Práctica 2: Tipologia y ciclo de vida de los datos

Contents

Descripción del dataset

Para esta práctica he escogido el dataset Heart Disease, disponible en UCI Machine Learning Repository (https://archive.ics.uci.edu/ml/datasets/Heart+Disease). Presenta información sobre pacientes hospitalarios y sus enfermedades cardícacas.

Está fechado en 1988 y contiene cuatro bases de datos: Cleveland, Hungria, Suiza y Long Beach. En concreto, utilizaremos la base de datos de Cleveland.

Se trata de un conjunto de datos con atributos categóricos, de números enteros y reales. Así mismo, contiene valores faltantes.

Presenta 303 registros y los siguiente 14 atributos:

- Edad: Edad del sujeto
- Sexo: Sexo del sujeto.
  - 0 para femenino, 1 para masculino
- Tipo de dolor de pecho
  - 1 para angina de pecho típica
  - 2 para angina de pecho atípica
  - 3 para dolor no proveniente de angina
  - 4 para angina asintomática
- Presión sanguinea en reposo
  - Expresada en mmHg
- Colesterol en suero
  - Expresado en mg/dl
- Nivel de azúcar en sangre en ayunas. Relativo a 120 mg/dl
  - 0 para superior a 120 mg/dl

  - 1 para inferior a 120 mg/dl

- ECG en reposo.
  - 0 para normal
  - 1 para Anormalidad en la onda ST-Tx
  - 2 para Hipertrofia del ventriculo izquierdo
- Maxíma frecuencia cardíaca del sujeto.
- Angina inducida por el ejercicio
  - 0 para si
  - 1 para no
- Depresión en la onda ST inducida por ejercicio relativa al reposo. Se calcula sobre el gráfico del electrocardiograma y se mide en milímetros
- Pico de segmento ST durante ejercicio
  - 1 para ascendente
  - 2 para plano
  - 3 para descendente
- Número de vasos sánguineos mayores vistos en fluoroscopia.
  - Entre 0 y 3
- Forma de talasemia
  - 3 para normal
  - 6 defecto irreversible
  - 7 para defecto reversible
- Diagnóstico de enfermedad cardíaca.
  - de 0 a 5, siendo 0 ausencia y 5 enfermedad cardiaca grave

A partir de esta conjunto de datos se plantea la problemática de determinar si los hombres sufren más enfermedades cardíacas que las mujeres. Además, se podrá proceder a crear modelos de regresión que permitan predecir la gravedad de la enfermedad cardiaca a partir del estado de salud del paciente y contrastes de hipótesis que ayuden a identificar propiedades interesantes de las muestras que puedan ser inferidas respecto a la población.

Estos análisis adquieren una gran relevancia en el ámbito médico. Por ejemplo, a la llegada de un paciente al departamento de cardiología de un hospital, se podría valer de estos análisis para utilizarlos como soporte a la hora de diagnosticar y tratar al paciente.

Limpieza de los datos

Antes de nada, configuraremos el directorio de trabajo para que sea la carpeta desde dónde se ejecuta este notebook.

```
#Set del directorio de trabajo
setwd(dirname(rstudioapi::getSourceEditorContext()$path))
getwd()
```

```
## [1] "/Users/daniel/Personal/heart-data-analysis/src"
```

Previamente a la limpieza de los datos, cargamos el fichero en el que se encuentran. Dado que no tiene cabeceras, se las añadiremos también

```r
# Carga de datos
data <- read.csv("../data/processed.cleveland.data")

# Columna de nombres de variables

columns <- c("Age",
             "Sex",
             "Chest_Pain_Type",
             "Resting_Blood_Pressure",
             "Serum_Cholesterol",
             "Fasting_Blood_Sugar",
             "Resting_ECG",
             "Max_Heart_Rate_Achieved",
             "Exercise_Induced_Angina",
             "ST_Depression_Exercise",
             "Peak_Exercise_ST_Segment",
             "Num_Major_Vessels_Flouro",
             "Thalassemia",
             "Diagnosis_Heart_Disease")


#Agregar columnas al dataset
colnames(data) <- columns

# Mostrar primeras filas

head(data[,1:5])
```

```
##   Age Sex Chest_Pain_Type Resting_Blood_Pressure Serum_Cholesterol
## 1  67   1               4                    160               286
## 2  67   1               4                    120               229
## 3  37   1               3                    130               250
## 4  41   0               2                    130               204
## 5  56   1               2                    120               236
## 6  62   0               4                    140               268
```

