    Exposure), family = "poisson", alpha=1)

plot(cv_lasso_3)
plot(lasso_reg_3.5)
```

```
#####################################################################
#                          RIDGE POISSON                            #
#####################################################################


#####################################
#      RIDGE SIN INTERACCION        #
#####################################

train.x_4 = model.matrix(ClaimNb ~ Power + CarAge + DriverAge + Brand + Gas
    + Region, data=train)
set.seed(2002)
folds = createFolds(train$ClaimNb, 5, list=FALSE)

set.seed(3)
ptn=Sys.time()
cv_ridge_4 = cv.glmnet(train.x_4, y = train$ClaimNb, offset = log(train$
    Exposure),family = "poisson",alpha = 0,nfolds = 5,foldid = folds,maxit
    =10^3,nlambda = 50)
ptn_4 = Sys.time() - ptn
ptn_4

optimal_lambda_4=cv_ridge_4$lambda.min; optimal_lambda_4
opt.lam_4 = c(cv_ridge_4$lambda.min, cv_ridge_4$lambda.1se)
coef(cv_ridge_4, s = opt.lam_4)

ridge_reg_4 = glmnet(train.x_4, y =train$ClaimNb, offset = log(train$
    Exposure), family = "poisson", alpha=0,lambda = optimal_lambda_4)
ridge_reg_4$dev.ratio
ridge_reg_4$nulldev

ridge_reg_4.5 = glmnet(train.x_4, y =train$ClaimNb, offset = log(train$
    Exposure), family = "poisson", alpha=0)

plot(cv_ridge_4)
plot(ridge_reg_4.5)

#####################################
#RIDGE INTERACCION POWER Y BRAND#
#####################################

train.x_5 = model.matrix(ClaimNb ~ Power + CarAge + DriverAge + Brand + Gas
    + Region + Power*Brand, data=train)
set.seed(2002)
folds = createFolds(train$ClaimNb, 5, list=FALSE)

set.seed(3)
ptn=Sys.time()
cv_ridge_5 = cv.glmnet(train.x_5, y = train$ClaimNb, offset = log(train$
    Exposure),family = "poisson",alpha = 0,nfolds = 5,foldid = folds,maxit
    =10^3,nlambda = 50)
ptn_5 = Sys.time() - ptn
```

```
ptn_5

optimal_lambda_5=cv_ridge_5$lambda.min; optimal_lambda_5
opt.lam_5 = c(cv_ridge_5$lambda.min, cv_ridge_5$lambda.1se)
coef(cv_ridge_5, s = opt.lam_5)

ridge_reg_5 = glmnet(train.x_5, y =train$ClaimNb, offset = log(train$
    Exposure), family = "poisson", alpha=0,lambda = optimal_lambda_5)
ridge_reg_5$dev.ratio
ridge_reg_5$nulldev

ridge_reg_5.5 = glmnet(train.x_5, y =train$ClaimNb, offset = log(train$
    Exposure), family = "poisson", alpha=0)

plot(cv_ridge_5)
plot(ridge_reg_5.5)

###################################
#RIDGE INTERACCION VARIABLES CAT#
###################################

train.x_6 = model.matrix(ClaimNb ~ CarAge + DriverAge + Power*Region +
    Power*Brand + Power*Gas +  Region* Brand + Region* Gas + Brand*Gas ,
    data=train)
set.seed(2002)
folds = createFolds(train$ClaimNb, 5, list=FALSE)

set.seed(3)
ptn_6=Sys.time()
cv_ridge_6 = cv.glmnet(train.x_6, y = train$ClaimNb, offset = log(train$
    Exposure),family = "poisson",alpha = 0,nfolds = 5,foldid = folds,maxit
    =10^3,nlambda = 50)
ptn_6 = Sys.time() - ptn
ptn_6

optimal_lambda_6=cv_ridge_6$lambda.min; optimal_lambda_6
opt.lam_6 = c(cv_ridge_6$lambda.min, cv_ridge_6$lambda.1se)
coef(cv_ridge_6, s = opt.lam_6)

ridge_reg_6 = glmnet(train.x_6, y =train$ClaimNb, offset = log(train$
    Exposure), family = "poisson", alpha=0,lambda = optimal_lambda_6)
ridge_reg_6$dev.ratio
ridge_reg_6$nulldev

ridge_reg_6.5 = glmnet(train.x_6, y =train$ClaimNb, offset = log(train$
    Exposure), family = "poisson", alpha=0)

plot(cv_ridge_6)
plot(ridge_reg_6.5)



