R Notebook

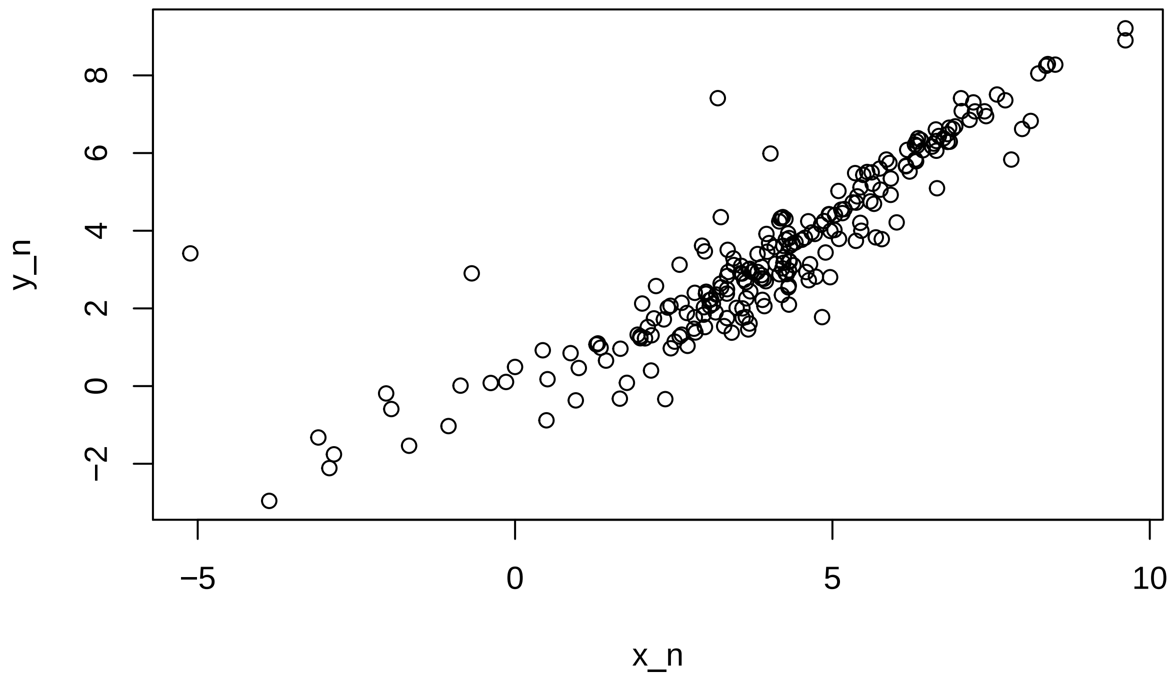
1. El Mercado asegurado en México opera en diferentes sectores. Siete de estos sectores son: Accidentes y enfermedades (ACC), Agricultura y ganadería (AGR), automóviles (AUT), gastos médicos mayores (MED), Incendios (FIR), responsabilidad civil y riesgos profesionales (LIA) y salud (HEA). Es de interés para las compañías de seguros predecir el monto de reclamación (Yi) en términos de la prima cobrada (Xi), medida en millones de pesos. La comisión nacional de seguros y fianzas junta la información de todas las compañías de seguro año con año para cada uno de los 32 etados de la republica y en algunos casos para el extranjero.

Pata el año 2010 se cuenta con n=228 registros clasificados por sector asegurados.

a) Realiza una gráfica de dispersión entre Xi vs Yi. Comenta sobre la posible relación entre estas dos variables

Se observa una relación positiva de los datos, es decir al incrementarse las primas emitidas el número de siniestro de igual forma se ve incrementado. Es por ello que es factible que podamos realizar una regresión lineal a los mismos

```r
#Cargamos los datos y graficamos
datos<-read.csv("FES2010c.csv",header=TRUE)
nombres<-c("Entidad","Operacion","Prima","Siniestro","z1","z2","z3","z4","z5","z6","z7")
names(datos)<-nombres
n<-nrow(datos)

x_n<-log(datos$Prima)
y_n<-log(datos$Siniestro)

plot(x_n,y_n)
```



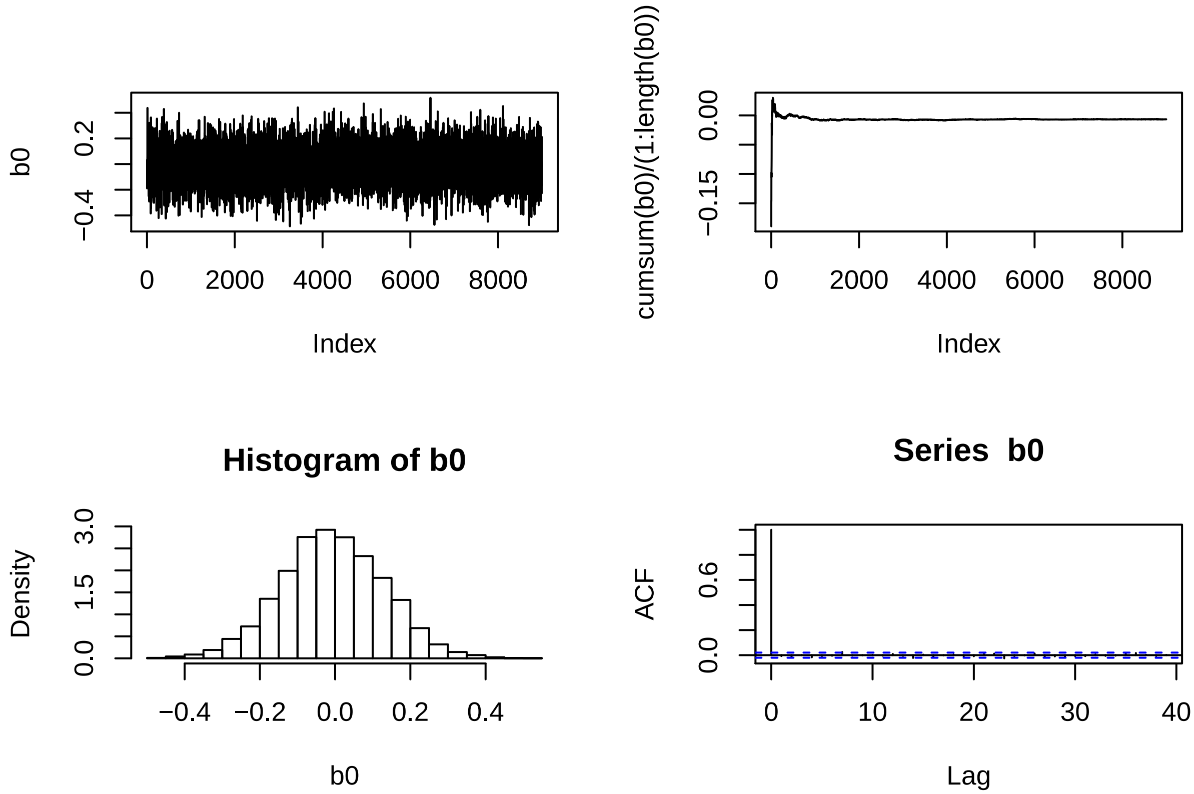
b) Ajusta un modelo de regresión lineal normal a los datos

