Tp3

Trabajo práctico 3

Primero cargamos todas las librerías que vamos a usar para los dos ejercicios de la práctica.

```
library(ISLR)
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loading required package: foreach
```

```
## Loaded glmnet 2.0-16
```

```
library(ggplot2)
library(glmnet)
library(faraway)
library(MASS)
library(GGally)
```

```
##
## Attaching package: 'GGally'
```

```
## The following object is masked from 'package:faraway':
##
##     happy
```

Ejercicio 1

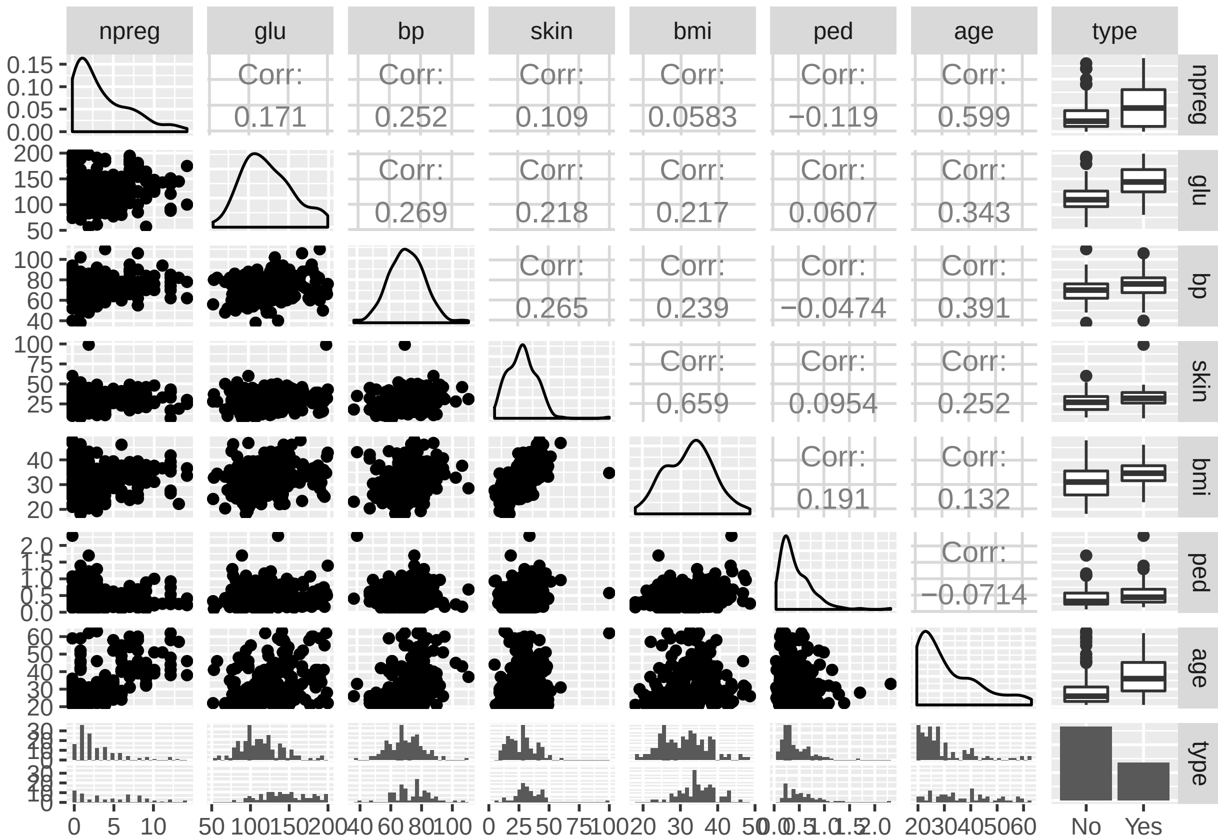
Utilizamos para el análisis de este ejercicio los datos de Pima, del paquete MASS

```
datos<-Pima.tr
```

a) Realizamos un gráfico ggpairs para poder realizar un primer análisis de las variables en estudio

```
ggpairs(datos)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Por los boxplot de la columna correspondiente a la variable *type*, parecería que las variables *npreg*, *glu*, *bmi* y *age* discriminan mas los diferentes tipos.

b) Realizamos un ajuste a un modelo de regresión logística para poder predecir diabetes utilizando todas las variables

