Tp2

Trabajo práctico 2

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'
```

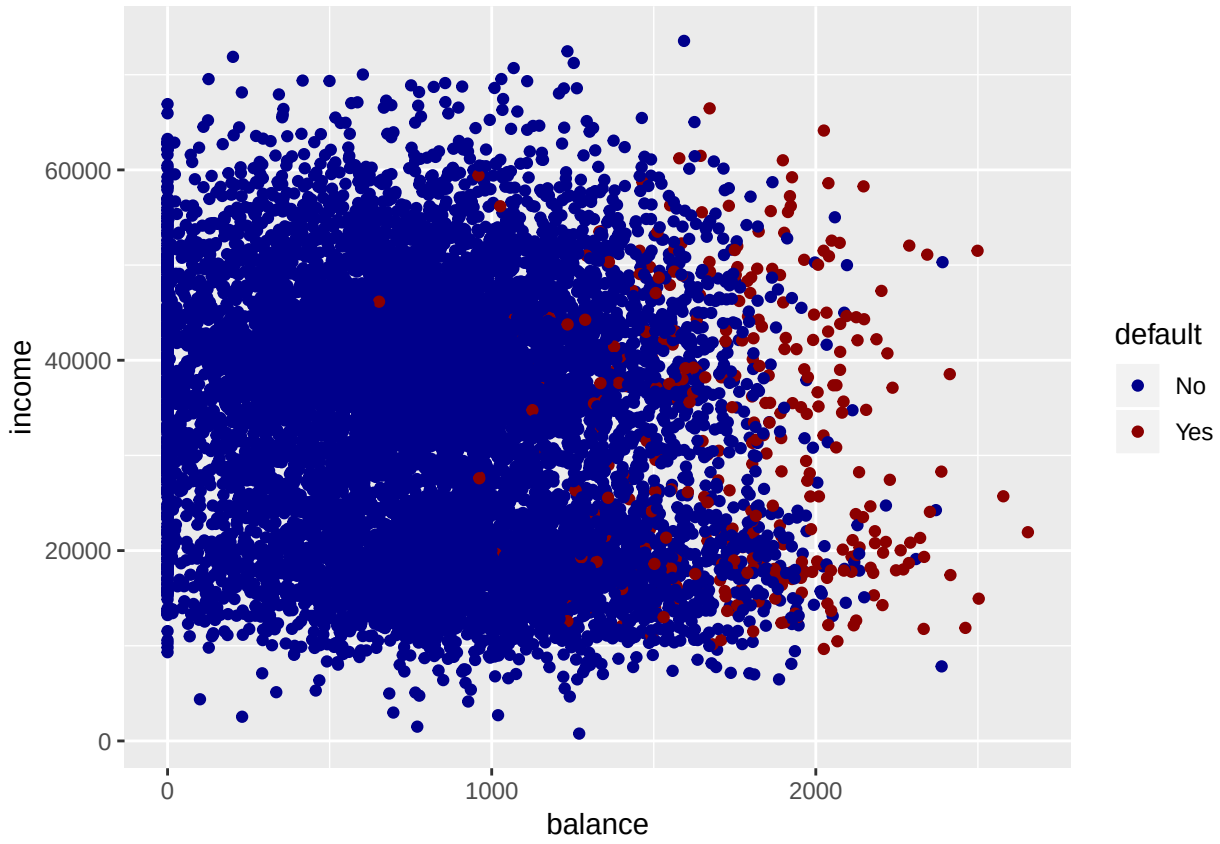
```
## The following object is masked from 'package:faraway':
##
##     happy
```

