

Práctica 2: ¿Cómo realizar la limpieza y análisis de datos?

Balpreet Kaur Singh/Daniel Cañete Román

16 de junio 2023

- 0. Introducción y pasos iniciales
- 1. Descripción del dataset
- 2. Integración y selección
- 3. Limpieza de los datos
- 4. Análisis de los datos
 - 4.1. Selección de los grupos de datos
 - 4.2. Comprobación de la normalidad y homogeneidad de la varianza.
 - 4.3. Aplicación de pruebas estadísticas para comparar los grupos de datos.
- 5. Representación de los resultados
- 6. Resolución del problema
- 7. Código
- 8. Vídeo

0. Introducción y pasos iniciales

Esta práctica se ha realizado bajo el contexto de la asignatura **Tipología y ciclo de vida de los datos**, perteneciente al Máster en Ciencia de Datos de la Universitat Oberta de Catalunya. En ella, se aplican técnicas de **limpieza y análisis de datos** mediante el lenguaje de programación R.

Para facilitar la lectura se incorpora parte del enunciado utilizando el formato blockquote

Tras analizar el dataset resultante en la PRA1, se ha seleccionado el propuesto en el enunciado por tener mayor riqueza a nivel educativo.

Antes de empezar con los apartados propuestos en el enunciado de la PRA, se realizan pasos previos necesarios: carga de datos y librerías utilizadas:

1. Descripción del dataset

¿Por qué es importante y qué pregunta/problema pretende responder?

En primer lugar, se estudia la documentación disponible en <https://www.kaggle.com/datasets/rashikrahmanpritom/heart-attack-analysis-prediction-dataset> (<https://www.kaggle.com/datasets/rashikrahmanpritom/heart-attack-analysis-prediction-dataset>) donde se puede apreciar:

- Se trata de un estudio ataques al corazón donde existen datos de problemas de corazón donde se
- Licencia CC0: Public Domain (<https://creativecommons.org/publicdomain/zero/1.0/>). Sin copyright, lo cual nos habilita a utilizarlo sin problemas.
- El dataset tiene las siguientes columnas:
 - Age: Edad del paciente
 - Sex : Sexo del paciente
 - exang: angina inducida por el ejercicio (1 = sí; 0 = no)
 - ca: número de venas principales
 - cp : Tipo de dolor torácico
 - Valor 1: angina típica
 - Valor 2: angina atípica

- Valor 3: dolor no anginoso
- Valor 4: asintomático
- trtbps: presión arterial en reposo (en mm Hg)
- chol: colesterol en mg/dl obtenido a través del sensor BMI
- fbs: azúcar en sangre en ayunas > 120 mg/dl (1 = verdadero; 0 = falso)
- rest_ecg : resultados electrocardiográficos en reposo
 - Valor 0: normal
 - Valor 1: tener anomalías en la onda ST-T (inversiones de la onda T y/o elevación o depresión del ST > 0,05 mV)
 - Valor 2: mostrar hipertrofia ventricular izquierda probable o definitiva según los criterios de Estes
- thalach: frecuencia cardíaca máxima alcanzada
- slp: se refiere a la pendiente de una línea de tendencia trazada en un gráfico.
- target: 0= menos posibilidades de ataque al corazón 1= más posibilidades de ataque al corazón

Corroboremos estos datos e indaguemos un poco más:

```
head(ds) %>% gt()
```

age	sex	cp	trtbps	chol	fbs	restecg	thalachh	exng	oldpeak	slp	caa	thall	output
63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
57	1	0	140	192	0	1	148	0	0.4	1	0	1	1

```
summary (ds)
```

```

##      age          sex          cp          trtbps
##  Min.   :29.00   Min.   :0.0000   Min.   :0.000   Min.   : 94.0
##  1st Qu.:47.50  1st Qu.:0.0000  1st Qu.:0.000   1st Qu.:120.0
##  Median :55.00  Median :1.0000  Median :1.000   Median :130.0
##  Mean   :54.37  Mean   :0.6832  Mean   :0.967   Mean   :131.6
##  3rd Qu.:61.00  3rd Qu.:1.0000  3rd Qu.:2.000   3rd Qu.:140.0
##  Max.   :77.00  Max.   :1.0000  Max.   :3.000   Max.   :200.0
##      chol          fbs          restecg        thalachh
##  Min.   :126.0   Min.   :0.0000   Min.   :0.0000   Min.   : 71.0
##  1st Qu.:211.0  1st Qu.:0.0000  1st Qu.:0.0000  1st Qu.:133.5
##  Median :240.0  Median :0.0000  Median :1.0000  Median :153.0
##  Mean   :246.3  Mean   :0.1485  Mean   :0.5281  Mean   :149.6
##  3rd Qu.:274.5  3rd Qu.:0.0000  3rd Qu.:1.0000  3rd Qu.:166.0
##  Max.   :564.0  Max.   :1.0000  Max.   :2.0000  Max.   :202.0
##      exng          oldpeak        slp          caa
##  Min.   :0.0000   Min.   :0.00   Min.   :0.000   Min.   :0.0000
##  1st Qu.:0.0000  1st Qu.:0.00  1st Qu.:1.000  1st Qu.:0.0000
##  Median :0.0000  Median :0.80  Median :1.000  Median :0.0000
##  Mean   :0.3267  Mean   :1.04  Mean   :1.399  Mean   :0.7294
##  3rd Qu.:1.0000  3rd Qu.:1.60  3rd Qu.:2.000  3rd Qu.:1.0000
##  Max.   :1.0000  Max.   :6.20  Max.   :2.000  Max.   :4.0000
##      thall          output
##  Min.   :0.000   Min.   :0.0000
##  1st Qu.:2.000  1st Qu.:0.0000
##  Median :2.000  Median :1.0000
##  Mean   :2.314  Mean   :0.5446
##  3rd Qu.:3.000  3rd Qu.:1.0000
##  Max.   :3.000  Max.   :1.0000

```

```
str(ds)
```

```

## 'data.frame': 303 obs. of 14 variables:
## $ age     : int 63 37 41 56 57 57 56 44 52 57 ...
## $ sex     : int 1 1 0 1 0 1 0 1 1 1 ...
## $ cp      : int 3 2 1 1 0 0 1 1 2 2 ...
## $ trtbps  : int 145 130 130 120 120 140 140 120 172 150 ...
## $ chol    : int 233 250 204 236 354 192 294 263 199 168 ...
## $ fbs     : int 1 0 0 0 0 0 0 1 0 ...
## $ restecg : int 0 1 0 1 1 1 0 1 1 1 ...
## $ thalachh: int 150 187 172 178 163 148 153 173 162 174 ...
## $ exng   : int 0 0 0 0 1 0 0 0 0 0 ...
## $ oldpeak : num 2.3 3.5 1.4 0.8 0.6 0.4 1.3 0 0.5 1.6 ...
## $ slp    : int 0 0 2 2 2 1 1 2 2 2 ...
## $ caa    : int 0 0 0 0 0 0 0 0 0 0 ...
## $ thall  : int 1 2 2 2 2 1 2 3 3 2 ...
## $ output : int 1 1 1 1 1 1 1 1 1 1 ...

```

```
print ("Valores ausentes:")
```

```
## [1] "Valores ausentes:"
```

```
colSums(is.na(ds))
```

```

##      age      sex      cp      trtbps      chol      fbs      restecg      thalachh
##          0       0       0       0       0       0       0       0       0
##      exng      oldpeak      slp      caa      thall      output
##          0       0       0       0       0       0

```

```
print("Cadenas vacías")
```

```
## [1] "Cadenas vacías"
```

```
colSums(ds=="")
```

```

##      age      sex      cp      trtbps      chol      fbs      restecg      thalachh
##      0        0        0        0        0        0        0        0        0
##      exng    oldpeak      slp      caa      thall      output
##      0        0        0        0        0        0        0

print (paste (nrow(ds) , " observaciones"))
## [1] "303 observaciones"
print ("Valores diferentes de cp: ")
## [1] "Valores diferentes de cp: "
unique (ds$cp)
## [1] 3 2 1 0
print ("Valores diferentes de restecg: ")
## [1] "Valores diferentes de restecg: "
unique (ds$restecg)
## [1] 0 1 2
print ("Valores diferentes de slp: ")
## [1] "Valores diferentes de slp: "
unique (ds$slp)
## [1] 0 2 1
print ("Valores diferentes de caa: ")
## [1] "Valores diferentes de caa: "
unique (ds$caa)
## [1] 0 2 1 3 4
print ("Valores diferentes de thall: ")
## [1] "Valores diferentes de thall: "
unique (ds$thall)
## [1] 1 2 3 0
print ("Valores diferentes de sex: ")
## [1] "Valores diferentes de sex: "
unique (ds$sex)
## [1] 1 0
print ("Valores diferentes de fbs: ")
## [1] "Valores diferentes de fbs: "
unique (ds$fbs)
## [1] 1 0
print ("Valores diferentes de exng: ")
## [1] "Valores diferentes de exng: "
unique (ds$exng)
## [1] 0 1

```

Hay 303 observaciones, inicialmente no hay valores ausentes. Respecto los 0, estos son valores aceptables entre los valores disponibles de atributo. Es decir, los valores “0” no significan ausencia de valor para el atributo específico.