```
ajuste<-glm(type~., data=datos, family=binomial)
summary(ajuste)
```

```
##
## Call:
## glm(formula = type ~ ., family = binomial, data = datos)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9830  -0.6773  -0.3681   0.6439   2.3154
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -9.773062   1.770386  -5.520 3.38e-08 ***
## npreg        0.103183   0.064694   1.595  0.11073
## glu          0.032117   0.006787   4.732 2.22e-06 ***
## bp          -0.004768   0.018541  -0.257  0.79707
## skin        -0.001917   0.022500  -0.085  0.93211
## bmi          0.083624   0.042827   1.953  0.05087 .
## ped          1.820410   0.665514   2.735  0.00623 **
## age          0.041184   0.022091   1.864  0.06228 .
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 256.41  on 199  degrees of freedom
## Residual deviance: 178.39  on 192  degrees of freedom
## AIC: 194.39
##
## Number of Fisher Scoring iterations: 5
```

Las estimaciones de los coeficientes son:

```
summary(ajuste)$coef[,1]
```

```
##  (Intercept)        npreg          glu           bp         skin
## -9.773061533  0.103183427  0.032116823 -0.004767542 -0.001916632
##          bmi          ped          age
##  0.083623912  1.820410367  0.041183529
```

Resultan significativos los coeficientes correspondientes a las variables: *glu*, *bmi*, *ped* y *age*.

c) A partir del ajuste realizado en el ítem anterior, y utilizando los datos de la salida, construimos intervalos de confianza de nivel 0.95 para cada uno de los coeficientes del modelo.

```
IC<-matrix(0,ncol=2,nrow=8)

for(i in 1:8)
{
 IC[i,]<-c(summary(ajuste)$coef[i,1]-qnorm(0.975)*summary(ajuste)$coef[i,2],
      summary(ajuste)$coef[i,1]+qnorm(0.975)*summary(ajuste)$coef[i,2])
}

colnames(IC)<-c("LI","LS")
rownames(IC)<-c("beta 0:", "beta 1:", "beta 2:", "beta 3:",
                "beta 4:", "beta 5:", "beta 6:", "beta 7:")

IC
```

```
##                    LI          LS
## beta 0: -1.324295e+01 -6.30316870
## beta 1: -2.361478e-02  0.22998164
## beta 2:  1.881396e-02  0.04541969
## beta 3: -4.110673e-02  0.03157164
## beta 4: -4.601492e-02  0.04218166
## beta 5: -3.152456e-04  0.16756307
## beta 6:  5.160273e-01  3.12479340
## beta 7: -2.113992e-03  0.08448105
```

d) Construimos la tabla de confusión entre la clasificación observada y la clasificación predicha por nuestro modelo:

```
#predecimos la probabilidad de que tenga diabetes con el modelo
prediccion<-predict(ajuste, type="response")

#clasificamos dependiendo si la probabilidad de que tenga diabetes es mayor a 0.5
predichos<-ifelse(prediccion>0.5,"Yes","No")

#armamos la tabla
```

```r
observados<-datos$type
confusion<-table(predichos,observados)
confusion
```

```
##          observados
## predichos  No Yes
##       No  116  29
##       Yes  16  39
```

Ahora, a partir de la tabla calculamos el porcentaje de aciertos (precisión de predicción)

```r
aciertos<-(confusion[1,1]+confusion[2,2])/length(predichos)
aciertos*100
```

```
## [1] 77.5
```

e) Repetimos el ítem anterior sobre la muestra de testeo

```r
test<-Pima.te

prediccion<-predict(ajuste,newdata=test, type="response")

predichos<-ifelse(prediccion>0.5,"Yes","No")

observados<-test$type

confusion<-table(predichos,observados)
confusion
```

```
##          observados
## predichos  No Yes
##       No  200  43
##       Yes  23  66
```