#####################################################################
#                         ELASTIC NET POISSON                       #
#####################################################################
```

```
###################################
#      E.N. SIN INTERACCION       #
###################################

train.x_7 = model.matrix(ClaimNb ~ Power + CarAge + DriverAge + Brand + Gas
    + Region, data=train)
set.seed(2002)
folds = createFolds(train$ClaimNb, 5, list=FALSE)

set.seed(3)
ptn=Sys.time()
cv_elastic_7 = cv.glmnet(train.x_7, y = train$ClaimNb, offset = log(train$
    Exposure),family = "poisson",alpha = 0.5,nfolds = 5,foldid = folds,maxit
    =10^3,nlambda = 50)
ptn_7 = Sys.time() - ptn
ptn_7

optimal_lambda_7=cv_elastic_7$lambda.min; optimal_lambda_7
opt.lam_7 = c(cv_elastic_7$lambda.min, cv_elastic_7$lambda.1se)
coef(cv_elastic_7, s = opt.lam_7)

elastic_reg_7 = glmnet(train.x_7, y =train$ClaimNb, offset = log(train$
    Exposure), family = "poisson", alpha=0.5,lambda = optimal_lambda_7)
elastic_reg_7$dev.ratio
elastic_reg_7$nulldev

elastic_reg_7.5 = glmnet(train.x_7, y =train$ClaimNb, offset = log(train$
    Exposure), family = "poisson", alpha=0.5)

plot(cv_elastic_7)
plot(elastic_reg_7.5)

###################################
#E.N. INTERACCION POWER Y BRAND #
###################################

train.x_8 = model.matrix(ClaimNb ~ Power + CarAge + DriverAge + Brand + Gas
    + Region + Power*Brand, data=train)
set.seed(2002)
folds = createFolds(train$ClaimNb, 5, list=FALSE)

set.seed(3)
ptn=Sys.time()
cv_elastic_8 = cv.glmnet(train.x_8, y = train$ClaimNb, offset = log(train$
    Exposure),family = "poisson",alpha = 0.5,nfolds = 5,foldid = folds,maxit
    =10^3,nlambda = 50)
ptn_8 = Sys.time() - ptn
ptn_8

optimal_lambda_8=cv_elastic_8$lambda.min; optimal_lambda_8
opt.lam_8 = c(cv_elastic_8$lambda.min, cv_elastic_8$lambda.1se)
coef(cv_elastic_8, s = opt.lam_8)
```

```
elastic_reg_8 = glmnet(train.x_8, y =train$ClaimNb, offset = log(train$
    Exposure), family = "poisson", alpha=0.5,lambda = optimal_lambda_8)
elastic_reg_8$dev.ratio
elastic_reg_8$nulldev

elastic_reg_8.5 = glmnet(train.x_8, y =train$ClaimNb, offset = log(train$
    Exposure), family = "poisson", alpha=0.5)

plot(cv_elastic_8)
plot(elastic_reg_8.5)

#####################################
#E.N. INTERACCION VARIABLES CAT #
#####################################

train.x_9 = model.matrix(ClaimNb ~ CarAge + DriverAge + Power*Region +
    Power*Brand + Power*Gas +  Region* Brand + Region* Gas + Brand*Gas ,
    data=train)
set.seed(2002)
folds = createFolds(train$ClaimNb, 5, list=FALSE)

set.seed(3)
ptn_9=Sys.time()
cv_elastic_9 = cv.glmnet(train.x_9, y = train$ClaimNb, offset = log(train$
    Exposure),family = "poisson",alpha = 0.5,nfolds = 5,foldid = folds,maxit
    =10^3,nlambda = 50)
ptn_9 = Sys.time() - ptn
ptn_9

