

```
rm(list = ls(all.names = TRUE))
gc()
```

```
##          used (Mb) gc trigger (Mb) max used (Mb)
## Ncells 401848 21.5      828665 44.3    638942 34.2
## Vcells 730757  5.6      8388608 64.0   1633308 12.5
```

Nuestra base de datos es sobre el diagnostico de la enfermedad de Párkinson. Está base de datos esta compuesta por grabaciones de voz de 31 personas, 23 de ellas padecen la enfermedad de Párkinson y 8 no la padecen. Cada persona cuenta con 6 grabaciones de voz, en total tenemos 195 grabaciones de voz. El objetivo es hacer una regresion logística mediante estadística bayesiana.

Para fines de este proyecto trataremos cada grabación de voz como una persona, es decir, en la base de datos contamos con 195 grabaciones de voz.

Veamos las variables con las que contamos.

Podemos ver que contamos con la siguientes variables.

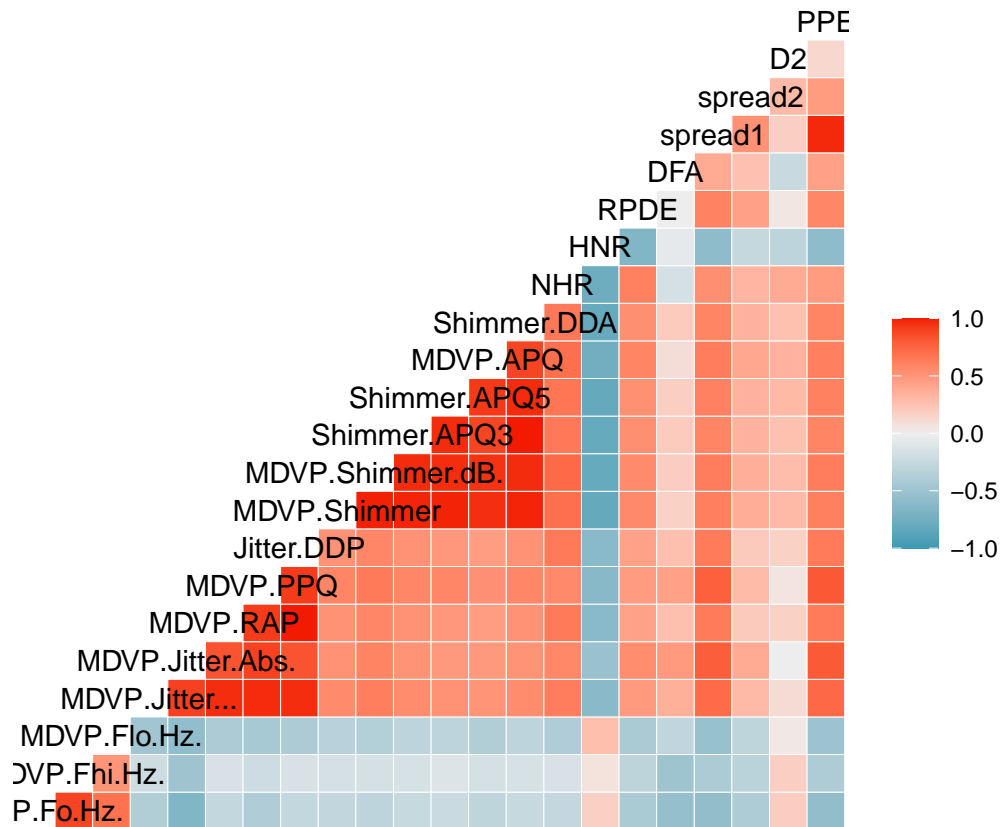
- name - son las grabaciones que tenemos por cada persona (6 por persona)
- MDVP:Fo(Hz) - Nota mque es más comoda para el ser humano.
- MDVP:Fhi(Hz) - Máxima nota que en promedio puede hacer el ser humano.
- MDVP:Flo(Hz) - Minima nota que en promedio puede hacer el ser humano.
- MDVP:Jitter(%), MDVP:Jitter(Abs) - Representa en porcentajes o en términos relativos la variación del período
- MDVP:RAP, MDVP:PPQ - Para minimizar los errores acusticos, se promedian 3 o 5 períodos consecutivos respectivamente.
- Jitter:DDP - Denota la diferencia absoluta promedio de las diferencias entre los ciclos de fluctuación.
- MDVP:Shimmer, MDVP:Shimmer(dB),Shimmer:APQ3,Shimmer:APQ5,MDVP:APQ,Shimmer:DDA - Son medidas analogas a Jitter, aunque estan medidas por la amplitud y no por la frecuencia.
- NHR,HNR - Es la relación de ruido a armónicos y la relación de armónicos a ruido de las señales acústicas.
- status - Pacientes que padecen Párkinson (1), (0) - No padecen Párkison.
- RPDE,D2 - Dos medidas no lineales, la entropía de densidad del período de recurrencia y la dimensión de correlación respectivamente.
- DFA - Análisis de fluctuación sin tendencia
- PPE - Entropía del período de tono.
- spread1,spread2 - Medidas no lineales de variación de frecuencia fundamental.

Cabe señalar que MDVP es un programa usado para medir distintos parámetros de grabaciones de sonidos, se ocupan para la medicina y tienen rangos de valores normales

Análisis descriptivo.

Vamos a ver si de alguna manera podemos seleccionar variables que no tengan tanta importancia en nuestra base de datos.

Debido a que la mayoría de la base de datos, tenemos más de la mitad de los pacientes que padecen la enfermedad de Párkinson, y con fines para un mejor clasificación, seleccionaremos proporciones cercanas del número de pacientes enfermos y paciente no enfermos.



```
## 0 1
## 25 99
```

En la grafica de correlacion, vemos que las variables MDVP.Fo.Hz, MDVP.Fhi.Hz, MDVP.Flo.Hz y DFA. Además notamos que hay 25 personas que no padecen Párkinson y 99 que si la padecen.