Los campos no son exactamente como aparece en la documentación, pero se puede encontrar una descripción más actualizada en <https://www.kaggle.com/code/namanmanchanda/heart-attack-eda-prediction-90-accuracy> (<https://www.kaggle.com/code/namanmanchanda/heart-attack-eda-prediction-90-accuracy>):

**Cam- Descripción
po**

Tipo

age Edad del paciente

int (29-77)

sex	Sexo del paciente	int (0 ó 1). No se indica qué significa cada cuál
cp	Tipo de dolor torácico <ul style="list-style-type: none"> • Valor 1: angina típica • Valor 2: angina atípica • Valor 3: dolor no anginoso • Valor 4: asintomático 	int (0, 1, 2 ó 3)
trbps	Presión arterial en reposo (en mm Hg)	int (94-200)
chol	Colesterol en mg/dl obtenido a través del sensor BMI	int (126-564)
fbs	Azúcar en sangre en ayunas > 120 mg/dl (1 = verdadero; 0 = falso)	int (0 ó 1)
restecg	Electrocardiograma en reposo <ul style="list-style-type: none"> • Valor 0: normal • Valor 1: tener anomalías en la onda ST-T (inversiones de la onda T y/o elevación o depresión del ST > 0,05 mV) • Valor 2: mostrar hipertrofia ventricular izquierda probable o definitiva según los criterios de Estes 	int (0, 1 ó 2)
thalachh	Frecuencia cardíaca máxima alcanzada	int (71-202)
exng	Angina inducida por el ejercicio (1 = sí; 0 = no)	int (0 ó 1)
oldpeak	Pico anterior	num (0-6.2)
slp	Tendency Slope. Se refiere a la pendiente de una línea de tendencia.	int (0, 1 ó 2)
caa	Número de venas principales	int (0, 1, 2, 3 ó 4)
thall	Resultado de la prueba de esfuerzo con talio. Esta prueba es para comprobar si el flujo sanguíneo a través de las arterias coronarias es normal o no.	int (0, 1, 2 ó 3)
output	Variable objetivo. Es una codificación, no se define el significado de cada valor. Pero asumimos que : <ul style="list-style-type: none"> • 0, no predispuesto a tener ataque cardíaco • 1, predispuesto a tener ataque cardíaco 	int (0 ó 1)

Vemos que tenemos:

- Una variable objetivo categórica: output
- Variables categóricas: sex, cp, fbs, restecg, exng, slp, caa y thall
- Variables numéricas: age, trbps, chol, thalachh y oldpeak

¿Por qué es importante y qué pregunta/problemática pretende responder?

Ahora conocemos un poco mejor los datos, podemos responder a la pregunta. Se trata de un dataset con información clínica de pacientes que están predispuestos o no a tener una angina de pecho. Evidentemente es muy interesante para crear un modelo predictivo, pero también para conocer qué variables hacen que una persona tenga más posibilidades de padecer una angina de pecho.

Nota: Hay algunos campos que no conocemos su significado concreto, pero aún así se pueden utilizar para el modelo. Si se detectan que son variables relacionadas con la variable objetivo, un experto puede describir su significado.

2. Integración y selección

Integración y selección de los datos de interés a analizar. Puede ser el resultado de adicionar diferentes datasets o una subselección útil de los datos originales, en base al objetivo que se quiera conseguir.

La muestra no tiene un número elevado de registros que puedan dar problemas en el rendimiento en computación de análisis. En caso de que si la hubiera, habría que ver alguna manera de limitar los datos, al menos para facilitar los cálculos en fase de desarrollo (por ejemplo, limitar los atributos mediante PCA).

Respecto las observaciones y los datos en general, cuanto mayor observaciones tengamos, mejor nos acercaremos a los resultados y conclusiones reales. Pero en nuestro caso, la muestra es pequeña. Por tanto, debemos de estudiar la muestra disponible para sacar las mejores conclusiones posibles.

Organizamos y buscamos duplicados:

```
# Cambio de tipo variables clasificadorias
ds$sex <- as.factor(ds$sex)
ds$cp <- as.factor(ds$cp)
ds$fbs <- as.factor(ds$fbs)
ds$restecg <- as.factor(ds$restecg)
ds$exng <- as.factor(ds$exng)
ds$slp <- as.factor(ds$slp)
ds$caa <- as.factor(ds$caa)
ds$thall <- as.factor(ds$thall)
ds$output <- as.factor(ds$output)

# Cambio de variables numericas
ds$age <- as.numeric (ds$age)
ds$trtbps <- as.numeric (ds$trtbps)
ds$chol <- as.numeric (ds$chol)
ds$thalachh <- as.numeric (ds$thalachh)
ds$oldpeak <- as.numeric (ds$oldpeak)

# Ordenación de los campos (objetivo, clasificadorias y numéricas)
ds <- ds [, c ("output", "sex", "cp", "fbs", "restecg", "exng", "slp", "caa", "thall", "age", "trtbps",
"chol", "thalachh", "oldpeak")]

# Comprobamos y quitamos Los duplicados:
print(paste ("Duplicados:", nrow(ds[duplicated(ds), ]), "duplicados"))

## [1] "Duplicados: 1 duplicados"
ds<-ds[!duplicated(ds), ]
print(paste ("Duplicados despues de limpieza:", nrow(ds[duplicated(ds), ]), "duplicados"))

## [1] "Duplicados despues de limpieza: 0 duplicados"
```

Se considera poco probable que haya dos observaciones exactamente iguales y, aunque en un caso real sería recomendable que lo valorase un experto, se ha supuesto una errata a la hora de crear el dataset y se ha eliminado la observación repetida.

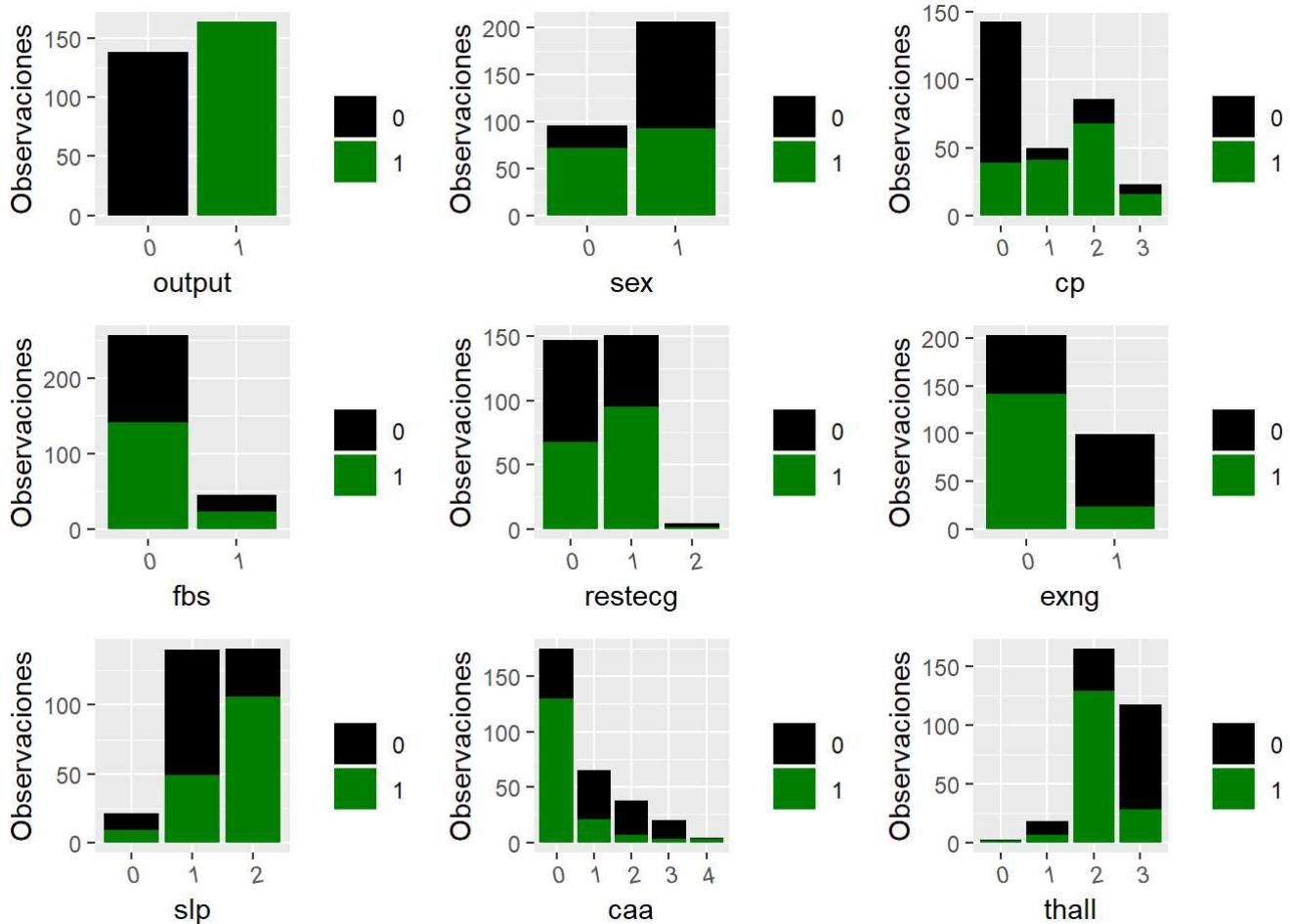
Vamos a estudiar un poco más los campos categóricos comparándolos con la variable objetivo:

```

myHistCat <- function(campo) {
  myPlot <- ggplot(ds, aes(x=ds[,campo], fill = output)) +
    geom_bar() + labs(x = campo, y = "Observaciones") +
    guides(fill = guide_legend(title = "")) +
    scale_fill_manual(values = c("black", "#008000")) +
    theme(axis.text.x = element_text(angle = 10, vjust = 1, hjust=0.5))
  return (myPlot)
}

grid.arrange(myHistCat("output"), myHistCat("sex"), myHistCat("cp"),
             myHistCat("fbs"), myHistCat("restecg"), myHistCat("exng"),
             myHistCat("slp"), myHistCat("caa"), myHistCat("thall"),
             ncol = 3, nrow = 3)

```



Sorprende que hay más personas predispuestas de las que no (164/138). Eso hace pensar que **la muestra está sesgada**, seguramente se ha obtenido entre pacientes que tenían alguna patología. Por eso se considera que **el estudio puede servir para detectar factores de riesgo, pero no se pueden sacar conclusiones a nivel de población general**,

Se observan también algunos valores posibles de variables con muy pocas observaciones, por lo que será difícil que sean significativos en un modelo predictivo. Este es el caso, por ejemplo:

