Analisis Exploratorio y Predicción de Enfermedades del Corazón

Javier Aguirre

31/12/2021

Indíce

Datos

Los datos han sido recogidos de kaggle del dataset llamado "Heart Failure Prediction Dataset" de el siguiente link: https://www.kaggle.com/fedesoriano/heart-failure-prediction

Las enfermedades cardiovasculares son la causa número 1 de muerte globalmente, aproximadamente con una tasa de mortalidad de 17.9 personas globalmente, lo que es el 31% de las muertes a nivel mundial. Este dataset contiene 11 características que se pueden usar para predecir posibles enfermedades del corazón.

La gente con enfermedades cardiovasculares o que tienen alto riesgo de enfermedades necesitan una detección precoz y tratamiento. La exploración puede aportar información valiosa en encontrar patrones y un modelo eficaz de machine learning puede ayudar a la prevención. Por eso, se procederá a la exploración de datos y a construir un modelo de machine learning.

Información de los datos

Age: edad del paciente en años

Sex: sexo del paciente [M: Hombre (Male), F: Mujer (Female)]

ChestPainType: Tipo de dolor de pecho [TA: Angina Típica (Typical Angina), ATA: Angina Atípica (Atypical Angina), NAP: Dolor No-Angina (Non-Anginal Pain), ASY: Asintomático (Asymptomatic)]

RestingBP: presion sanguínea en reposo mm Hg

Cholesterol: colesterol en suero [mm/dl]

FastingBS: azúcar en sangre en ayunas [1: si FastingBS $>$ 120 mg/dl, 0: si no]

RestingECG: resultados de electrocardiograma en reposo [Normal: Normal, ST: tener una anormalidad en las ondas ST-T (inversion en la onda T y/o elevación o depresion en ST de $>$ 0.05 mV), LVH: mostrado una probable o definitiva hipertrofia en el ventrículo izquierdo por el criterio de Estes]

MaxHR: máxima pulsación obtenida [valor numérico entre 60 y 202]

ExerciseAngina: angina inducida por ejercicio [Y: Si, N: No]

Oldpeak: depresión del ST inducida por el ejercicio relativo al descanso [valor numérico medido en depresión]

ST_Slope: pendiente del segmento ST de ejercicio máximo [Up: cuesta arriba, Flat: plano, Down: cuesta abajo]

HeartDisease: clase de enfermedad del corazón [1: Enfermo, 0: Normal]

Estudio descriptivo general

Primero vamos a cargar los datos: (eliminamos la fila 450 ya que contenia un dato faltante y todas las variables nominales las sustituiremos por numeros usando la funcion matrix). Hay que tener en cuenta que cuando tenemos valores faltantes o outliers hay que ser rigurosos a la hora de proponer una solución. En este caso teniendo en cuenta que tenemos 917 muestras y sólo en 1 hay datos faltantes, borrarla es una buena opción ya que no tiene una gran importancia en el dataset.

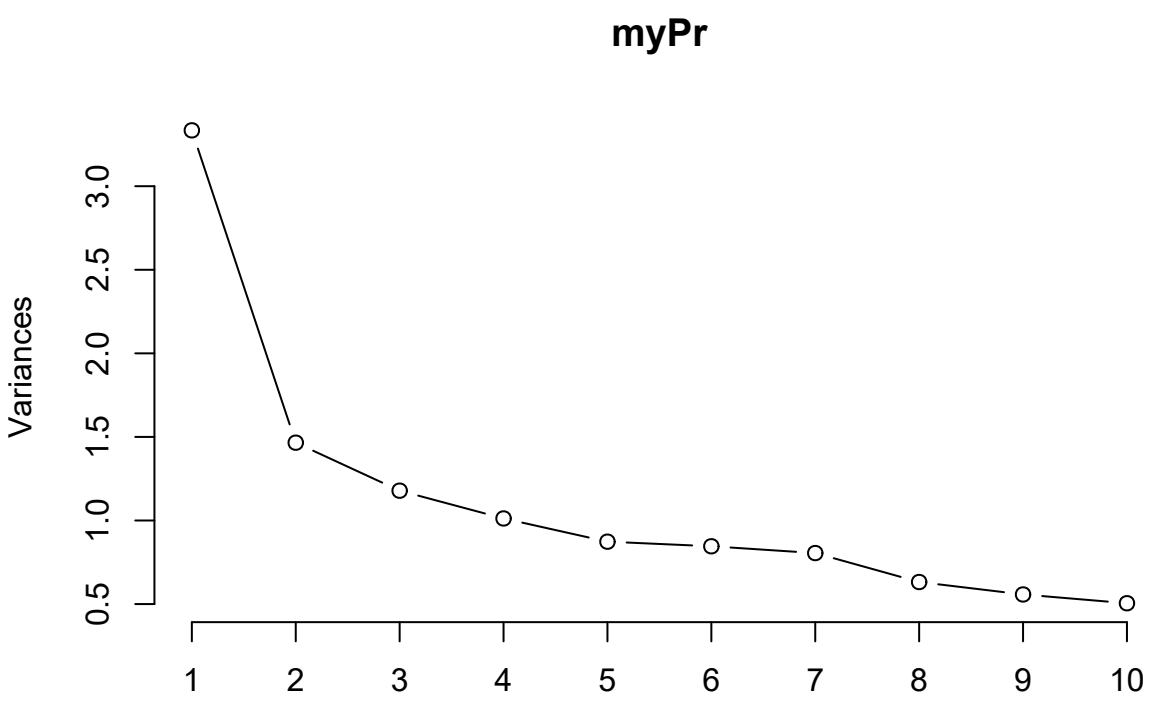
```
data <- read.csv("heart.csv")
data<-data[-450,]
data<-data.matrix(data)
```

Dada la naturaleza de los datos, es importante identificar que componentes estan relacionados entre ellos y con las enfermedades cardíacas. Por eso, vamos a proceder a aplicar la tecnica del PCA (Principal component analysis) para reducir la dimensionalidad de los datos y entender mejor las relaciones de nuestro dataset.

```
myPr <- prcomp(data[,c(1:12)], scale = TRUE)
summary(myPr)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6     PC7
## Standard deviation     1.8260 1.2106 1.0855 1.00607 0.93452 0.91963 0.89728
## Proportion of Variance 0.2778 0.1221 0.0982 0.08435 0.07278 0.07048 0.06709
## Cumulative Proportion  0.2778 0.4000 0.4982 0.58253 0.65530 0.72578 0.79287
##                           PC8     PC9    PC10    PC11    PC12
## Standard deviation     0.79494 0.74662 0.71058 0.65603 0.60071
## Proportion of Variance 0.05266 0.04645 0.04208 0.03587 0.03007
## Cumulative Proportion  0.84553 0.89199 0.93406 0.96993 1.00000
```

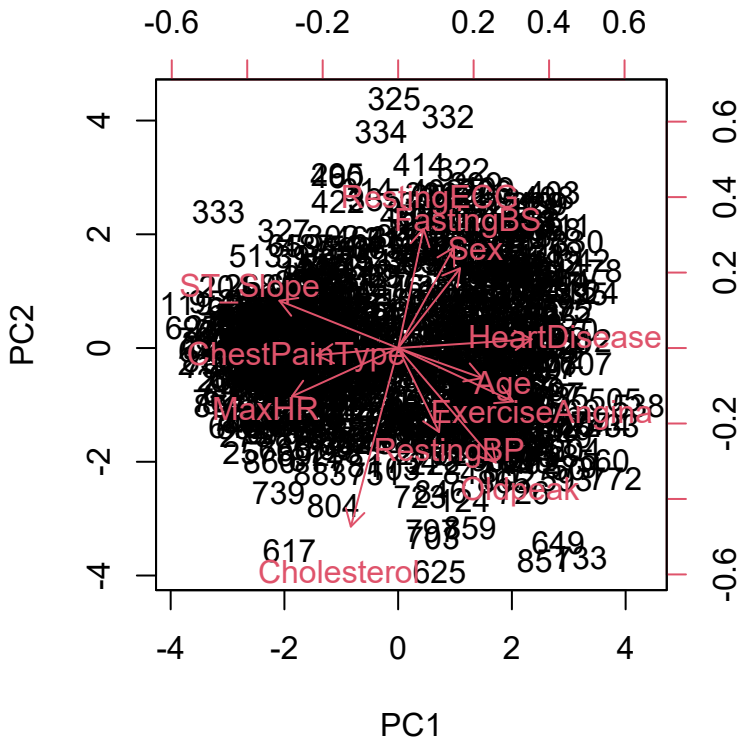
```
plot(myPr, type = "l")
```



Se puede ver en la figura de arriba la importancia de las dos primeras componenetes lo que explica casi la mitad de la variabilidad de los datos.