Primero haremos nuestro modelo logit con herramienta de estadística frecuentista y después lo haremos con herramienta bayesiana.

Selección de variables

Hagamos nuestra selección de variables mediante el método por pasos.

Este método compara el criterio de optimización al agregar (forward), quitar (backward), o ambos (both) una variable al modelo que se analiza.

Como base, usaremos el modelo saturado y el modelo que solamente tiene el intercept.

Probemos con AIC y forward

```
modelStepfAIC=step(glm(status~1, family = binomial(link="logit"), data=df), direction = c("forward"), s
summary(modelStepfAIC)
```

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

```
## glm(formula = status ~ spread1 + D2 + MDVP.Flo.Hz. + NHR + HNR +
##     DFA + MDVP.Jitter.Abs., family = binomial(link = "logit"),
##     data = df)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.0492   0.0000   0.0025   0.2043   1.7242
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -6.035e+01  2.597e+01  -2.324  0.02014 *
## spread1        2.958e+00  1.257e+00   2.354  0.01857 *
## D2             8.217e+00  3.016e+00   2.725  0.00644 **
## MDVP.Flo.Hz.   -6.037e-03  1.238e-02  -0.488  0.62581
## NHR            1.502e+03  5.604e+02   2.681  0.00735 **
## HNR            1.415e+00  5.622e-01   2.517  0.01185 *
## DFA            3.085e+01  1.425e+01   2.165  0.03037 *
## MDVP.Jitter.Abs. -1.213e+05  6.786e+04  -1.788  0.07373 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 124.652  on 123  degrees of freedom
## Residual deviance:  45.966  on 116  degrees of freedom
## AIC: 61.966
##
## Number of Fisher Scoring iterations: 10
```

Podemos notar que la selección de variables sugiere que juguemos con:

spread1, D2, MDVP.Flo.Hz., NHR, HNR, DFA, MDVP.Jitter.Abs.

Ahora intentemos con BIC forward

```
##
## Call:
## glm(formula = status ~ spread1 + D2 + MDVP.Flo.Hz., family = binomial(link = "logit"),
##     data = df)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.37986   0.02113   0.13102   0.39140   1.87670
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   4.069367  4.394957   0.926 0.354489
## spread1        2.213994  0.625955   3.537 0.000405 ***
## D2             6.438111  2.033508   3.166 0.001545 **
## MDVP.Flo.Hz.  -0.022376  0.009483  -2.360 0.018297 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 124.652  on 123  degrees of freedom
```

```
## Residual deviance: 62.987 on 120 degrees of freedom
## AIC: 70.987
##
## Number of Fisher Scoring iterations: 7
```

Aquí nos sugiere que juguemos con 3 variables: *spread1*, *MDVP.Flo.Hz.*, *D2*

Ahora intentemos con el BIC backward

```
##
## Call:
## glm(formula = status ~ MDVP.Flo.Hz. + MDVP.Jitter... + MDVP.RAP +
##      MDVP.PPQ + MDVP.Shimmer.dB. + Shimmer.APQ5 + NHR + spread2 +
##      D2, family = binomial(link = "logit"), data = df)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.49145   0.00002   0.00954   0.26219   2.04343
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -1.566e+01  6.462e+00  -2.424  0.01537 *
## MDVP.Flo.Hz.    -4.998e-02  1.718e-02  -2.909  0.00363 **
## MDVP.Jitter...  -7.373e+03  2.649e+03  -2.784  0.00537 **
## MDVP.RAP        4.994e+03  2.200e+03   2.270  0.02320 *
## MDVP.PPQ        7.200e+03  3.043e+03   2.366  0.01798 *
## MDVP.Shimmer.dB. 1.119e+02  5.604e+01   1.997  0.04588 *
## Shimmer.APQ5    -1.648e+03  7.772e+02  -2.120  0.03401 *
## NHR             4.222e+02  2.256e+02   1.872  0.06123 .
## spread2         2.694e+01  1.060e+01   2.542  0.01103 *
## D2              8.658e+00  3.553e+00   2.437  0.01483 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 124.652 on 123 degrees of freedom
## Residual deviance:  44.899 on 114 degrees of freedom
## AIC: 64.899
##
## Number of Fisher Scoring iterations: 9
```

el modelo plantea más variables a consederar, las cuales son : *MDVP.Flo.Hz.*,*MDVP.Jitter...*,*MDVP.RAP*,
MDVP.PPQ, *MDVP.Shimmer.dB.*,*Shimmer.APQ5*,*NHR*,*spread2*, *D2*

Poder predictivo

Vamos probar estos métodos, y ver cual nos conviene basandonos en el porcentaje de error, esto será con ayuda de validación cruzada.

La razón es porque nuestra base de datos es pequeña, y consiste de la siguiente manera.

1. Para $k=1, \dots, K$. Se elimina la k -ésima parte y el ajuste del modelo se realiza con las $k-1$ partes restante cada elemento en la k -ésima parte se clasifica con el modelo ajustando.
2. Usando la clasificación obtenida se calcula un porcentaje de error.
3. Este procedimiento se repite $M=100$ o 1000 veces. Al final, se promedian los M porcentajes de error de clasificación

Usemos la validación cruzada para nuestros posibles modelos hechos en la selección de variables.

Veamos el porcentaje de error.

```
cv_StepfAIC <- colSums(cv_tmp)/ndatos  
mean(cv_StepfAIC)
```

```
## [1] 0.1123387
```

```
cv_StepfAIC1 <- colSums(cv_tmp1)/sum(df$status==1)  
mean(cv_StepfAIC1)
```

```
## [1] 0.05808081
```

```
cv_StepfAIC0 <- colSums(cv_tmp0)/sum(df$status==0)  
mean(cv_StepfAIC0)
```

```
## [1] 0.3272
```

Notamos que el porcentaje de error global es del 11% mientras que para el grupo con Párkison el porcentaje de error es pequeño (5%), aunque para los que no padecen de esta enfermedad el porcentaje es muy grande.