optimal_lambda_9=cv_elastic_9$lambda.min; optimal_lambda_9
opt.lam_9 = c(cv_elastic_9$lambda.min, cv_elastic_9$lambda.1se)
coef(cv_elastic_9, s = opt.lam_9)

elastic_reg_9 = glmnet(train.x_9, y =train$ClaimNb, offset = log(train$
    Exposure), family = "poisson", alpha=0.5,lambda = optimal_lambda_9)
elastic_reg_9$dev.ratio
elastic_reg_9$nulldev

elastic_reg_9.5 = glmnet(train.x_9, y =train$ClaimNb, offset = log(train$
    Exposure), family = "poisson", alpha=0.5)

plot(cv_elastic_9)
plot(elastic_reg_9.5)


#############################################################
#                          DEVIANZA                         #
#############################################################

devianza=rep(0,9)
devianza[1]=deviance(lasso_reg_1);devianza[2]=deviance(lasso_reg_2);
    devianza[3]=deviance(lasso_reg_3)
devianza[4]=deviance(ridge_reg_4);devianza[5]=deviance(ridge_reg_5);
    devianza[6]=deviance(ridge_reg_6)
devianza[7]=deviance(elastic_reg_7);devianza[8]=deviance(elastic_reg_8);
```

```
    devianza[9]=deviance(elastic_reg_9)

devianza

#########################################################
#      AUXILIARES PARA CALCULAR AIC, AICc, BIC      #
#########################################################

#se extraen datos para calcular con Excel

cv_aicc <- function(fit, lambda = 'lambda.1se'){
    whlm <- which(fit$lambda == fit[[lambda]])
    with(fit$glmnet.fit,
         {
            tLL <- nulldev - nulldev * (1 - dev.ratio)[whlm]
            k <- df[whlm]
            n <- nobs
            return(list('tLL'=tLL,
                        'nulldev'=nulldev,
                        'k'=k,
                        'n'=n))
         })
}


AICc.min_1_aux <- cv_aicc(cv_lasso_1, 'lambda.min')
AICc.min_2_aux <- cv_aicc(cv_lasso_2, 'lambda.min')
AICc.min_3_aux <- cv_aicc(cv_lasso_3, 'lambda.min')

AICc.min_4_aux <- cv_aicc(cv_ridge_4, 'lambda.min')
AICc.min_5_aux <- cv_aicc(cv_ridge_5, 'lambda.min')
AICc.min_6_aux <- cv_aicc(cv_ridge_6, 'lambda.min')

AICc.min_7_aux <- cv_aicc(cv_elastic_7, 'lambda.min')
AICc.min_8_aux <- cv_aicc(cv_elastic_8, 'lambda.min')
AICc.min_9_aux <- cv_aicc(cv_elastic_9, 'lambda.min')

AIC_min_aux=cbind(AICc.min_1_aux,AICc.min_2_aux,AICc.min_3_aux,AICc.min_4_
    aux,
                  AICc.min_5_aux,AICc.min_6_aux,AICc.min_7_aux,AICc.min_8_
                      aux,
                  AICc.min_9_aux)
AIC_min_aux


#########################################################
#                    PLOTS Y OTROS                      #
#########################################################

lambdas = matrix(0,nrow = 9, ncol = 2)
lambdas[1,]=opt.lam_1;lambdas[2,]=opt.lam_2;lambdas[3,]=opt.lam_3
lambdas[4,]=opt.lam_4;lambdas[5,]=opt.lam_5;lambdas[6,]=opt.lam_6
lambdas[7,]=opt.lam_7;lambdas[8,]=opt.lam_8;lambdas[9,]=opt.lam_9

lambdas
```

```
plot(cv_lasso_1,main="Lasso-modelo 1",line=3)

dev.off()
win.graph(width=10,height=10)
par(mfrow=c(4,2))

plot(cv_lasso_2,main="Lasso-modelo 2",line=3)
plot(cv_lasso_3,main="Lasso-modelo 3",line=3)
plot(cv_ridge_4,main="Ridge-modelo 1",line=3)
plot(cv_ridge_5,main="Ridge-modelo 2",line=3)
plot(cv_ridge_6,main="Ridge-modelo 3",line=3)
plot(cv_elastic_7,main="Elastic Net-modelo 1",line=3)
plot(cv_elastic_8,main="Elastic Net-modelo 2",line=3)
plot(cv_elastic_9,main="Elastic Net-modelo 3",line=3)