- 4 venas principales en caa, donde hay solo 5 observaciones.
- 0 en la prueba de esfuerzo, donde hay solo dos observaciones.
- 2 en restecg (hipertrofia ventricular izquierda probable o definitiva según los criterios de Estes) donde solo hay 4 observaciones.

```

for (i in 1:9) {
  vName <- names(ds)[i]
  print (vName)
  print (table(ds[,i], ds$output))
}

```

```

## [1] "output"
##
##      0   1
##  0 138   0
##  1   0 164
## [1] "sex"
##
##      0   1
##  0  24  72
##  1 114  92
## [1] "cp"
##
##      0   1
##  0 104  39
##  1    9  41
##  2   18  68
##  3    7  16
## [1] "fbs"
##
##      0   1
##  0 116 141
##  1  22  23
## [1] "restecg"
##
##      0   1
##  0  79  68
##  1  56  95
##  2    3  1
## [1] "exng"
##
##      0   1
##  0   62 141
##  1   76  23
## [1] "slp"
##
##      0   1
##  0   12   9
##  1   91  49
##  2   35 106
## [1] "caa"
##
##      0   1
##  0   45 130
##  1   44  21
##  2   31   7
##  3   17   3
##  4    1   3
## [1] "thall"
##
##      0   1
##  0    1   1
##  1   12   6
##  2   36 129
##  3   89  28

```

Viendo las tablas por cada variable, podemos observar lo siguiente:

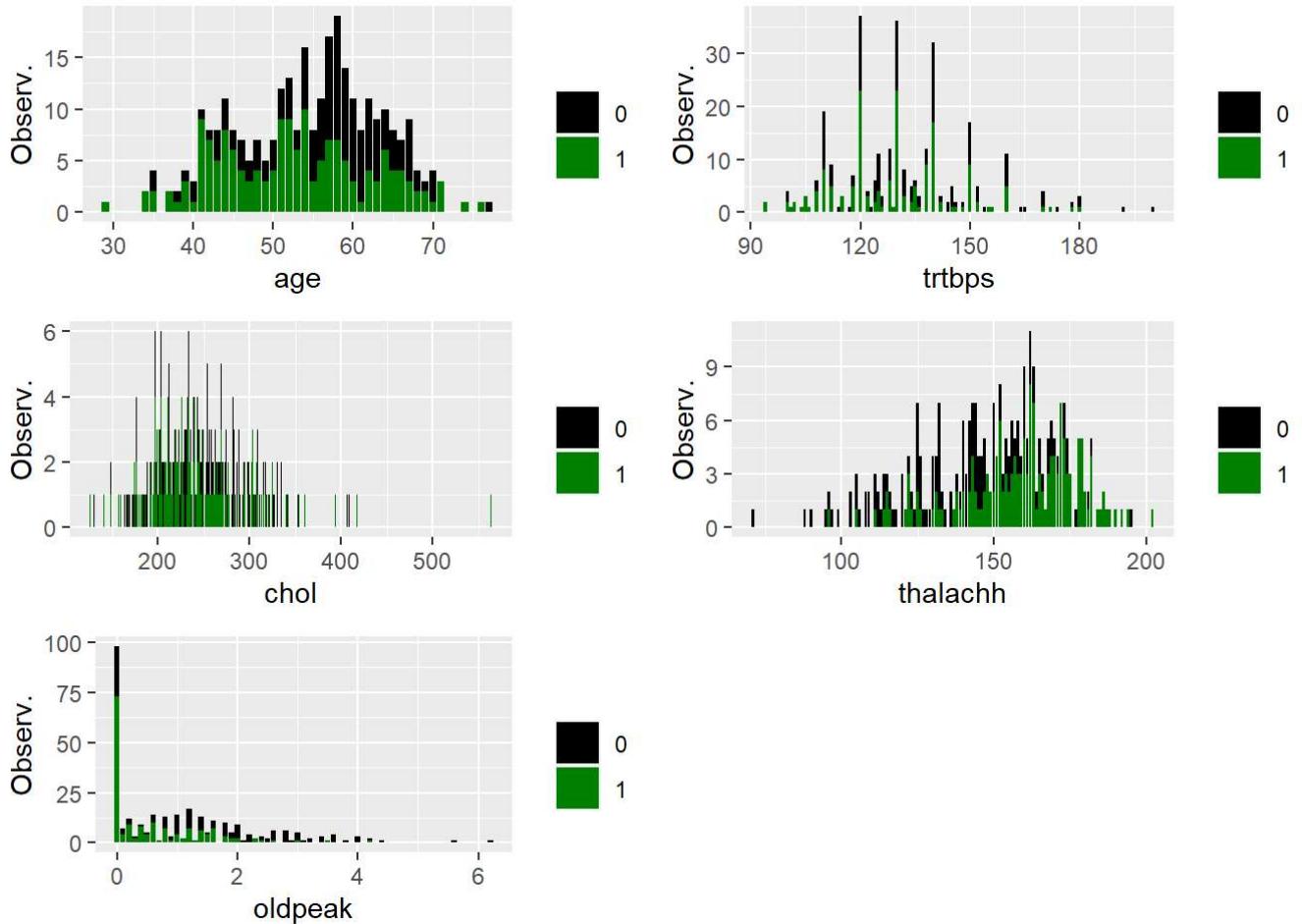
- sex: tenemos un total de 96 personas de sexo tipo “0” y 206 personas de sexo tipo “1”. Y si pasamos a profundizar más en estos totales, podemos ver que tenemos 72 casos de los 96 que tienen ataque cardiaco. Y 92 casos de los 206 tienen ataque cardiaco.

- cp: vemos que tenemos un total de 143 personas de cp tipo "0" (no padecen ningún tipo de Chest Pain listados), de los cuales 39 tienen ataque cardíaco. Después tenemos 50 personas de cp tipo "1" (los cuales padecen dolor típico de angina), entre los cuales 41 son personas que padecen ataque cardíaco. Tenemos 86 personas de sexo tipo "2" (los cuales padecen dolor no anginoso), entre los cuales 68 son personas que padecen ataque cardíaco. Y por último, de las 32 personas que padecen dolor de pecho son debido a dolores asintomático, solamente 16 padecen ataque cardíaco.
- Visualizando la grafica y las cantidades numericas del sexo, vemos que tenemos un total de 96 personas de sexo tipo "0" y 206 personas de sexo tipo "1". Y si pasamos a profundizar más en estos totales, podemos ver que tenemos 72 casos de los 96 de Sexo "0" tienen ataque cardiaco.Y 92 casos de los 206 de Sexo "1" tienen ataque cardiaco.
- fbs: el cual significa fasting blood sugar (glucemia en ayunas), podemos decir que 54,86% ($141/(116+141)$) de las personas no padecen glucemia en ayunas tiene ataque cardíaco. En cambio, el porcentaje de tener el ataque cardíaco baja a 51,11% ($23/(22+23)$) para las personas que padecen glucemia en ayunas. Y sumando las 2 categorías, tenemos que la probabilidad de tener ataque cardíaco es 54,3% ($((141+23)/(302))$) independientemente de si la persona ha tenido glucemia en ayunas o no. Por tanto, en resumen, las personas que no padecen glucemia en ayunas tienen mas probabilidad de tener un ataque cardíaco.
- restecg: el cual significa resultados electrocardiográficos en reposo, podemos decir que 46,25% ($68/(68+79)$) de las personas que tienen un resultado electrocardiográficos en reposo NORMAL pueden padecer un ataque cardíaco. Y este porcentaje sube a 62,91% en el caso de personas con anomalía en la onda ST-T. Y es 25% la probabilidad de que una persona que muestra hipertrofia ventricular izquierda probable o definitiva según los criterios de Estes padezca un ataque cardíaco. Y sumando ($((68+95+1)/(79+68+56+95+3+1))$), como esperamos, hay una probabilidad de 54,3% de que una persona padezca un ataque cardíaco independientemente de tipo de resultados electrocardiográficos en reposo. En resumen, hay más probabilidad de que una persona con anomalía en la onda ST-T padezca un ataque cardíaco comparado a los 2 otros tipos de resultados electrocardiográficos en reposo. Y podemos decir de que es mas probable de una una persona con resultado NORMAL padezca un ataque cardíaco que una persona que muestre hipertrofia ventricular izquierda probable o definitiva según los criterios de Estes.
- exng: que significa angina inducida por el ejercicio (exercise induced angina), podemos decir 69,45% de los casos de personas de tener angina no inducida por el ejercicio tienen ataque cardíaco. En cambio, si hablamos de personas con angina inducida por el ejercicio, este porcentaje de padecer el ataque cardíaco baja a 23%. Claramente analizamos que las personas con angina no inducida por el ejercicio es mas que doble de probable que la persona con angina inducida por ejercicio.
- slp: podemos observar que las personas que cubran el "slope" tipo 0 tienen una probabilidad de 42,85% de sufrir el ataque cardíaco y las personas que cubren slope tipo "1" tiene una probabilidad de padecer el ataque cardíaco en un 35%. Las personas de slope tipo "2" tienen una probabilidad de 75,17% de tener el ataque cardíaco. Por tanto, Slope tipo "2" tiene más probabilidad de tener el ataque cardíaco en comparación a los otros 2 tipos y el slope tipo "1" es la que menos probabilidad.
- caa: podemos decir que 74,28% para personas con 0 venas principales (major vessels) tienen ataque cardiaco, 32,3% para personas con 1 vena principal, 18,42% para personas con 2 venas principales, 15% para personas con 3 venas principales y 75% para personas con 4 venas principales.
- thall: tenemos que hay 50% probable de que una persona con Thallium Stress Test result "0" padezca un ataque cardiaco, 33,33% para personas con resultado "1" de prueba de Thallium Stress, 78,18% para personas con resultado "2" de prueba de Thallium Stress y 23,93% para personas con resultado "3" de prueba de Thallium Stress.