Intentemos ver lo que pasa con nuestro siguiente modelo *BIC forward*

```
cv_StepfBIC <- colSums(cv_tmp)/ndatos  
mean(cv_StepfBIC)
```

```
## [1] 0.1223387
```

```
cv_StepfBIC1 <- colSums(cv_tmp1)/sum(df$status==1)  
mean(cv_StepfBIC1)
```

```
## [1] 0.05090909
```

```
cv_StepfBIC0 <- colSums(cv_tmp0)/sum(df$status==0)  
mean(cv_StepfBIC0)
```

```
## [1] 0.4052
```

Notamos que el error global es un poco más alto, el error para los pacientes con Párkinson es parecido al primer modelo, mientras que aumenta el error para los que no padecen esta enfermedad.

Por último intentemos con el modelo *BIC barckward*

Con su respectivo porcentaje de error.

```
cv_StepbBIC <- colSums(cv_tmp)/ndatos  
mean(cv_StepbBIC)
```

```
## [1] 0.1325806
```

```
cv_StepbBIC1 <- colSums(cv_tmp1)/sum(df$status==1)  
mean(cv_StepbBIC1)
```

```
## [1] 0.07060606
```

```
cv_StepbBIC0 <- colSums(cv_tmp0)/sum(df$status==0)  
mean(cv_StepbBIC0)
```

```
## [1] 0.378
```

Sigue en aumento el error global, y ahora el error para los que padecen de Párkinson es mayor, aunque baja el error para los que no padecen de esta enfermedad.

Ahora comparemos los modelos y seleccionemos el que tenga el menor error global.

```
c(mean(cv_StepfAIC),mean(cv_StepfBIC),mean(cv_StepbBIC))
```

```
## [1] 0.1123387 0.1223387 0.1325806
```

El modelo que elegimos es el de AIC, ya que tiene un porcentaje de error del 11%, además de que el porcentaje de que nos equivoquemos respecto a los pacientes que padecen de Párkinson es del 5%.

Con Bayesiana

Veamos el contraste con estadística Bayesiana.

Primero ajustemos nuestra matriz de diseño, escalando los datos.

```
library(rjags)
D1 <- read_csv("C:/Users/Lenovo-L430/Downloads/D1.csv") # Pacientes con Párkinson
D0 <- read_csv("C:/Users/Lenovo-L430/Downloads/D0.csv") # Pacientes sin Párkinson

D1<-subset(D1, select = -c(X1, name)) #eliminamos columnas que no nos interesan.
D0<-subset(D0, select = -c(X1, name))

datos<-rbind(D1,D0)
datos$status<-as.numeric(datos$status)

X=scale(datos[, -17], center = TRUE, scale = TRUE)
head(X)
```

```
##      MDVP.Fo.Hz. MDVP.Fhi.Hz. MDVP.Flo.Hz. MDVP.Jitter... MDVP.Jitter.Abs.
## [1,] -0.6991483 -0.2873975 -0.99597807 1.5237523 1.7418976
## [2,] -0.6924815 -0.7417254 -0.06725673 -0.7169020 -0.2881334
## [3,] -1.0060601 -1.2718426 -0.31224625 -0.9305342 -0.2881334
## [4,] -1.2873233 -0.8589655 -0.60518422 0.3661637 1.2343899
## [5,] -1.3036628 -1.1299813 -0.61749783 0.2717681 1.2343899
## [6,] -1.3800756 -1.2258391 -0.73256478 0.3115136 1.2343899
##      MDVP.RAP MDVP.PPQ Jitter.DDP MDVP.Shimmer MDVP.Shimmer.dB.
## [1,] 1.0449832 2.5215531 1.0424704 1.8856865 2.0709815
## [2,] -0.7742183 -0.5483961 -0.7719820 -0.7410393 -0.8043772
## [3,] -0.8672938 -0.7228251 -0.8707469 -0.7799748 -0.8646994
## [4,] 0.3934552 0.5853919 0.3962657 -0.2804601 -0.2916384
## [5,] 0.1819202 0.5853919 0.1789829 0.4270275 0.3517985
## [6,] 0.2919184 0.6202777 0.2946790 0.3453578 0.2914763
##      Shimmer.APQ3 Shimmer.APQ5 MDVP.APQ Shimmer.DDA NHR HNR
## [1,] 1.5903012 2.6789046 1.1565033 1.5896449 1.08716035 -0.6329432
## [2,] -0.8265188 -0.7499944 -0.6468888 -0.8265323 -0.65843268 0.6078608
## [3,] -0.7403883 -0.7359223 -0.7526433 -0.7404044 -0.94464899 1.2538646
## [4,] -0.3200718 -0.2183822 -0.2394558 -0.3206740 -0.15456137 -0.3820773
## [5,] 0.3138482 0.4820951 0.5698443 0.3143764 -0.20679857 -0.3659755
## [6,] 0.2845638 0.3507556 0.3138071 0.2845187 -0.07729385 -0.5102476
##      RPDE DFA spread1 spread2 D2 PPE
## [1,] -0.6808802 1.7451305 1.3531724 0.8209206 0.06105229 1.4759942
## [2,] 1.0472611 0.7521167 0.3248712 0.6948244 -1.44783606 0.3553017
## [3,] 1.4417865 0.7356224 -0.3428258 -0.3649717 -0.73872877 -0.3826384
## [4,] 1.2332830 0.9359794 0.4947049 1.6991095 0.13222719 0.6599263
## [5,] 0.5800565 1.4186990 1.1042039 1.6737039 0.50477641 1.2716505
## [6,] 0.9266534 1.3033749 1.1688662 2.2120490 1.21312699 1.3418917
```

```
apply(X, 2, sd)
```

```
##      MDVP.Fo.Hz.      MDVP.Fhi.Hz.      MDVP.Flo.Hz.      MDVP.Jitter...
##              1              1              1              1
## MDVP.Jitter.Abs.      MDVP.RAP      MDVP.PPQ      Jitter.DDP
##              1              1              1              1
##      MDVP.Shimmer MDVP.Shimmer.dB.      Shimmer.APQ3      Shimmer.APQ5
##              1              1              1              1
##      MDVP.APQ      Shimmer.DDA      NHR      HNR
##              1              1              1              1
##      RPDE      DFA      spread1      spread2
##              1              1              1              1
##      D2      PPE
##              1              1
```