```r
aciertos<-(confusion[1,1]+confusion[2,2])/length(predichos)
aciertos*100
```

```
## [1] 80.12048
```

f) Estimamos la probabilidad de que una nueva observación caiga en la categoría de diabetes, utilizando la función *predict*

```r
nuevadata<-c( 2 ,100,  70, 20, 26,  0.24 , 30)
ND<-rbind(test[1:7],nuevadata)[333,] #para que la nueva observación concerve los
# nombres de las variables

probabilidad_diabetes<-predict(ajuste,newdata=ND, type="response")
probabilidad_diabetes
```

```
##        333
## 0.05312865
```

g) Con el ajuste realizado, buscamos analizar cómo cambian los odds de una mujer cuando la glucosa aumenta en 10 unidades y todas los demás valores permanecen constantes.

Primero analizamos cuanto vale la diferencia de log odds en el caso en que la glucosa aumenta en 10 unidades y todas las demás variables permanecen constantes (Desarrollo en el pizarrón). La diferencia de log odds será $10\beta_{glu}$

El valor estimado para este caso será:

```r
dif_log_odd<-summary(ajuste)$coef[3,1]*10
```

En segundo lugar analizamos cuanto vale el cociente de los odds en el caso en que la glucosa aumenta en 10 unidades y todas las demás variables permanecen constantes (Desarrollo en el pizarrón). La diferencia de log odds será $e^{10\beta_{glu}}$

El valor estimado para este caso será:

```r
cociente_odd<-exp(summary(ajuste)$coef[3,1]*10)
```

Ahora buscamos un intervalo de confianza de nivel aproximado 0.95 para dichos odds. Para esto necesitamos conocer las varianzas estimadas de nuestros estimadores $\hat{\beta}$. Podemos conseguirlos de la salida de R, pero también vamos a calcularlos.

```r
#de la salida de R
sigma_beta_sombrero<-summary(ajuste)$coef[,2]

#haciendo la cuenta a ver si da lo mismo
X<-model.matrix(ajuste)
pi<-predict(ajuste,type="response")
n<-nrow(datos)
W_sombrero<-diag(pi*(1-pi))

sigma_betas<-sqrt(diag(solve(t(X)%*%W_sombrero%*%X)))
```

Los valores de W también podíamos conseguirlos de la salida

```r
W<-diag(ajuste$weights) #es lo mismo que con la cuenta
```

Entonces, calculamos los dos intervalos pedidos:

```r
ic_beta<-c(exp(summary(ajuste)$coef[3,1]-qnorm(0.975)*exp(summary(ajuste)$coef[3,2])),
      exp(summary(ajuste)$coef[3,1]+qnorm(0.975)*exp(summary(ajuste)$coef[3,2])))

ic_diferencia<-10*ic_beta
ic_cociente<-exp(ic_diferencia)
```

h) Creamos la función que nos permite calcular un intervalo de confianza para la probabilidad de éxito en un punto x_0:

```r
intervalo_probabilidad<-function(salida,alfa,x0) #importante respetar el orden en x0
{
  d<-as.matrix(c(1,x0))
  psi<-t(d)%*%(summary(salida)$coef[,1])
  X<-model.matrix(salida)
  W<-diag(salida$weights)
  var<-solve(t(X)%*%W%*%X)
  e<-qnorm(1-alfa/2)*sqrt(t(d)%*%var%*%(d))
  ic<-c(1/(1+exp(-psi+e)),1/(1+exp(-psi-e)))
  return(ic)
}
```

i) Utilizando la función creada en el ítem anterior, construimos un intervalo para la probabilidad de que un individuo con los datos del ítem f) tenga diabetes.