dev.off()

par(mfrow=c(1,2))
plot_glmnet(lasso_reg_1.5, xvar = "lambda",col=1:300)
plot_glmnet(lasso_reg_1.5, xvar = "norm",col=1:300)
title(main= list("Lasso modelo 1",cex=2.5), line = -2, outer = TRUE)

par(mfrow=c(1,2))
plot_glmnet(lasso_reg_2.5, xvar = "rlambda",col=1:300)
plot_glmnet(lasso_reg_2.5, xvar = "norm",col=1:300)
title(main= list("Lasso modelo 2",cex=2.5), line = -2, outer = TRUE)

par(mfrow=c(1,2))
plot_glmnet(lasso_reg_3.5, xvar = "rlambda",col=1:300)
plot_glmnet(lasso_reg_3.5, xvar = "norm",col=1:300)
title(main= list("Lasso modelo 3",cex=2.5), line = -2, outer = TRUE)

par(mfrow=c(1,2))
plot_glmnet(ridge_reg_4.5, xvar = "rlambda",col=1:300)
plot_glmnet(ridge_reg_4.5, xvar = "norm",col=1:300)
title(main= list("Ridge modelo 1",cex=2.5), line = -2, outer = TRUE)

par(mfrow=c(1,2))
plot_glmnet(ridge_reg_5.5, xvar = "rlambda",col=1:300)
plot_glmnet(ridge_reg_5.5, xvar = "norm",col=1:300)
title(main= list("Ridge modelo 2",cex=2.5), line = -2, outer = TRUE)

par(mfrow=c(1,2))
plot_glmnet(ridge_reg_6.5, xvar = "rlambda",col=1:300)
plot_glmnet(ridge_reg_6.5, xvar = "norm",col=1:300)
title(main= list("Ridge modelo 3",cex=2.5), line = -2, outer = TRUE)

par(mfrow=c(1,2))
plot_glmnet(elastic_reg_7.5, xvar = "rlambda",col=1:300)
plot_glmnet(elastic_reg_7.5, xvar = "norm",col=1:300)
title(main= list("Elastic Net modelo 1",cex=2.5), line = -2, outer = TRUE)

par(mfrow=c(1,2))
plot_glmnet(elastic_reg_8.5, xvar = "rlambda",col=1:300)
plot_glmnet(elastic_reg_8.5, xvar = "norm",col=1:300)
```

```
title(main= list("Elastic Net modelo 2",cex=2.5), line = -2, outer = TRUE)

par(mfrow=c(1,2))
plot_glmnet(elastic_reg_9.5, xvar = "rlambda",col=1:300)
plot_glmnet(elastic_reg_9.5, xvar = "norm",col=1:300)
title(main= list("Elastic Net modelo 3",cex=2.5), line = -2, outer = TRUE)

dev.off()

plot(cv_lasso_1$glmnet.fit, "norm",    label=TRUE, main="Lasso modelo 1",
    line=3)
plot(cv_lasso_1$glmnet.fit, "lambda", label=TRUE, main="Lasso modelo 1",
    line=3)
plot(cv_lasso_2$glmnet.fit, "norm",    label=TRUE, main="Lasso modelo 2",
    line=3)
plot(cv_lasso_2$glmnet.fit, "lambda", label=TRUE, main="Lasso modelo 2",
    line=3)
plot(cv_lasso_3$glmnet.fit, "norm",    label=TRUE, main="Lasso modelo 3",
    line=3)
plot(cv_lasso_3$glmnet.fit, "lambda", label=TRUE, main="Lasso modelo 3",
    line=3)
plot(cv_ridge_4$glmnet.fit, "norm",    label=TRUE, main="Ridge modelo 1",
    line=3)
plot(cv_ridge_4$glmnet.fit, "lambda", label=TRUE, main="Ridge modelo 1",
    line=3)
plot(cv_ridge_5$glmnet.fit, "norm",    label=TRUE, main="Ridge modelo 2",
    line=3)
plot(cv_ridge_5$glmnet.fit, "lambda", label=TRUE, main="Ridge modelo 2",
    line=3)
plot(cv_ridge_6$glmnet.fit, "norm",    label=TRUE, main="Ridge modelo 3",
    line=3)
plot(cv_ridge_6$glmnet.fit, "lambda", label=TRUE, main="Ridge modelo 3",
    line=3)
plot(cv_elastic_7$glmnet.fit, "norm",    label=TRUE, main="Elastic net
    modelo 1",line=3)
plot(cv_elastic_7$glmnet.fit, "lambda", label=TRUE, main="Elastic net
    modelo 1",line=3)
plot(cv_elastic_8$glmnet.fit, "norm",    label=TRUE, main="Elastic net
    modelo 2",line=3)
plot(cv_elastic_8$glmnet.fit, "lambda", label=TRUE, main="Elastic netmodelo
    2",line=3)
plot(cv_elastic_9$glmnet.fit, "norm",    label=TRUE, main="Elastic netmodelo
    3",line=3)
plot(cv_elastic_9$glmnet.fit, "lambda", label=TRUE, main="Elastic netmodelo
    3",line=3)