Selección de variables

Haremos una selección de variables, basandonos en nuestro modelo saturado, y respecto a los histogramas tomaremos una decisión para nuestro modelo.

```
data <- list(y =datos$status,
             x1=X[, "MDVP.Fo.Hz."],
             x2=X[, "MDVP.Fhi.Hz."],
             x3=X[, "MDVP.Flo.Hz."],
             x4=X[, "MDVP.Jitter..."],
             x5=X[, "MDVP.Jitter.Abs."],
             x6=X[, "MDVP.RAP"],
             x7=X[, "MDVP.PPQ"],
             x8=X[, "Jitter.DDP"],
             x9=X[, "MDVP.Shimmer"],
             x10=X[, "MDVP.Shimmer.dB."],
             x11=X[, "Shimmer.APQ3"],
             x12=X[, "Shimmer.APQ5"],
             x13=X[, "MDVP.APQ"],
             x14=X[, "Shimmer.DDA"],
             x15=X[, "NHR"],
             x16=X[, "HNR"],
             x17=X[, "RPDE"],
             x18=X[, "DFA"],
             x19=X[, "spread1"],
             x20=X[, "spread2"],
             x21=X[, "D2"],
             x22=X[, "PPE"],
             n=length(datos$status)
)

param <- c("betas", "alpha")

inits <- function(){  list(
  "betas" = rnorm(22,0,1)
) }
```

Ajustando el modelo.

```
setwd("C:/Users/Lenovo-L430/Downloads")
```

```

fit <- jags.model("ddexp.bug", data, inits, n.chains=3)

## Compiling model graph
##   Resolving undeclared variables
##   Allocating nodes
## Graph information:
##   Observed stochastic nodes: 124
##   Unobserved stochastic nodes: 23
##   Total graph size: 5641
##
## Initializing model

update(fit, 5000)
sample <- coda.samples(fit, param, n.iter=10000, thin=1)

