Universidad de Costa Rica
Escuela de Matemática

Modelos Lineales

Proyecto

Profesor:

Luis Barboza

Estudiantes:

Amanda Coto Jara B42137

Esteban Molina Calvo B34311

Índice

1. Introducción

El problema de la modelación de frecuencia y severidad en el área de los seguros de no-vida, ha sido abarcado ampliamente y es fundamental en la práctica actuarial, esto pues es importante porque las aseguradoras se pueden preparar para eventualidades y entonces tener reservas y demás prácticas de la administración de riesgo.

Una forma de abarcar este problema, es con la utilización de modelos lineales y el método de mínimos cuadrados, esto para la estimación de parámetros visto desde un enfoque frecuentista. Pero si se prefiere un enfoque bayesiano, se puede usar simulación para ajustar los parámetros del modelo lineal. En el presente proyecto se describirán los métodos usuales para el ajuste de modelos lineales, desde el enfoque bayesiano aplicado en el área actuarial de seguros.

2. Metodología

Primero se explicarán los modelos lineales generales a utilizar en el proyecto, estos son el Poisson y el Normal, para esto se seguirá la contrucción de Agresti en [1].

El modelo de Poisson es para datos de conteo, por lo tanto, se aplicará a los datos de frecuencia pues tienen esta naturaleza. El modelo asume un componente aleatorio de Poisson y usa el logaritmo como función de enlace. De esta forma se obtiene el modelo: $\eta_i = log(\mu_i) = \sum_{j=1}^{p} \beta_j x_{ij}$. Donde x_{ij} es el valor de la característica j para la observación i y β_j son los parámetros que se deben ajustar al modelo. Entonces, el valor esperado está dado por $\mu_i = exp(\eta_i)$.

Para la severidad se utilizará un modelo lineal ordinario con componente aleatorio normal y la identidad como función de enlace. Por lo tanto, se obtiene el modelo:

$$y_i = \sum_{j=1}^{p} \beta_j x_{ij} + \epsilon_i$$

Donde y_i es la variable objetivo, en este caso la severidad, x_{ij} y β_j son lo mismo que en el modelo anterior y ϵ_i es un error con esperanza 0 y varianza σ^2.

Como se explica en [2], la diferencia entre el ajuste frecuentista de un modelo lineal, o un modelo lineal generalizado, es que el cálculo de los parámetros se realiza con minímos cuadrados en el caso frecuentista; mientras que el ajuste bayesiano requiere un muestreo vía Markov Chain Monte Carlo. Esto da cambia la inferencia que se realiza de los parámetros como se vera más adelante.

Por lo tanto, se necesita entender los algoritmos a utilizar para obtener los parámetros de los modelos lineales escogidos, para la frecuencia y la severidad. Los algoritmos de Markov Chain Monte Carlo o MCMC son métodos de simulación utilizados en modelación Bayesiana. Según lo explicado en [5], la idea de estos métodos es que para obtener muestras de una distribución π, que puede ser la distribución posterior, es suficiente producir una cadena de Markov cuya distribución estacionaria sea π. La teoría de cadenas de Markov que le da fundamentación a este algoritmo es básica y no se ahondará en este proyecto.

Si tal cadena existe, se obtiene que para los pasos x_t de la cadena: $\frac{1}{T}\sum_{t=1}^{T} g(x_t)$ converge a $\mathbb{E}[g(X)]$ sin importar el valor inicial, donde T es la cantidad de pasos de la cadena y g es una función integrable, en este caso π. Debido a que la cadena converge en una cantidad usualmente alta de pasos, se debe tomar un periodo de *burn-in* que serían los pasos de la cadena previos a la covergencia.

Uno de los algoritmos clásicos del muestreo MCMC que se utilizan para la obtención de los parámetros β, es el algoritmo de Metropolis-Hastings. Sea f la función objetivo, es decir, de la que se quiere obtener muestras, y q la densidad condicional, se sigue la descripción del algoritmo de [6]. Entonces dado x_t

1. Generar una muestra de $Y_t \sim q(y \mid x_t)$
2. Calcular $\rho(x_t, Y_t) = min\left(\frac{f(Y_t)q(x_t|Y_t)}{f(x_t)q(Y_t|x_t)}, 1\right)$
3. Tomar
$$x_t = \begin{cases} Y_t \text{ con probabilidad } \rho(x_t, Y_t) \\ x_t \text{ con probabilidad } 1 - \rho(x_t, Y_t) \end{cases}$$

Para realizar el paso 3, se obtiene una muestra de $U \sim Uniforme(0, 1)$ y se utiliza como la probabilidad obtenida. Esto se realiza para $t = 1, ..., T$.

Como se menciona en [3], el algoritmo de Metropolis-Hastings puede ser ineficiente. Esto sucede porque la cadena en ocasiones tarda mucho en converger, pues existe un movimiento en zigzag repetitivo a través de la cadena. Aunque existan formas de corregir esto usando el mismo algoritmo, no hay mejoría si la distribución objetivo tiene muchas dimensiones.

Este es el caso en el presente problema, pues el parámetro $\beta = (\beta_1, ..., \beta_p)$ que se quiere calcular tiene bastantes entradas, como se verá en la sección de resultados. Por lo tanto, el mismo [3] propone la utilización del algoritmo Hamiltonian Monte Carlo como solución a la ineficiencia mencionada. La idea es añadirle una variable de momentum al modelo Metropolis-Hastings, que se va actualizando conforme avanza la cadena, esto con la idea de permitirle a la cadena moverse más rapidamente por la distribución. Este será el algoritmo que se utilizara en el presente proyecto.