Estudiemos los valores numéricos:

```
myHistNum <- function(campo) {
  myPlot <- ggplot(ds, aes(ds[,campo], fill = output)) +
    geom_bar() + labs(x = campo, y = "Observ.") +
    guides(fill = guide_legend(title = ""))
    scale_fill_manual(values = c("black", "#008000"))
  return (myPlot)
}

grid.arrange(myHistNum("age"), myHistNum("trtbps"), myHistNum("chol"),
             myHistNum("thalachh"), myHistNum("oldpeak"),
             ncol = 2, nrow = 3)
```



Al analizar las distribuciones en los histogramas, podemos decir que las observaciones son mas frecuentes cuando:

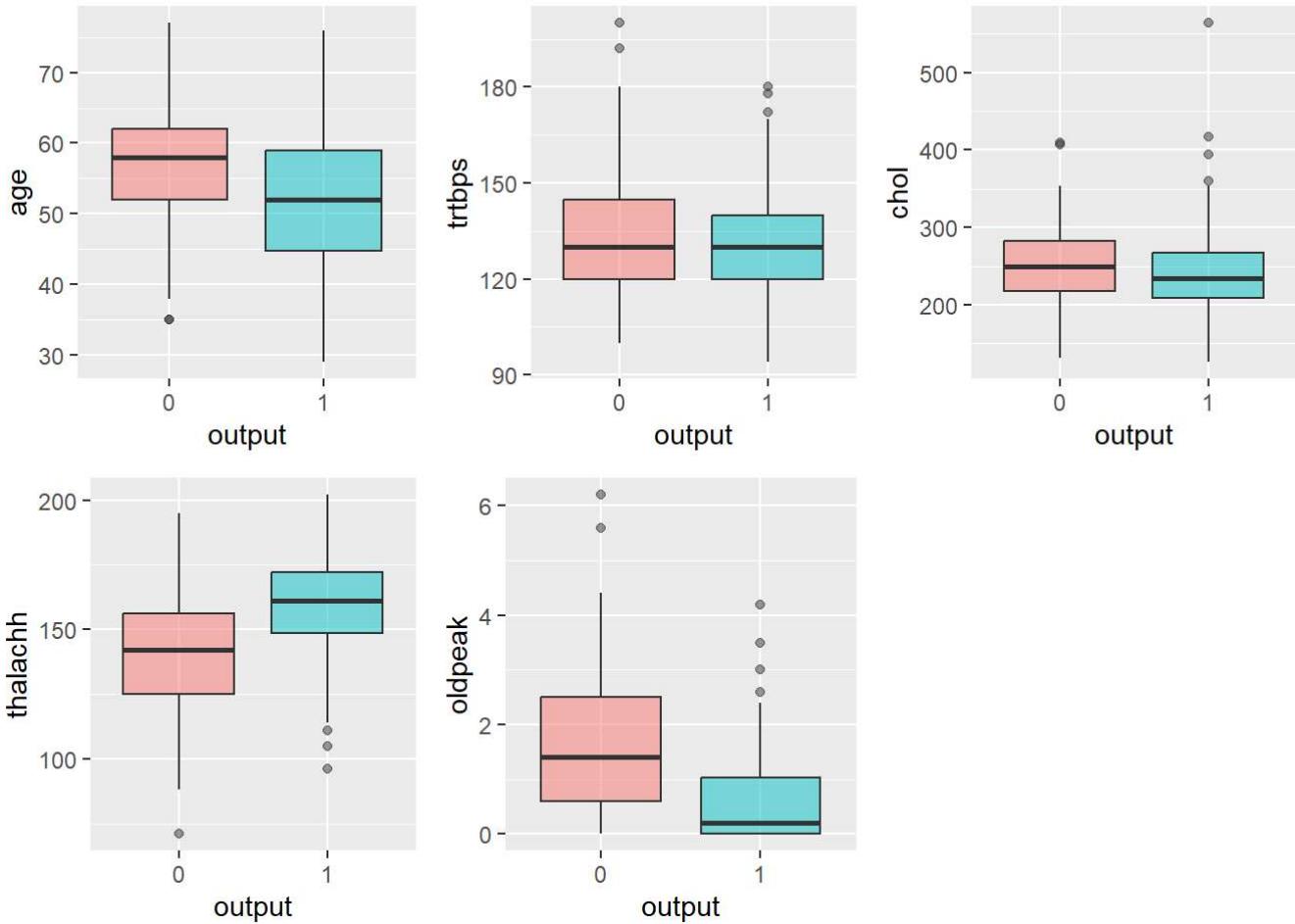
- Picos anteriores (oldpeak) son más cercano a 0
- Frecuencia cardiaca (thalachh) con valores superiores a 140 y inferiores a 180.
- Presión arterial en reposo está entre 100 y 160 Hg.
- Colesterol esta entre 150 y 350 mg/dl segun BMI sensor
- Y edad entre 40 y 70 años.

Los histogramas presentados no nos permiten diferenciar comportamientos distintos dependiendo de la variable objetivo.

Veamos en diagrams de cajas:

```
myCaja <- function(campo) {
  myPlot <- ggplot(ds, aes(x=output, y=ds[,campo], fill=output)) +
    geom_boxplot(alpha=0.5) +
    labs(x = "output", y = campo) +
    theme(legend.position="none")
  return (myPlot)
}

grid.arrange(myCaja("age"), myCaja("trtbps"), myCaja("chol"),
             myCaja("thalachh"), myCaja("oldpeak"),
             ncol = 3, nrow = 2)
```

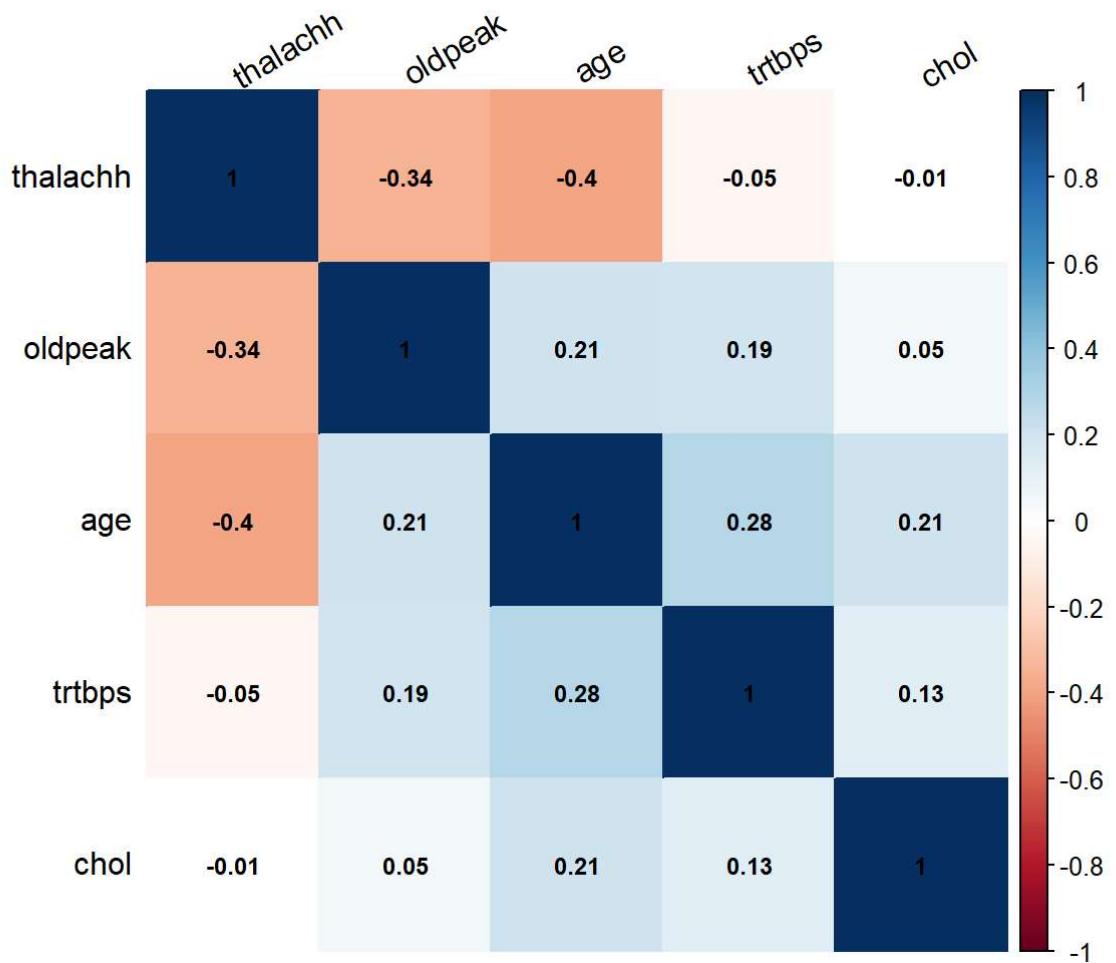


Se pueden intuir alguna correlación con la variable objetivo. Por ejemplo, parece que es más probable tener ataque al corazón (output = 1) cuando:

- Picos anteriores (oldpeak) son más cercano a 0
- Frecuencia cardiaca (thalachh) con valores entre 150y 170.

Estudiemos la correlación entre las variables. Si encontramos alguna muy relacionadas podemos quitarlas del modelo y reducir su complejidad:

```
n = c("age", "trtbps", "chol", "thalachh", "oldpeak")
factores= ds %>% select(all_of(n))
res<-cor(factores)
corrplot(res,method="color",tl.col="black", tl.srt=30, order = "AOE",
number.cex=0.75,sig.level = 0.01, addCoef.col = "black")
```



```
matriz <- cor(ds[,n])
corrplot(matriz, type="upper", order="hclust", tl.col="black", tl.srt=45)
```



Observamos que no hay una alta correlación entre las variables por lo que no prescindiremos de ninguna.