```

viazualización.

```

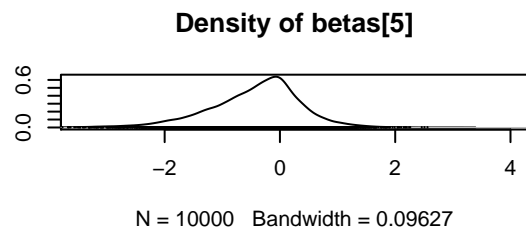
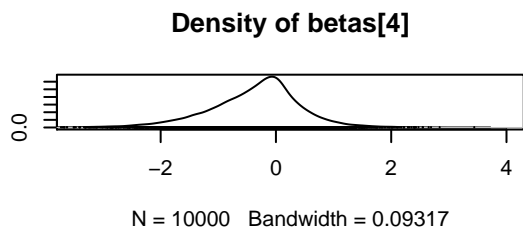
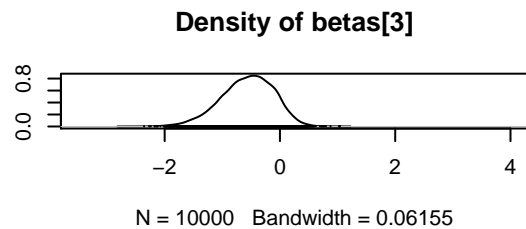
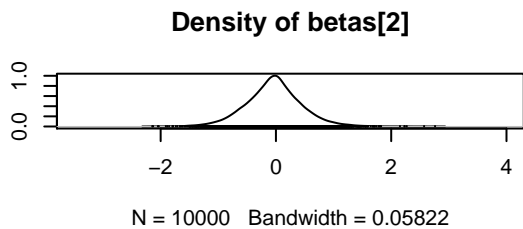
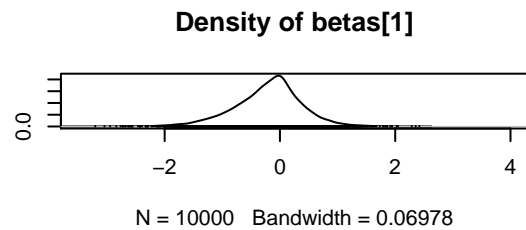
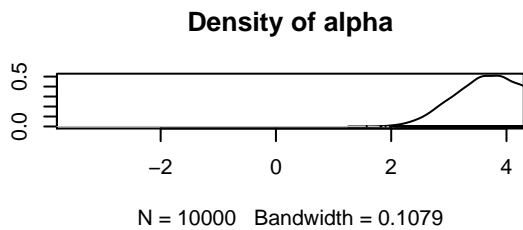
##
## Iterations = 6001:16000
## Thinning interval = 1
## Number of chains = 3
## Sample size per chain = 10000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##           Mean      SD Naive SE Time-series SE
## alpha      3.97415 0.8224 0.004748      0.019046
## betas[1]   -0.20731 0.5903 0.003408      0.010823
## betas[2]   -0.03778 0.4856 0.002804      0.007245
## betas[3]   -0.56662 0.4564 0.002635      0.006279
## betas[4]   -0.36853 0.8104 0.004679      0.017696
## betas[5]   -0.43378 0.8040 0.004642      0.015714
## betas[6]    0.58480 0.9031 0.005214      0.021354
## betas[7]    0.14009 0.7710 0.004451      0.015758
## betas[8]    0.54962 0.8940 0.005161      0.020883
## betas[9]    0.07859 0.8294 0.004788      0.017540
## betas[10]   0.11955 0.8519 0.004918      0.018599
## betas[11]  -0.14628 0.7795 0.004500      0.015685
## betas[12]   0.16371 0.7940 0.004584      0.015634
## betas[13]   0.35553 0.8416 0.004859      0.017062
## betas[14]  -0.18117 0.7888 0.004554      0.015598
## betas[15]   1.55193 1.2270 0.007084      0.034199
## betas[16]   0.31125 0.6369 0.003677      0.011807
## betas[17]  -0.34432 0.4051 0.002339      0.004767
## betas[18]  -0.05155 0.4267 0.002463      0.006147
## betas[19]   0.56572 0.7397 0.004271      0.015341
## betas[20]   0.53865 0.4088 0.002360      0.004934
## betas[21]   1.30797 0.5933 0.003425      0.008433
## betas[22]   0.72920 0.7969 0.004601      0.017272
##
## 2. Quantiles for each variable:
##
##           2.5%      25%      50%      75%  97.5%
## alpha      2.5627  3.40271  3.89817  4.47491 5.8047

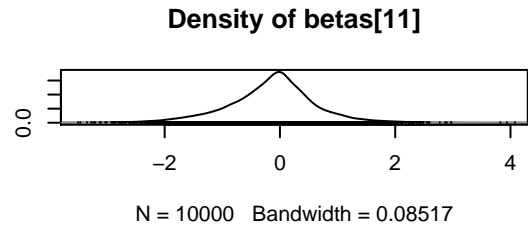
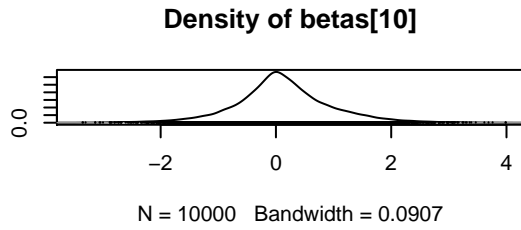
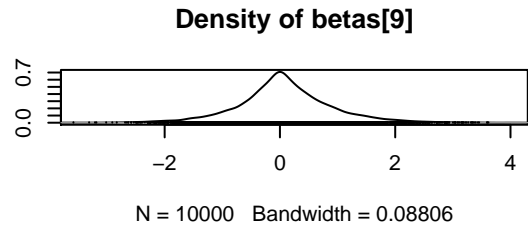
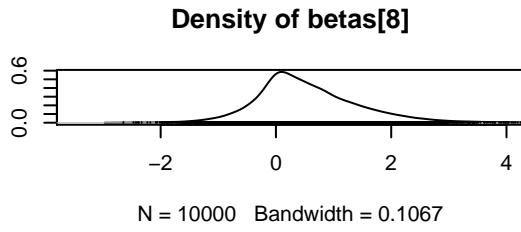
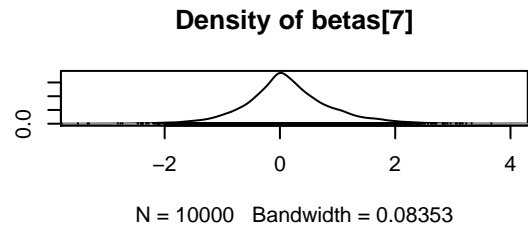
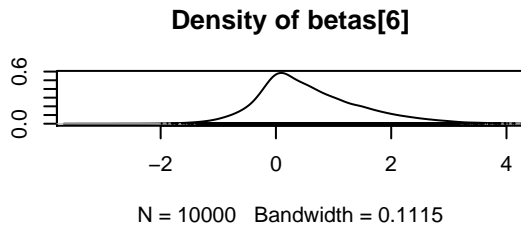
```

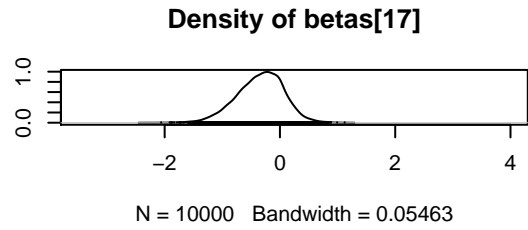
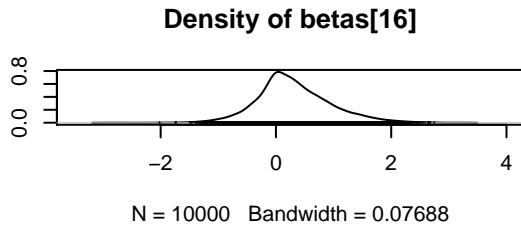
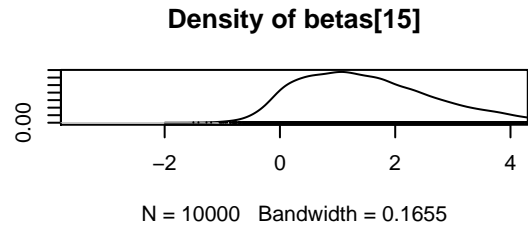
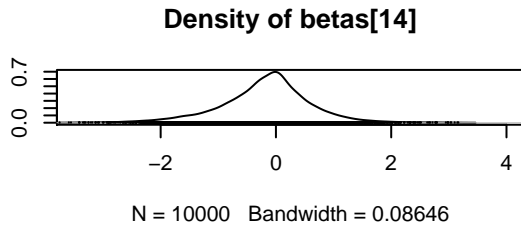
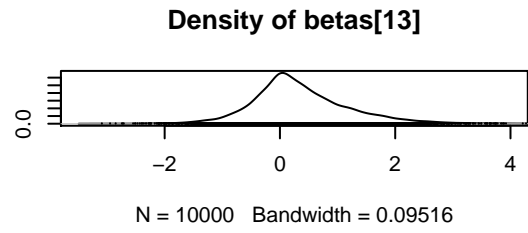
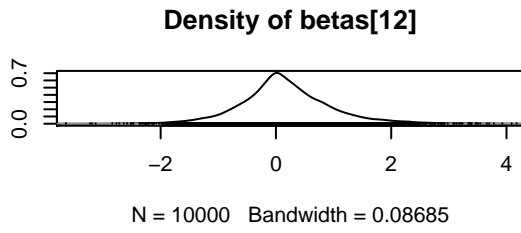