Ejercicio 1

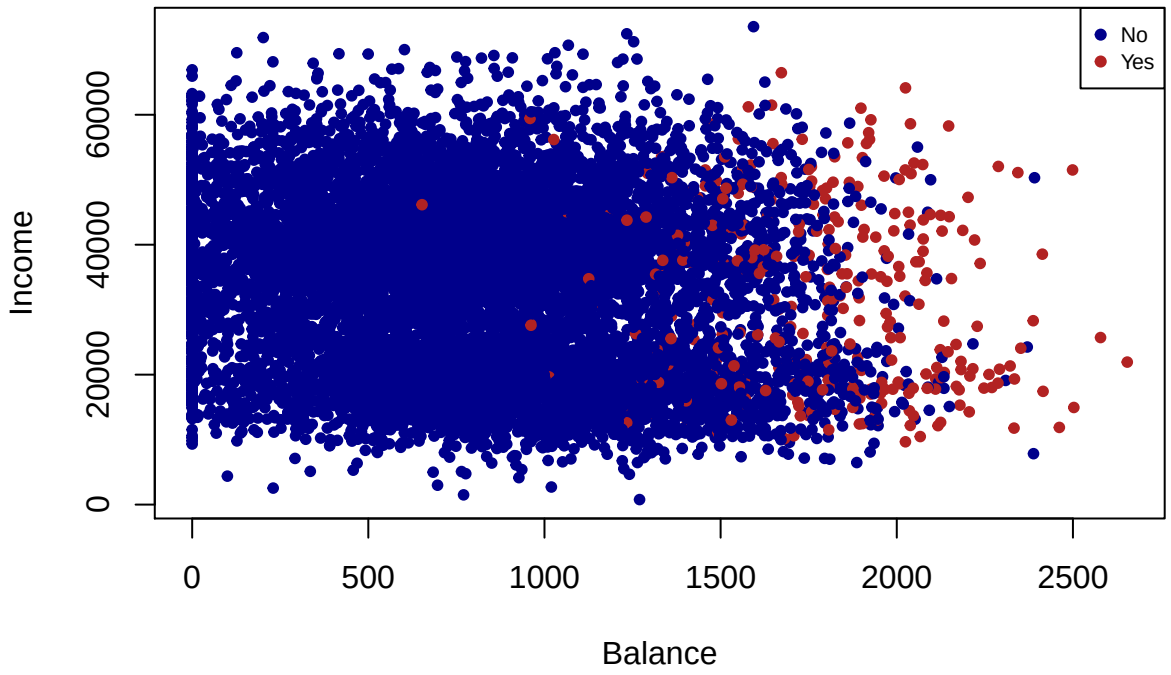
a) Graficamos las variables Balance vs. Income, pintando de diferente color según la variable Default.