Para la compresión de los resultados del modelo hay que entender bien los intervalos de credibilidad, pues para cada parámetro hay uno de estos intervalos. Estos se definen según [7] , para una distribución previa π, un intervalo de credibilidad C_x es α-creible si $1-\alpha \leq P(\theta \in C_x|x)$. Estos tinene la ventaja que a diferencia de los intervalos de confianza clásicos, que implican un paso aleatorio para alcanzar niveles de confianza nominal, utilizan la previa para evitar ese paso aleatorio. Así quitan ese ruido y aprovechan la información que ya contienen los datos, por lo que da más precisión al intervalo.

Para el diagnóstico de modelos que utilizan un ajuste vía MCMC hay dos factores fundamentales, la convergencia de las cadenas y la autocorrelación que presentan. Para verificar estos dos elementos existen herramientas que validan el análisis. Estas son el $\hat{R}$ que verifica convergencia y el número efectivo de la muestra n_{eff} que verifica la autocorrelación del modelo. Para la definición de los mismos se sigue la construcción de [3]. Sea m el número de cadenas y n el largo de cada una de estas, entonces primero hay que definir varios componentes:

- Sea β cada escalar que vamos a estimar. Entonces $\beta_{ij}, (i = 1, ..., n; j = 1, ..., m)$ es la simulación en el paso i, en la cadena j.
- Se define $\overline{\beta_{.j}} = \frac{1}{n} \sum_{i=1}^{n} \beta_{ij}$
- Entonces $\overline{\beta_{..}} = \frac{1}{m} \sum_{j=1}^{m} \overline{\beta_{.j}}$
- Además $s_j^2 = \frac{1}{n-1} \sum_{i=1}^{n} (\beta_{ij} - \overline{\beta_{.j}})^2$
- Con esto se define: $B = \frac{n}{m-1} \sum_{j=1}^{m} (\overline{\beta_{.j}} - \overline{\beta_{..}})^2$ y $W = \frac{1}{m} \sum_{j=1}^{n} s^2$

- Con esto se puede estimar la varianza marginal de la posterior como $\hat{v}ar(\beta|y) = \frac{n-1}{n}W + \frac{1}{n}B$

Con esta definición de la varianza se puede demostrar que es insesgada cuando hay estacionalidad en la cadena. Lo que sucede es, que esta definición sobreestima la varianza porque asume que la distribución inicial tiene la dispersión apropiada. Por otro lado W subestima la varianza marginal de la posterior, esto pues cada s_j^2 no ha tenido el tiempo necesario para converger a la distribución objetivo, pero se espera que ambas converjan a $var(\beta|y)$ cuando $n \to \infty$. Por esta razón, para monitorear la varianza se define $\hat{R} = \sqrt{\frac{\hat{v}ar(\beta|y)}{W}}$, que asintóticamente converge a uno. Por lo tanto, entre más cerca está $\hat{R}$ de uno, para cada parámetro, es una señal de convergencia del modelo.

Para el tamaño de muestra efectivo, tenemos que en presencia de autocorrelación se define como $n_{eff} = N(1 + 2\sum_{t=1}^{\infty}\rho_t)^{-1}$ donde ρ_t es la autocorrelación de la secuencia de β con lag t, y $N = mn$ es el tamaño de la muestra. Note que es necesario para el cálculo del mismo una serie infinita, pero esto no es un problema, pues desde el inicio se necesita simular cadenas lo suficientemente largas para aproximar la convergencia asintótica a la distribución objetivo. Una herramienta útil es la proporción $\frac{n_{eff}}{N}$ pues simplemente hace referencia a $(1 + 2\sum_{t=1}^{\infty}\rho_t)^{-1}$, entonces entre más pequeño sea esa proporción, más autocorrelación habrá en la cadena y por lo tanto habrá una mezcla deficiente.

3. Resultados

Tanto en la implementación del modelo poisson, como para el modelo normal, se utilizaron los datos recolectados por el Swedish Committee on the Analysis of Risk Premium que consisten en 2182 observaciones de portafolios de seguros vehiculares en Suecia, la misma fue obtenida de [4]. Contiene 7 variables, Kilometros, Zona, Tipo, Bono, Meses, Reclamos y Severidad de distintos portafolios de seguros. Las variables dependientes a estudiar son Severidad y Reclamos.

3.1. Severidad

Iniciando con el modelo para la severidad, se realizó primero una transformación en los datos. Esta fue log(x+1), con el fin de estabilizar varianza

de los datos y poder utilizar un modelo normal. Se procedió a realizar una selección de variables mediante el método Forward Stepwise Selection. Según estos resultados se determino que lo niveles de la variables zona, kilometros y la variable meses son las que conforma el mejor modelo para la severidad.

Para la realización del muestreo vía MCMC se realizaron 5 cadenas de markov con 5000 mil pasos cada una. El periodo de burn in fue de 2500 y se dejaron los restantes 2500 para el Monte Carlo. La previa de cada beta del modelo es una normal con media 0 y varianza de 2.5. Se realizó un análisis de la convergencia de los parámetros.

Se realiza un gráfico de la proporción n_{eff} del modelo:

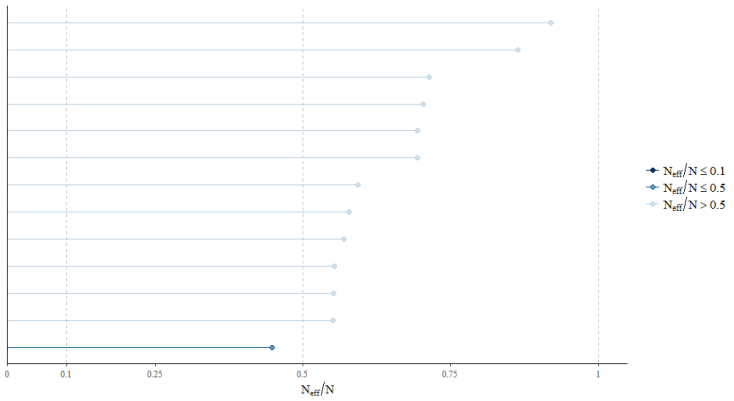


Figura 1: n_{eff} ratio de los parámetros. Fuente: Elaboración propia.