A continuación el biplot del PCA. Este plot se utiliza para poder ver las dos componentes principales y la ubicacion de cada variable en la correlación:

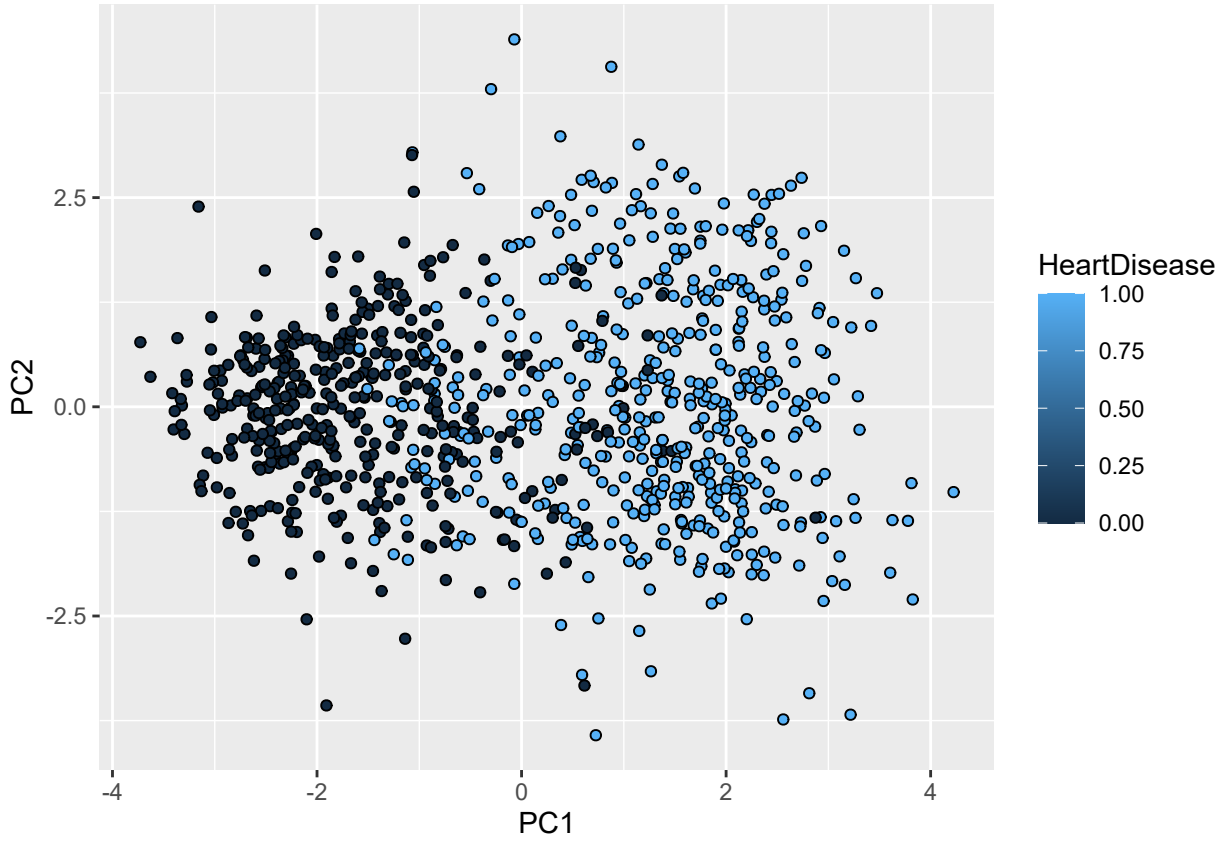
```
biplot(myPr, scale = 0)
```



Se aprecian en el biplot 4 clusters diferentes formandose. El primero es *age*, *exerciseAngina*, *RestingBP*, *Oldpeak* y *HeartDisease*. El segundo es *ST_Slope*, *ChestPainType* y *MaxHR*. El tercero es *RestingECG*, *FastingBS* y *sex*. Finalmente *cholesterol* el colesterol está separado del resto de los datos.

Aparte del biplot, podemos coger los componenetes principales (PC1 y PC2) y ver su relación con respecto a la variable de enfermedad del corazón.

```
data2 <- cbind(data, myPr$x)
library(ggplot2)
data2<-data.frame(data2)
ggplot(data2, aes(PC1, PC2, col = HeartDisease, fill = HeartDisease)) +
    geom_point(shape = 21, col = "black")
```



Aunque la separacion no sea perfecta, se puede ver claramente que el PC1 es capaz de separar bastante bien los enfermos de los normales. Lo cual indica que dentro de PC1 los elementos aunque no perfectos, son predictores de la enfermedad.

Además aqui la correlacion entre variables y componentes principales:

```
cor(data[,c(1:11)], data2[, 13:16])
```

```
##                       PC1         PC2         PC3          PC4
## Age             0.5079719 -0.12038242  0.58277626 -0.009322981
## Sex             0.3742622  0.31974956 -0.21235529  0.183598823
## ChestPainType  -0.4928044 -0.02847799  0.43057980  0.261864270
## RestingBP       0.2474142 -0.33676947  0.59360508 -0.297610160
## Cholesterol    -0.2861961 -0.71611740 -0.08918543 -0.152930257
## FastingBS       0.3371618  0.40310425  0.38598492  0.410501086
## RestingECG      0.1535823  0.47741650  0.03908531 -0.637645561
## MaxHR          -0.6418674 -0.19365398 -0.10166928  0.302166743
## ExerciseAngina  0.6957261 -0.21428204 -0.23674360 -0.182041563
## Oldpeak         0.5898550 -0.45900554 -0.06374329  0.140296079
## ST_Slope       -0.7186344  0.18978271  0.09092857 -0.235238079
```