```
datos<-Default

#con ggplot
gg<-ggplot(data=datos, aes(x=balance,y=income,col=default))+
  geom_point()+
  scale_color_manual(values=c("darkblue", "darkred"))
gg
```



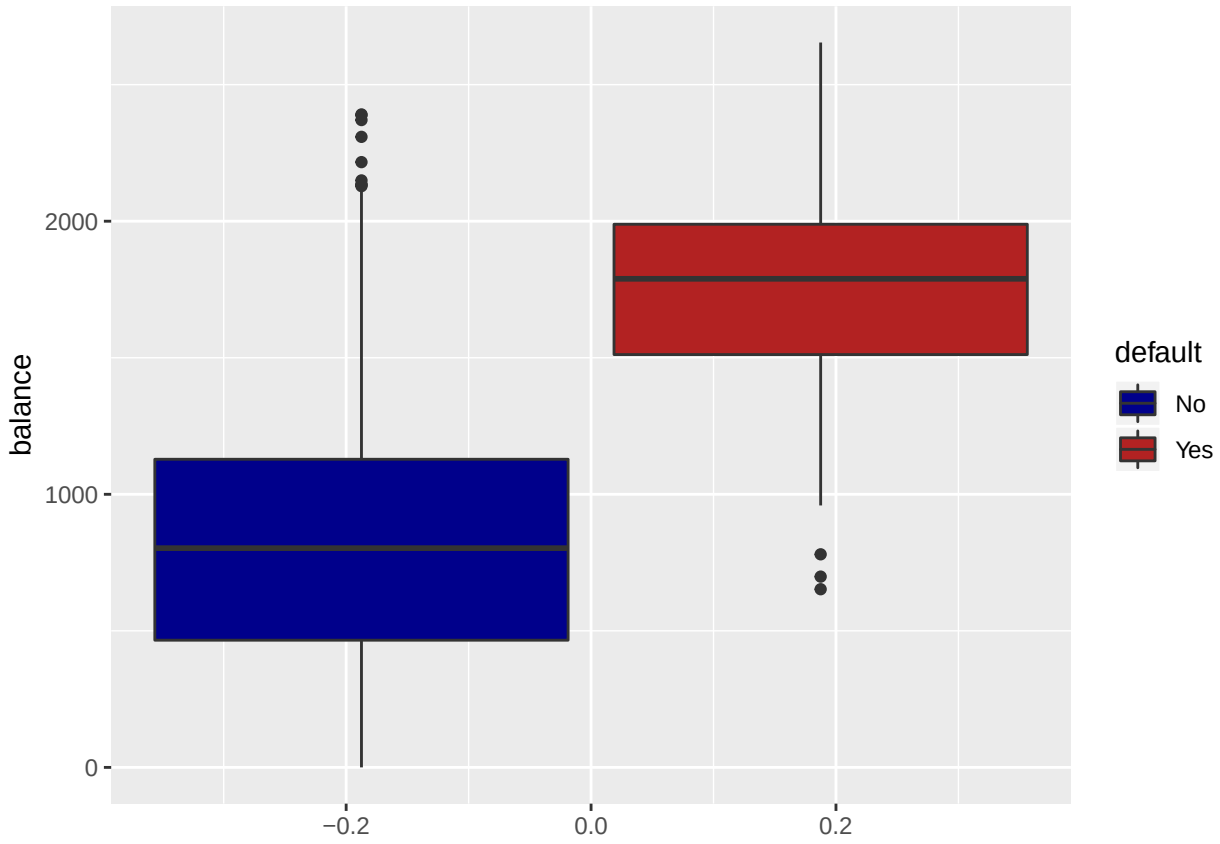