Podemos observar en el gráfico que el valor más pequeño está por encima de 0.25, y los demás están por encima de 0.5, por lo que podemos concluir que la autocorrelación de cada cadena no es preocupante. Esto da indicios de que hay una buena mezcla de las cadenas.

Además podemos ver los valores $\hat{R}$ del modelo:

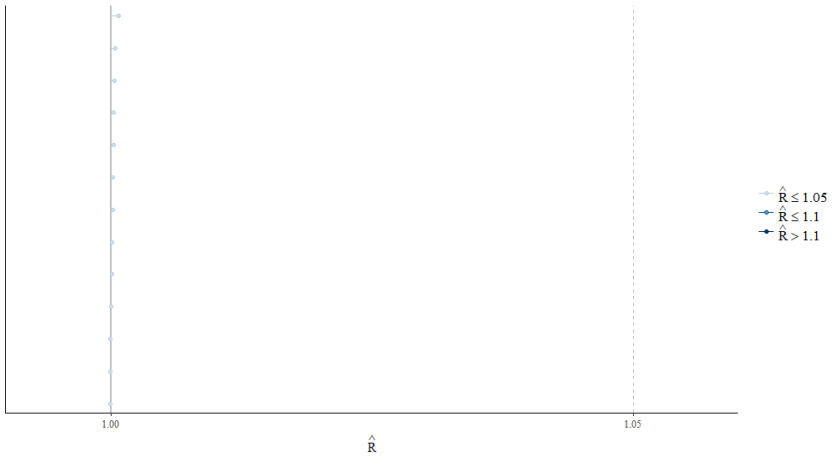
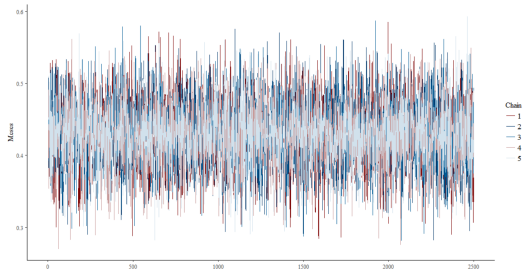


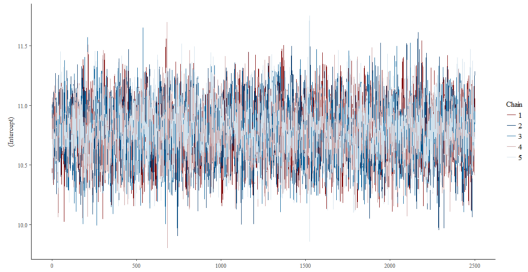
Figura 2: $\hat{R}$ de los parámetros. Fuente: Elaboración propia.

Este nos muestra cómo los valores para todos los parámetros están cerca de uno. Esto nos indica que las cadenas están en equilibrio y todas convergen en un carácter asintótico.

Ahora se observa los trace plots de los parámetros:



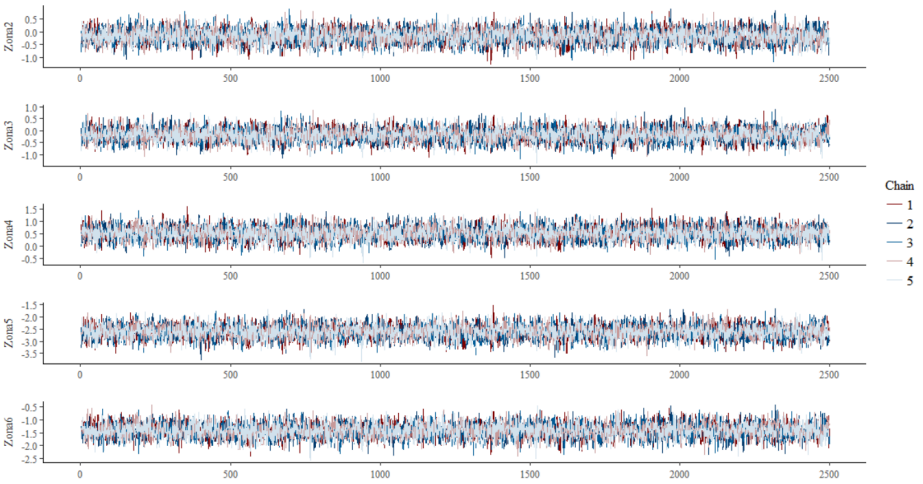
(a) Meses



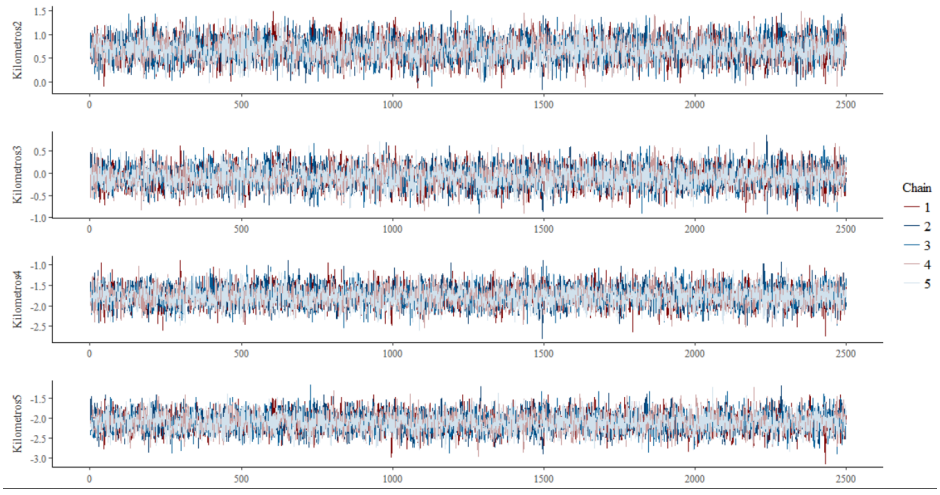
(b) Intercepto

Figura 3: Trace plots de los parámetros de la variable meses y el intercepto. Fuente: Elaboración propia.