3. Limpieza de los datos

3.1. ¿Los datos contienen ceros o elementos vacíos? Gestiona cada uno de estos casos

3.2. Identifica y gestiona los valores extremos

En el punto anterior se ha analizado el dataset y observando los resultados, ya podemos obtener algunas conclusiones sin necesidad de desarrollar más código:

- **No se aprecian valores ausentes.** Asumiendo que los datos son recogidos sin errores alguno, podemos afirmar que no hay ausencia de valores ya estos se suelen identificar con:
 - “na” -> No se encuentra dicha cadena en el dataset
 - “0” -> En el caso de estudio, la mayoría de las variables tiene un significado concreto para ese valor y son recolectados y/o insertados sin transformación previa alguna.
- En los diagramas de cajas, vemos varios valores fuera de los rangos normales. Estos son valores extremos, pero no tienen por qué ser erroneos (outliers), por lo que sería recomendable consultar a un experto antes de descartarlos. En nuestro caso entendemos todos como válidos excepto un caso que destaca: una **observación con extremadamente elevado, que está muy separada de los rangos normales.**

La gestión de valores atípicos (outlier management) es la ciencia de investigar y aplicar un tratamiento adecuado a los valores atípicos en los datos. Puede ser tentador simplemente eliminar los registros donde hay valores atípicos en el conjunto de datos, pero no siempre es el mejor enfoque. El método de tratamiento de valores atípicos puede variar de un caso a otro y debe discutirse con la empresa antes de finalizar el método. Existen diferentes enfoques, como **reemplazar el valor atípico con el valor medio o la mediana** o, en algunos casos, **descartar la observación con el valor atípico sospechoso para evitar cualquier sesgo en ellos.** Tendemos a eliminar los valores atípicos si se deben a errores de entrada de datos causados por errores humanos, errores de procesamiento de datos.

Vamos a gestionar el último valor extremo de Colestrol, del cual estamos seguros que es valor extremo inaceptable optando por sustituirlo por la media:

```
max(ds$chol)
## [1] 564
ds [ds$chol==max(ds$chol), ]
##      output sex cp fbs restecg exng slpcaa thall age trtbps chol thalachh
## 86      1   0   2   0       0   0   1   0     3   67   115   564      160
##      oldpeak
## 86      1.6
# El ultimo valor extremo de cholestrol Lo sustituimos por La media al estar seguros de que es un valor outlier erroneo.
ds [ds$chol==max(ds$chol), ]$chol <- mean (ds$chol)
```

Con el siguiente código podríamos detectar otros valores extremos y candidatos a outliers. Como se ha comentado, lo ideal es revisar con un experto la estrategia a seguir:

```
ds %>%
  arrange(desc(chol)) %>%
  slice(1:8)

##      output sex cp fbs restecg exng slpcaa thall age trtbps chol thalachh oldpeak
## 1      1   0   2   1       0   0   2   1     2   65   140   417      157      0.8
## 2      0   0   0   0       0   1   1   2     3   56   134   409      150      1.9
## 3      0   0   0   0       0   0   1   3     3   63   150   407      154      4.0
## 4      1   0   0   0       0   0   1   0     2   62   140   394      157      1.2
## 5      1   0   2   0       0   0   2   0     2   65   160   360      151      0.8
## 6      1   0   0   0       1   1   2   0     2   57   120   354      163      0.6
## 7      0   1   0   0       1   1   1   1     3   55   132   353      132      1.2
## 8      1   0   1   0       1   0   2   0     2   55   132   342      166      1.2
```

4. Análisis de los datos

4.1. Selección de los grupos de datos

Selección de los grupos de datos que se quieren analizar/comparar (p. ej., si se van a comparar grupos de datos, ¿cuáles son estos grupos y qué tipo de análisis se van a aplicar?)

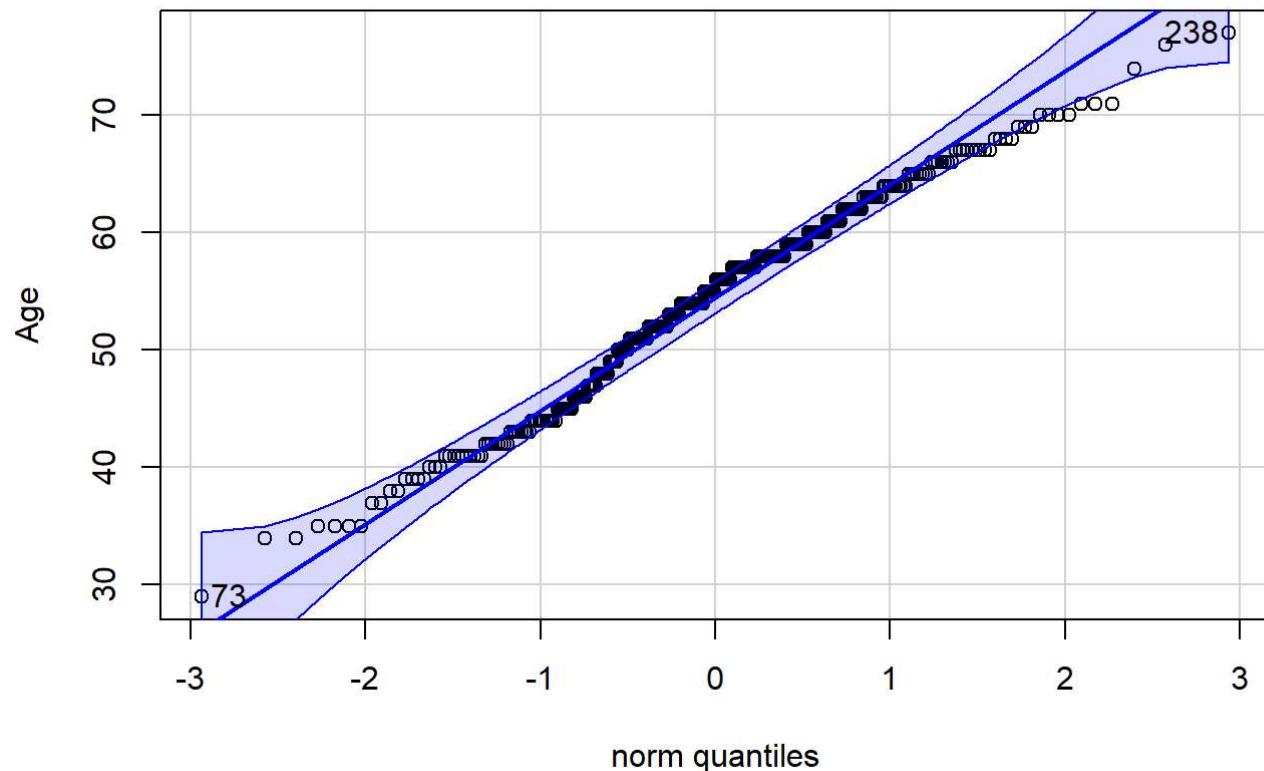
Teniendo en cuenta a **output como la variable objetivo**, procederemos a compararla con los diferentes atributos. En los siguientes puntos procederemos a hacer las comparaciones entre siguientes grupos y estrategias:

- Output y atributos categóricos ['sex', 'cp', 'fbs', 'restecg', 'exng', 'slp', 'caa', 'thall']: tenemos **dos variables categóricas** y haremos un test de χ^2 para ver si existen diferencias significativas entre los grupos definidos. Pero en nuestro caso, tenemos una **muestra pequeña**, por tanto, **usaremos Fisher Test**.
- Output y variables numéricas nominal ['age', 'trtbps', 'chol', 'thalachh', 'oldpeak']: comparamos una **variable categórica con otra numérica**. La estrategia será distinta según:
 - si cumple normalidad y homocedasticidad con output -> aplicamos tStudent,
 - si no -> test de Mann-Whitney.

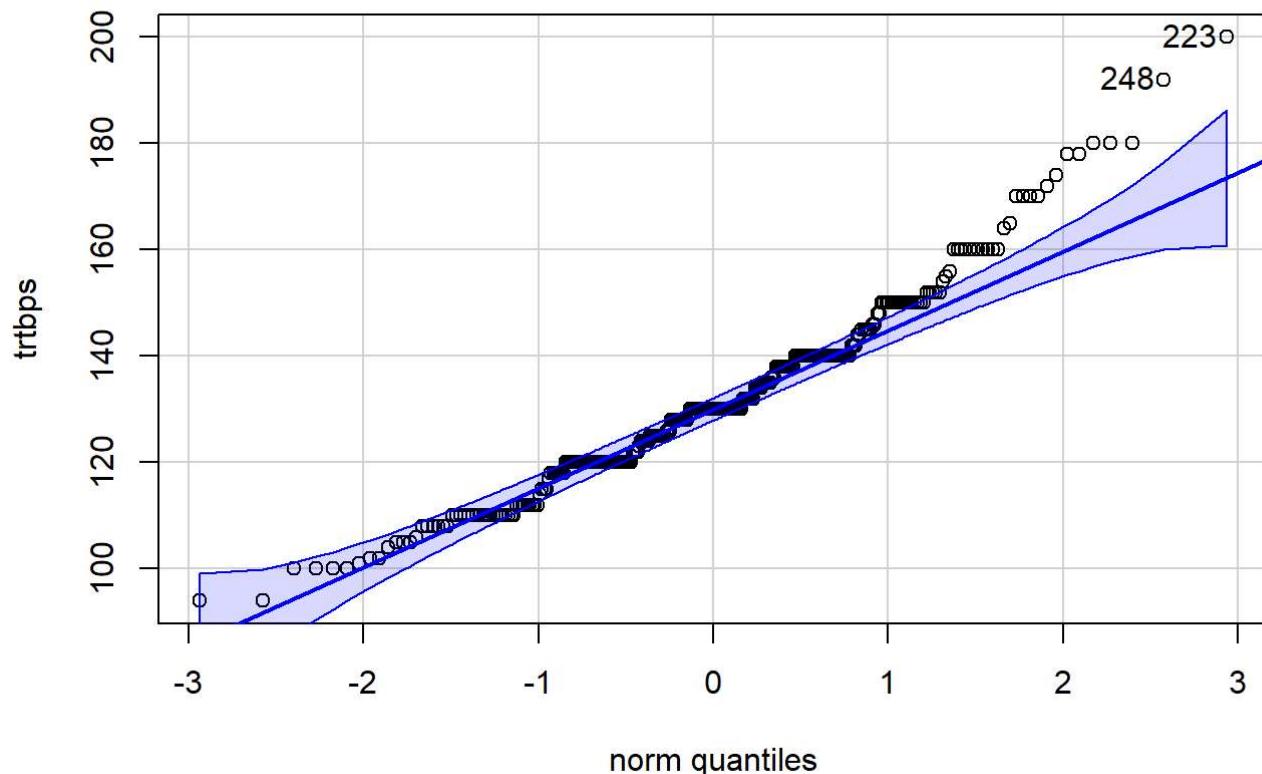
4.2. Comprobación de la normalidad y homogeneidad de la varianza.