```
#con plot
colores<-c("darkblue", "firebrick")
plot(datos$balance,datos$income,col=colores[datos$default],
     pch=20, xlab="Balance", ylab="Income")
legend(
  x ="topright",
  legend = levels(datos$default),
  col = colores,
  pch = 19, # same as pch=20, just smaller
  cex = .7 # scale the legend to look attractively sized
)
```



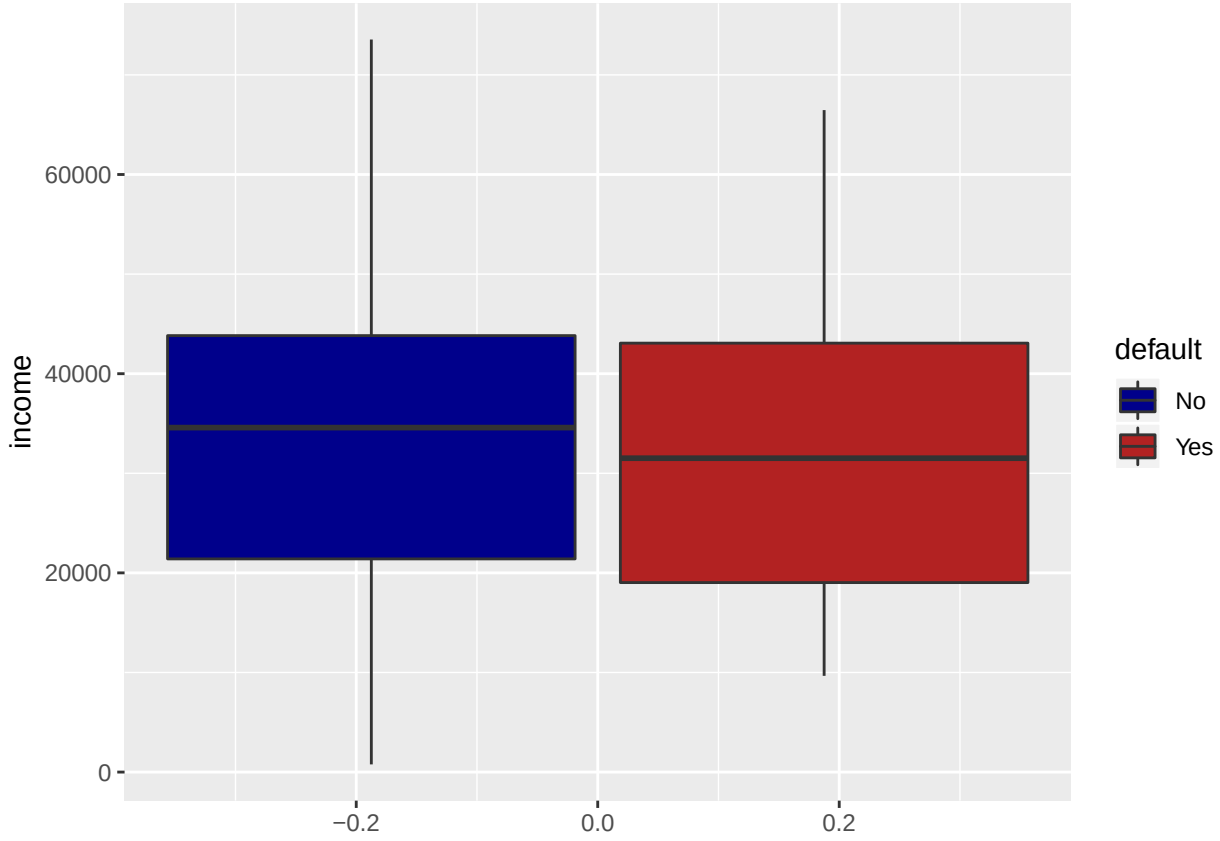
En el gráfico se observa que hay una gran diferencia en las observaciones de alto valor de Balance (Default tiende a ser "Yes") y las de bajo valor, mientras que si se