Se puede notar en las figuras 3 y 4 que las cadenas están explorando la misma región para cada parámetro sin sobresaltos importantes. Además hay una buena mezcla de los parámetros, lo cual es buena señal de la convergencia de los mismos, y confirma lo que se está viendo en los gráficos de la figura 1 y 2. Por lo tanto, se determina que hay un buen ajuste por parte del modelo, o lo que es lo mismo, se tiene certeza de que los valores de los parámetros son correctos.



(a) Niveles de Zona



(b) Niveles de Kilometros

Figura 4: Trace plots de los parámetros de cada nivel de las variables Zona y Kilometros. Fuente: Elaboración propia.

Ahora que existe certeza de la convergencia del método, se analizan los intervalos de credibilidad de los parámetros:

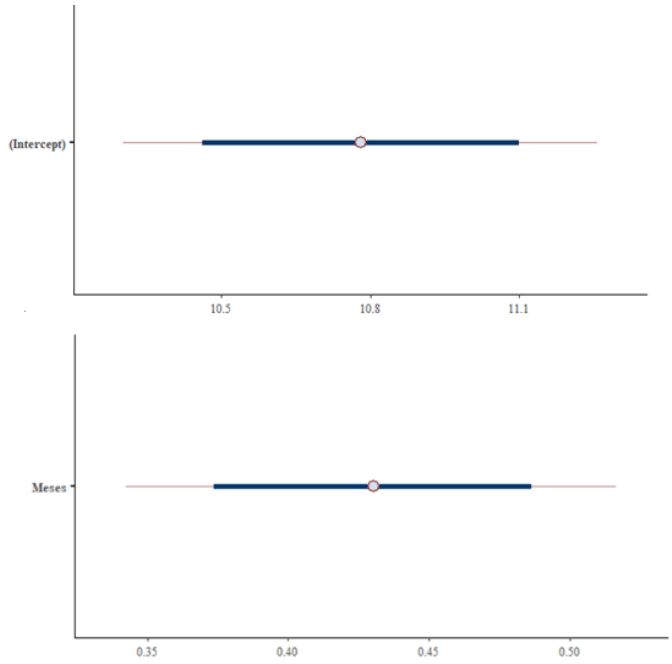


Figura 5: Intervalos de credibilidad de los parámetros de la variable meses y el intercepto. Fuente: Elaboración propia.

En la figura 5 se puede observar de la muestra, que hay un 95 % de probabilidades de que el valor del intercepto esté entre 10.3 y 11.3. Esto quiere decir que la varianza de la muestra es pequeña, por lo que confirmamos que la transformación utilizada fue útil. Por otro lado, para el parámetro de meses la varianza es aún menor, pues el valor tiene 95 % de probabilidades de estar entre 0.3 y 0.5.

Ahora en la figura 6 se observan los intervalos de credibilidad para los niveles de la variable zona y los niveles de la variable kilómetros. De igual forma la varianza de los parámetros es bastante pequeña y tenemos una alta precisión del 95 % para los valores. La mayoría son cercanos a 0, pero se puede notar que los niveles 5 y 6 de la variable zona, además de los niveles 4 y 5 de la variable kilómetros, son los que más peso tienen al cambiar del nivel base de sus correspondientes variables. Se nota además que el cambio es negativo por lo que si se forma parte de estos niveles la severidad disminuye.

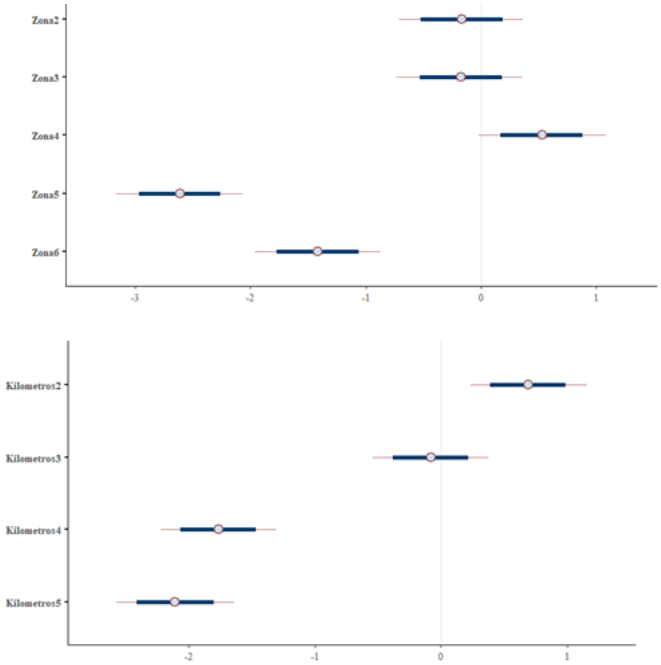


Figura 6: Intervalos de credibilidad de los parámetros de cada nivel de las variables Zona y Kilometros. Fuente: Elaboración propia.

3.2. Frecuencia

Para el modelo de frecuencia, se determinó utilizar un modelo de Poisson. Se realizó una selección de variables mediante el método Forward Stepwise Selection. Según estos resultados se decidió que lo niveles de la variable bono y variable meses son las que conforman el mejor modelo para la frecuencia de los reclamos.

Para la realización del muestreo vía MCMC se realizaron 5 cadenas de markov con 5000 mil pasos cada una. El periodo de burn in fue de 2500 y se dejaron los restantes 2500 para el Monte Carlo. La previa de cada beta del modelo es una normal con media 0 y varianza de 2.5. Se realizó un análisis de la convergencia de los parámetros.