```r
intervalo_probabilidad(ajuste,0.05,nuevadata)
```

```
## [1] 0.02327956 0.11667826
```

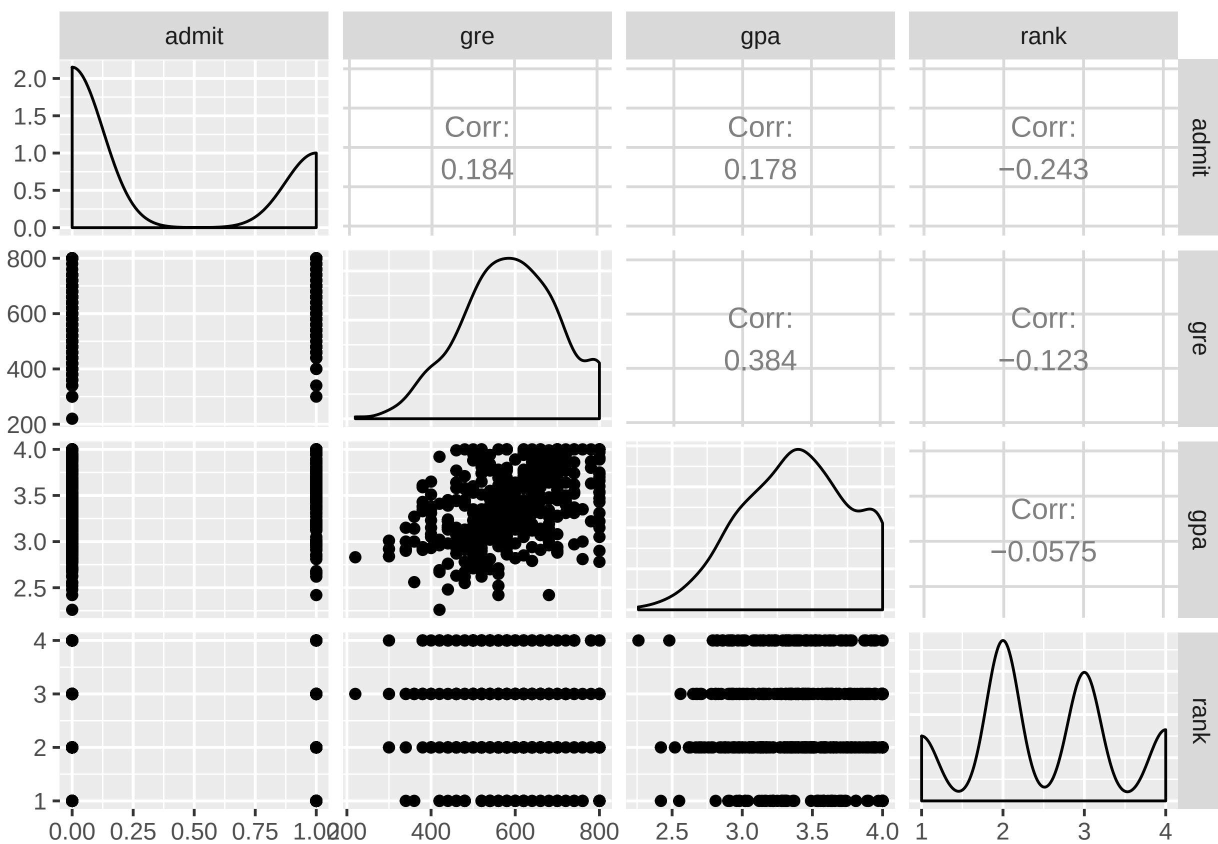
```r
#OTRA FORMA
# intervalo de confianza usando el método delta

intervalo_probabilidad_delta<-function(salida,alfa,x0, pi)
{
  d<-as.matrix(c(1,x0))
  psi<-t(d)%*%(summary(salida)$coef[,1])
  X<-model.matrix(salida)
  W<-diag(salida$weights)
  var<-solve(t(X)%*%W%*%X)
  e<-qnorm(1-alfa/2)*sqrt(t(d)%*%var%*%(d))*pi*(1-pi)
  ic<-c(pi-e,pi+e)
  return(ic)
}