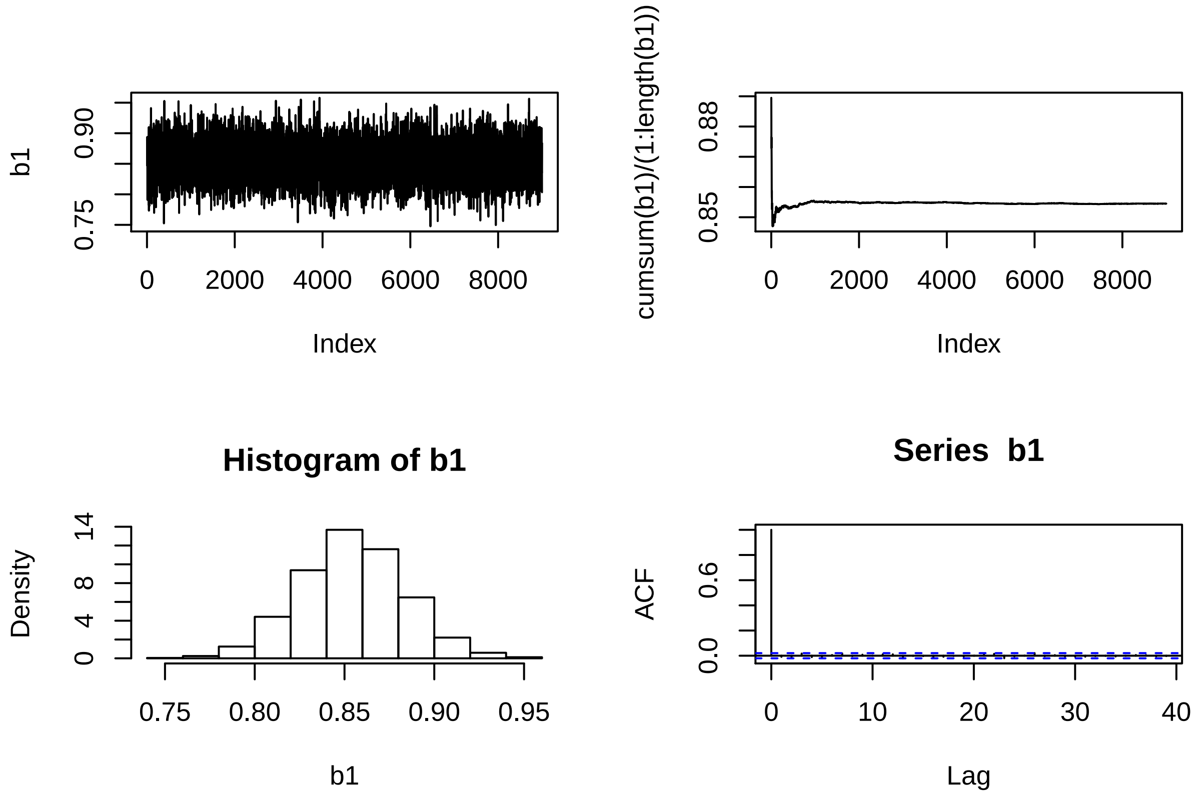
```
#Modelo Normal
#-Defining data-
  data<-list("n"=n, "y"=y_n,"x"=x_n)
  #-Defining inits-
  inits<-function(){list(beta=rep(0,2),tau=1,yf=rep(0,n))}
  #-Selecting parameters to monitor-
  parameters<-c("beta","tau","yf")
  #-Running code-
  exa_b<-bugs(data,inits,parameters,model.file="Exa_b.txt",            n.iter=10000,n.chains=1,n.burnin=
```

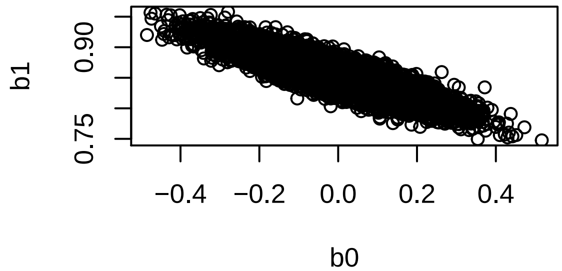
En relación a los coeficientes obtenidos por el modelo para alfa(b0), podemos ver en la grafica que toma valores igual a cero. Por lo que nos llevaría a pensar que no es significante para el modelo.

Por otra parte el coeficiente para beta(b1), no toma el valor igual a cero, por lo que diriamos que este coeficiente si es significativo para nuestra variable.

```
out_b<-Result(exa_b)
```


b0
0.2
−0.4
0
2000
4000
6000
8000
Index
cumsum(b0)/(1:length(b0))
0.00
−0.15
0
2000
4000
6000
8000
Index
Histogram of b0
Density
3.0
1.5
0.0
−0.4
−0.2
0.0
0.2
0.4
b0
Series b0
ACF
0.6
0.0
0
10
20
30
40
Lag


b1
0.75
0.90
0
2000
4000
6000
8000
Index
cumsum(b1)/(1:length(b1))
0.85
0.88
0
2000
4000
6000
8000
Index
Histogram of b1
Density
0
4
8
14
0.75
0.80
0.85
0.90
0.95
b1
Series b1
ACF
0.0
0.6
0
10
20
30
40
Lag



El estimador puntual para alfa es -0.006651439 con un intervalo al 95% de confianza igual a (-0.27860000, 0.2564025), el cual contiene al cero, por lo cual podemos concluir que no es un coeficiente significativo.

El estimador puntual para beta es igual 0.854498111 con intervalo de confianza igual a (0.79719750,0.9125025). Por lo cual este coeficiente es significativo.

Adicional se puede observar que al graficar las betas, hay una correlación negativa. Por lo que beta, sería suficiente para explicar el modelo.

Adicional podemos decir que por cada incremento de x, y se ve incrementado beta veces en escala logaritmica.

```
#OpenBUGS
 out_b.sum<-exa_b$summary
#print(out_b.sum)
 head(out_b.sum)
```

```
##                 mean         sd        2.5%        25%        50%
## beta[1] -0.006651439 0.13603743 -0.27860000 -0.0971975 -0.0090565
## beta[2]  0.854498111 0.02915915  0.79719750  0.8351000  0.8546000
## tau      0.979907456 0.09241861  0.80660000  0.9161000  0.9772000
## yf[1]    4.174111867 1.00824756  2.20397499  3.5190000  4.1750000
## yf[2]    3.386130074 1.02598579  1.36795000  2.6977500  3.3830000
## yf[3]    2.019799346 1.01883502 -0.00324205  1.3417500  2.0080000
##               75%     97.5%
## beta[1] 0.0859275 0.2564025
## beta[2] 0.8739000 0.9125025
## tau     1.0420000 1.1680000
## yf[1]   4.8540000 6.1450500
```

```
## yf[2]    4.0630000 5.4410750
## yf[3]    2.6960000 4.0100000
```