Se realiza un gráfico de la proporción n_{eff} del modelo:

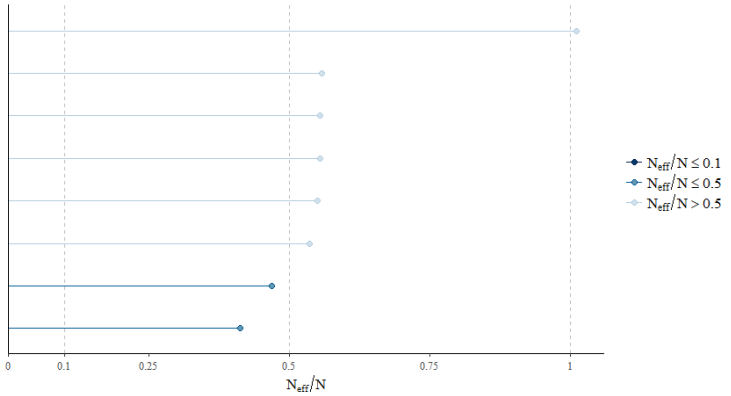


Figura 7: n_{eff} ratio de los parámetros. Fuente: Elaboración propia.

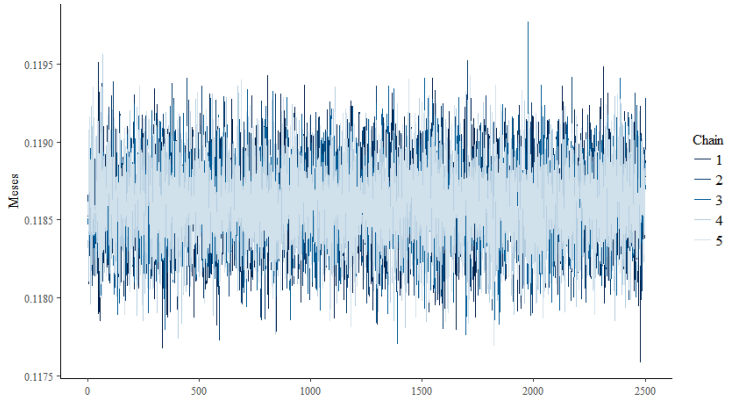
Podemos observar en la figura 7 que los valores más pequeños están por encima de 0.25, y los demás están por encima de 0.5, por lo que podemos concluir que la autocorrelación de la cadena no es alta. Por lo que parece haber una buena mezcla de las cadenas

Además podemos ver los valores $\hat{R}$ del modelo:

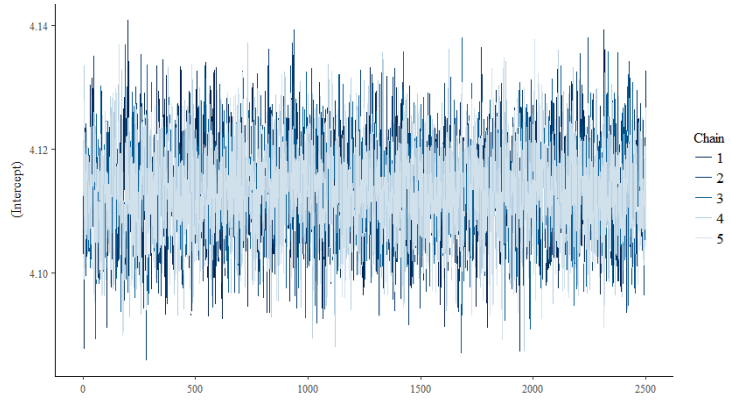
Figura 8: $\hat{R}$ de los parámetros. Fuente: Elaboración propia.

Este nos muestra como los valores para todos los parámetros están muy cerca de uno. Entonces que las cadenas están en equilibrio y todas convergen en un carácter asintótico, al igual que el caso del modelo de severidad.

Ahora se observa los trace plots de los parámetros:



(a) Meses



(b) Intercepto

Figura 9: Trace plots de los parámetros de la variable meses y el intercepto. Fuente: Elaboración propia.

Se puede notar en las figuras 9 y 10 que las cadenas están explorando la misma región para cada parámetro. Además hay una buena mezcla en las cadenas lo cual es señal de la convergencia de las mismos, y confirma lo que se esta viendo en los gráficos de la figura 7 y 8. Por lo tanto, se determina que también hay buen ajuste por parte del modelo, o lo que es lo mismo, se tiene certeza de que los valores de los parámetros son correctos.

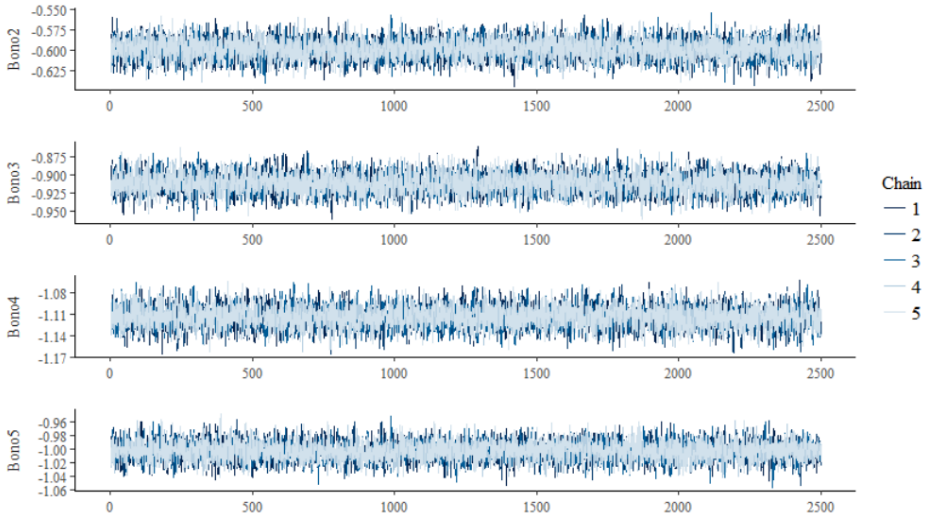


Figura 10: Trace plots de los parámetros de cada nivel de las variable Bono. Fuente: Elaboración propia.