intervalo_probabilidad_delta(ajuste,0.05,nuevadata,probabilidad_diabetes)
```

```
## [1] 0.01005778 0.09619952
```

Ejercicio 2

a) Cargamos los datos y hacemos un primer análisis exploratorio de las variables:

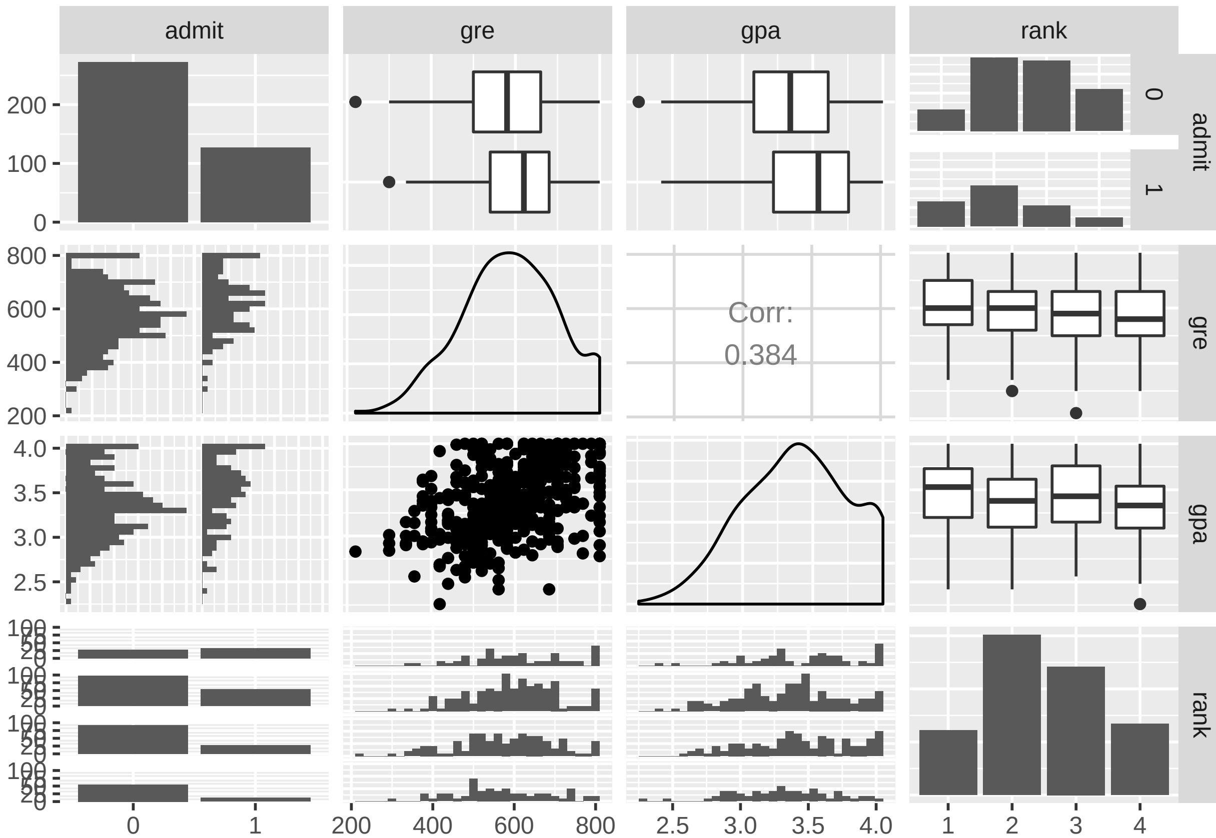
```r
datos<-read.csv("binary.csv")
ggpairs(datos)
```



Como las variables *rank* y *admit* son variables categóricas, podremos analizar mejor los datos si las transformamos a factores