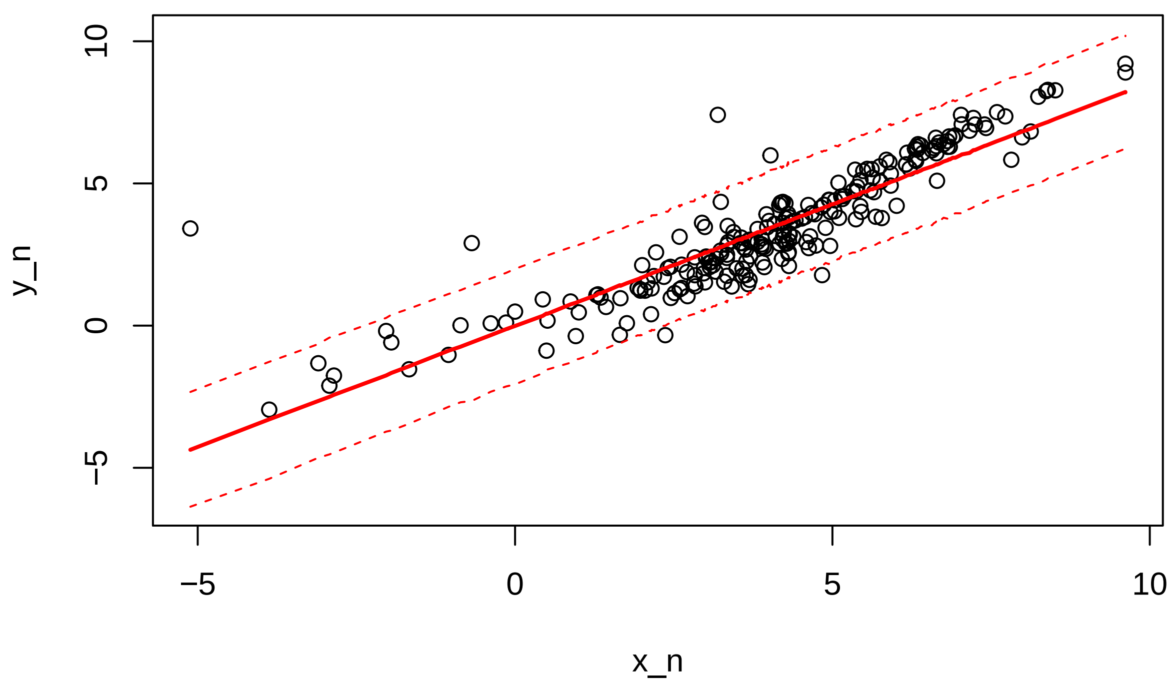
```
  #DIC
  out_b.yf<-out_b.sum[grep("yf",rownames(out_b.sum)),]
  out_b.dic<-exa_b$DIC
  seudoR_b<-cor(y_n,out_b.yf[,1])
  print(out_b.dic)
```

```
## [1] 655.6
```

```
  print(seudoR_b)
```

```
## [1] 0.8935554
```

```
#Predictions
or<-order(x_n)
ymin<-min(y_n,out_b.yf[,c(1,3,7)])
ymax<-max(y_n,out_b.yf[,c(1,3,7)])

par(mfrow=c(1,1))
plot(x_n,y_n,ylim=c(ymin,ymax))
#Modelo 1
lines(x_n[or],out_b.yf[or,1],lwd=2,col=2)
lines(x_n[or],out_b.yf[or,3],lty=2,col=2)
lines(x_n[or],out_b.yf[or,7],lty=2,col=2)
```



c)Ajusta un modelo lineal generalizado gamma

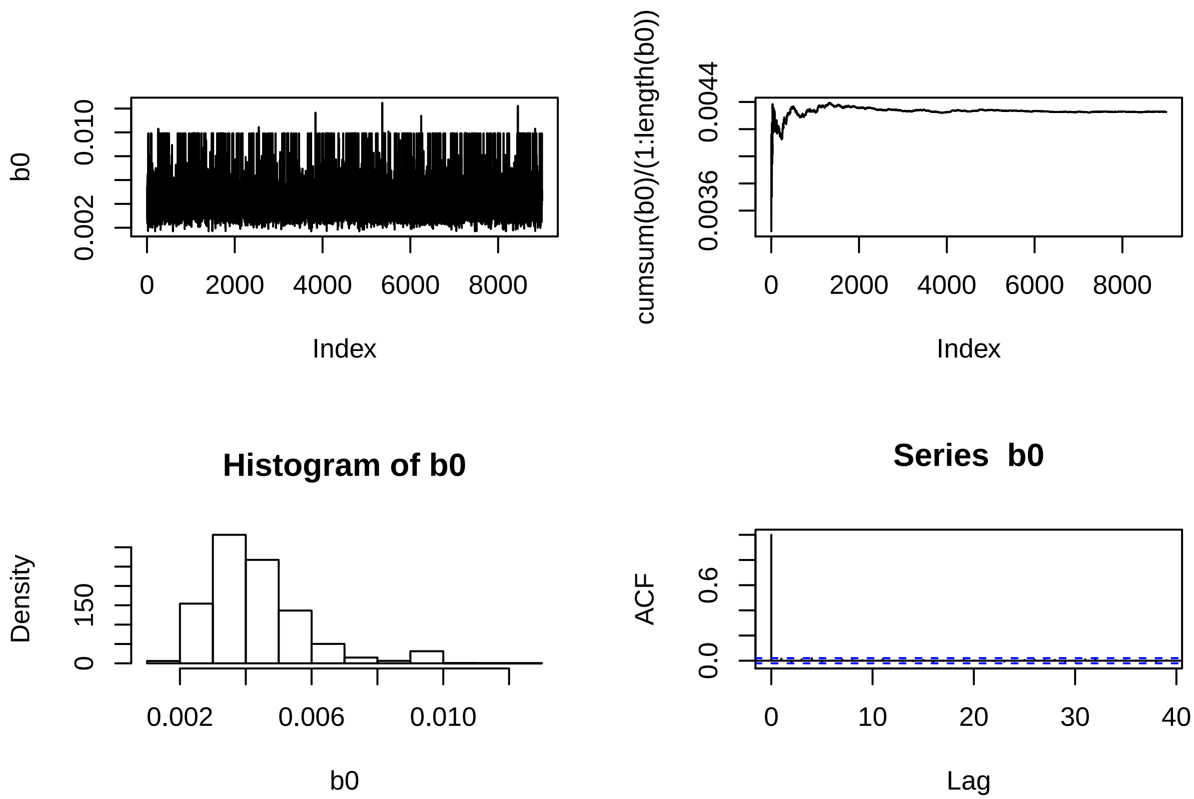
```r
#Modelo de regresión gamma

y_g<-(datos$Siniestro)/1000
x_g<-(datos$Prima)/1000

#-Defining data-
data<-list("n"=n, "y"=y_g,"x"=x_g)
#-Defining inits-

inits<-function(){list(beta=rep(1,2),r=1,yf=rep(0,n))}

#-Selecting parameters to monitor-
parameters<-c("beta","r","yf")
#-Running code-


exa_c<-bugs(data,inits,parameters,model.file="Exa_c.txt",           n.iter=10000,n.chains=1,n.burnin=1
```

El estimador para alfa(b0), toma valores mayores a cero. Nos llevaría a pensar que es significante para el modelo.