Ahora que existe certeza de la convergencia del método, se analizan los intervalos de credibilidad de los parámetros:

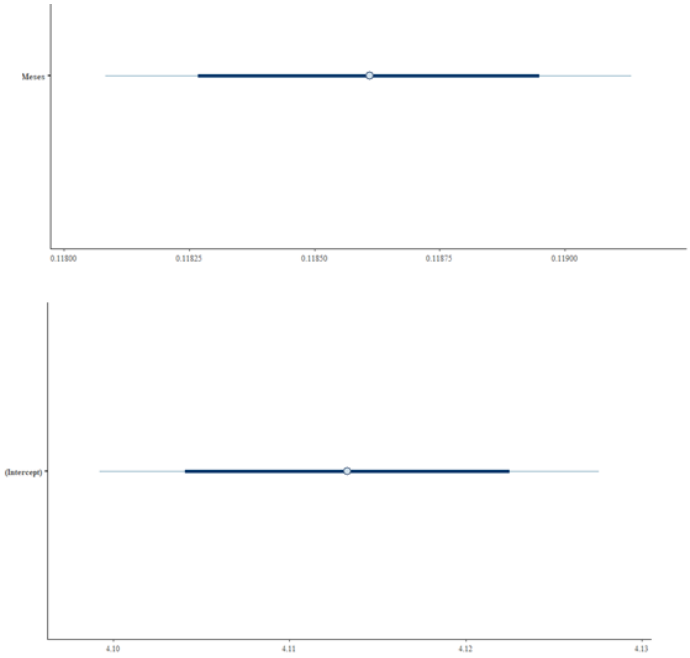


Figura 11: Intervalos de credibilidad de los parámetros de la variable meses y el intercepto. Fuente: Elaboración propia.

En la figura 11 se puede observar desde la muestra, que hay un 95 % de probabilidades de que el valor del intercepto esté entre 4.10 y 4.13. Esto quiere decir que la varianza de la muestra es mínima, por lo que se tiene bastante precisión en la estimación del parámetro. Por otro lado, para el parámetro de meses la varianza es aún menor pues sus valores pues hay 95 % de probabilidades de que el valor esté entre 0.118 y 0.119.

Por otro lado en la figura 12 se observan los intervalos de credibilidad para los niveles de la variable bono. De igual forma varianza de los parámetros es bastante pequeña y tenemos una alta precisión del 95 % para los valores. Todos son cercanos a 0, pero se puede notar que los niveles 4 y 5 son las que más peso tienen al cambiar del nivel base. Se nota además que el cambio es negativo por lo que si se forma parte de estos niveles la frecuencia disminuye. Lo cual es consistente con la información, pues cada aumento en el nivel bono significa que la persona lleva más tiempo sin chocar.

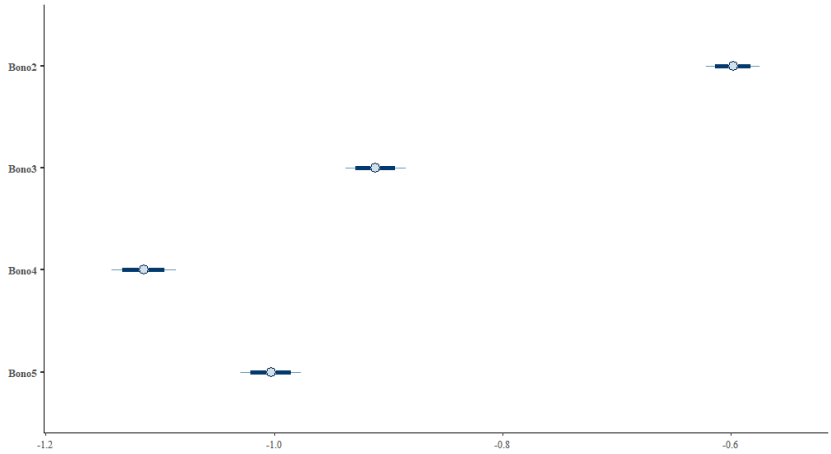


Figura 12: Intervalos de credibilidad de los parámetros de cada nivel de las variables Zona y Kilometros. Fuente: Elaboración propia.

4. Conclusiones

- Como muestran los resultados, con respecto a la severidad, al realizar la transformación se pierde interpretabilidad de las pérdidas, pero se gana bastante en el ajuste del modelo. Además fue conveniente para el estudio del impacto de cada covariable. Con el análisis que se realizó se puede concluir que los niveles más altos de la variable kilómetros y de la variable zona indica que la severidad en el portafolio es más baja. Por una parte, los niveles más altos de la variable kilómetros indica que

los individuos que están en el portafolio han recorrido más distancia en sus vehículos, por lo que se concluye que más allá de estar expuestos a siniestros durante más tiempo, entre más distancias se recorra la severidad de los choques es menor. Por otra parte, los niveles más altos de la variable zona hacen referencia de zonas rurales de Suecia y la isla Gotland, también de carácter rural. Por lo que nos permite pensar que los portafolios de zonas urbanas, es donde suelen suceder choques con severidades altas más usualmente, y por lo tanto, son más riesgosos. Sería interesante estudiar las interacciones entre variables, para determinar cómo se comportan los portafolios de choferes con más distancias recorridas según las zonas por las que se desplazan, pero esto generaría una cantidad muy amplia de variables. Por lo que a la hora de realizar el muestreo se necesita una capacidad computacional de la que se carece. Esto es una limitante de la métodologia empleada, se requiere mucho tiempo y poder computacional para llevar a cabo el muestreo correctamente.