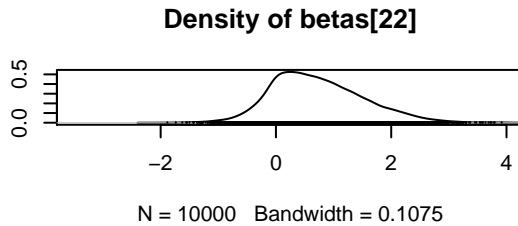
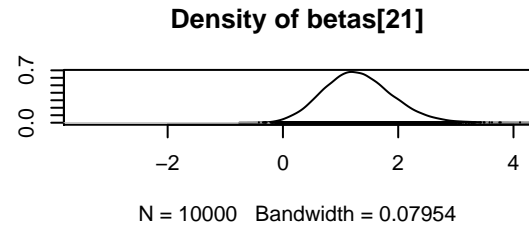
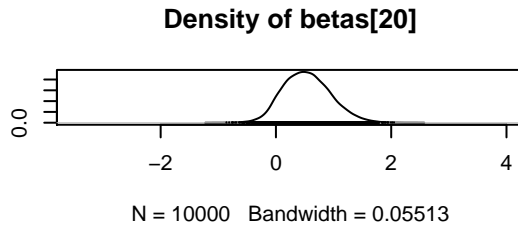
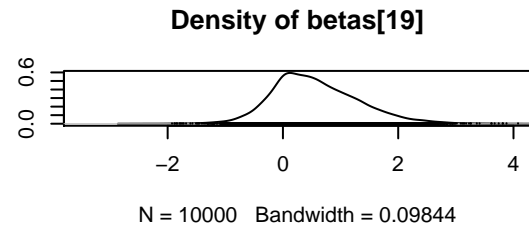
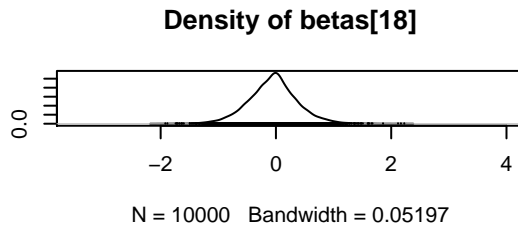
```
## betas[1] -1.4955 -0.54292 -0.15194 0.15045 0.8943
## betas[2] -1.0188 -0.32996 -0.03697 0.24858 0.9686
## betas[3] -1.5405 -0.86371 -0.53606 -0.23533 0.2281
## betas[4] -2.1962 -0.81070 -0.26680 0.11508 1.1028
## betas[5] -2.2537 -0.88208 -0.32207 0.07447 0.9924
## betas[6] -0.8889 -0.02058 0.42481 1.08725 2.6972
## betas[7] -1.3643 -0.29037 0.08653 0.53960 1.8320
## betas[8] -0.9849 -0.02382 0.41491 1.03609 2.6090
## betas[9] -1.6100 -0.36455 0.04223 0.51045 1.8801
## betas[10] -1.6207 -0.34046 0.07548 0.56076 1.9681
## betas[11] -1.8553 -0.55927 -0.09188 0.28700 1.4196
## betas[12] -1.3684 -0.28394 0.10324 0.57899 1.9570
## betas[13] -1.1184 -0.15637 0.23147 0.78916 2.3051
## betas[14] -1.9488 -0.59560 -0.12211 0.26349 1.3445
## betas[15] -0.2791 0.62700 1.38158 2.28548 4.3178
## betas[16] -0.8559 -0.08474 0.23916 0.67912 1.7453
## betas[17] -1.1931 -0.60692 -0.31484 -0.05968 0.3870
## betas[18] -0.9112 -0.31298 -0.04380 0.20342 0.8270
## betas[19] -0.6952 0.04643 0.47798 1.02457 2.2030
## betas[20] -0.1859 0.24348 0.51807 0.80844 1.3959
## betas[21] 0.2340 0.89478 1.27597 1.68513 2.5644
## betas[22] -0.6033 0.14994 0.63655 1.23176 2.4498
```

```
par(mfrow=c(3,2))
densplot(sample[,1:23], xlim = c(-3.5,4.0))
```









Podemos notar que x_3 = MDVP.Flo.Hz. , x_{15} =NHR , x_{17} = RPDE, x_{20} =spread2 , x_{21} =D2 no estan centradas en el cero, por lo que seleccionaremos estas variables.

```
data1 <- list(y =datos$status,
              x1=X[, "MDVP.Flo.Hz."],
              x2=X[, "NHR"],
              x3=X[, "RPDE"],
              x4=X[, "spread2"],
              x5=X[, "D2"],
              n=length(datos$status)
)

param <- c("betas", "alpha")

inits <- function(){ list(
  "betas" = rnorm(5,0,1)
) }

setwd("C:/Users/Lenovo-L430/Downloads")

fit1 <- jags.model("primerbay.bug", data1,inits, n.chains=3)

## Compiling model graph
##   Resolving undeclared variables
##   Allocating nodes
## Graph information:
##   Observed stochastic nodes: 124
```