Elementos vacios

Encontramos algunos elementos vacios en los datos, denotados por el carácter '?'. Para tratarlos, primeramente los substituiremos por "NA", de tal forma que R pueda identificarlos como valores faltantes

```r
# Sustitución de "?" por NA
data[data == "?"] <- NA

# Visualización de datos faltantes

sapply(data, function(x) sum(is.na(x)))
```

```
##                       Age                       Sex          Chest_Pain_Type
##                         0                         0                         0
##   Resting_Blood_Pressure         Serum_Cholesterol       Fasting_Blood_Sugar
##                         0                         0                         0
```

```
##              Resting_ECG  Max_Heart_Rate_Achieved  Exercise_Induced_Angina
##                        0                        0                        0
##   ST_Depression_Exercise Peak_Exercise_ST_Segment Num_Major_Vessels_Flouro
##                        0                        0                        4
##              Thalassemia  Diagnosis_Heart_Disease
##                        2                        0
```

Vemos que el conjunto de datos contiene 2 registros faltantes para el campo "Thalassemia" y 4 para el campo "Num_Majors_Vessels_Fluoro".

Dado que són relativamente pocos, procederemos eliminar las filas con estos valores faltantes.

```
# Reescribimos el dataframe "data" por el mismo sin valores faltantes.
data <- na.omit(data)

# Visualizamos el número de registros tras borrar aquellos con valores faltantes
nrow(data)
```

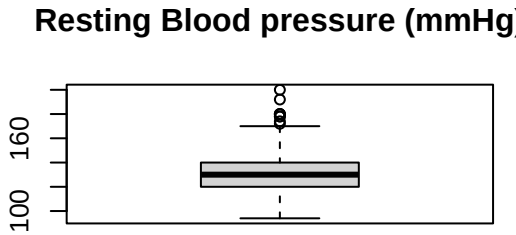
```
## [1] 296
```

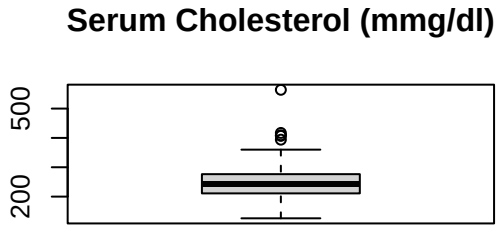
Esta vez, el dataset se reduce de 303 a 296 registros.

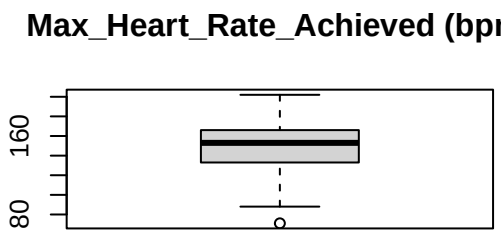
Outliers

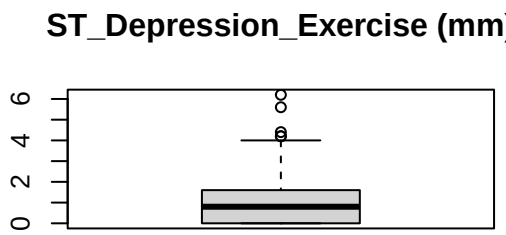
Para encontrar los valores extremos, los representaremos graficamente mediante la función boxplot, la cual también nos los devuelve

```
par(mfrow = c(2, 2))
blood_pressure_outliers <- boxplot(data$Resting_Blood_Pressure, main = 'Resting Blood pressure (mmHg)' )
serum_cholesterol_outliers <- boxplot(data$Serum_Cholesterol, main = 'Serum Cholesterol (mmg/dl)' )
heart_rate_outliers <- boxplot(data$Max_Heart_Rate_Achieved, main = 'Max_Heart_Rate_Achieved (bpm)' )
st_depression_outliers <- boxplot(data$ST_Depression_Exercise, main = 'ST_Depression_Exercise (mm)' )
```









Encontramos diversos outliers en el conjunto de datos. Sin embargo, dado que mis conocimientos en cardiología són nulos, no puedo afirmar si se trata de valores correctos, erróneos o incompatibles con la vida. De esta manera, voy a optar por no eliminar los valores extremos.