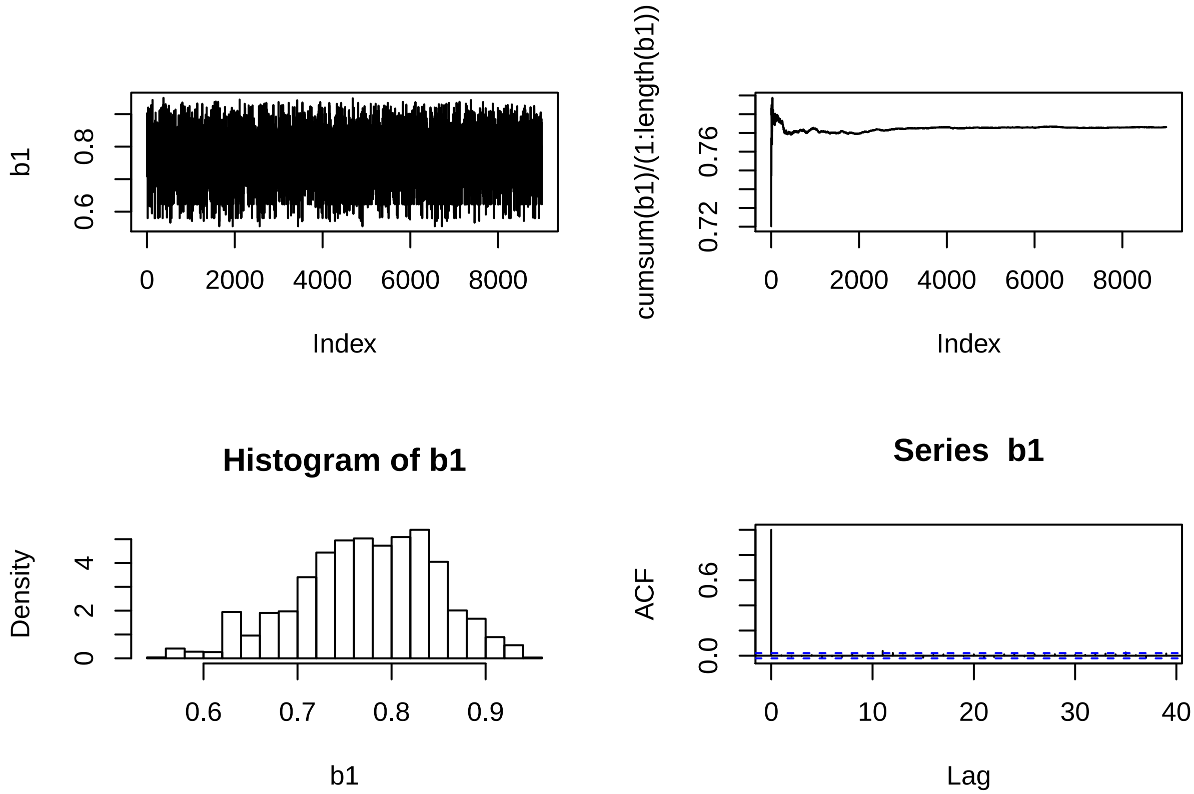
El coeficiente beta(b1), no toma el valores igual a cero, por lo que es coeficiente si es significativo para nuestra modelo.

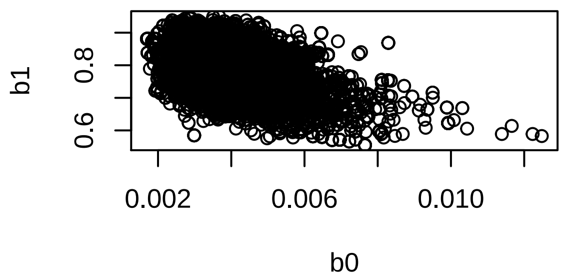
Adicionalmente se puede observar en las gráficas que la convergencia para alfa, fue mucho mas dificil de lograr en la simulación, haciendo incluso que el computo de la misma, fuese mucho mas tardado que en el ejercicio anterior.

```r
out_c<-Result(exa_c)
```



b0
0.002
0.010
0
2000
4000
6000
8000
Index
cumsum(b0)/(1:length(b0))
0.0036
0.0044
0
2000
4000
6000
8000
Index
Histogram of b0
Density
0
150
0.002
0.006
0.010
b0
Series b0
ACF
0.0
0.6
0
10
20
30
40
Lag


b1
0.6
0.8
0
2000
4000
6000
8000
Index
cumsum(b1)/(1:length(b1))
0.72
0.76
0
2000
4000
6000
8000
Index
Histogram of b1
Density
0
2
4
0.6
0.7
0.8
0.9
b1
Series b1
ACF
0.0
0.6
0
10
20
30
40
Lag



El estimador puntual para alfa es 0.00432663 con un intervalo al 95% de confianza igual a (0.0023400000, 0.00992400), el cual no contiene al cero.

El estimador puntual para beta es igual 0.77313010 con intervalo de confianza igual a (0.6226000000,0.90280750). Por lo cual este coeficiente es significativo.

Adicional se puede observar que al graficar las betas, no hay una correlación. Ambos modelos son significativos.

En el caso particular por cada unidad de cambio en "x", y

```
  #OpenBUGS
  out_c.sum<-exa_c$summary
 #print(out_b.sum)
  head(out_c.sum)
```

```
##                 mean          sd         2.5%        25%      50%       75%
## beta[1] 0.00432663 0.001556149 0.0023400000 0.00329175 0.004021 0.0049550
## beta[2] 0.77313010 0.073666323 0.6226000000 0.72617500 0.778000 0.8273000
## r       0.82611563 0.068570949 0.6970974998 0.77857500 0.823750 0.8728000
## yf[1]   0.10604856 0.117516819 0.0014238748 0.02490500 0.067830 0.1469250
## yf[2]   0.04531818 0.049170418 0.0005914975 0.01062000 0.029535 0.0636525
## yf[3]   0.01268569 0.014314734 0.0001531975 0.00279650 0.007988 0.0171750
##              97.5%
## beta[1] 0.00992400
## beta[2] 0.90280750
## r       0.96670750
## yf[1]   0.42481998
## yf[2]   0.17350000
```

```
## yf[3]   0.05277025
```

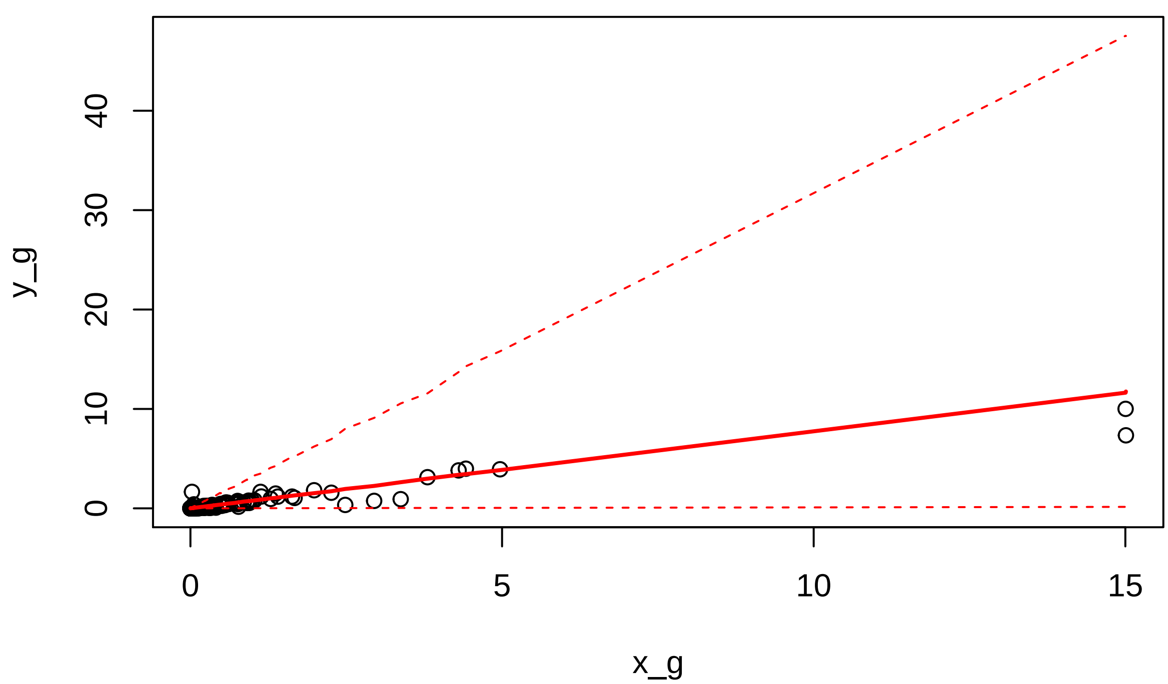
```r
#DIC
out_c.yf<-out_c.sum[grep("yf",rownames(out_c.sum)),]
out_c.dic<-exa_c$DIC
seudoR_c<-cor(y_g,out_c.yf[,1])
print(out_c.dic)
```

```
## [1] -790.5526
```

```r
print(seudoR_c)
```

```
## [1] 0.9574803
```

```r
#Predictions
or<-order(x_g)
ymin<-min(y_g,out_c.yf[,c(1,3,7)])
ymax<-max(y_g,out_c.yf[,c(1,3,7)])

par(mfrow=c(1,1))
plot(x_g,y_g,ylim=c(ymin,ymax))
#Modelo 1
lines(x_g[or],out_c.yf[or,1],lwd=2,col=2)
lines(x_g[or],out_c.yf[or,3],lty=2,col=2)
lines(x_g[or],out_c.yf[or,7],lty=2,col=2)
```