Análisis de relaciones entre variables

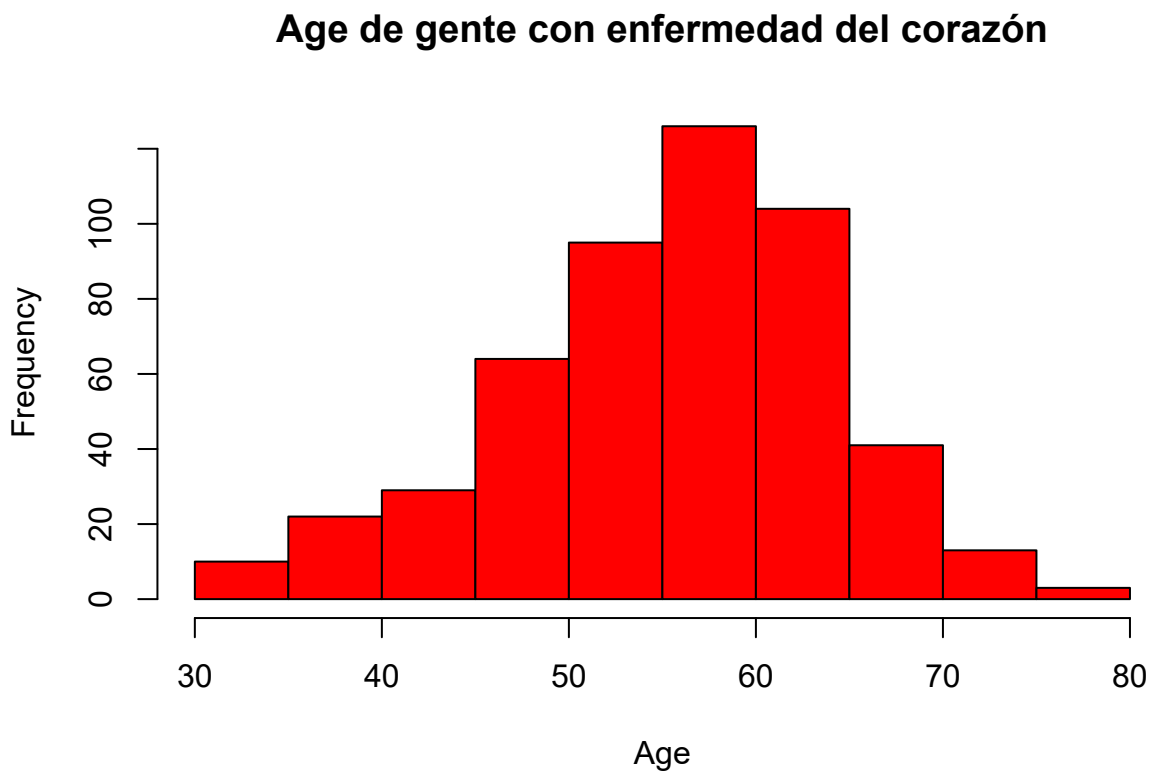
Primero de todo vamos a crear nuevos dataframes para hacer analisis específicos de variables y compararemos las variables con *HeartDisease*. En concreto vamos a elegir las variables con mayores puntuaciones en el analisis de PCA y menores puntuaciones (*Age*,*ExerciseAngina*,*RestingBP*,*Oldpeak*,*ST_Slope*,*MaxHR*):

```
dataAge<-data.frame(data[,12],data[,1])
names(dataAge)<-c("HeartDisease","Age")
dataAngina<-data.frame(data[,12],data[,9])
names(dataAngina)<-c("HeartDisease","ExerciseAngina")
dataResting<-data.frame(data[,12],data[,4])
names(dataResting)<-c("HeartDisease","RestingBP")
dataPeak<-data.frame(data[,12],data[,10])
names(dataPeak)<-c("HeartDisease","Oldpeak")
dataSlope<-data.frame(data[,12],data[,11])
names(dataSlope)<-c("HeartDisease","ST_Slope")
dataMaxHR<-data.frame(data[,12],data[,8])
names(dataMaxHR)<-c("HeartDisease","MaxHR")
```

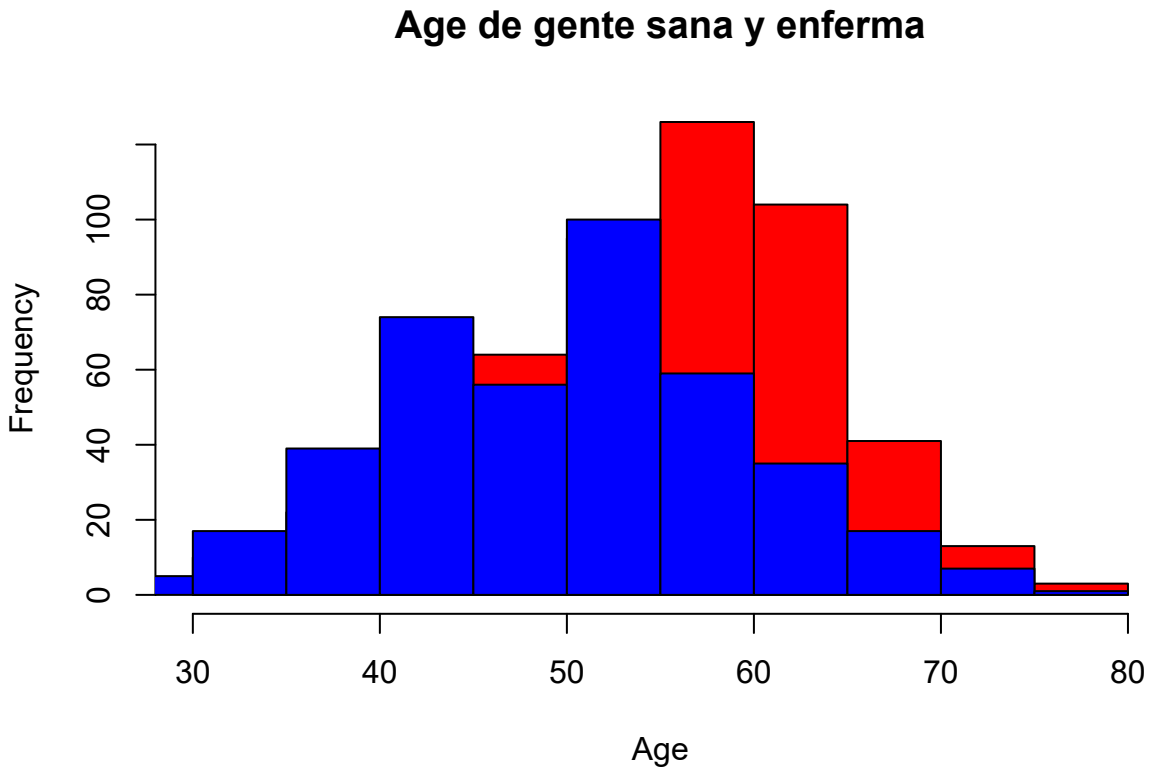
Ahora podemos comenzar con el estudio.

Empezamos comparando la edad de los enfermos con la de los sanos para ver si como aparecia en el PCA tienen relación alguna.*age*:

```
#Age Study
datHealthy<-subset(dataAge, HeartDisease == 0)
datUnhealthy<-subset(dataAge, HeartDisease == 1)
hist(main="Age de gente con enfermedad del corazón", xlab="Age", as.numeric(unlist(datUnhealthy[,2])),
```