Análisis de los datos

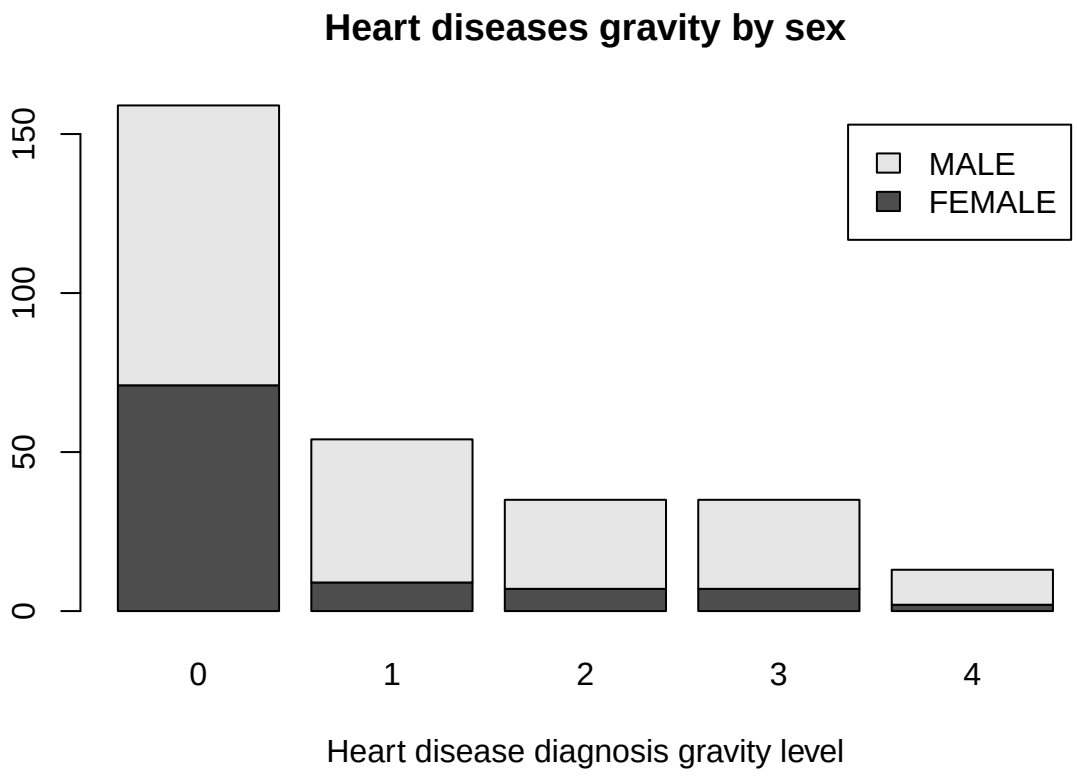
Selección de datos

Será interesante seleccionar los datos por sexo. De esta manera, más adelante, podremos comparar algunas variables en función de este parámetro. Lo realizamos mediante la función subset de R. Así mismo, crearemos la variable Sex_Label a partir del campo Sex, dónde 0 y 1 corresponden a 'FEMALE' y 'MALE' respectivamente. Esto nos permitira una mejor visualización de los datos

```
data$Sex_Label<-ifelse(data$Sex>0,'MALE','FEMALE')
female_data <- subset(data, Sex_Label == 'FEMALE')
male_data <- subset(data, Sex_Label == 'MALE')
```

Podemos ver como se distribuye la gravedad de la enfermedad cardíaca por sexo. Observamos a simple vista, que en los niveles de mayor gravedad, la mayoría de afectados son hombres.

```
gravity_by_sex <- table(data$Sex_Label, data$Diagnosis_Heart_Disease)
barplot(gravity_by_sex, legend=rownames(gravity_by_sex), xlab='Heart disease diagnosis gravity level', m
```



Comprobación de la normalidad y homogeneidad de la varianza

Comprobaremos estas condiciones para las siguientes variables numéricas:

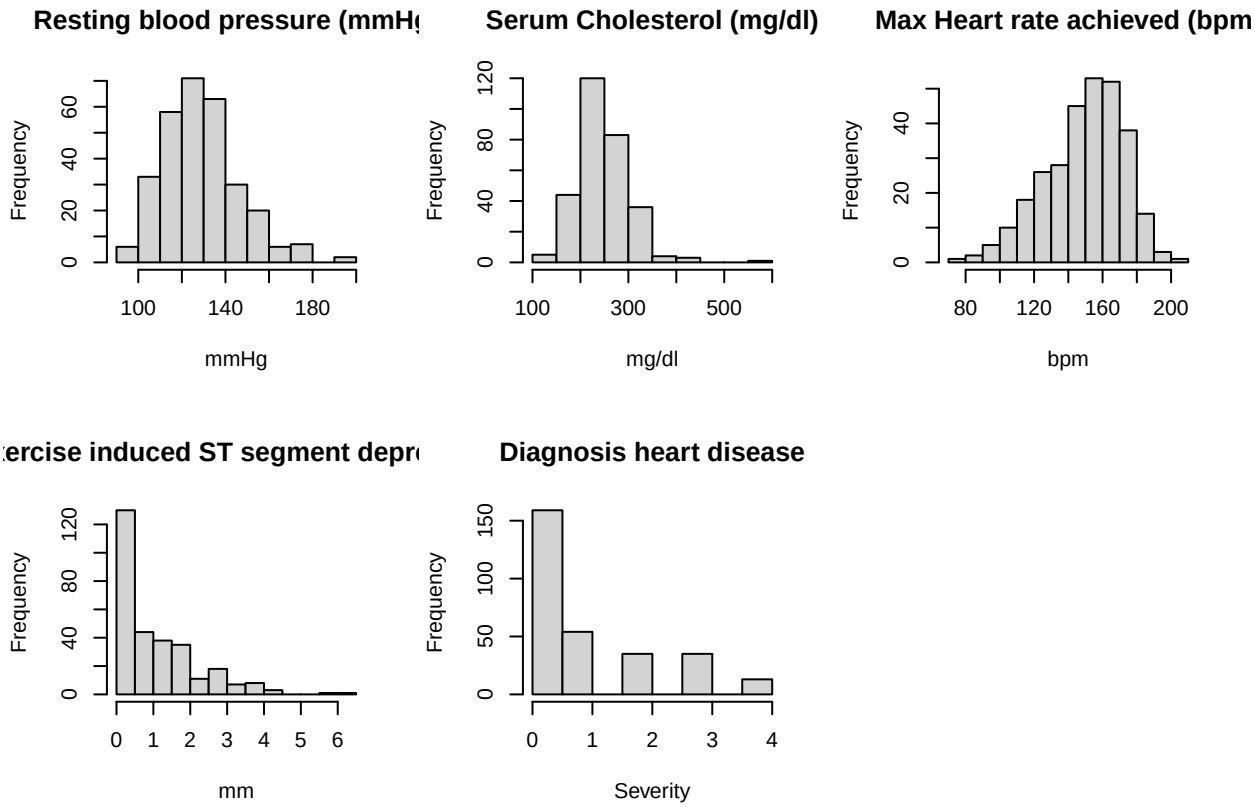
- Resting blood pressure
- Serum cholesterol
- Max heart rate achieved
- Exercise induced ST wave Depression
- Diagnosis heart disease