analiza la variable income, no hay marcada diferencia entre las dos categorías de Default para los distintos valores de income. Podría tener mayor capacidad predictiva la variable Balance.

b) Realizamos boxplot, para cada una de las dos variables, separando en las distintas categorías de la variable Default.

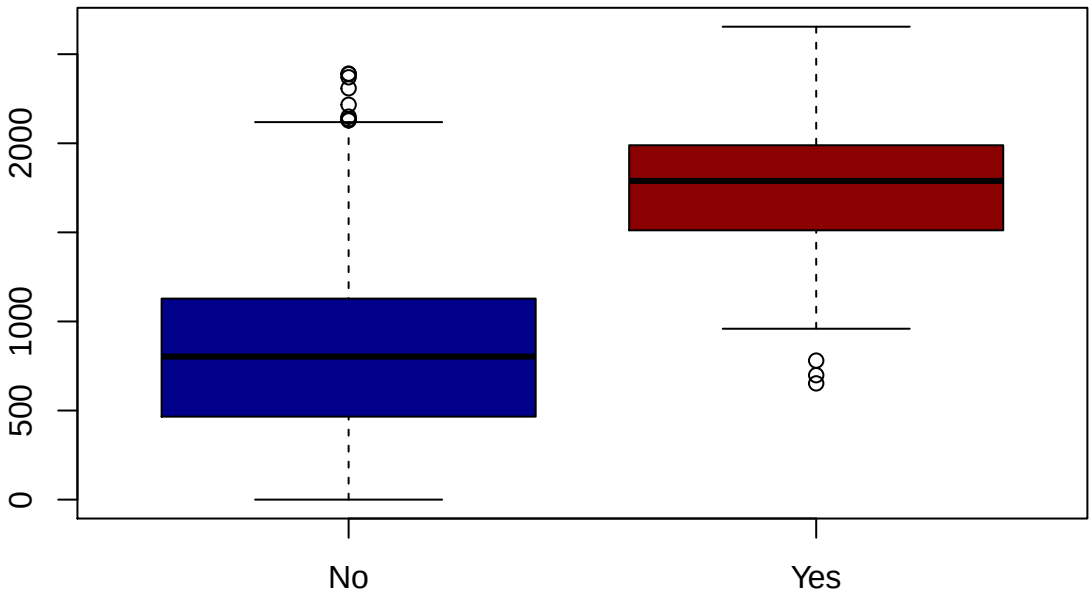
```
#con ggplot
gg<-ggplot(data=datos, aes(y=balance,fill=default))+
  geom_boxplot()+
  scale_fill_manual(values=c("darkblue", "firebrick"))
gg
```



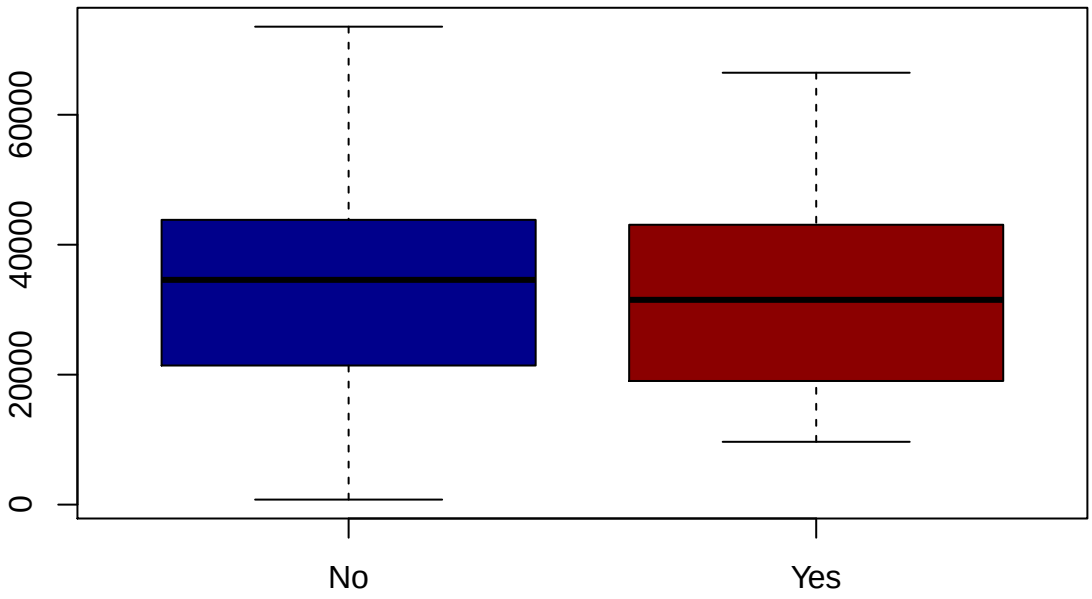