d) Compara los modelos del inciso b y c.

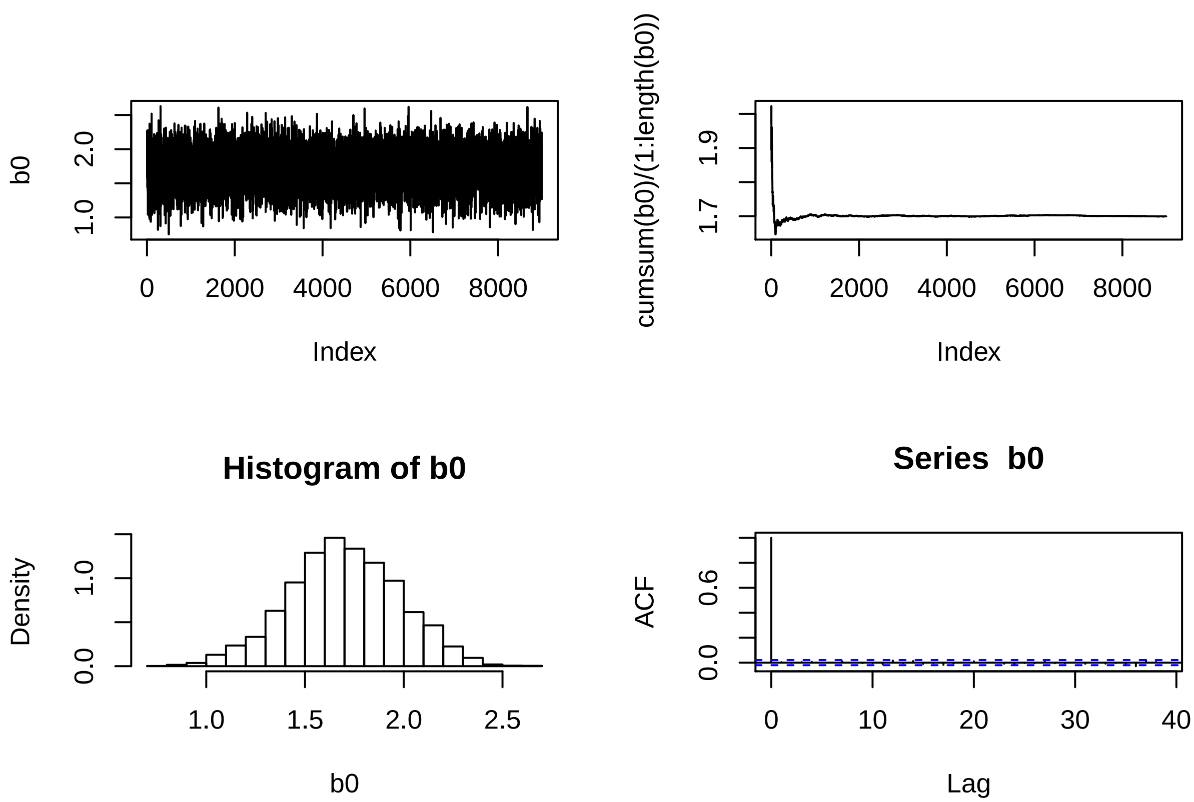
En cuanto al valor al criterio de comparación pseudo R2ˆ podríamos decir que el mejor modelo que ajusta a los datos es el modelo gamma.

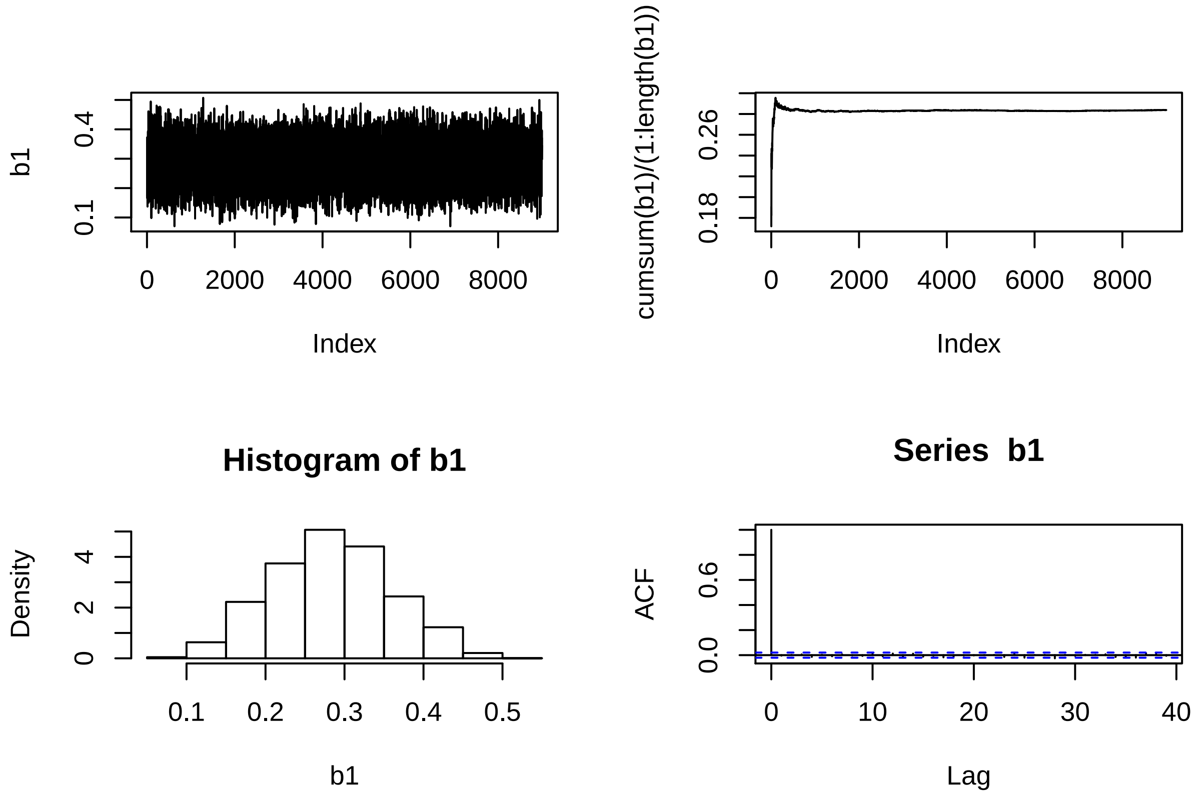
Llama mi atención que para montos grandes de siniestros la banda del intervalo de confianza sea muy grande. En el negocio de seguro, para los siniestros pocos probables los costos suele ser muy elevados.