```r
datos$rank<-factor(datos$rank)
datos$admit<-factor(datos$admit)

ggpairs(datos) #se entiende un poco mejor
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



b) Realizamos el ajuste a un logístico usando todas las variables explicativas

```r
ajuste<-glm(admit~., data=datos, family=binomial)
summary(ajuste)
```

```
##
## Call:
## glm(formula = admit ~ ., family = binomial, data = datos)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6268  -0.8662  -0.6388   1.1490   2.0790
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.989979   1.139951  -3.500 0.000465 ***
## gre          0.002264   0.001094   2.070 0.038465 *
```

```
## gpa          0.804038   0.331819   2.423 0.015388 *
## rank2       -0.675443   0.316490  -2.134 0.032829 *
## rank3       -1.340204   0.345306  -3.881 0.000104 ***
## rank4       -1.551464   0.417832  -3.713 0.000205 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 499.98  on 399  degrees of freedom
## Residual deviance: 458.52  on 394  degrees of freedom
## AIC: 470.52
##
## Number of Fisher Scoring iterations: 4
```

Cuando la variable *GRE* aumenta en una unidad, el log odds de la admisión aumenta en $\hat{\beta}_{gre}$. Observando la salida del ajuste, este valor es 0.00226.

En cuanto a la interpretación de las estimaciones de los coeficientes relacionados con la variable rank, si observamos la matriz de diseño, o la salida del ajuste, podemos ver que el modelo toma como nivel basal al caso en que el rango es 1, es decir, si agregamos variables dummies el modelo sería el siguiente: $logodds = \beta_0 + \beta_1 GRE + \beta_2 GPA + \beta_3 D_2 + \beta_4 D_3 + \beta_5 D_4$

Donde las variables D_i valen 1 cuando la variable *rank* vale i. Entonces, cuando la variable *rank* vale 1, todas las dummies valen 0, y el modelo será $logodds = \beta_0 + \beta_1 GRE + \beta_2 GPA$

Por lo tanto, β_3 será el cambio en el log odds cuando paso de rank 1 a rank 2. Lo mismo para los otros coeficientes referidos a la variable *rank*.

c) Como se vió en la teoría, $\pi = \frac{1}{1+e^{-x^t\beta}}$, por lo tanto, una estimación de la probabilidad será $\hat{\pi} = \frac{1}{1+e^{-x^t\hat{\beta}}}$.

Buscamos $\hat{\pi}$ para los distintos niveles de la variable *rank* cuando las otras variables toman como valor la media muestral.

```
m_gre<-mean(datos$gre)
m_gpa<-mean(datos$gpa)

beta_sombrero<-ajuste$coefficients

r1<-c(1,m_gre,m_gpa,0,0,0)
r2<-c(1,m_gre,m_gpa,1,0,0)
r3<-c(1,m_gre,m_gpa,0,1,0)
r4<-c(1,m_gre,m_gpa,0,0,1)

pi1<-1/(1+exp(-t(beta_sombrero)%*%r1))
pi2<-1/(1+exp(-t(beta_sombrero)%*%r2))
pi3<-1/(1+exp(-t(beta_sombrero)%*%r3))
pi4<-1/(1+exp(-t(beta_sombrero)%*%r4))
c(pi1,pi2,pi3,pi4)
```

```
## [1] 0.5166016 0.3522846 0.2186120 0.1846684
```

El valor predicho para cada una de estas observaciones será 1 si $\hat{\pi}_i$ es mayor a 0.5, o 0 en caso contrario.

```
pred1<-1*(pi1>0.5)
pred2<-1*(pi2>0.5)
pred3<-1*(pi3>0.5)
pred4<-1*(pi4>0.5)
```

```r
c(pred1,pred2,pred3,pred4)
```

```
## [1] 1 0 0 0
```