Comprobamos la normalidad y homogeneidad de la varianza:

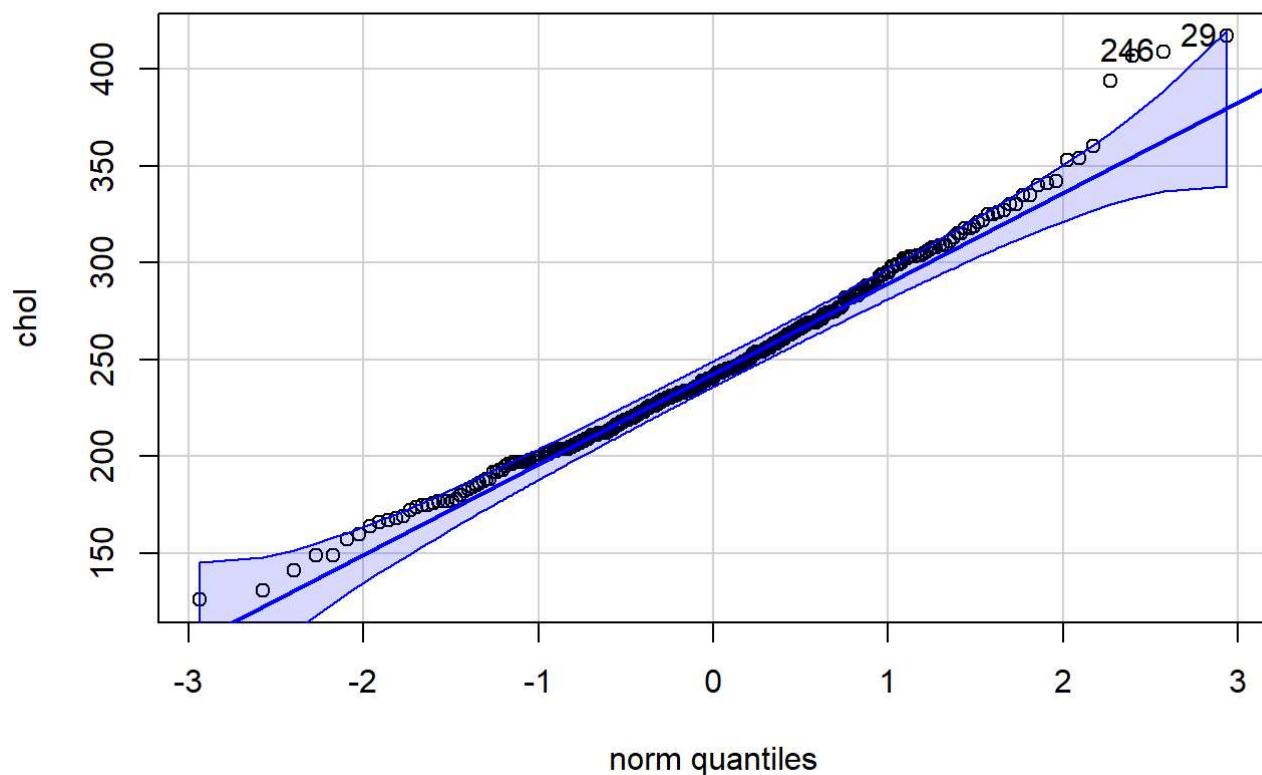
```
ds$age %>% qqPlot(dist="norm", ylab = "Age")
```



```
## [1] 73 238
ds$trtbps %>% qqPlot(dist="norm", ylab = "trtbps")
```

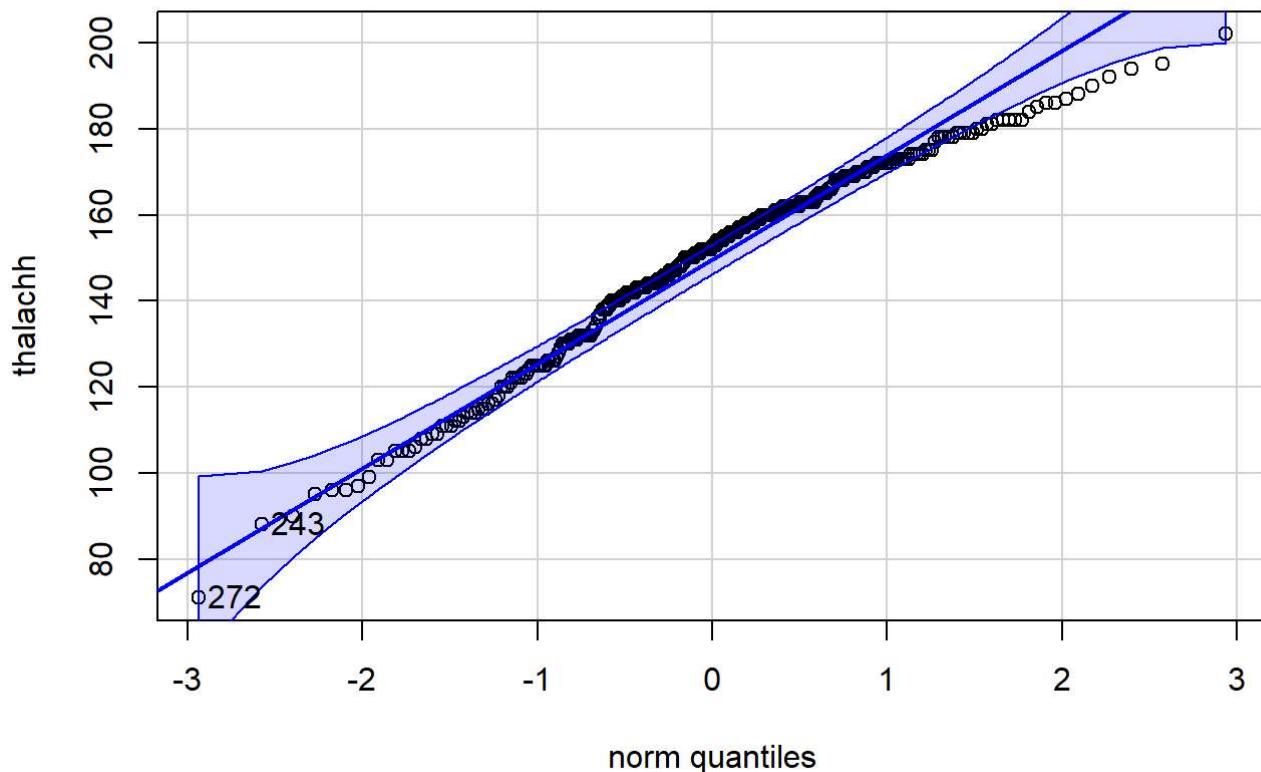


```
## [1] 223 248  
ds$chol %>% qqPlot(dist="norm", ylab = "chol")
```



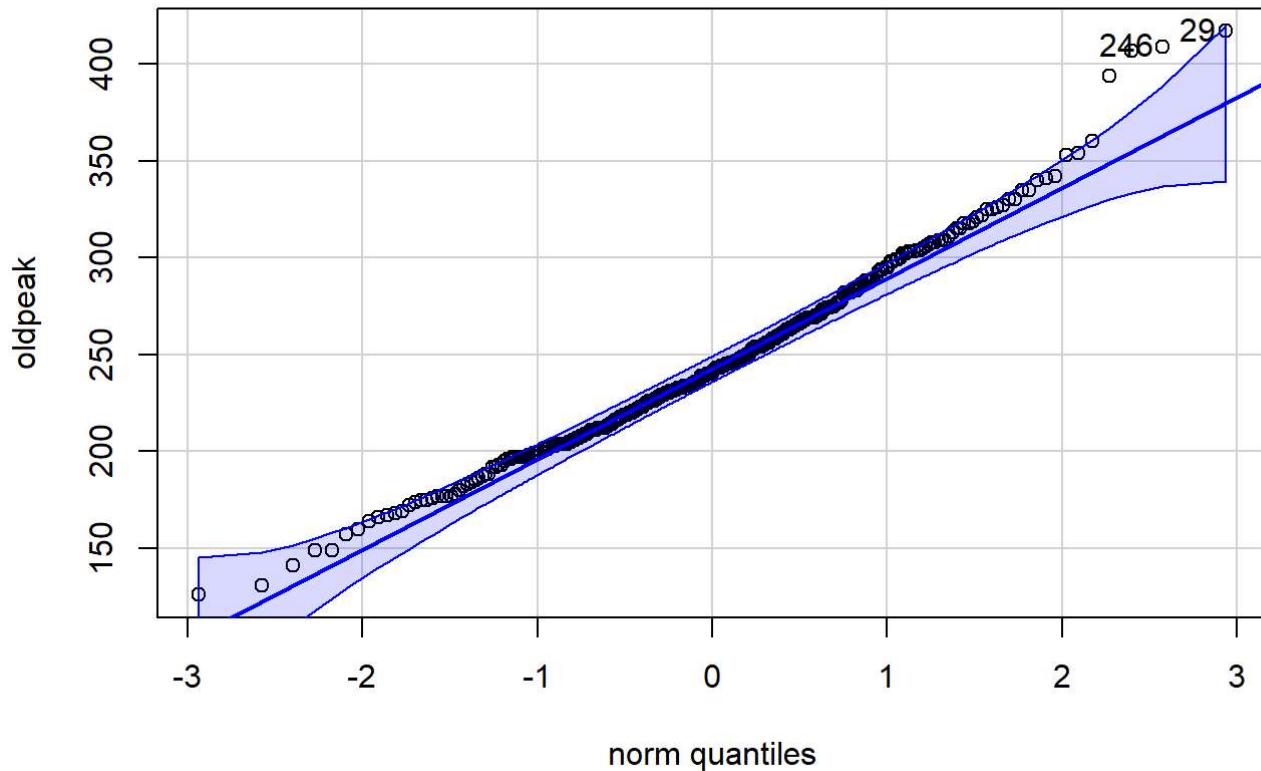
```
## [1] 29 246
```

```
ds$thalachh %>% qqPlot(dist="norm", ylab = "thalachh")
```



```
## [1] 272 243
```

```
ds$chol %>% qqPlot(dist="norm", ylab = "oldpeak")
```



```

## [1] 29 246
shapiro.test(ds$age)
##
##  Shapiro-Wilk normality test
##
## data: ds$age
## W = 0.98664, p-value = 0.006745
shapiro.test(ds$trtbp)
##
##  Shapiro-Wilk normality test
##
## data: ds$trtbp
## W = 0.96573, p-value = 1.419e-06
shapiro.test(ds$chol)
##
##  Shapiro-Wilk normality test
##
## data: ds$chol
## W = 0.98277, p-value = 0.001079
shapiro.test(ds$thalachh)
##
##  Shapiro-Wilk normality test
##
## data: ds$thalachh
## W = 0.97679, p-value = 8.268e-05
shapiro.test(ds$oldpeak)
##
##  Shapiro-Wilk normality test
##
## data: ds$oldpeak
## W = 0.84522, p-value < 2.2e-16