```
gg<-ggplot(data=datos, aes(y=income,fill=default))+
  geom_boxplot()+
  scale_fill_manual(values=c("darkblue", "firebrick"))
gg
```




```
#con plot
boxplot(datos$balance~datos$default,col=c("darkblue", "darkred"))
```




```
boxplot(datos$income~datos$default,col=c("darkblue", "darkred"))
```



En estos gráficos se observa de forma más marcada lo observado en el ítem anterior: para la variable income, no hay mayores diferencias entre default si o no, mientras que para la variable Balance se observa una diferencia, lo cual puede hacer que dicha variable sea mucho mejor para predecir el default.

c) Según lo pedido, creamos la función vecinos.k, que dado un vector de observaciones x, un valor k, y una observación x_0, devuelve los k índices de los k vecinos más cercanos a x_0.

```
vecinos.k<-function(x,k,x0)
{
  distancia<-c()
  for(i in 1:length(x))
  {
    distancia[i]<-sqrt(sum(x[i]-x0)^2)
  }
  minimos<-sort(distancia,index.return=TRUE)$ix
  return(minimos[1:k])
}
```

d) Según lo pedido, creamos una función *clasificador* que utilice la función anterior y luego, conociendo las etiquetas y clasifique al punto x_0.

```
clasificador<-function(x,y,k,x0)
{
  y_elegidos<-y[vecinos.k(x,k,x0)]
  prediccion<-ifelse(mean(y_elegidos=="No")>0.5,"No","Si")
  return(prediccion)
}


#si no quiero depender de saber que hay adentro de y, o hay mas clases...

clasificador2<-function(x,y,k,x0)
{
  prediccion<-which.max(table(y[vecinos.k(x,k,x0)]))
  return(names(prediccion))
}
```

e) Utilizamos la función definida en el ítem anterior para clasificar 3 puntos, uno con balance 1500, otro con 1700 y el tercero con 2000

```
clasificador(datos$balance,datos$default,5,1500)
```

```
## [1] "No"
```

```
clasificador(datos$balance,datos$default,5,1700)
```

```
## [1] "No"
```

```
clasificador(datos$balance,datos$default,5,2000)
```

```
## [1] "Si"
```

f) Ahora separamos al azar la muestra en una muestra de entrenamiento y otra de testeo.

```
set.seed(123)
train<-sample(1:10000,8000)
datos.E<-datos[train,]
datos.T<-datos[-train,]
```

g) Utilizamos las muestras ya separadas para clasificar las observaciones de la muestra de testeo, con un k=5, y poder luego medir el porcentaje de observaciones mal clasificadas, utilizando para clasificar la variable Balance.

```
prediccion<-c()
for(i in 1:nrow(datos.T))
{
  prediccion[i]<-clasificador(datos.E$balance,datos.E$default,5,datos.T$balance[i])
}

mean(prediccion!=datos.T$default)
```

```
## [1] 0.0365
```

h) Repetimos el ítem anterior pero clasificando según la variable Income

```
prediccion<-c()
for(i in 1:nrow(datos.T))
{
  prediccion[i]<-clasificador(datos.E$income,datos.E$default,5,datos.T$income[i])
}

mean(prediccion!=datos.T$default)
```

```
## [1] 0.03
```

i) Utilizamos la librería glmnet, y la función glm para realizar un ajuste de regresión logística para predecir la la probabilidad de que un cliente tenga un default a partir de la variable balance.

```
logistica<-glm(default~balance,data=datos, family =binomial, subset = train)
summary(logistica)
```

```
##
## Call:
## glm(formula = default ~ balance, family = binomial, data = datos,
##     subset = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
```

```
## -2.3692  -0.1406  -0.0532  -0.0194   3.8115
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.102e+01  4.207e-01  -26.19   <2e-16 ***
## balance      5.756e-03  2.562e-04   22.47   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2394.2  on 7999  degrees of freedom
## Residual deviance: 1259.2  on 7998  degrees of freedom
## AIC: 1263.2
##
## Number of Fisher Scoring iterations: 8
```

Buscamos predecir la probabilidad de Default, dados los datos de testing, y luego calcular la proporción de datos mal clasificados para poder compararla con la proporción utilizando vecinos más cercanos. Utilizamos la función *predict* con el tipo "response" para que nos devuelva la probabilidad estimada con este ajuste.

```
datosnuevos<-as.data.frame(datos.T$balance)
names(datosnuevos)<-"balance"
# para que funcione la funcion predict, los datos nuevos
# deben ser de la forma data frame con los mismos nombres...

predichos<-predict(logistica,newdata=datosnuevos,type = "response")
```

Como criterio de clasificación, si la probabilidad de default es menor a 0.5 diremos que es "no".

```
pred<-ifelse(predichos>0.5,"Yes","No")
mean(pred!=datos.T$default)
```

```
## [1] 0.029
```

j) Repetimos el ítem anterior pero utilizando las variables Balance e Income para predecir.

```
logistica<-glm(default~balance+income,data=datos, family =binomial, subset = train)
datosnuevos<-as.data.frame(cbind(datos.T$balance,datos.T$income))
names(datosnuevos)<-c("balance","income")

predichos<-predict(logistica,newdata=datosnuevos,type = "response")