e)

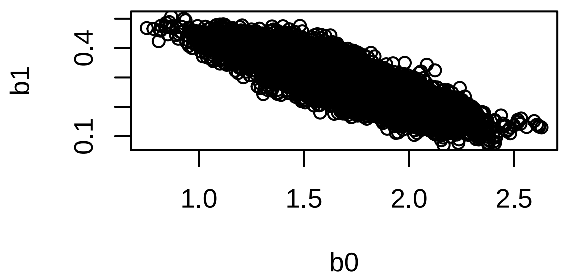
```r
#Modelo Normal

#-Defining data-
data<-list("n"=n, "y"=y_n,"x"=x_n,
           "z2"=datos$z2,
           "z3"=datos$z3,
           "z4"=datos$z4,
           "z5"=datos$z5,
           "z6"=datos$z6,
           "z7"=datos$z7)
#-Defining inits-
inits<-function(){list(beta=rep(0,2),tau=1,yf=rep(0,n),a=rep(0,6),b=rep(0,6))}

#-Selecting parameters to monitor-
parameters<-c("beta","a","b","tau","yf")
#-Running code-

  exa_e<-bugs(data,inits,parameters,model.file="Exa_e.txt",            n.iter=10000,n.chains=1,n.burn
```

```r
out_e<-Result(exa_e)
```





b1
0.1
0.4
0
2000
4000
6000
8000
Index
cumsum(b1)/(1:length(b1))
0.18
0.26
0
2000
4000
6000
8000
Index
Histogram of b1
Density
0
2
4
0.1
0.2
0.3
0.4
0.5
b1
Series b1
ACF
0.0
0.6
0
10
20
30
40
Lag



El parámetro alfa para el model general toma relevancia, con un estimación puntual igual a 1.6995107 y un intervalo de confianza igual a (1.1380000, 2.23300000).

El parámetro beta, se ve disminuido, con una estimación puntual de 0.2838213 con un intervalo (0.1423975, 0.43050500)

Podemos observas que las alfas correspondientes a cada uno de los modelos son signifivas, salvo el correspondiente a z5, correspondiente al ramo de incendios.

Por otra parte al disminuir la beta general del modelo, se ve el efecto del ramo o sector que tiene sobre los siniestros. Donde el ramo de Gastos Médicos mayores y autos son los coeficientes con mayor peso.

```
#OpenBUGS
out_e.sum<-exa_e$summary
print(out_e.sum[1:20,])
```

```
##                mean         sd       2.5%       25%       50%       75%
## beta[1]  1.6995107 0.27820467  1.1380000  1.513000  1.69500  1.892000
## beta[2]  0.2838213 0.07522677  0.1423975  0.228500  0.28410  0.333400
## a[1]    -1.1963114 0.53559854 -2.1850250 -1.571000 -1.21300 -0.843150
## a[2]    -1.8401701 0.99461194 -3.9070000 -2.483000 -1.82150 -1.138000
## a[3]    -2.8029856 0.63188872 -4.0100250 -3.252000 -2.81400 -2.359000
## a[4]    -1.0090998 0.63494289 -2.2790250 -1.435000 -1.01500 -0.589075
## a[5]    -2.3448383 0.52715785 -3.3850000 -2.690000 -2.34200 -2.012000
## a[6]    -1.4994812 0.31579736 -2.1120000 -1.715000 -1.49700 -1.288000
## b[1]     0.3661070 0.14184972  0.0856900  0.269800  0.36940  0.465125
## b[2]     0.7011281 0.15563793  0.4150950  0.590100  0.69485  0.804725
## b[3]     0.8123143 0.12537823  0.5679800  0.723375  0.81250  0.905225
```

```
## b[4]      0.4187319 0.14135451  0.1369975  0.324375  0.42150  0.512500
## b[5]      0.5415224 0.13722498  0.2706000  0.451800  0.54100  0.633400
## b[6]      0.4695901 0.09566856  0.2756000  0.404900  0.47060  0.535800
## tau       1.5443022 0.14999375  1.2660000  1.441000  1.53700  1.642250
## yf[1]     3.1027847 0.82771756  1.4829750  2.536000  3.10850  3.665000
## yf[2]     2.8182417 0.82605050  1.1790000  2.272000  2.81300  3.376250
## yf[3]     2.3688039 0.82398047  0.7460400  1.813750  2.37400  2.930000
## yf[4]     2.4534468 0.83046814  0.8469600  1.894750  2.44900  3.017000
## yf[5]     2.7551546 0.83070543  1.1259000  2.186000  2.76400  3.314250
##                97.5%
## beta[1]   2.23300000
## beta[2]   0.43050500
## a[1]     -0.11252500
## a[2]     -0.01095225
## a[3]     -1.56500000
## a[4]      0.27253750
## a[5]     -1.30600000
## a[6]     -0.87189250
## b[1]      0.63460250
## b[2]      1.02702499
## b[3]      1.04700000
## b[4]      0.70141500
## b[5]      0.81471499
## b[6]      0.65380500
## tau       1.85200000
## yf[1]     4.71000000
## yf[2]     4.44400000
## yf[3]     3.97415000
## yf[4]     4.08505000
## yf[5]     4.37202500
```

```
head(out_e.sum)
```

```
##                 mean          sd       2.5%      25%      50%       75%
## beta[1]   1.6995107 0.27820467  1.1380000  1.5130  1.6950  1.892000
## beta[2]   0.2838213 0.07522677  0.1423975  0.2285  0.2841  0.333400
## a[1]     -1.1963114 0.53559854 -2.1850250 -1.5710 -1.2130 -0.843150
## a[2]     -1.8401701 0.99461194 -3.9070000 -2.4830 -1.8215 -1.138000
## a[3]     -2.8029856 0.63188872 -4.0100250 -3.2520 -2.8140 -2.359000
## a[4]     -1.0090998 0.63494289 -2.2790250 -1.4350 -1.0150 -0.589075
##                97.5%
## beta[1]   2.23300000
## beta[2]   0.43050500
## a[1]     -0.11252500
## a[2]     -0.01095225
## a[3]     -1.56500000
## a[4]      0.27253750
```