#####################################################################
#                              ZIP POISSON                          #
#####################################################################

form_zip_1 <- formula(ClaimNb ~ Power + CarAge + DriverAge + Brand + Gas +
    Region + offset(log(Exposure))|1)
```

```
ptn = Sys.time()
ZIP_model_1 = zeroinfl(form_zip_1,data = train,dist = "poisson",link="logit
    ")
ptn_ZIP = Sys.time() - ptn
ptn_ZIP

summary(ZIP_model_1)
AICc(ZIP_model_1,second.ord = FALSE)
AICc(ZIP_model_1, second.ord = TRUE)
useBIC(ZIP_model_1)


form_zip_2 <- formula(ClaimNb ~ Power + CarAge + DriverAge + Brand + Gas +
    Region + offset(log(Exposure))|Power + CarAge + DriverAge + Brand + Gas
    + Region + offset(log(Exposure)))

ptn = Sys.time()
ZIP_model_2 = zeroinfl(form_zip_2,data = train,dist = "poisson",link="logit
    ")
ptn_ZIP_2 = Sys.time() - ptn
ptn_ZIP_2

summary(ZIP_model_2)
AICc(ZIP_model_2,second.ord = FALSE)
AICc(ZIP_model_2, second.ord = TRUE)
useBIC(ZIP_model_2)

#####################################################################
#                          LASSO ZIP                                #
#####################################################################

set.seed(3)
folds = createFolds(train$ClaimNb, 5, list=FALSE)
set.seed(3)

ptn = Sys.time()
cv_lasso_1_zi = cv.zipath(ClaimNb ~ Power + CarAge + DriverAge + Brand +
    Gas + Region +offset(log(train$Exposure))|Power +
                              CarAge + DriverAge + Brand + Gas + Region +
                                  offset(log(train$Exposure)),data=train,
                          family = "poisson",nlambda=10)
ptn_ZIP_LASSO.1 = Sys.time() - ptn
ptn_ZIP_LASSO.1

plot(cv_lasso_1_zi)
cv_lasso_1_zi$residmat #matrix for cross-validated log-likelihood at each (
    count.lambda,zero.lambda)sequence
cv_lasso_1_zi$cv       #The mean cross-validated log-likelihood - a vector
    of length length(count.lambda)
cv_lasso_1_zi$cv.error #estimate of standard error of cv
cv_lasso_1_zi$lambda.which #index of (count.lambda,zero.lambda) that gives
    maximum cv.
cv_lasso_1_zi$lambda.optim #value of (count.lambda,zero.lambda) that gives
    maximum cv.
```

```
set.seed(3)
ptn=Sys.time()
lassoreg_1_zi = zipath(ClaimNb ~ Power + CarAge + DriverAge + Brand + Gas +
    Region +offset(log(train$Exposure))|Power +
                          CarAge + DriverAge + Brand + Gas + Region +offset
                              (log(train$Exposure)),data=train,
                       family = "poisson",nlambda=100)
ptn_ZIP_LASSO.2 = Sys.time() - ptn
ptn_ZIP_LASSO.2

options(max.print=1000000)

summary(lassoreg_1_zi)

minBic_1 <- which.min(AIC(lassoreg_1_zi))
coef(lassoreg_1_zi, minBic_1)