d) Buscamos las estimaciones de los cocientes de los odds cuando la variable x_i aumenta en una unidad y el resto permanece constante, para cada i de 1 a 5.

```r
codd1<-exp(ajuste$coef[1])
codd2<-exp(ajuste$coef[2])
codd3<-exp(ajuste$coef[3])
codd4<-exp(ajuste$coef[4])
codd5<-exp(ajuste$coef[5])
c(codd1,codd2,codd3,codd4,codd5)
```

```
## (Intercept)         gre         gpa       rank2       rank3
##   0.0185001   1.0022670   2.2345448   0.5089310   0.2617923
```

```r
IC<-matrix(0,ncol=2,nrow=5)

for(i in 1:5)
{
  IC[i,]<-exp(c(summary(ajuste)$coef[i,1]-qnorm(0.975)*summary(ajuste)$coef[i,2],
          summary(ajuste)$coef[i,1]+qnorm(0.975)*summary(ajuste)$coef[i,2]))
}
colnames(IC)<-c("LI","LS")
rownames(IC)<-c("exp{beta 1}", "exp{beta 2}", "exp{beta 3}",
                "exp{beta 4}", "exp{beta 5}")

IC
```

```
##                      LI        LS
## exp{beta 1} 0.001980825 0.1727834
## exp{beta 2} 1.000120237 1.0044184
## exp{beta 3} 1.166121956 4.2818768
## exp{beta 4} 0.273692172 0.9463578
## exp{beta 5} 0.133055086 0.5150889
```

Ningún intervalo contiene al 1, como era de esperarse.

e) Realizamos un análisis secuencial de la salida usando la función Anova

```r
anova(ajuste, test= "Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: admit
##
## Terms added sequentially (first to last)
##
##
##      Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                   399     499.98
## gre   1  13.9204       398     486.06 0.0001907 ***
## gpa   1   5.7122       397     480.34 0.0168478 *
## rank  3  21.8265       394     458.52 7.088e-05 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

f) Para observar que hace en cada caso, comparamos realizando un ajuste secuencial agregando de a una variable por vez.

```
mod1<-glm(admit ~ 1,  data = datos, family = "binomial")
mod2<-glm(admit ~ gre,  data = datos, family = "binomial")
mod3<-glm(admit ~ gre+gpa,  data = datos, family = "binomial")
mod4<-glm(admit ~ gre+gpa+rank,  data = datos, family = "binomial")
anova(mod1,mod2,test="LRT")
```

```
## Analysis of Deviance Table
##
## Model 1: admit ~ 1
## Model 2: admit ~ gre
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1       399     499.98
## 2       398     486.06  1    13.92 0.0001907 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(mod2, mod3, test="LRT")
```

```
## Analysis of Deviance Table
##
## Model 1: admit ~ gre
## Model 2: admit ~ gre + gpa
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       398     486.06
## 2       397     480.34  1   5.7122  0.01685 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(mod3, mod4, test="LRT")
```

```
## Analysis of Deviance Table
##
## Model 1: admit ~ gre + gpa
## Model 2: admit ~ gre + gpa + rank
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1       397     480.34
## 2       394     458.52  3   21.826 7.088e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#con Chisq

anova(mod1,mod2,test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: admit ~ 1
## Model 2: admit ~ gre
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1       399     499.98
## 2       398     486.06  1    13.92 0.0001907 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(mod2, mod3, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: admit ~ gre
## Model 2: admit ~ gre + gpa
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       398     486.06
## 2       397     480.34  1   5.7122  0.01685 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(mod3, mod4, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: admit ~ gre + gpa
## Model 2: admit ~ gre + gpa + rank
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1       397     480.34
## 2       394     458.52  3   21.826 7.088e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Observamos que los valores son los mismos haciendo el desarrollo de a un paso por vez, que el realizado en el ítem anterior. Los resultados usando el test Chisq son los mismos.

f) Comparamos todos los modelos posibles de 2 variables con el completo

```r
mod5<-glm(admit ~ gre+gpa,  data = datos, family = "binomial")
mod6<-glm(admit ~ gre+rank,  data = datos, family = "binomial")
mod7<-glm(admit ~ gpa+rank,  data = datos, family = "binomial")
anova(mod5,ajuste,test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: admit ~ gre + gpa
## Model 2: admit ~ gre + gpa + rank
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1       397     480.34
## 2       394     458.52  3   21.826 7.088e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(mod6,ajuste,test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: admit ~ gre + rank
## Model 2: admit ~ gre + gpa + rank
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       395     464.53
## 2       394     458.52  1   6.0143  0.01419 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(mod7,ajuste,test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: admit ~ gpa + rank
## Model 2: admit ~ gre + gpa + rank
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       395     462.88
## 2       394     458.52  1   4.3578  0.03684 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```