pred<-ifelse(predichos>0.5,"Yes","No")
mean(pred!=datos.T$default)
```

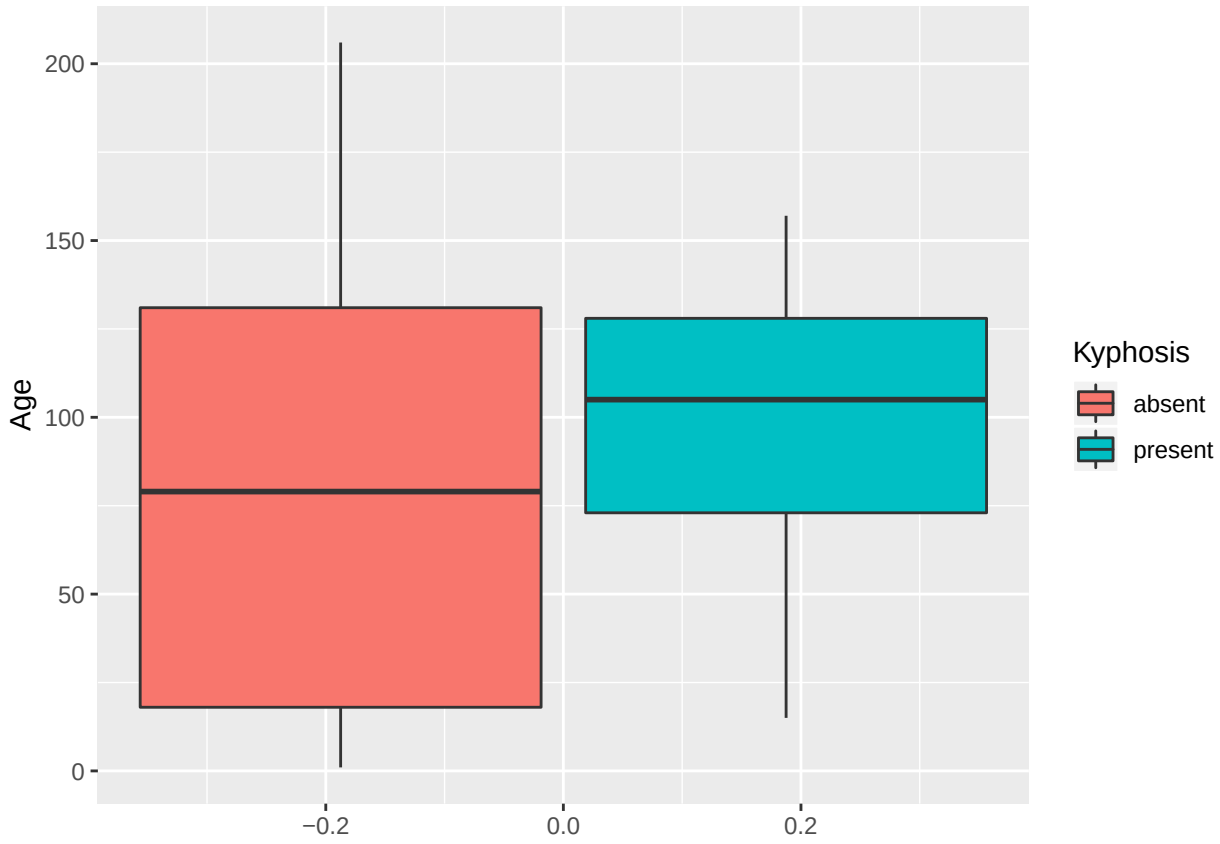
```
## [1] 0.029
```

Ejercicio 2

a) Graficamos la respuesta de incidencia en función de la edad del niño.

```
datos<-read.csv("kyphosis.csv")

gg<-ggplot(data=datos, aes(y=Age,fill=Kyphosis))+
  geom_boxplot()
gg
```



Realizamos un ajuste a un modelo logístico y observamos la salida del mismo

```
ajuste<-glm(Kyphosis~Age, data=datos, family = binomial)
summary(ajuste)
```

```
##
## Call:
## glm(formula = Kyphosis ~ Age, family = binomial, data = datos)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9023  -0.7397  -0.6028  -0.5521   1.9449
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.809351   0.530353  -3.412 0.000646 ***
## Age          0.005442   0.004822   1.129 0.259068
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 83.234  on 80  degrees of freedom
## Residual deviance: 81.932  on 79  degrees of freedom
## AIC: 85.932
##
## Number of Fisher Scoring iterations: 4
```

Según la salida, la variable Age no resulta significativa para predecir la probabilidad de la incidencia de la cifosis.

b) Realizamos un ajuste utilizando todas las variables estudiadas.