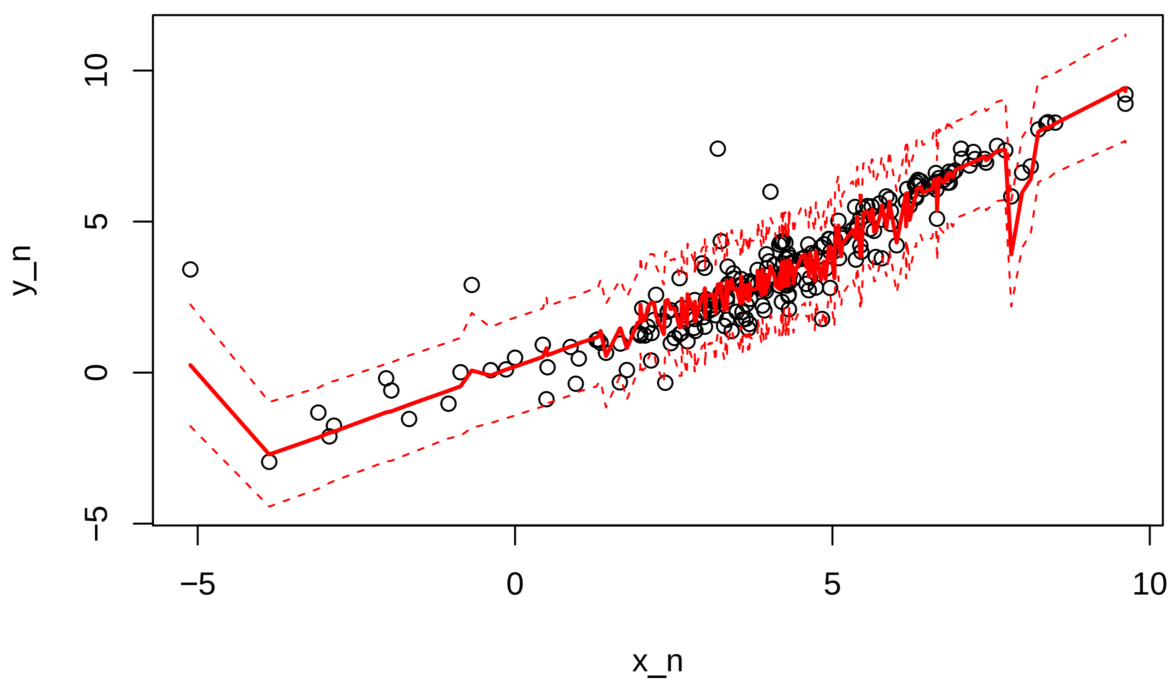
```
#DIC
out_e.yf<-out_e.sum[grep("yf",rownames(out_e.sum)),]
out_e.dic<-exa_e$DIC
seudoR_e<-cor(y_n,out_e.yf[,1])
print(out_e.dic)
```

```
## [1] 564.4
```

```
print(seudoR_e)
```

```
## [1] 0.9372898
```

```
#Predictions
or<-order(x_n)
ymin<-min(y_n,out_e.yf[,c(1,3,7)])
ymax<-max(y_n,out_e.yf[,c(1,3,7)])

par(mfrow=c(1,1))
plot(x_n,y_n,ylim=c(ymin,ymax))
#Modelo 1
lines(x_n[or],out_e.yf[or,1],lwd=2,col=2)
lines(x_n[or],out_e.yf[or,3],lty=2,col=2)
lines(x_n[or],out_e.yf[or,7],lty=2,col=2)
```



f)¿Cuál es la interpretación de alfa y beta en el modelo del inciso (e)? Que interpretación tiene

La alfa y beta del inciso e) refleja el peso de todas las variables en su conjunto, el peso de estas variables se ve reforzado o decrementado al combinarse con las alfas y betas para cada variable indicadora.

g)Ajusta un modelo lineal generalizado gamma para los montos de reclamación

```
#Modelo de regresión gamma

#-Defining data-
data<-list("n"=n, "y"=y_g,"x"=x_g,
              "z2"=datos$z2,
              "z3"=datos$z3,
```

```r
              "z4"=datos$z4,
              "z5"=datos$z5,
              "z6"=datos$z6,
              "z7"=datos$z7)
  #-Defining inits-

inits<-function(){list(beta=rep(1,2),a=rep(1,6),b=rep(1,6),r=1,yf=rep(0,n))}

#-Selecting parameters to monitor-
parameters<-c("beta","a","b","r","yf")
#-Running code-


exa_g<-bugs(data,inits,parameters,model.file="Exa_g.txt",            n.iter=10000,n.chains=1,n.burnin=1
```

Los estimadores para alfa y beta para el modelo general, en el cual estan incluidas todas las variables son relevantes, puesto que no contiene el cero en su respectivo intervalo de confianza.

Por otro lado, las alfas y betas relevantes, que ponderan mayor información por sector asegurador corresponden salud, agricultura y ganaderia

```r
  #OpenBUGS
  out_g.sum<-exa_g$summary
  print(out_g.sum[1:14,])
```

```
##                mean          sd        2.5%        25%        50%
## beta[1]  0.008601830 0.002900061  0.00668200  0.0069170  0.007210
## beta[2]  0.747444933 0.006922399  0.73090000  0.7482000  0.750600
## a[1]     0.907499189 0.015795057  0.89080000  0.8987000  0.901300
## a[2]     0.050489123 0.147530340 -0.16990500 -0.0520325  0.027535
## a[3]    -0.005085973 0.016005670 -0.02174025 -0.0146600 -0.009255
## a[4]     0.118193400 0.033635815  0.06465975  0.0944200  0.114600
## a[5]    -0.006525426 0.004415924 -0.01494000 -0.0081940 -0.006175
## a[6]    -0.004625503 0.005534735 -0.00718100 -0.0068530 -0.006703
## b[1]     1.074411556 0.007127976  1.06000000  1.0710000  1.077000
## b[2]     0.135670215 0.299578518 -0.34921250 -0.0741525  0.100900
## b[3]    -0.040844280 0.129494217 -0.26900000 -0.1427250 -0.043420
## b[4]    -0.513490216 0.175144004 -0.72280000 -0.6376000 -0.551800
## b[5]    -0.281002001 0.219519248 -0.57970000 -0.5035000 -0.280500
## b[6]     1.087390556 0.005757231  1.08400000  1.0850000  1.086000
##                75%       97.5%
## beta[1]  0.00905800  0.01550000
## beta[2]  0.75180000  0.75370000
## a[1]     0.90590000  0.94240000
## a[2]     0.12685000  0.39981500
## a[3]    -0.00059705  0.03638225
## a[4]     0.13730000  0.19710250
## a[5]    -0.00363475  0.00136640
## a[6]    -0.00651100  0.01208000
## b[1]     1.07800000  1.08500000
## b[2]     0.30630000  0.83501000
## b[3]     0.05572750  0.19640500
## b[4]    -0.43970000 -0.05856075
## b[5]    -0.12275000  0.10211500
## b[6]     1.08700000  1.10600000
```

```r
#head(out_g.sum)
```