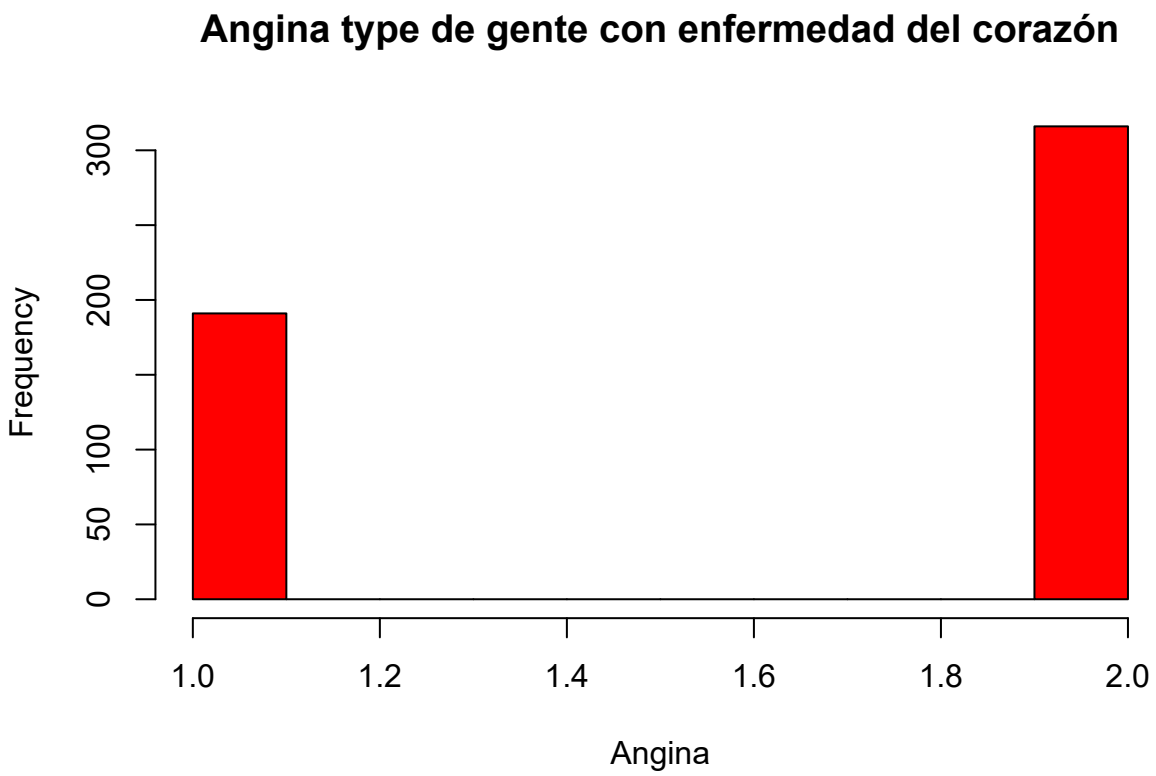
```
hist(main="Age de gente sana y enferma",xlab="Age",as.numeric(unlist(datUnhealthy[,2])), col='red')
hist(as.numeric(unlist(datHealthy[,2])),col='blue', add=TRUE)
```



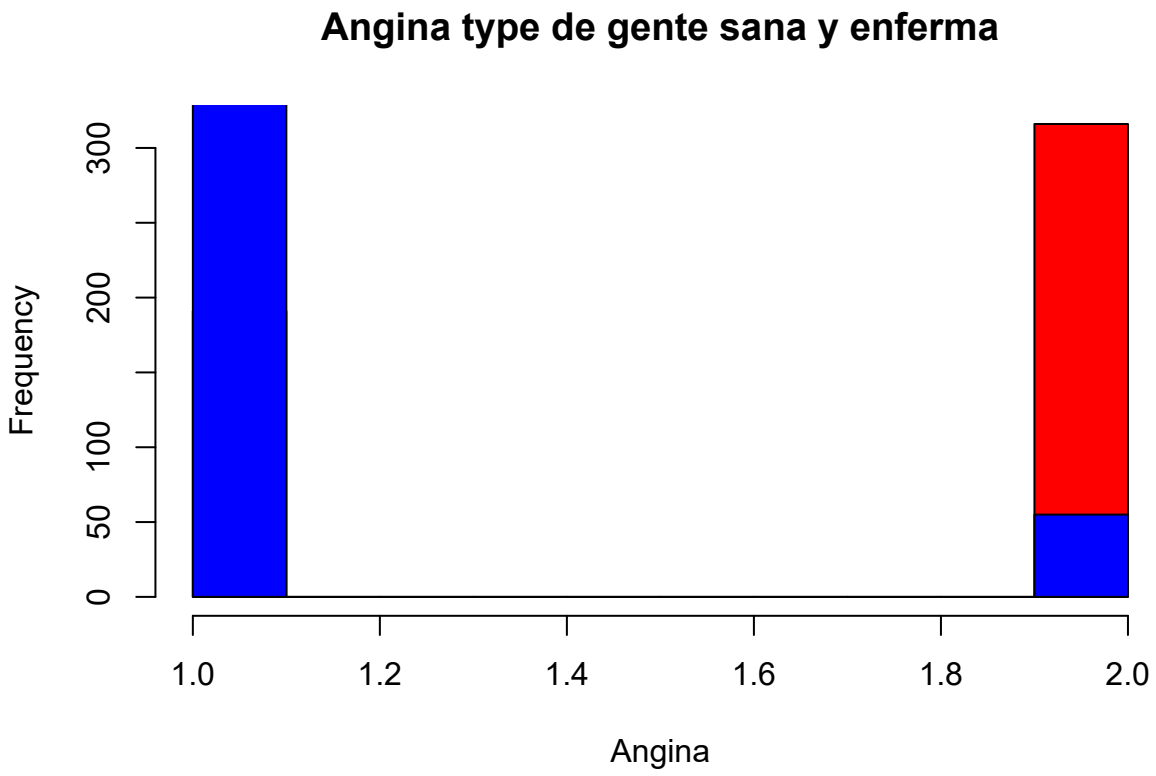
Es importante tener en cuenta que los datos originales no son uniformes si no que al haber más población de 50 años que jovenes, hay mayor cantidad de datos de gente adulta. Por lo tanto, es arriesgado decir que en gente muy mayor hay menos casos. Ahora, si se puede comparar la cantidad de gente sana y enferma y su distribución. Se puede apreciar como hay más frecuencia de casos de enfermedades del corazon en gente mayor que joven. Y como de la población total la gente sana esta más distribuida que los enfermos.

Analicemos ahora la angina (1 equivale a no tener angina y 2 equivale a tener angina):

```
#Angina study
datHealthy<-subset(dataAngina, HeartDisease == 0)
datUnhealthy<-subset(dataAngina, HeartDisease == 1)
hist(main="Angina type de gente con enfermedad del corazón",xlab="Angina", as.numeric(unlist(datUnhealth
```



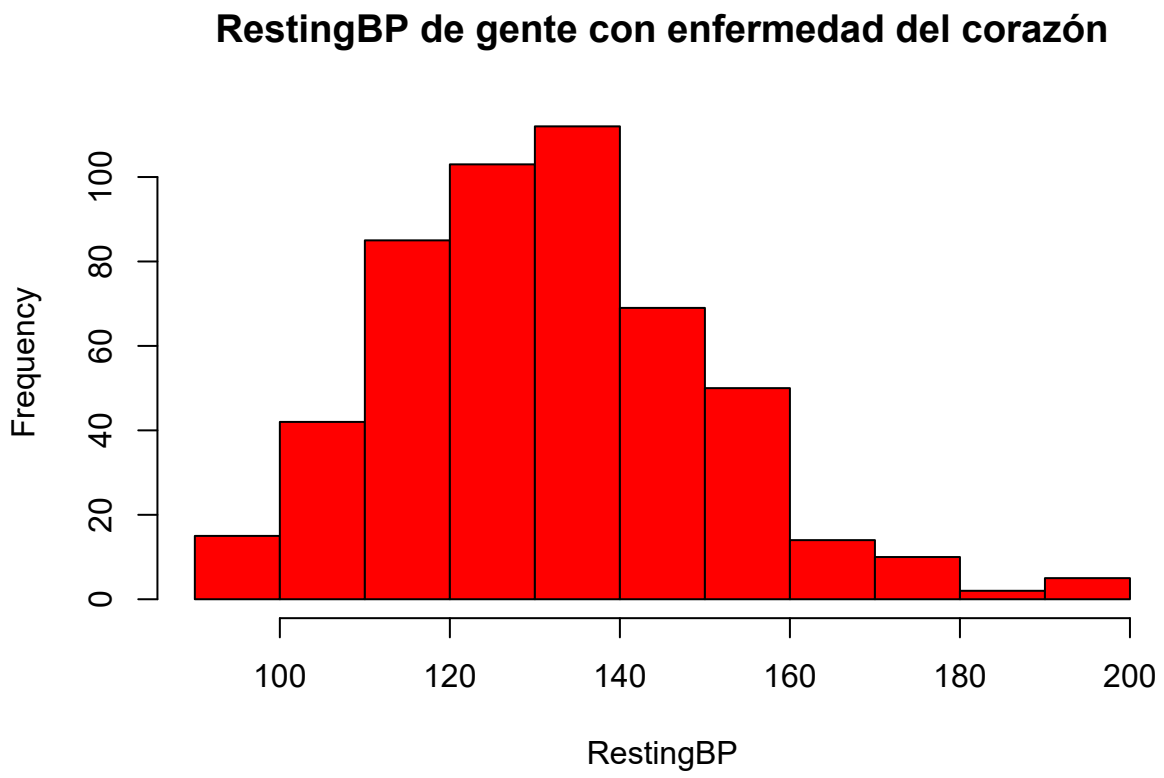
```
hist(main="Angina type de gente sana y enferma",xlab="Angina",as.numeric(unlist(datUnhealthy[,2])), col=
hist(as.numeric(unlist(datHealthy[,2])),col='blue', add=TRUE)
```