Con el objetivo de verificar la suposición de normalidad, emplearemos el test de Shapiro-Wilk. Se trata de unos métodos más portentes para constrastar la normalidad de una muestra. Asumiendo como hipótesis nula que la población está distribuida normalmente, si el p-valor es menor al nivel de significancia, generalmente 0.05, entonces la hipótesis nula es rechazada y se concluye que los datos no cuentan con una distribución normal. Por otra parte, si el p-valor es superior al nuvel de significancia de 0.05, se asume que los datos siguen una distribución normal.

Representamos gráficamente las variables seleccionadas

```
par(mfrow = c(2, 3))

hist(data$Resting_Blood_Pressure, main='Resting blood pressure (mmHg)', xlab ='mmHg')
hist(data$Serum_Cholesterol, main='Serum Cholesterol (mg/dl)', xlab ='mg/dl')
hist(data$Max_Heart_Rate_Achieved, main='Max Heart rate achieved (bpm)', xlab ='bpm')
hist(data$ST_Depression_Exercise, main='Exercise induced ST segment depression', xlab ='mm')
hist(data$Diagnosis_Heart_Disease, main='Diagnosis heart disease', xlab ='Severity')
```



Empleamos el test de Shapiro wilk para contrastar la normalidad de las variables

```
shapiro.test(data$Resting_Blood_Pressure)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data$Resting_Blood_Pressure
## W = 0.96614, p-value = 2.041e-06
```

```
shapiro.test(data$Serum_Cholesterol)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data$Serum_Cholesterol
## W = 0.94854, p-value = 1.14e-08
```

```
shapiro.test(data$Max_Heart_Rate_Achieved)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data$Max_Heart_Rate_Achieved
## W = 0.97652, p-value = 8.874e-05
```

```
shapiro.test(data$ST_Depression_Exercise)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data$ST_Depression_Exercise
## W = 0.84658, p-value < 2.2e-16
```

```
shapiro.test(data$Diagnosis_Heart_Disease)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data$Diagnosis_Heart_Disease
## W = 0.75596, p-value < 2.2e-16
```

Observamos que ninguna variable presenta normalidad, ya que sus p-valores son inferiores al valor de significancia de 0.05

Para comprobar la homocedasticidad, deberemos emplear el test de Fligner-Kulleen, ya que los datos no cumplen con la condición de normalidad. En esta prueba, la hipótesis nula asume igualdad de varianzas en los diferentes grupos de dato. Debido a esto, los p-valroes inferiores al nivel de significancia indicarán heterocedasticidad.

```
fligner.test(Resting_Blood_Pressure ~ Sex, data=data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Resting_Blood_Pressure by Sex
## Fligner-Killeen:med chi-squared = 1.1598, df = 1, p-value = 0.2815
```

```
fligner.test(Serum_Cholesterol ~ Sex, data=data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Serum_Cholesterol by Sex
## Fligner-Killeen:med chi-squared = 8.8518, df = 1, p-value = 0.002928
```

```
fligner.test(Max_Heart_Rate_Achieved ~ Sex, data=data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Max_Heart_Rate_Achieved by Sex
## Fligner-Killeen:med chi-squared = 7.1842, df = 1, p-value = 0.007355
```

```
fligner.test(ST_Depression_Exercise ~ Sex, data=data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  ST_Depression_Exercise by Sex
## Fligner-Killeen:med chi-squared = 8.5209, df = 1, p-value = 0.003511
```

```
fligner.test(Diagnosis_Heart_Disease ~ Sex, data=data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Diagnosis_Heart_Disease by Sex
## Fligner-Killeen:med chi-squared = 17.769, df = 1, p-value = 2.494e-05
```

Vemos que la variable Resting_blood_Pressure presenta varianzas estadísticamente similares para los diferentes sexos, ya que su p-valor de 0.28 es superior al valor de significancia 0.05.