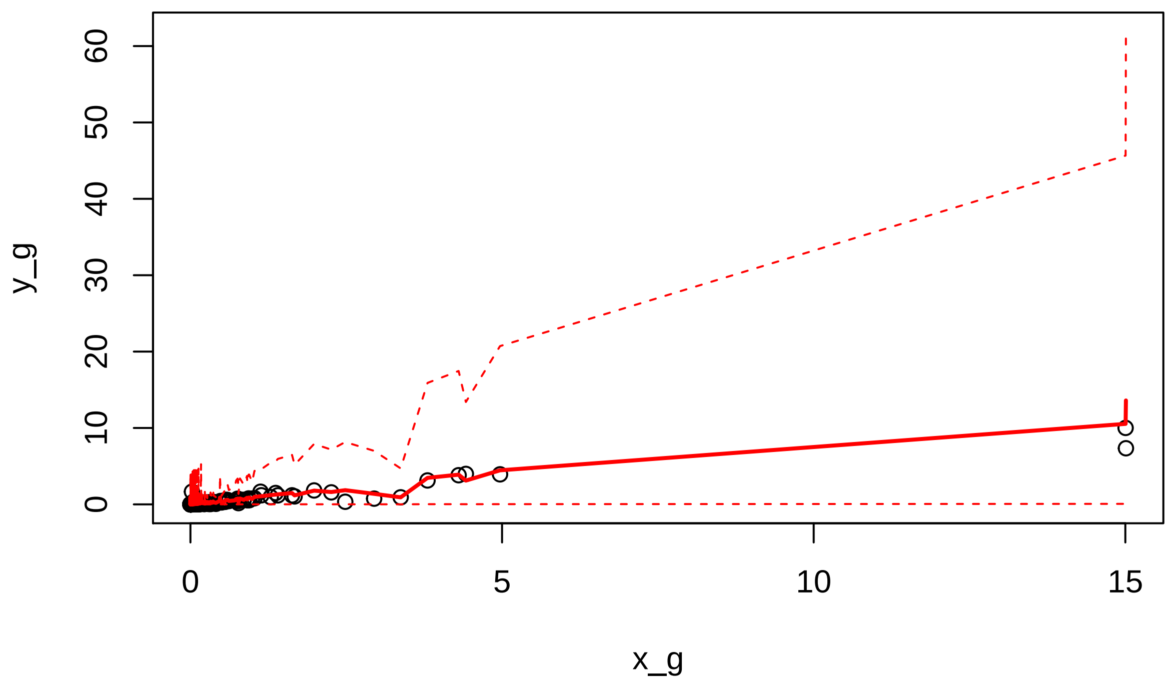
```r
#DIC
out_g.yf<-out_g.sum[grep("yf",rownames(out_g.sum)),]
out_g.dic<-exa_g$DIC
seudoR_g<-cor(y_g,out_g.yf[,1])
print(out_g.dic)
```

```
## [1] -250.2282
```

```r
print(seudoR_g)
```

```
## [1] 0.9093284
```

```r
#Predictions
or<-order(x_g)
ymin<-min(y_g,out_g.yf[,c(1,3,7)])
ymax<-max(y_g,out_g.yf[,c(1,3,7)])

par(mfrow=c(1,1))
plot(x_g,y_g,ylim=c(ymin,ymax))
#Modelo 1
lines(x_g[or],out_g.yf[or,1],lwd=2,col=2)
lines(x_g[or],out_g.yf[or,3],lty=2,col=2)
lines(x_g[or],out_g.yf[or,7],lty=2,col=2)
```



h) El mejor modelo que resulto de comparar los modelos gammas corresponde al que no desagrega los pesos por sector asegurador con una r^2 del 95% vs 90%. Perdiendo ajuste para valores cercanos a cero, pero cuidando el ajuste para valores grandes.

i) Conclusiones:

Podemos decir que dada la naturaleza de nuestros datos, donde “y” siempre toma valores positivos es mejor proponer modelos gamma.

El mejor modelo fue un modelo gamma con liga igual a mu, sin considerar la ponderación por ramo de los parámetros, ya que se perdio información para valores pequeños.

```
model
{
#Likelihood
for (i in 1:n) {
    y[i] ~ dnorm(mu[i],tau)
    mu[i]<-beta[1]+beta[2]*x[i]

}
#Priors
for (j in 1:2) { beta[j] ~ dnorm(0,0.001) }
tau ~ dgamma(0.001,0.001)
#Prediction
for (i in 1:n) { yf[i] ~ dnorm(mu[i],tau) }
}
```

```
model
{
#Likelihood
for (i in 1:n) {
    y[i] ~ dgamma(r,lambda[i])
        lambda[i]<-r/(mu[i])
    mu[i]<-beta[1]+beta[2]*x[i]

}
#Priors
for (j in 1:2) { beta[j] ~ dnorm(0,0.001) }
r ~ dunif(0,100)
#Prediction
for (i in 1:n) { yf[i] ~ dgamma(r,lambda[i]) }
}
```

```
model
{
#Likelihood
for (i in 1:n) {
    y[i] ~ dnorm(mu[i],tau)
    mu[i]<-beta[1]+beta[2]*x[i]
           +a[1]*z2[i]
           +a[2]*z3[i]
               +a[3]*z4[i]
               +a[4]*z5[i]
               +a[5]*z6[i]
               +a[6]*z7[i]
               +b[1]*x[i]*z2[i]
```

```
           +b[2]*x[i]*z3[i]
           +b[3]*x[i]*z4[i]
           +b[4]*x[i]*z5[i]
           +b[5]*x[i]*z6[i]
           +b[6]*x[i]*z7[i]
}
#Priors
for (j in 1:2) { beta[j] ~ dnorm(0,0.001) }
for (j in 1:6) { a[j] ~ dnorm(0,0.001)
                 b[j] ~ dnorm(0,0.001) }
tau ~ dgamma(0.001,0.001)
#Prediction
for (i in 1:n) { yf[i] ~ dnorm(mu[i],tau) }
}

model
{
#Likelihood
for (i in 1:n) {
    y[i] ~ dnorm(mu[i],tau)
    mu[i]<-beta[1]+beta[2]*x[i]
           +a[1]*z2[i]
           +a[2]*z3[i]
               +a[3]*z4[i]
               +a[4]*z5[i]
               +a[5]*z6[i]
               +a[6]*z7[i]
               +b[1]*x[i]*z2[i]
           +b[2]*x[i]*z3[i]
           +b[3]*x[i]*z4[i]
           +b[4]*x[i]*z5[i]
           +b[5]*x[i]*z6[i]
           +b[6]*x[i]*z7[i]
}
#Priors
for (j in 1:2) { beta[j] ~ dnorm(0,0.001) }
for (j in 1:6) { a[j] ~ dnorm(0,0.001)
                 b[j] ~ dnorm(0,0.001) }
tau ~ dgamma(0.001,0.001)
#Prediction
for (i in 1:n) { yf[i] ~ dnorm(mu[i],tau) }
}

model
{
#Likelihood
for (i in 1:n) {
    y[i] ~ dgamma(r,lambda[i])
        lambda[i]<-r/(mu[i])
    mu[i]<-beta[1]+beta[2]*x[i]
               +a[1]*z2[i]
           +a[2]*z3[i]
               +a[3]*z4[i]
               +a[4]*z5[i]
```

```
               +a[5]*z6[i]
               +a[6]*z7[i]
               +b[1]*x[i]*z2[i]
           +b[2]*x[i]*z3[i]
           +b[3]*x[i]*z4[i]
           +b[4]*x[i]*z5[i]
           +b[5]*x[i]*z6[i]
           +b[6]*x[i]*z7[i]
}
#Priors
for (j in 1:2) { beta[j] ~ dnorm(0,0.001) }
for (j in 1:6) { a[j] ~ dnorm(0,0.001)
                 b[j] ~ dnorm(0,0.001) }
r ~ dunif(0,100)
#Prediction
for (i in 1:n) { yf[i] ~ dgamma(r,lambda[i]) }
}
```