```

Vemos que en todos los casos, el valor P es inferior a 0,05, lo que significa que **no podemos asumir normalidad en ninguno de las variables numéricas.**

Ahora veamos homogeneidad de varianza mediante Levene Test (no FTest) ya que no cumplimos el requisito de normalidad:

```

leveneTest(age ~ output, data = ds)
## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value    Pr(>F)
## group     1  7.6349 0.006079 **
##          300
## ---
## Signif. codes:  0 '****' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
leveneTest(trtbp ~ output, data = ds)
## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value    Pr(>F)
## group     1  1.7911 0.1818
##          300
leveneTest(chol ~ output, data = ds)
## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value    Pr(>F)
## group     1  0.8995 0.3437
##          300
leveneTest(thalachh ~ output, data = ds)

```

```

## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value    Pr(>F)
## group     1  5.1661 0.02374 *
##            300
## ---
## Signif. codes:  0 '****' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

leveneTest(oldpeak ~ output, data = ds)

## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value    Pr(>F)
## group     1 32.438 2.934e-08 ***
##            300
## ---
## Signif. codes:  0 '****' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

La prueba de Levene es una estadística inferencial utilizada para evaluar la igualdad de varianzas de una variable calculada para dos o más grupos. Para esta prueba, un valor p de menos de 0,05 indica que, de hecho, hay suficiente variación en la muestra para explicar las posibles diferencias de medias.

Cuando tenemos un valor de p por encima de 0,05 (nivel de significancia), podemos concluir que existe una no diferencia significativa entre las varianzas de la muestra probada y, por tanto nos encontramos con homogeniedad en la varianza. Según esto, nos encontramos con:

- Homogenidad de las varianzas: trtbps, chol
- No homogenidad de las varianzas: age, thalachh, oldpeak

Por tanto, **incumplimos la asunciones de que el dataset sea normal y tenga homogeneidad de varianza.**

4.3. Aplicación de pruebas estadísticas para comparar los grupos de datos.

En función de los datos y el objetivo del estudio, aplicar pruebas de contraste de hipótesis, correlaciones, regresiones, etc. Aplicar al menos tres métodos de análisis diferentes.

Fisher test

Debido a que tenemos un dataset pequeño (<1000) y ya hemos visto que no podemos asegurar normalidad, tenemos que utilizar fisher test para analizar la correlación entre variables en vez de **Chi-cuadrado**.

Dado que vamos a hacer el mismo proceso con todas las variables clasificadoras, hacemos una función para facilitar código legible:

```

myFishertest <- function (campo) {
  counts <- table(ds$output, ds[,campo])
  print (campo)
  print (counts / rowSums(counts))
  print (fisher.test(counts))
}

```

Este método nos ofrece una tabla de frecuencias y el resultado del test de fisher que puede sacar conclusiones sobre la tabla.

Si el resultado del fisher test nos da un p-value por debajo de su nivel de significación (p. ej., 0,05), se rechaza la hipótesis nula, o sea, que hay relación entre las variables.

Aplicamos el método con todas las variables categóricas:

```
myFishertest ("sex")
```

```
## [1] "sex"
##
##          0         1
## 0 0.1739130 0.8260870
## 1 0.4390244 0.5609756
##
## Fisher's Exact Test for Count Data
##
## data: counts
## p-value = 1.01e-06
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
## 0.1502637 0.4734676
## sample estimates:
## odds ratio
## 0.2701939
```

```
myFishertest ("cp")
```

```
## [1] "cp"
##
##          0         1         2         3
## 0 0.75362319 0.06521739 0.13043478 0.05072464
## 1 0.23780488 0.25000000 0.41463415 0.09756098
##
## Fisher's Exact Test for Count Data
##
## data: counts
## p-value < 2.2e-16
## alternative hypothesis: two.sided
```

```
myFishertest ("fbs")
```

```
## [1] "fbs"
##
##          0         1
## 0 0.8405797 0.1594203
## 1 0.8597561 0.1402439
##
## Fisher's Exact Test for Count Data
##
## data: counts
## p-value = 0.746
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
## 0.433890 1.709881
## sample estimates:
## odds ratio
## 0.8605235
```

```
myFishertest ("restecg")
```

```
## [1] "restecg"
##
##          0         1         2
## 0 0.572463768 0.405797101 0.021739130
## 1 0.414634146 0.579268293 0.006097561
##
## Fisher's Exact Test for Count Data
##
## data: counts
## p-value = 0.004462
## alternative hypothesis: two.sided
```

```
myFishertest ("exng")
```

```
## [1] "exng"
##
##          0         1
## 0 0.4492754 0.5507246
## 1 0.8597561 0.1402439
##
## Fisher's Exact Test for Count Data
##
## data: counts
## p-value = 3.438e-14
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
## 0.07309196 0.23880340
## sample estimates:
## odds ratio
## 0.1340885
```

```
myFisherTest ("slp")
```

```
## [1] "slp"
##
##          0         1         2
## 0 0.08695652 0.65942029 0.25362319
## 1 0.05487805 0.29878049 0.64634146
##
## Fisher's Exact Test for Count Data
##
## data: counts
## p-value = 1.769e-11
## alternative hypothesis: two.sided
```

```
myFisherTest ("caa")
```

```
## [1] "caa"
##
##          0         1         2         3         4
## 0 0.326086957 0.318840580 0.224637681 0.123188406 0.007246377
## 1 0.792682927 0.128048780 0.042682927 0.018292683 0.018292683
##
## Fisher's Exact Test for Count Data
##
## data: counts
## p-value < 2.2e-16
## alternative hypothesis: two.sided
```

```
myFisherTest ("thall")
```

```
## [1] "thall"
##
##          0         1         2         3
## 0 0.007246377 0.086956522 0.260869565 0.644927536
## 1 0.006097561 0.036585366 0.786585366 0.170731707
##
## Fisher's Exact Test for Count Data
##
## data: counts
## p-value < 2.2e-16
## alternative hypothesis: two.sided
```

```
myFisherTest ("output")
```

```

## [1] "output"
##
##      0 1
##      0 1 0
##      1 0 1
##
## Fisher's Exact Test for Count Data
##
## data: counts
## p-value < 2.2e-16
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  3195.791      Inf
## sample estimates:
## odds ratio
##          Inf

```

Cuando su valor p esté por debajo de su nivel de significación (p. ej., 0,05), rechace la hipótesis nula. Los datos de la muestra son lo suficientemente fuertes como para concluir que existe una relación entre las variables categóricas en la población. Conocer el valor de una variable proporciona información sobre el valor de la otra variable.

A continuación, se resume las conclusiones obtenidas en la siguiente tabla:

	p-value	Relaci- onadas	Tabla de frecuen- cias
sex	= 1.01e-06	Sí	Podemos ver que el 43,9% de sexo “0” tuvieron heart attack, por debajo del sexo “1” con probabilidad de 56,09%. Tambien, gracias al test, podemos confirmar que la variable sexo afecta a la variable dependiente output.
cp	< 2.2e-16	Sí	Rechazamos la hipótesis nula y confirmamos de que la probabilidad de tener un ataque es más alto en cp=2 teniendo en consideración la grafica y estadísticas anteriores.Tambien, gracias al test Fisher, podemos confirmar que la variable cp afecta a la variable dependiente output.
fbs	= 0.746	No	No rechazamos la hipótesis nula y, por tanto, los diferentes tipos de cps tienen más o menos la misma probabilidad de tener un ataque infarto. Tambien, gracias al test Fisher, podemos confirmar que la variable fbs no afecta a al output de la variable dependiente.
restecg	= 0.004462	Sí	Rechazamos la hipótesis nula y confirmamos de que la probabilidad de tener un ataque es más alto en restecg=1 teniendo en cuenta la gráfica y estadísticas anteriores.
exng	= 3.438e-14	Sí	Rechazamos la hipótesis nula y confirmamos de que la probabilidad de tener un ataque es más alto en exng=0 teniendo en cuenta la gráfica y las estadísticas anteriores.
sip	= 1.769e-11	Sí	Rechazamos la hipótesis nula y confirmamos de que la probabilidad de tener un ataque es más alto en sip=2 teniendo en cuenta la gráfica y las estadísticas anteriores.
caa	< 2.2e-16	Sí	Rechazamos la hipótesis nula y confirmamos de que la probabilidad de tener un ataque es más alto en caa=0 teniendo en cuenta la gráfica y las estadísticas anteriores. Tambien, gracias al test Fisher, podemos confirmar que la variable caa afecta a la variable dependiente output.
thall	< 2.2e-16	Sí	Rechazamos la hipótesis nula y confirmamos de que la probabilidad de tener un ataque es más alto en thall=2 teniendo en cuenta la gráfica y las estadísticas anteriores.
output	< 2.2e-16	Sí	Rechazamos la hipótesis nula, como era de esperar ya que comparamos la misma variable

Whitney

Ahora vamos a estudiar los atributos numéricos versus el output mediante Whitney Wilcoxon Rank sum Test.