```
ajuste_todas<-glm(Kyphosis~Age+Number+Start, data=datos, family = binomial)
summary(ajuste_todas)
```

```
##
## Call:
## glm(formula = Kyphosis ~ Age + Number + Start, family = binomial,
##     data = datos)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3124  -0.5484  -0.3632  -0.1659   2.1613
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.036934   1.449575  -1.405  0.15996
## Age          0.010930   0.006446   1.696  0.08996 .
## Number       0.410601   0.224861   1.826  0.06785 .
## Start       -0.206510   0.067699  -3.050  0.00229 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 83.234  on 80  degrees of freedom
## Residual deviance: 61.380  on 77  degrees of freedom
## AIC: 69.38
##
## Number of Fisher Scoring iterations: 5
```

Se observa que si tenemos en cuenta todas las variables, todas resultan significativas con un nivel de 0.1, pero solo Start resulta significativa con un nivel de 0.05.

c) Ajustamos ahora a un modelo de regresión logística como en el ítem a) pero agregando un término cuadrático.

```
Age2<-datos$Age^2
datos2<-cbind(datos,Age2)

ajuste2<-glm(Kyphosis~Age+Age2, data=datos2, family = binomial)
summary(ajuste2)
```

```
##
## Call:
## glm(formula = Kyphosis ~ Age + Age2, family = binomial, data = datos2)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0079  -0.8412  -0.4155  -0.2209   2.3920
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.7702901  1.1511211  -3.275  0.00106 **
```

```
## Age          0.0700351  0.0269840   2.595  0.00945 **
## Age2        -0.0003652  0.0001478  -2.471  0.01349 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 83.234  on 80  degrees of freedom
## Residual deviance: 72.739  on 78  degrees of freedom
## AIC: 78.739
##
## Number of Fisher Scoring iterations: 5
```

Vemos que de esta manera, la variable Age sí resulta significativa.

d) Repetimos el ítem b) agregando un término cuadrático para la variable Age.

```
ajuste3<-glm(Kyphosis~., data=datos2, family = binomial)
summary(ajuste3)
```

```
##
## Call:
## glm(formula = Kyphosis ~ ., family = binomial, data = datos2)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.23573  -0.51241  -0.24509  -0.06108   2.35495
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -4.3835660  2.0548871  -2.133  0.03291 *
## Age          0.0816412  0.0345292   2.364  0.01806 *
## Number       0.4268659  0.2365134   1.805  0.07110 .
## Start       -0.2038421  0.0706936  -2.883  0.00393 **
## Age2        -0.0003965  0.0001905  -2.082  0.03737 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 83.234  on 80  degrees of freedom
## Residual deviance: 54.428  on 76  degrees of freedom
## AIC: 64.428
##
## Number of Fisher Scoring iterations: 6
```

e) Ajustamos el modelo pedido

```
Number2<-datos$Number^2
datos3<-cbind(datos2,Number2)

ajuste4<-glm(Kyphosis~Number2+Age*Number, data=datos3, family = binomial)
summary(ajuste4)
```

```
##
## Call:
## glm(formula = Kyphosis ~ Number2 + Age * Number, family = binomial,
```

```
##     data = datos3)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3565  -0.6941  -0.4340  -0.1464   2.1894
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -7.762002   3.004345  -2.584  0.00978 **
## Number2     -0.049078   0.075632  -0.649  0.51640
## Age          0.026445   0.018239   1.450  0.14708
## Number       1.512401   0.883205   1.712  0.08682 .
## Age:Number  -0.004075   0.003735  -1.091  0.27523
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 83.234  on 80  degrees of freedom
## Residual deviance: 69.612  on 76  degrees of freedom
## AIC: 79.612
##
## Number of Fisher Scoring iterations: 5
```