- Por el lado de la frecuencia hay resultados interesantes. Recordemos que la variable Bono segmenta a los portafolios en niveles que van de 1 a 5, donde un portafolio en nivel 1 contiene personas que suelen utilizar el seguro, y el nivel 5 contiene personas que no suelen utilizarlos. Por lo que su importancia a la hora de estudiar la frecuencia es clara. Lo interesante es el que nivel donde el cambio con respecto al nivel base es más amplio, no es el nivel 5, sino el nivel 4. Esto nos indica que estos portafolios son mejores, que de hecho no es lo que se espera, por lo que el análisis es valioso. Se tiene la limitante en el modelo Poisson de asumir independencia de las observaciones, cuando quizás estén correlacionadas.

- El muestreo vía MCMC es una alternativa interesante a la opción frecuentista del ajuste de un modelo lineal generalizado. Se pasa de optimizar una función, a realizar una muestra de los parámetros desde la misma distribución de estos, utlizando la información que tenemos. Dando así más precisión a la hora de hacer la inferencia sobre los resultados. Estos detalles se notan con los intervalos de credibilidad brindados por el modelo, donde todas tienen una varianza baja y se aprovecha la información que se tiene de la base de datos para aumentar la precisión del análisis. Esto es de vital importancia para el ejercicio actuarial, pues entre más precisos sean los análisis mejores serán las estimaciones de pérdidas, reservas, etc, y mejor gestión de riesgo se puede realizar.

- La implementación del algoritmo de Hamilton Monte Carlo para llevar

a cabo el muestreo del MCMC es una herramienta muy poderosa. Es bastante útil pues en casos donde la autocorrelación de la cadena es fuerte, el algoritmo brilla. También es muy versátil pues en caso de que la autocorrelación sea baja el algoritmo trabaja de forma usual y tarda lo mismo que el algoritmo de Metropolis-Hastings, pero a la hora de que la autocorrelación aumenta, sus resultados son impresionantes, pues en una misma cantidad de pasos donde el algoritmo de Metropolis Hastings esta lejos de converger, este ya lo ha hecho. Para la base de datos en cuestión la autocorrelación del modelo de poisson era bastante alta, y esto causaba que las cadenas no se mezclaran bien con el algoritmo Metropolis-Hastings. A la hora de implementar este método la mejora es sustancial y se obtienen los resultados mostrados en el documento, donde la convergencia y la mezcla es clara.

Referencias

[1] Agresi. A. (2015) *Foundations of Linear and Generalized Linear Models.* John Wiley & Sons, Inc., Hoboken, New Jersey

[2] Cowles, M.K. (2013) *Applied Bayesian Statistics With R and OpenBUGS Examples*, Springer.

[3] Gelman, A., Carlin, J.B., Stern, H., Dunson, D., Vehtari, A., Rubin, D. (2014) *Bayesian Data Analysis*, Springer, Texts in Statistical Science Series, 3nd edition.

[4] Herzberg, A.M., Andrews, D.F. *Data: A Collection of Problems from Many Fields for the Student and Research Worker*, Springer Series in Statistics.

[5] Marin, J.M., Robert, C. (2014) *Bayesian Essentials with R*, Springer Texts in Statistics, 2nd edition.

[6] Robert, C., Casella, G. (2009) *Introducing Monte Carlo Methods with R*, Springer, Use R!.

[7] Robert, C. (2001) *The Bayesian Choice*, Springer Texts in Statistics.

5. Anexos

```
```{r, warning=FALSE, comment=FALSE}
load("C:/Users/esteb/Desktop/Actuariales/Modelos lineales/Proyecto X/swautoins.r
library(ggplot2)
library(ggpubr)

```



```{r}
swautoins$Insured = swautoins$Insured/100000
swautoins$Insured = swautoins$Insured * 365/12

datos <- cbind(Kilometros = swautoins$Kilometres, Zona = swautoins$Zone, Bono =
datos <- as.data.frame(datos)

datos$Kilometros <- as.factor(datos$Kilometros)
datos$Zona <- as.factor(datos$Zona)
datos$Bono <- as.factor(datos$Bono)
datos$Tipo <- as.factor(datos$Tipo)

```
```{r}
scatterSeveridad <- ggplot(datos, aes(x = Meses, y = Severidad)) + geom_point()
scatterSeveridad
scatterFrec <- ggplot(datos, aes(x = Meses, y = Reclamos)) + geom_point() + them
scatterFrec
```
```{r}
hf <- density(datos$PromedioFrec)
hs <- density(datos$PromedioSev)

frecuencia <- ggplot(datos, aes(x = PromedioFrec)) + geom_histogram(binwidth = h

severidad <- ggplot(datos, aes(PromedioSev)) + geom_histogram(binwidth = hs$bw)

histogramas <- ggarrange(frecuencia, severidad, labels = c("Frecuencia", "Severi
histogramas
```
```

```
#Boxplots Importantes
```{r}
fill <- "#4271AE"
line <- "#1F3552"
severidad.km <- ggplot(datos, aes(x=Kilometros, y= PromedioSev)) +
  geom_boxplot(fill = fill, colour = line, alpha = 0.7) +
  scale_y_continuous(name = "Severidad Promedio") +
  scale_x_discrete(name = "Kilometros") +
  theme_bw()
severidad.zone <- ggplot(datos, aes(x=Zona, y= PromedioSev)) +
  geom_boxplot(fill = fill, colour = line, alpha = 0.7) +
  scale_y_continuous(name = "Severidad Promedio") +
  scale_x_discrete(name = "Zona") +
  theme_bw()
severidad.bonus <- ggplot(datos, aes(x=Bono, y= PromedioSev)) +
  geom_boxplot(fill = fill, colour = line, alpha = 0.7) +
  scale_y_continuous(name = "Severidad Promedio") +
  scale_x_discrete(name = "Bono") +
  theme_bw()
severidad.make <- ggplot(datos, aes(x=Tipo, y= PromedioSev)) +
  geom_boxplot(fill = fill, colour = line, alpha = 0.7) +
  scale_y_continuous(name = "Severidad Promedio") +
  scale_x_discrete(name = "Tipo de Vehículo") +
  theme_bw()