Sin embargo, las variables Serum_Cholesterol, Max_Heart_Rate_Achieved y ST_Depression_Exercise presentan heterocedasticidad por sexo.

Pruebas estadísticas

Correlación

Procederemos a realizar un análisis de correlación entre las distintas variables para determinar cuáles de ellas ejercen una mayor influencia sobre la gravedad de la enfermedad cardiáca. Dado que los datos no siguen una distribución normal, nos fijaremos en el coeficiente de correlación de Spearman

```
numeric_data <- data[, c('Resting_Blood_Pressure', 'Age', 'Serum_Cholesterol', 'Max_Heart_Rate_Achieved

res <- cor(numeric_data, method = 'spearman')
round(res, 2)
```

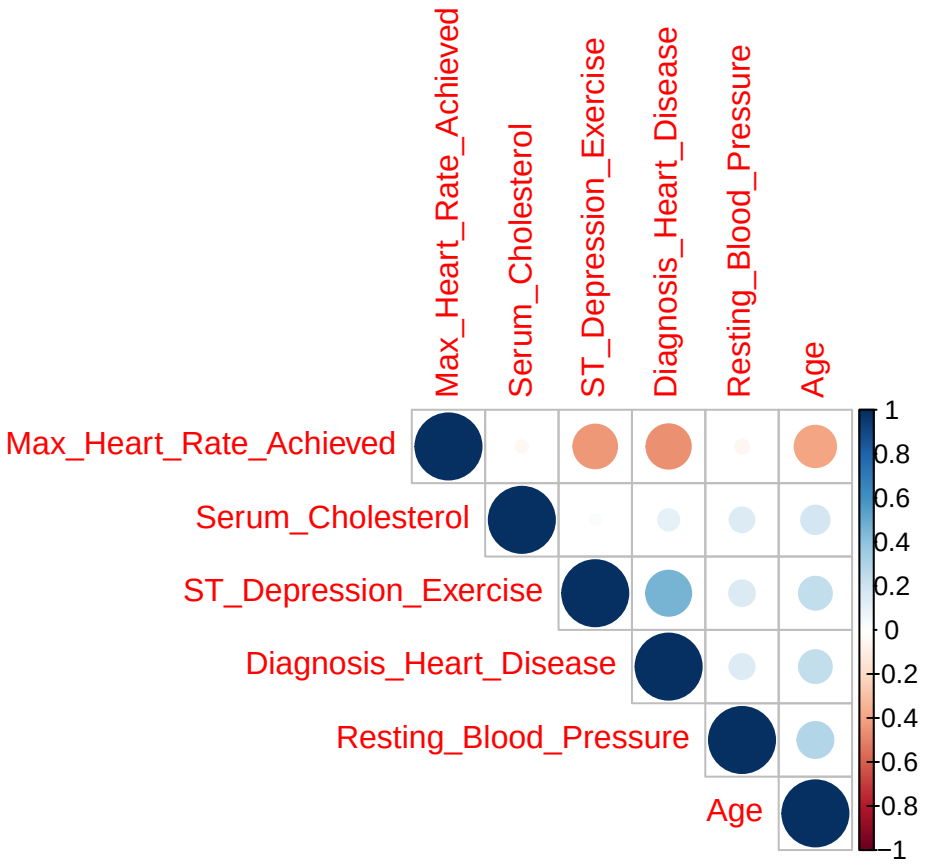
```
##                         Resting_Blood_Pressure   Age Serum_Cholesterol
## Resting_Blood_Pressure                    1.00  0.30              0.14
## Age                                       0.30  1.00              0.18
## Serum_Cholesterol                         0.14  0.18              1.00
## Max_Heart_Rate_Achieved                  -0.05 -0.39             -0.04
## ST_Depression_Exercise                    0.15  0.25              0.03
## Diagnosis_Heart_Disease                   0.14  0.25              0.10
##                         Max_Heart_Rate_Achieved ST_Depression_Exercise
## Resting_Blood_Pressure                    -0.05                   0.15
## Age                                       -0.39                   0.25
## Serum_Cholesterol                         -0.04                   0.03
## Max_Heart_Rate_Achieved                    1.00                  -0.44
## ST_Depression_Exercise                    -0.44                   1.00
## Diagnosis_Heart_Disease                   -0.45                   0.47
##                         Diagnosis_Heart_Disease
## Resting_Blood_Pressure                     0.14
```

```
## Age                                0.25
## Serum_Cholesterol                  0.10
## Max_Heart_Rate_Achieved           -0.45
## ST_Depression_Exercise             0.47
## Diagnosis_Heart_Disease            1.00
```

```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
corrplot(res, type="upper", order="hclust")
```