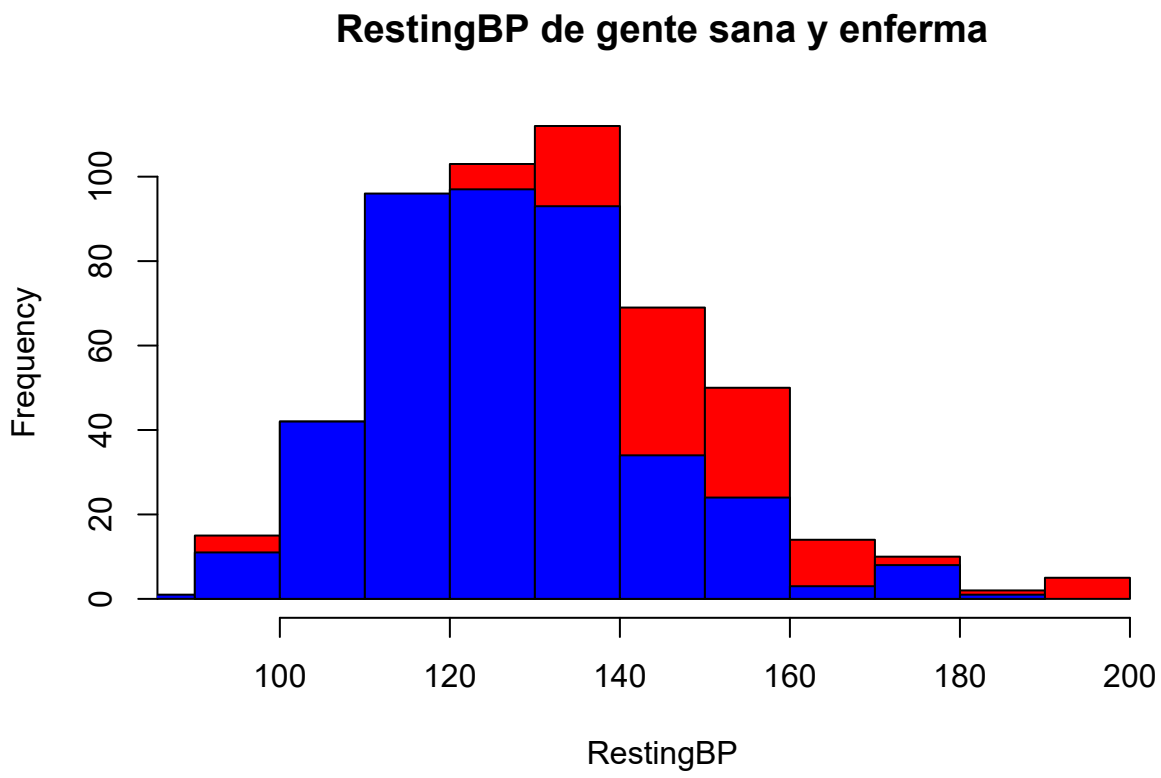
De nuevo, para evitar sesgos compararemos los casos de enfermos con los casos sanos para obtener una buena comparativa. Podemos ver como la gente sana apenas tiene angina, mientras que la gente enferma tiene en gran cantidad angina. Esto no quiere decir que si tienes enfermedades del corazón vayas a tener angina, pero si que se puede ver que en gran medida y comparando gente sana y enferma, el ratio de tener angina y estar enfermo es de 300 frente a 50 de gente sana. Por lo tanto, la angina parece un buen predictor de enfermedades del corazón.

Analicemos ahora la presion sanguínea en reposo (RestingBP):

```
#RestingBP study
datHealthy<-subset(dataResting, HeartDisease == 0)
datUnhealthy<-subset(dataResting, HeartDisease == 1)
hist(main="RestingBP de gente con enfermedad del corazón",xlab="RestingBP", as.numeric(unlist(datUnheal
```



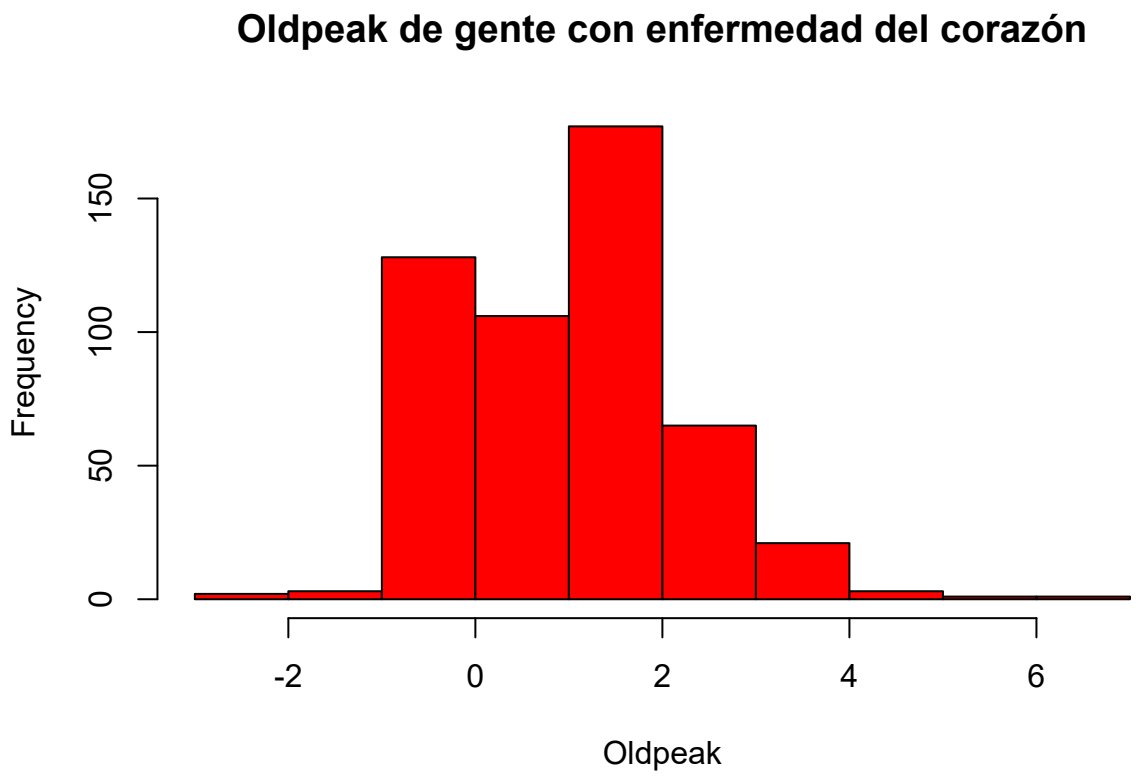
```
hist(main="RestingBP de gente sana y enferma",xlab="RestingBP",as.numeric(unlist(datUnhealthy[,2])), co
hist(as.numeric(unlist(datHealthy[,2])),col='blue', add=TRUE)
```