km.zone <- ggarrange(severidad.km, severidad.zone, labels = c("Kilometros", "Zon
bonus.make <- ggarrange(severidad.bonus, severidad.make, labels = c("Bonus", "Ti
km.zone
bonus.make
```
```{r}
fill <- "#4271AE"
line <- "#1F3552"
frec.km <- ggplot(datos, aes(x=Kilometros, y= PromedioFrec)) +
  geom_boxplot(fill = fill, colour = line, alpha = 0.7) +
  scale_y_continuous(name = "Frecuencia Promedio") +
  scale_x_discrete(name = "Kilometros") +
  theme_bw()
frec.zone <- ggplot(datos, aes(x=Zona, y= PromedioFrec)) +
  geom_boxplot(fill = fill, colour = line, alpha = 0.7) +
```
```

```
  scale_y_continuous(name = "Frecuencia Promedio") +
  scale_x_discrete(name = "Zona") +
  theme_bw()
frec.bonus <- ggplot(datos, aes(x=Bono, y= PromedioFrec)) +
  geom_boxplot(fill = fill, colour = line, alpha = 0.7) +
  scale_y_continuous(name = "Frecuencia Promedio") +
  scale_x_discrete(name = "Bono") +
  theme_bw()
frec.make <- ggplot(datos, aes(x=Tipo, y= PromedioFrec)) +
  geom_boxplot(fill = fill, colour = line, alpha = 0.7) +
  scale_y_continuous(name = "Frecuencia Promedio") +
  scale_x_discrete(name = "Tipo de Vehículo") +
  theme_bw()

km.zone <- ggarrange(frec.km, frec.zone, labels = c("Kilometros", "Zona"), ncol =
bonus.make <- ggarrange(frec.bonus, frec.make, labels = c("Bonus", "Tipo de Vehí

km.zone
bonus.make
```

```{r}
library(leaps)
library(rstanarm)
library(bayesplot)
```

#Seleccion de Variables
```{r}
set.seed(666)
conjunto <- regsubsets(Reclamos ~ Kilometros+Zona+Bono+Tipo+Meses, data = datos,
resumen <- summary(conjunto)
which(resumen$bic == min(resumen$bic))
min(resumen$bic)
resumen$bic
```

#Modelo Poisson
```{r}
modeloFrec <- stan_glm(Reclamos ~ Bono+Meses, data = datos, family = poisson, al


```

```
```

```{r}
resumen <- as.array(modeloFrec)
summary(modeloFrec)
```


```{r}
mcmc_intervals(resumen, pars = c("Bono2", "Bono3", "Bono4", "Bono5"), prob = 0.8
  prob_outer = 0.95)
mcmc_intervals(resumen, pars = c("Meses"), prob = 0.8,
  prob_outer = 0.95)
mcmc_intervals(resumen, pars = c("(Intercept)"),prob = 0.8,
  prob_outer = 0.95)
```

```{r}
mcmc_trace(resumen,
           pars = c("Bono2", "Bono3", "Bono4", "Bono5"),
           facet_args = list(ncol = 1, strip.position = "left"))

mcmc_trace(resumen,
           pars = c("Meses"),
           facet_args = list(ncol = 1, strip.position = "left"))

mcmc_trace(resumen,
           pars = c("(Intercept)"),
           facet_args = list(ncol = 1, strip.position = "left"))
```

```{r}
rhats <- rhat(modeloFrec)
mcmc_rhat(rhats)

ratios_cp <- neff_ratio(modeloFrec)
mcmc_neff(ratios_cp, size = 2)
```

#Selección de Variables
```{r}
```

```
conjunto <- regsubsets(log(Severidad+1) ~ Kilometros+Zona+Bono+Tipo+Meses, data =
resumen <- summary(conjunto)
which(resumen$bic == min(resumen$bic))
min(resumen$bic)
resumen$bic
```

#Modelo Severidad

```{r}
modeloSev <- stan_glm(log(Severidad+1) ~ Zona+Meses+Kilometros, data = datos, fa
```

```{r}
summary(modeloSev)
```

```{r}
resumen <- as.array(modeloSev)
color_scheme_set("red")
mcmc_intervals(resumen, pars = c("Zona2", "Zona3", "Zona4", "Zona5", "Zona6"), p
  prob_outer = 0.95)
mcmc_intervals(resumen, pars = c("Kilometros2", "Kilometros3", "Kilometros4", "K
  prob_outer = 0.95)
mcmc_intervals(resumen, pars = c("Meses"), prob = 0.8,
  prob_outer = 0.95)
mcmc_intervals(resumen, pars = c("(Intercept)"),prob = 0.8,
  prob_outer = 0.95)
```

```{r}

mcmc_trace(resumen,
           pars = c("Zona2", "Zona3", "Zona4", "Zona5", "Zona6"),
           facet_args = list(ncol = 1, strip.position = "left"))

mcmc_trace(resumen,
           pars = c("Kilometros2", "Kilometros3", "Kilometros4", "Kilometros5"),
           facet_args = list(ncol = 1, strip.position = "left"))

mcmc_trace(resumen,
           pars = c("Meses"),
```

```
           facet_args = list(ncol = 1, strip.position = "left"))

mcmc_trace(resumen,
           pars = c("(Intercept)"),
           facet_args = list(ncol = 1, strip.position = "left"))
```

```{r}
rhats <- rhat(modeloSev)
mcmc_rhat(rhats)

ratios_cp <- neff_ratio(modeloSev)
mcmc_neff(ratios_cp, size = 2)
```