No se observan grandes correlaciones en el conjunto de datos, ya que ninguna supera el valor de 0.5. Sin embargo, las variables que más correlacion tienen con la gravedad de la enfermedad cardíaca (Diagnosis_Heart_Disease) són la depresión de la onda ST inducida por el ejercicio (ST_Depression_Exercise) y la máxima frecuencia cardíaca obtenida (Max_Heart_Rate_Achieved)

Regresión logistica

Realizaremos un modelo de regresión lineal para predecir la gravedad de la enfermedad cardíaca a partir de las siguienntes variables: Age, Sex, Resting_blood_pressure, Serum_cholesterol,Max_heart_rate achieved y ST_Depression_exercise

Primeramente, crearemos una nueva variable llamada 'target' a partir de Diagnosis_Heart_disease. De esta forma, si Diagnosis_Heart_disease > 0, el paciente presenta enfermedad cardíaca y 'target' $= 1$.

```r
#Variable dicotómica target
data$target<-ifelse(data$Diagnosis_Heart_Disease>0,TRUE,FALSE)
```

Seeguidamente dividiremos el conjunto de datos en subconjuntos de entreno y test, dedicando un 80% de este al entreno, y el 20% restante a test del modelo. Para ello, utilizaremos la libreria caTools.

```r
library(caTools)
set.seed(123)
split <- sample.split(data$Diagnosis_Heart_Disease, SplitRatio = 0.8)
train <- subset(data, split == TRUE)
test <- subset(data, split == FALSE)
```

Seguidamente, obtendremos el modelo con las variables seleccionadas

```r
model=glm(target ~ Age+Sex+Resting_Blood_Pressure+Serum_Cholesterol, data=train,family = "binomial")
summary(model)
```

```
##
## Call:
## glm(formula = target ~ Age + Sex + Resting_Blood_Pressure + Serum_Cholesterol,
##     family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9259  -1.0232  -0.3769   0.9921   1.9590
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)            -8.181708   1.626675  -5.030 4.91e-07 ***
## Age                     0.070037   0.018685   3.748 0.000178 ***
## Sex                     1.812126   0.367401   4.932 8.13e-07 ***
## Resting_Blood_Pressure  0.011465   0.009006   1.273 0.203016
## Serum_Cholesterol       0.005562   0.003073   1.810 0.070336 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 325.79  on 235  degrees of freedom
## Residual deviance: 277.21  on 231  degrees of freedom
## AIC: 287.21
##
## Number of Fisher Scoring iterations: 4
```

A continuación, realizaremos predicciones con el modelo y crearemos la matrix de confusión.

```r
predicciones = predict(model, type='response')
table(train$target, predicciones>0.5)
```

```
##
##         FALSE TRUE
##   FALSE    90   37
##   TRUE     38   71
```

Calculamos la precisión del modelo. Obtenemos un valor de 0.68, no muy alto, pero estadísticamente significante. Seguramente podríamos mejorar el modelo para obtener un modelo mejor, buscando qué variables tienen mayor peso y descartando aquellas que no lo tengan.

```
(90+71)/236
```

```
## [1] 0.6822034
```

Validación cruzada

En la validación cruzada los datos originales se dividen en un número de subconjuntos mutuamente exclusivos de tamaños similares. De esta forma, el entrenamiento y evaluación del modelo se realizan tantas veces como subconjuntos hayamos escogido, a partir de todas las combinaciones posibles entre estos.

Encontraremos el rendimiento del modelo como el promedio del rendimiento entre todas las evaluaciones.

Una de las ventajas de la validación cruzada es que nos permite evaluar el sobreajuste del modelo. Esto quiere decir que el modelo se ajusta muy bien a los datos con los que ha sido entrenado, pero no es capaz de extrapolar nuevos resultados adecuadamente. Mediante validación cruzada, podemos ver si el modelo rinde igual con todas las combinaciones de datos, o si se ha ajustado demasiado a alguna de ellas, en cuyo caso deberemos replantearlo.