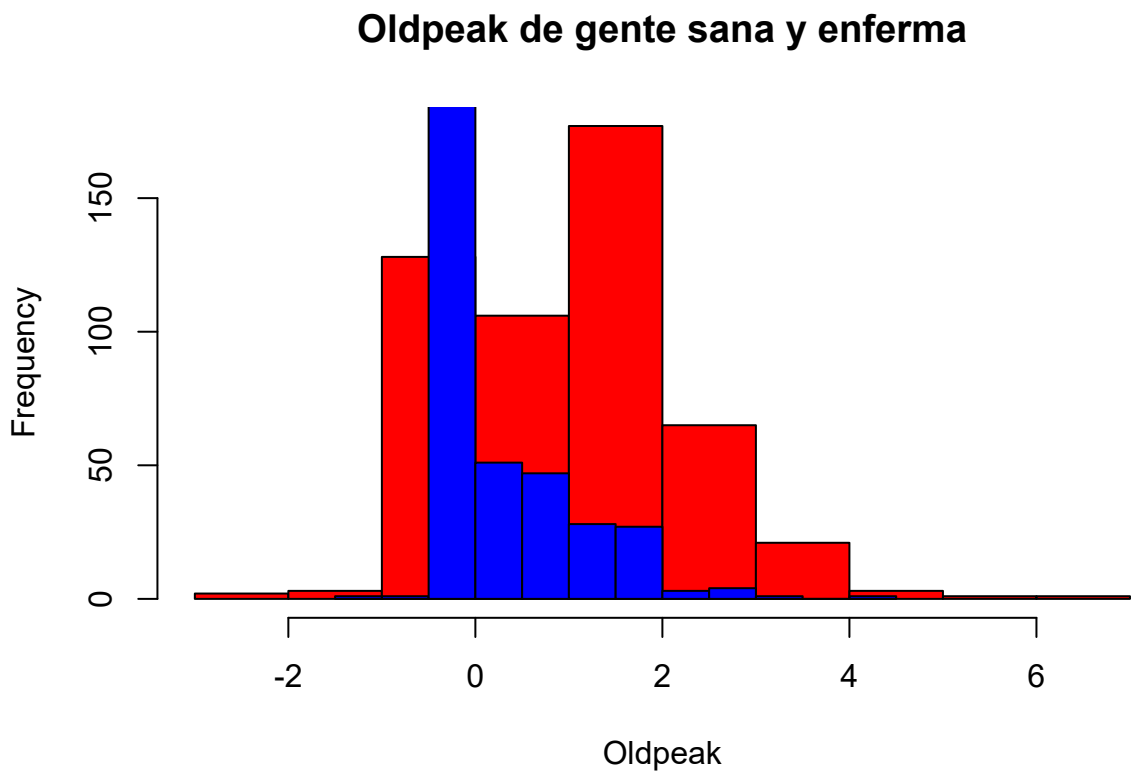
En este caso se puede ver que presiones sanguineas más altas tienden a tener mayores enfermedades del corazón. Ahora, tampoco se ve una proporción muy significativa.

Ahora analicemos el Oldpeak (depresión del ST inducida por el ejercicio relativo al descanso):

```
#Oldpeak study
datHealthy<-subset(dataPeak, HeartDisease == 0)
datUnhealthy<-subset(dataPeak, HeartDisease == 1)
hist(main="Oldpeak de gente con enfermedad del corazón",xlab="Oldpeak", as.numeric(unlist(datUnhealthy[
```



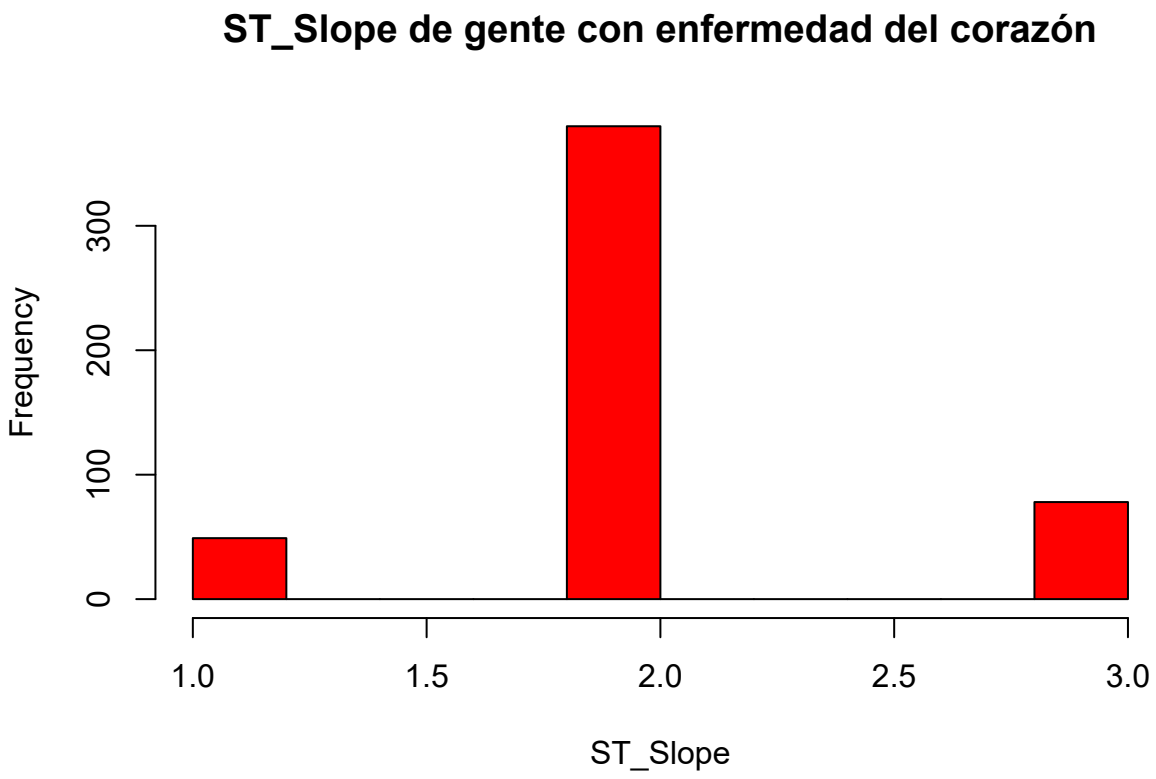
```
hist(main="Oldpeak de gente sana y enferma",xlab="Oldpeak", as.numeric(unlist(datUnhealthy[,2])), col='
hist(as.numeric(unlist(datHealthy[,2])),col='blue', add=TRUE)
```