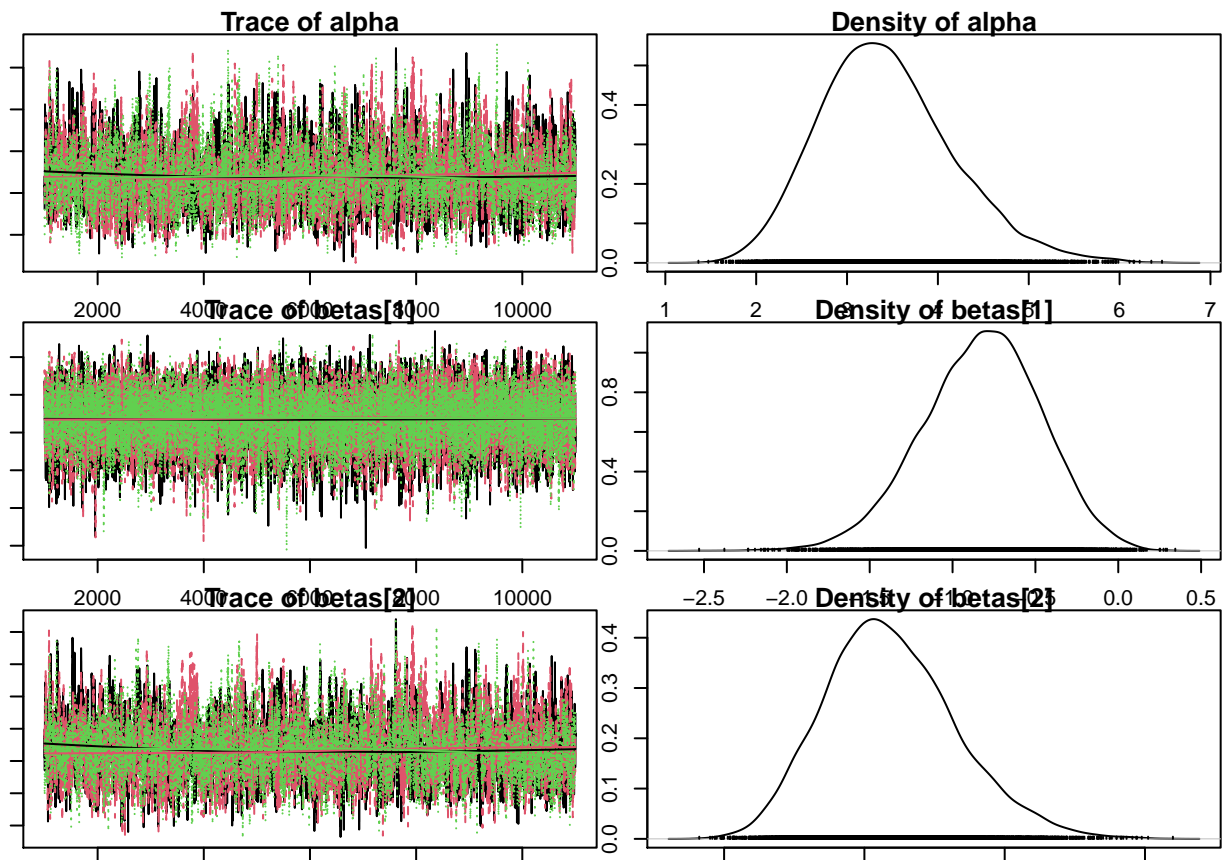
```

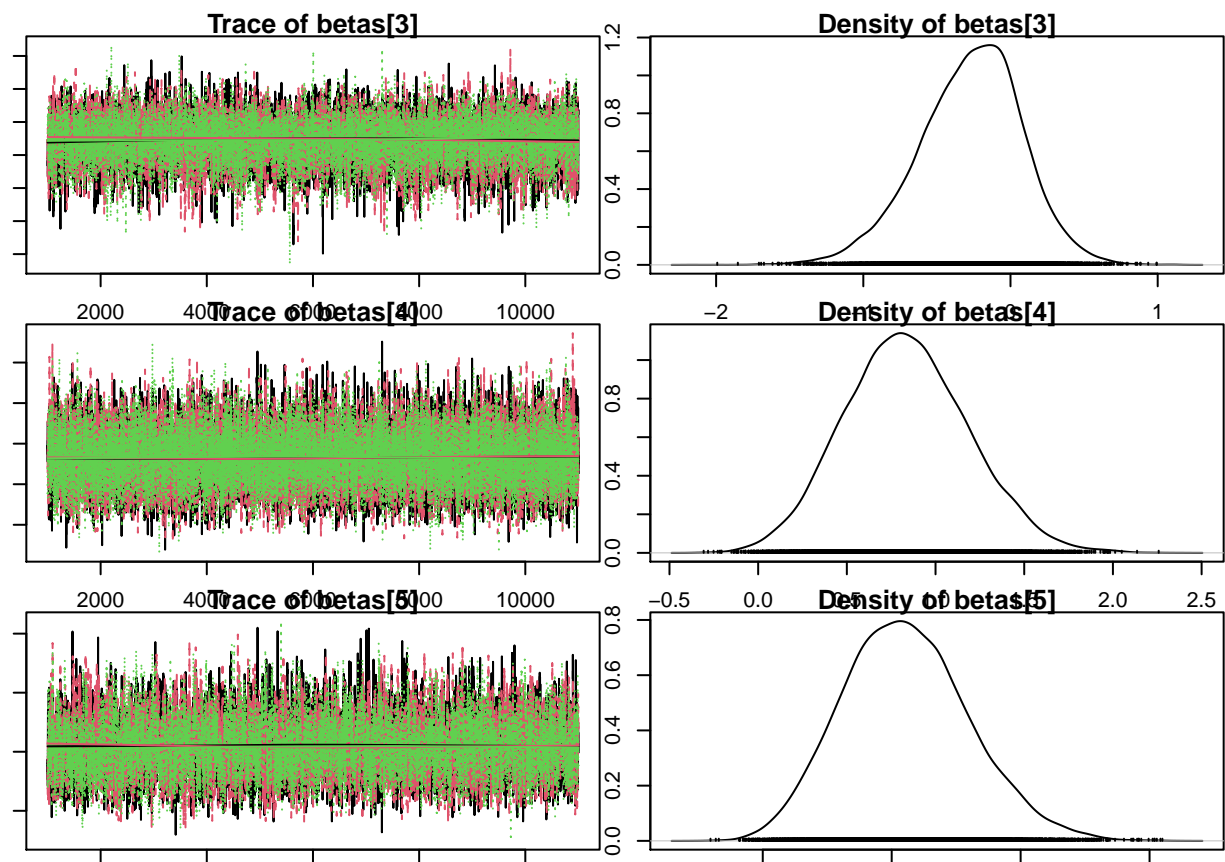
##      Unobserved stochastic nodes: 6
##      Total graph size: 1617
##
## Initializing model
update(fit,5000)
sample1 <- coda.samples(fit1, param, n.iter=10000, thin=1)

par(mar=c(1,1,1,1))
summary(sample1)

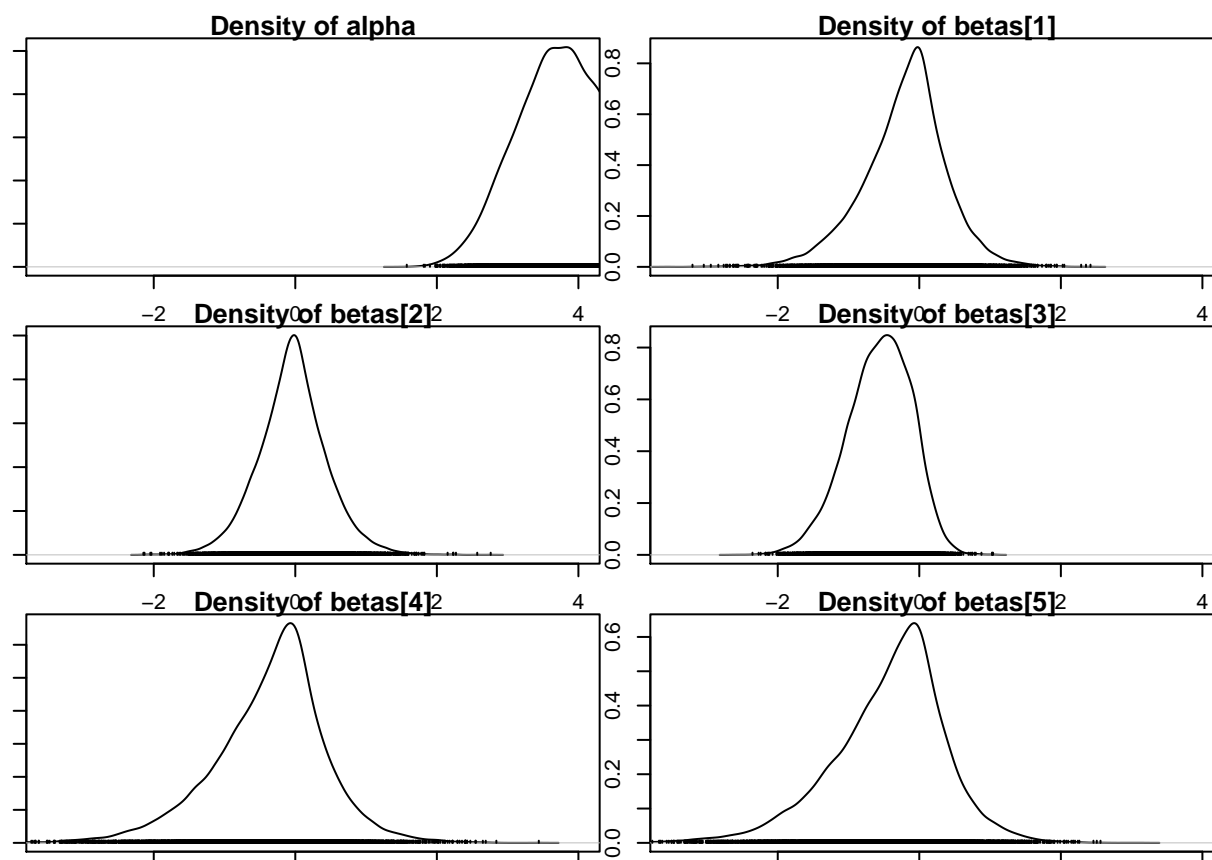
##
## Iterations = 1001:11000
## Thinning interval = 1
## Number of chains = 3
## Sample size per chain = 10000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##           Mean      SD Naive SE Time-series SE
## alpha      3.4317 0.7377 0.004259      0.016629
## betas[1]  -0.8252 0.3609 0.002084      0.004309
## betas[2]   2.3754 0.9402 0.005428      0.020944
## betas[3]  -0.2743 0.3527 0.002036      0.004525
## betas[4]   0.8347 0.3509 0.002026      0.003552
## betas[5]   1.1085 0.4903 0.002831      0.006909
##
## 2. Quantiles for each variable:
##
##           2.5%      25%      50%      75%      97.5%
## alpha      2.1529  2.9115  3.3723  3.88133  5.0646
## betas[1]  -1.5708 -1.0588 -0.8098 -0.57613 -0.1597
## betas[2]   0.7174  1.7141  2.3069  2.96567  4.4226
## betas[3]  -1.0246 -0.4984 -0.2516 -0.03456  0.3734
## betas[4]   0.1769  0.5939  0.8239  1.06567  1.5436
## betas[5]   0.2153  0.7635  1.0871  1.42605  2.1317
plot(sample1)

```





```
par(mfrow=c(3,2))
densplot(sample[,1:6], xlim = c(-3.5,4.0))
```



```
dic0=dic.samples(fit, n.iter = 1e3)
dic0
```

```
## Mean deviance: 62.74
## penalty 10.34
## Penalized deviance: 73.08
```

```
dic1=dic.samples(fit1, n.iter = 1e3)
dic1
```

```
## Mean deviance: 67.63
## penalty 5.416
## Penalized deviance: 73.05
```