Primeramente definiremos las especificaciones para realizar la validación cruzada mediante la función trainControl del paquete caret, el cual contiene funciones útiles en analítica predictiva y varios modelos y algoritmos de machine learning

Haremos validación cruzada en 5 folds, ya que el tamaño del dataset puede no ser suficiente para realizar muchos subconjuntos de datos. Mediante el parámetro savePredictions = all indicaremos que queremos almacenar todas las predicciones para cada iteración. Así mismo, el parámetro classProbs=TRUE indica que queremos obtener también las probabilidades de cada clase, además del valor predicho.

```
library(caret)
```

```
## Loading required package: ggplot2
```

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

```
# Configuración de validación cruzada con 5 folds
kfoldSpecs <- trainControl(method="cv", number=5, savePredictions="all", classProbs=TRUE, verboseIter=TR
```

Seguidamente, utilizaremos la función train del paquete caret para entrenar el modelo con la validación cruzada definida previamente. Especificaremos la función, los datos, el método (glm, regresión logística) y pasaremos al objeto de configuración kFoldsSpecs definido

```
# Preprocesado necesario: convertir la variable target en factor. Requerido por el paquete 'caret'
train$target<-ifelse(train$target==TRUE,'yes','no')
train$target<-as.factor(train$target)

test$target<-ifelse(test$target==TRUE,'yes','no')
test$target<-as.factor(test$target)
```

Tras haber realizado este procesamiento, construimos el modelo y aplicamos la validación cruzada.

```
# Se contruye el modelo
kfoldModel <- train(target ~ Age+Sex+Resting_Blood_Pressure+Serum_Cholesterol, data=train,
                method="glm",
                family="binomial",
                trControl=kfoldSpecs)
```

```
## + Fold1: parameter=none
## - Fold1: parameter=none
## + Fold2: parameter=none
## - Fold2: parameter=none
## + Fold3: parameter=none
## - Fold3: parameter=none
## + Fold4: parameter=none
## - Fold4: parameter=none
## + Fold5: parameter=none
## - Fold5: parameter=none
## Aggregating results
## Fitting final model on full training set
```

```
# Se muestra el resumen
print(kfoldModel)
```

```
## Generalized Linear Model
##
## 236 samples
##   4 predictor
##   2 classes: 'no', 'yes'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 189, 188, 190, 188, 189
## Resampling results:
##
##   Accuracy   Kappa
##   0.6736933  0.3443385
```

Observamos que hemos realizado 5 folds, de 189 registros cada uno, excepto uno de 188 y hemos obtenido una precisión del 0.66.

Con este nuevo modelo creado a partir de validación cruzada, realizamos nuevas predicciones sobre los datos de test y hallamos la matriz de confusión.

```
kfoldPredictions <- predict(kfoldModel, newdata=test)
confusionMatrix(data=kfoldPredictions, test$target)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction no yes
##        no  23  14
##        yes  9  14
##
```

```
##                Accuracy : 0.6167
##                  95% CI : (0.4821, 0.7393)
##     No Information Rate : 0.5333
##     P-Value [Acc > NIR] : 0.1218
##
##                   Kappa : 0.2212
##
##  Mcnemar's Test P-Value : 0.4042
##
##             Sensitivity : 0.7188
##             Specificity : 0.5000
##          Pos Pred Value : 0.6216
##          Neg Pred Value : 0.6087
##              Prevalence : 0.5333
##          Detection Rate : 0.3833
##    Detection Prevalence : 0.6167
##       Balanced Accuracy : 0.6094
##
##        'Positive' Class : no
##
```

Observamos que obtenemos un resultado similar al anterior, con una precisión del 0.61. La matrix de confusión nos permite obtener también muchas otras métricas, como la sensibilidad y la especificidad. No obtenemos un modelo muy bueno, pero es mejor que un clasificador aleatorio. Podríamos mejorar el modelo de diversas formas. Por ejemplo, podríamos incluir mas variables, o aumentar el tamaño del dataset.

Inferencia

Comprobaremos si existen diferencias significativas entre la gravedad de la enfermedad cardíaca (Diagnosis_Heart_Disease) por sexo. Para ello emplearemos el test de Wilcoxon, debido que los datos no cumplen las condiciones de normalidad y homocedasticidad

- H_0 = Los hombres presentan mayor gravedad de enfermedad cardíaca
- H_1 = No existen diferencias significativas en la gravedad de la enfermedad cardíaca entre hombres y mujeres