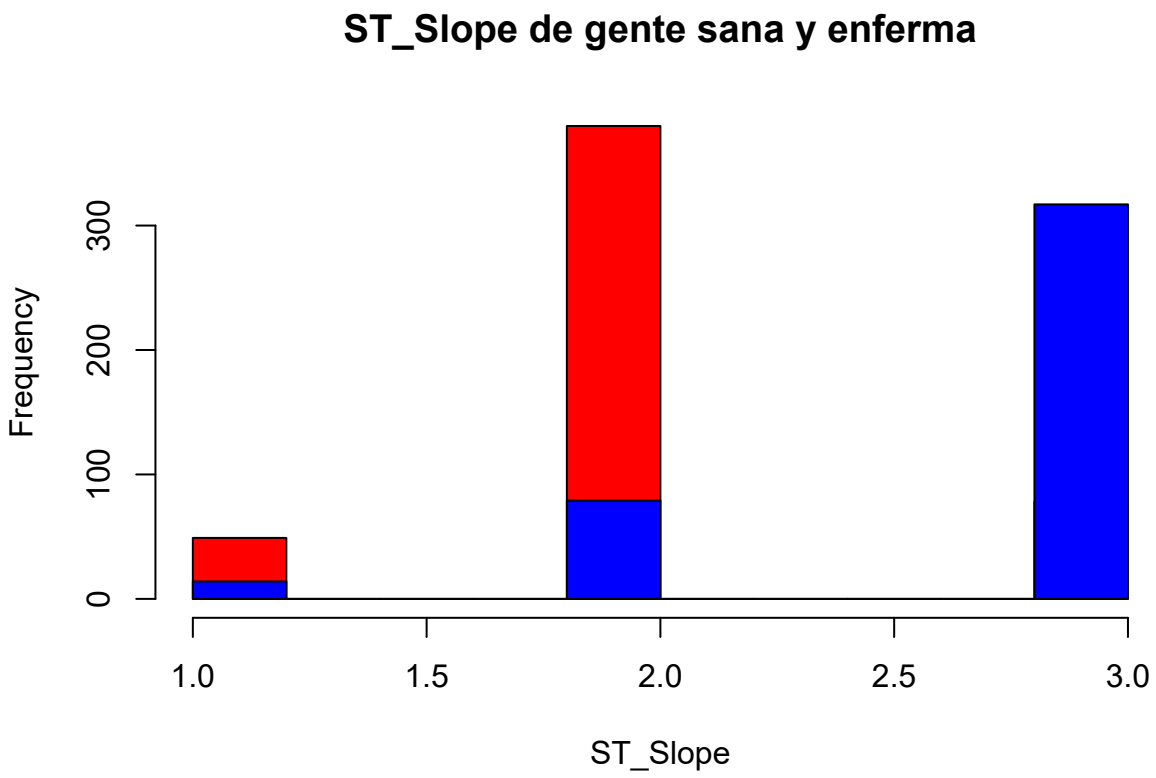
En el caso del Oldpeak si podemos ver de nuevo una relación. De hecho, se puede ver como a medida que el oldpeak es mayor aumenta su frecuencia en el caso de los enfermos mientras que en el caso de los sanos pasa lo opuesto, cuando el oldpeak es menor es cuando aumenta su frecuencia. Por ello, el oldpeak parece ser un buen predictor.

Ahora analicemos el ST_Slope (pendiente del segmento ST de ejercicio máximo, 1 es cuesta abajo, 2 es recto y 3 es cuesta arriba):

```
#ST_slope study
datHealthy<-subset(dataSlope, HeartDisease == 0)
datUnhealthy<-subset(dataSlope, HeartDisease == 1)
hist(main="ST_Slope de gente con enfermedad del corazón",xlab="ST_Slope", as.numeric(unlist(datUnhealthy
```



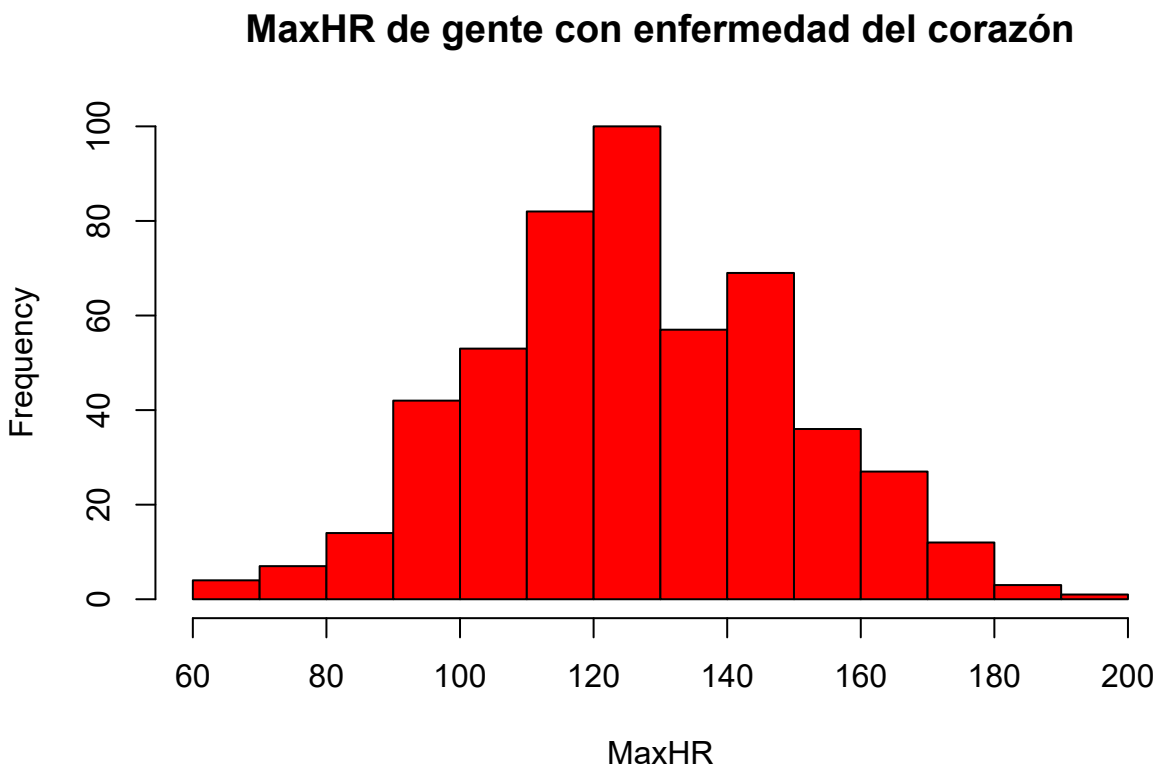
```
hist(main="ST_Slope de gente sana y enferma",xlab="ST_Slope",as.numeric(unlist(datUnhealthy[,2])), col=
hist(as.numeric(unlist(datHealthy[,2])),col='blue', add=TRUE)
```



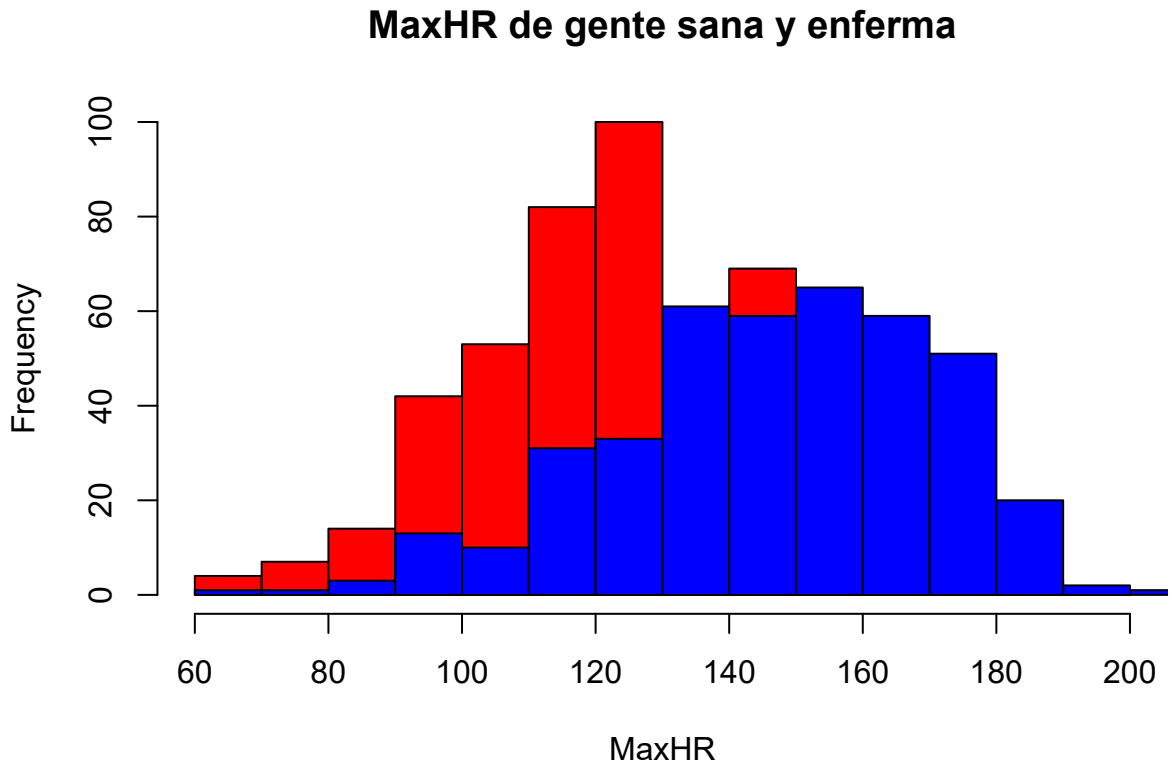
Los resultados son bastante claros en el ST_Slope. Se puede ver como dentro de los enfermos la mayoría de enfermos estan en recto y pocos casos en cuesta arriba o cuesta abajo. Por otro lado en el caso de los sanos los resultados son justo los contrarios. En recto hay muy poca gente y cuesta arriba mucha. Parece que el ST_Slope es un buen predictor de enfermedades del corazón.