AIC_ZIP_LASSO=AIC(lassoreg_1_zi)[minBic_1]; AIC_ZIP_LASSO
BIC_ZIP_LASSO=BIC(lassoreg_1_zi)[minBic_1]; BIC_ZIP_LASSO
logLik(lassoreg_1_zi)[minBic_1]
plot(lassoreg_1_zi,xvar = c("norm"),label="TRUE")
plot(lassoreg_1_zi,xvar = c("lambda"),label="TRUE")
lassoreg_1_zi$lambda.count
lassoreg_1_zi$coefficients
p=AIC_ZIP_LASSO/2 +logLik(lassoreg_1_zi)[minBic_1];p
p=32
n=lassoreg_1_zi$n;n
AIC_c_ZIP_LASSO = AIC_ZIP_LASSO+(2*p*(p+1))/(n-p-1);AIC_c_ZIP_LASSO
attributes(lassoreg_1_zi)
plot(lassoreg_1_zi$aic,type="l", col=90,xlab="Indice de lambda", ylab="AIC"
    , lwd=3,main = "Estimacion de AIC", sub="ZIP-LASSO")

x= c(1:100)
data1 = data.frame(x,lassoreg_1_zi$lambda.count); data2 = data.frame(x,
    lassoreg_1_zi$lambda.zero)
cols = c("indice", "lambda")
colnames(data1) = cols
colnames(data2) = cols
aux = ggplot() +
   geom_line(data = data1, aes(x = indice, y = lambda ), color = "blue") +
   geom_line(data = data2, aes(x = indice, y=lambda), color = "red") +
   xlab('indice') +
   ylab('lambda')

print(aux)



#####################################################################
#                  ELASTIC NET, ALFA DE 0 A 1                       #
#####################################################################

ENfun = function(alfa){
   train.x_1 = model.matrix(ClaimNb ~ Power + CarAge + DriverAge + Brand +
       Gas + Region, data=train)
```

```
    set.seed(2002)
    folds = createFolds(train$ClaimNb, 5, list=FALSE)
    set.seed(3)
    ptn=Sys.time()
    cv_ = cv.glmnet(train.x_1, y = train$ClaimNb, offset = log(train$
        Exposure),family = "poisson",alpha = alfa,
                    nfolds = 5,foldid = folds,maxit=10^3,nlambda = 50)
    ptn_1 = Sys.time() - ptn
    ptn_1

    optimal_lambda=cv_$lambda.min; optimal_lambda
    opt.lam = c(cv_$lambda.min, cv_$lambda.1se)
    coeficiente=coef(cv_, s = opt.lam)

    reg = glmnet(train.x_1, y =train$ClaimNb, offset = log(train$Exposure),
        family = "poisson", alpha=alfa,
                 lambda = optimal_lambda)
    reg$dev.ratio
    reg$nulldev

    reg_1.5 = glmnet(train.x_1, y =train$ClaimNb, offset = log(train$
        Exposure), family = "poisson", alpha=alfa)
    plot(cv_)
    plot(reg_1.5)

    devianza.cal=deviance(reg)
    return(list("coeficientes"=coeficiente,"optimal_lambda"=optimal_lambda,"
        devianza"=devianza.cal,"cv_aicc" =
                   cv_aicc(cv_, 'lambda.min')))
}

EN0.0=ENfun(0.0)
EN0.1=ENfun(0.1)
EN0.2=ENfun(0.2)
EN0.3=ENfun(0.3)
EN0.4=ENfun(0.4)
EN0.5=ENfun(0.5)
EN0.6=ENfun(0.6)
EN0.7=ENfun(0.7)
EN0.8=ENfun(0.8)
EN0.9=ENfun(0.9)
EN1.0=ENfun(1.0)


#######################################################
#                    TIEMPO TOTAL                     #
#######################################################

ptn_completo = Sys.time() - ptn_total
ptn_completo

#############################################################################################
#                                          FIN                                              #
#############################################################################################
```