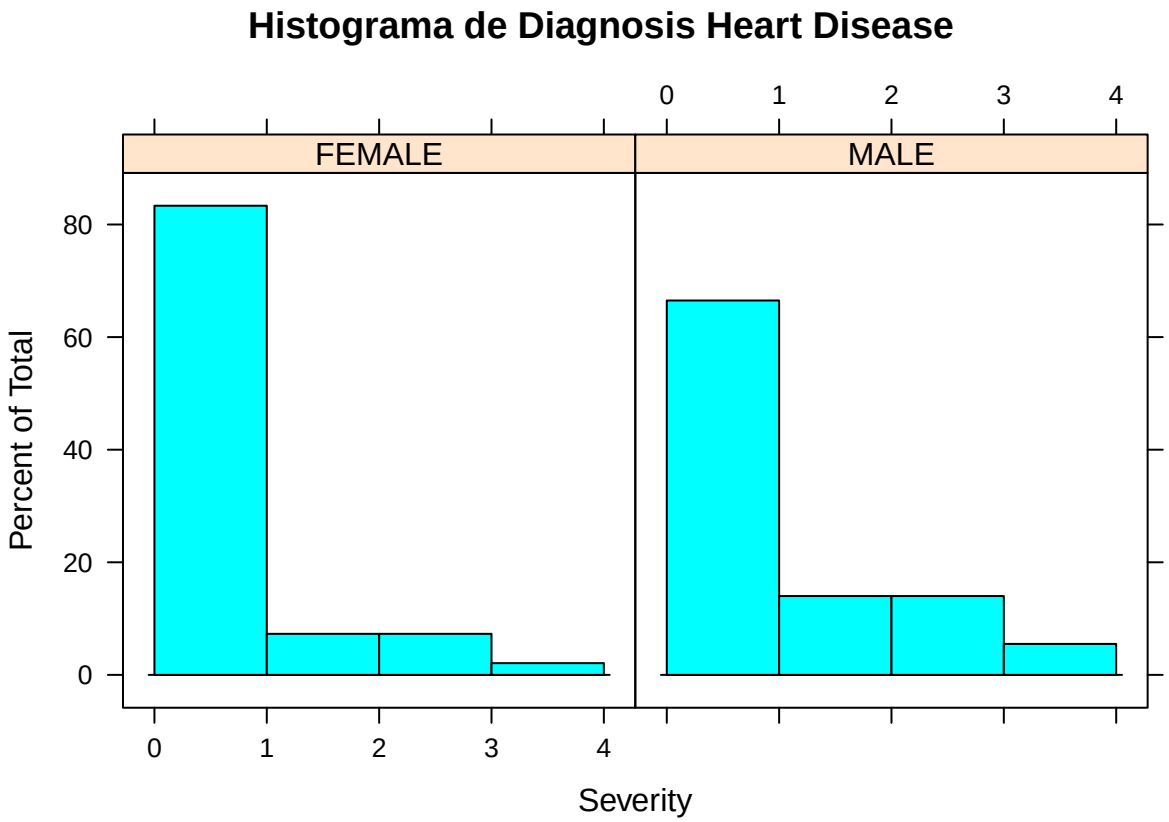
```
wilcox.test(Diagnosis_Heart_Disease ~ Sex, data=data)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  Diagnosis_Heart_Disease by Sex
## W = 6761.5, p-value = 6.672e-06
## alternative hypothesis: true location shift is not equal to 0
```

En este caso sí se observan diferencias significativas en la gravedad de la enfermedad cardíaca por sexo, ya que el p-valor obtenido por el test es inferior al valor de significancia 0.05. Aceptaremos la hipótesis nula.

Visualizamos los histogramas de Diagnosis_heart_disease para hombres y mujeres y vemos que los hombres presentan un diagnóstico más grave de enfermedad cardíaca, como nos confirma el test de Wilcoxon realizado previamente.

```
library(lattice)
histogram(~Diagnosis_Heart_Disease|Sex_Label, data=data, main="Histograma de Diagnosis Heart Disease",
```



Resolución del problema

Gracias al análisis y modelado de los datos que hemos realizado, podemos extraer las siguientes conclusiones, entre otras.

Primeramente, observamos que no existen correlaciones muy marcadas entre las variables. Podríamos destacar la correlación entre ST_Depression_Exercise y Heart Disease, con un valor de 0.47, y la correlación entre Max_Heart_Rate_achieved y Hart_Disease_Diagnósis, con un valor de -0.47.

De esta forma, podemos afirmar que mientras mayor es la depresión de la onda ST inducida por el ejercicio, mayor suele ser la gravedad de la enfermedad cardíaca. También podemos afirmar que a mayor frecuencia cardíaca máxima obtenida, menor suele ser la gravedad de la enfermedad cardíaca.

Por otra parte, el modelo de regresión logística obtenido permite predecir con una precisión aceptable si el paciente presentará enfermedad cardíaca a partir de su edad, sexo, colesterol y presión sanguínea en reposo. De esta forma, con solamente cuatro variables bastante básicas hemos obtenido un modelo qué, sin tener un rendimiento increíble, muestra el potencial de estas técnicas estadísticas.

Así mismo, hemos realizado una inferencia estadística mediante el test de Wilcoxon, ya que los datos no cumplen las condiciones de normalidad y homocedasticidad. Gracias a esto, podemos concluir que los hombres sufren enfermedades cardíacas de mayor gravedad que las mujeres.

Previamente a estos análisis, hemos realizado la limpieza del dataset, eliminando los registros con valores faltantes. Por otra parte, los valores extremos los hemos dejado en el conjunto de datos, al no tener

conocimientos suficientes sobre su validez.

En conclusión, hemos podido extraer conclusiones relevantes de los datos, y así mismo, he obtenido un gran aprendizaje de R, que desconocía previamente, así de como emplearlo para analizar datos.