Para finalizar analicemos el MaxHR (máxima pulsación del corazón).

```
#MaxHR study
datHealthy<-subset(dataMaxHR, HeartDisease == 0)
datUnhealthy<-subset(dataMaxHR, HeartDisease == 1)
hist(main="MaxHR de gente con enfermedad del corazón",xlab="MaxHR", as.numeric(unlist(datUnhealthy[,2]))
```



```r
hist(main="MaxHR de gente sana y enferma",xlab="MaxHR",as.numeric(unlist(datUnhealthy[,2])), col='red')
hist(as.numeric(unlist(datHealthy[,2])),col='blue', add=TRUE)
```



En el caso de la máxima puntuación se puede ver como gente con mayores niveles de pulsación tiende a estar más sana mientras que gente enferma tiende a tener menores niveles de pulsación. Por lo que también parece un buen indicador de enfermedades del corazón.

Conclusion

Parece que la *angina*, *oldpeak* y *ST_Slope* son las variables que más claramente predicen las enfermedades del corazón. Además la máxima pulsación *MaxHR* y la edad *age* tambien parecen bastante buenas predictoras.

Modelo predictivo

Después de haber explorado la base de datos y haber extraido conclusiones, es hora de intentar predecir dados unos datos iniciales si un paciente tiene o tendrá enfermedades del corazón. Para ello, utilizaremos el conocido algoritmo de K-nn.

Aprendizaje supervisado K-nn

A continuación, dividiremos el conjunto de datos en train y test. Tomaremos el 80% de los datos para entrenar y el 20% restante para test. Utilizaremos la librería "caret".

```
library(caret)
```

```
## Loading required package: lattice
```

```
data[,12]<-factor(data[,12])

# Obtener los índices del conjunto de train
trainIndex <- createDataPartition(data[,12], p = .8, list = FALSE, times = 1)
# Seleccionar las instancias correspondientes
data.train <- data[trainIndex, ]
data.test <- data[-trainIndex, ]
```

Es importante comprobar que nuestras clases estén correctamente distribuidas, para ello podemos ver la proporción de HeartDisease en los datos generales, en el train y en el test:

```
# Distribución original
prop.table(table(data[,12]))
```

```
##
##         1         2
## 0.4471101 0.5528899
```

```
# Distribución en el conjunto de train
prop.table(table(data.train[,12]))
```

```
##
##         1         2
## 0.4604905 0.5395095
```

```
# Distribución en el conjunto de test
prop.table(table(data.test[,12]))
```

```
##
##         1         2
## 0.3934426 0.6065574
```

Entrenamiento y Resultados de K-nn

Vamos a crear un modelo para clasificar mediante k-NN.

```
train.ctrl <- trainControl(method = "none")
knn_fit <- train(HeartDisease ~., data = data.train,
                 method = "knn",
                 trControl=train.ctrl,
                 preProcess = c("center", "scale"),
                 tuneGrid = data.frame(k=1))
```

```
## Warning in train.default(x, y, weights = w, ...): You are trying to do
## regression and your outcome only has two possible values Are you trying to do
## classification? If so, use a 2 level factor as your outcome column.
```

```
knn_fit
```

```
## k-Nearest Neighbors
##
## 734 samples
##  11 predictor
##
## Pre-processing: centered (11), scaled (11)
## Resampling: None
```

Para entrenar el clasificador k-NN, debemos pasarle "method = knn" al método train(). "HeartDisease ~." indica que HeartDisease será la clase. En este caso es conveniente normalizar, ya que tenemos valores numéricos que pueden tener diferentes distribuciones y vamos a utilizar distancias. También hemos establecido el número de vecinos (k=1) mediante el parámetro "tuneGrid". En nuestro caso, como solo tenemos HeartDisease con valores de 1 o 2 nuestra k será igual a 1.

Una vez construido el modelo lo evaluamos con los datos del conjunto test:

```
knnPredict <- predict(knn_fit, newdata = data.test)
knnPredict
```

```
##   [1] 1 1 2 2 1 1 1 1 1 1 1 1 2 1 2 1 2 2 1 2 1 1 1 1 1 1 1 1 1 2 1 2 1 1 2
##  [38] 2 1 1 1 1 2 2 2 2 2 1 1 1 2 1 2 2 2 2 2 2 1 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2
##  [75] 2 2 2 2 2 2 1 2 2 2 2 2 2 2 1 2 1 2 2 1 2 2 2 2 1 2 2 2 2 1 1 2 2 2 2 2 2
## [112] 2 2 2 1 2 2 2 1 2 1 2 2 1 1 1 1 2 1 1 1 1 2 2 2 2 1 2 1 1 2 2 2 2 2 1 2 1
## [149] 1 2 1 2 1 2 1 2 1 2 2 2 1 1 1 2 1 1 1 1 2 2 2 1 1 2 1 2 2 1 1 1 2 1 1
```

Utilizamos la matriz de confusión para ver los resultados estadísticos:

```
confusionMatrix(table(knnPredict, data.test[,12]))
```

```
## Confusion Matrix and Statistics
##
##
## knnPredict  1  2
##          1 63 18
##          2 11 91
```

```
##
##                Accuracy : 0.8415
##                  95% CI : (0.7804, 0.8912)
##     No Information Rate : 0.5956
##     P-Value [Acc > NIR] : 5.335e-13
##
##                   Kappa : 0.6759
##
##  Mcnemar's Test P-Value : 0.2652
##
##             Sensitivity : 0.8514
##             Specificity : 0.8349
##          Pos Pred Value : 0.7778
##          Neg Pred Value : 0.8922
##              Prevalence : 0.4044
##          Detection Rate : 0.3443
##    Detection Prevalence : 0.4426
##       Balanced Accuracy : 0.8431
##
##        'Positive' Class : 1
##
```

Como conclusión, se puede apreciar como con K-nn hemos obtenido una precisión del 84.15% lo que indica que con una fiabilidad bastante alta se pueden predecir enfermedades del corazón. Es importante destacar que en el ámbito médico una fiabilidad del 84.15% no es admisible ya que no se puede utilizar como diagnóstico fiable. Por eso, otros posibles métodos de clasificación o una clasificación con k-nn optimizando los parámetros es necesaria para poder obtener una gran precisión. Es muy posible que utilizando el método y los parámetros adecuados se pueda obtener una puntuación mayor.