```
str(ds[10:14])
```

```

## 'data.frame': 302 obs. of 5 variables:
## $ age      : num 63 37 41 56 57 57 56 44 52 57 ...
## $ trtbps   : num 145 130 130 120 120 140 140 120 172 150 ...
## $ chol     : num 233 250 204 236 354 192 294 263 199 168 ...
## $ thalachh: num 150 187 172 178 163 148 153 173 162 174 ...
## $ oldpeak  : num 2.3 3.5 1.4 0.8 0.6 0.4 1.3 0 0.5 1.6 ...
wilcox.test(age ~ output, data=ds)

##
## Wilcoxon rank sum test with continuity correction
##
## data: age by output
## W = 14394, p-value = 4.626e-05
## alternative hypothesis: true location shift is not equal to 0

wilcox.test(trtbps ~ output, data=ds)

##
## Wilcoxon rank sum test with continuity correction
##
## data: trtbps by output
## W = 12931, p-value = 0.03223
## alternative hypothesis: true location shift is not equal to 0

wilcox.test(chol ~ output, data=ds)

##
## Wilcoxon rank sum test with continuity correction
##
## data: chol by output
## W = 12924, p-value = 0.03351
## alternative hypothesis: true location shift is not equal to 0

wilcox.test(thalachh ~ output, data=ds)

##
## Wilcoxon rank sum test with continuity correction
##
## data: thalachh by output
## W = 5725, p-value = 1.398e-13
## alternative hypothesis: true location shift is not equal to 0

wilcox.test(oldpeak ~ output, data=ds)

##
## Wilcoxon rank sum test with continuity correction
##
## data: oldpeak by output
## W = 16723, p-value = 3.347e-13
## alternative hypothesis: true location shift is not equal to 0

```

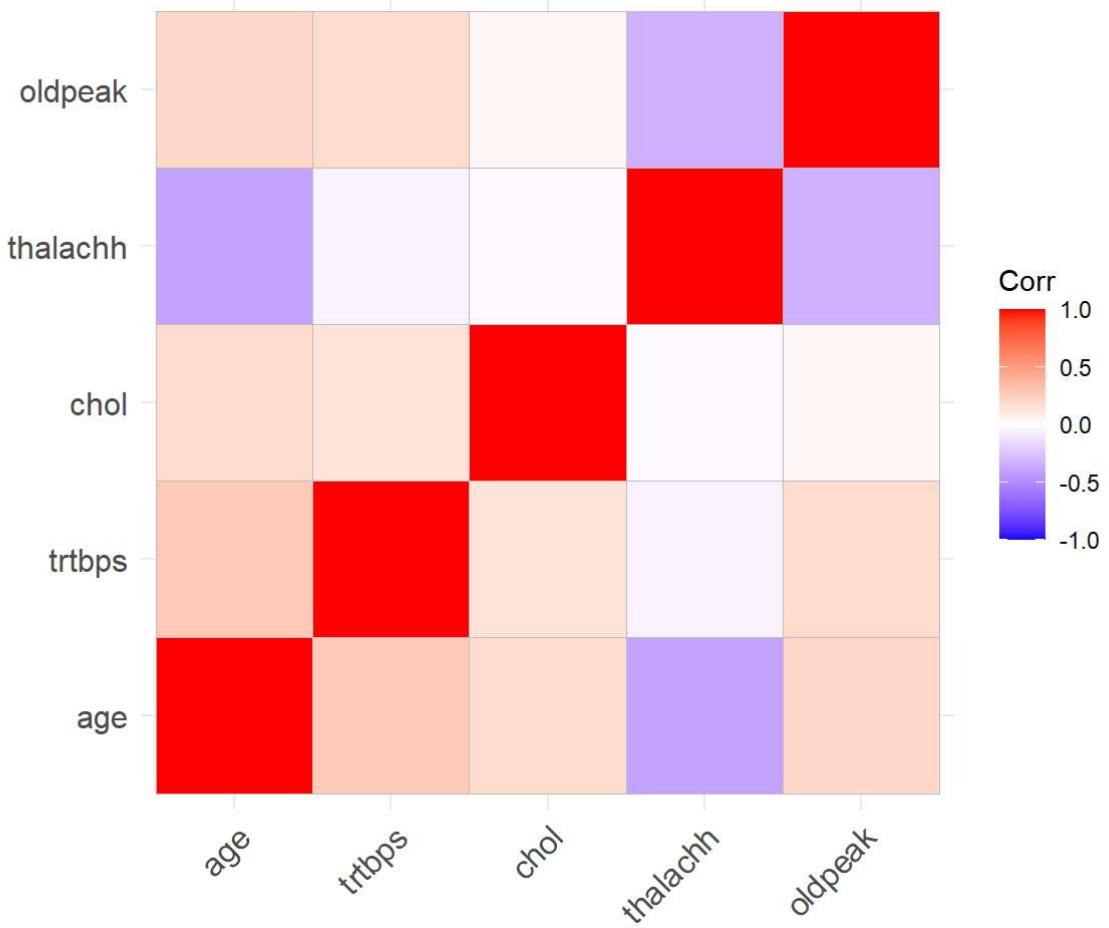
Con un nivel de significancia de .05, rechazamos hipótesis nula y podemos concluir que todos los atributos numéricos y la salida en el conjunto de datos provienen de poblaciones no idénticas. También que la diferencia entre la media de la población de salida y el atributo numérico correspondiente es estadísticamente significativa.

Análisis de componentes principales

```

numerical_data <- ds[10:14]
data_normalized <- scale(numerical_data)
corr_matrix <- cor(data_normalized)
ggcorrplot(corr_matrix)

```



```
data.pca <- princomp(corr_matrix)
summary(data.pca)
```

```
## Importance of components:
##                               Comp.1    Comp.2    Comp.3    Comp.4   Comp.5
## Standard deviation      0.7237748 0.4117721 0.3571720 0.29405280 0
## Proportion of Variance 0.5772801 0.1868501 0.1405836 0.09528626 0
## Cumulative Proportion  0.5772801 0.7641302 0.9047137 1.00000000 1
```

Los resultados anteriores muestran cuán importantes son estos 4 componentes de PCA considerando lo siguiente:

- El componente 1 explica el 57,70% de la varianza total.
- El componente 2 explica el 18,87% de la varianza total.
- El componente 3 explica el 14,56% de la varianza total.
- El componente 4 explica el 8,56% de la varianza total.

Extra: es genial tener los primeros 4 componentes, pero ¿qué significan realmente? Esto se puede responder explorando cómo se relacionan con cada columna utilizando las cargas de cada componente principal.

```
data.pca$loadings[, 1:4]
##                               Comp.1    Comp.2    Comp.3    Comp.4
## age        0.51853385 0.31246809 0.2593289 0.5990149
## trtbps     0.21024875 -0.04094717 0.7647701 -0.5929054
## chol       0.04928446 0.76618907 -0.3729530 -0.4627945
## thalachh   -0.69149215 -0.08694997 0.1089146 0.0598035
## oldpeak    0.45423314 -0.55324591 -0.4437554 -0.2681215
```

¿Cómo interpretamos estos resultados? Por ejemplo, centrándonos en el Componente 1, debemos mencionar que el atributo "thalachh" impacta negativamente en el PCA y los otros factores de forma positiva. Y el mismo enfoque se aplica a los otros componentes.

5. Representación de los resultados

Representación de los resultados a partir de tablas y gráficas. Este apartado se puede responder a lo largo de la práctica, sin necesidad de concentrar todas las representaciones en este punto de la práctica.

En el desarrollo del ejercicio se han ido utilizando diagramas representativos. Para seleccionar los más interesantes en cada caso, se ha utilizado como referencia la siguiente url: <https://www.data-to-viz.com/> (<https://www.data-to-viz.com/>)

6. Resolución del problema

A partir de los resultados obtenidos, ¿cuáles son las conclusiones? ¿Los resultados permiten responder al problema?

En base a los resultados antes expuestos, tenemos las siguientes conclusiones. Tenemos más probabilidad de tener un caso de ataque cardíaco:

- Sexo 0 tiene más probabilidad de tener un ataque cardíaco.
- Cp tipo 2 tiende a tener más probabilidad de ataque cardíaco.
- Fbs con anomalía en la onda ST-T tiene 62% de probabilidad de tener el ataque cardíaco.
- Respecto variable angina inducida por el ejercicio (exng), podemos decir 69,45% de los casos de personas de tener angina no inducida por el ejercicio tienen ataque cardiaco. Este baja a 23% cuando el ataque cardíaco es debido a angina inducida por el ejercicio.
- Slope tipo "2" tiene más probabilidad de tener el ataque cardíaco.
- Respecto la caa, personas con 0 venas principales tienen más probabilidad de tener el ataque cardiaco.
- Respecto thall, decir que personas con resultado "2" de prueba de Thalium Stress tiene una probabilidad de 78,18% de tener el ataque cardíaco.

Y es más probable tener ataque al corazón (output = 1) cuando:

- Picos anteriores (oldpeak) son más cercano a 0
- Frecuencia cardiaca (thalachh) con valores entre 150y 170.
- Presión arterial en reposo está entre 100 y 140 Hg.
- Colesterol esta entre 205 y 265 mg/dl según BMI sensor
- Y edad entre 45 y 59 años.

Después de comprobar qqtest, shapiro test y levene test comprobamos que incumplimos la asunciones de que el dataset sea normal y tenga homogeneidad de varianza.

Después de hacer Fisher Test, pudimos comprobar que los datos de la muestra son lo suficientemente fuertes como para concluir que existe una relación entre las variables categóricas en la población. Y conocer el valor de una variable proporciona información sobre el valor de la otra variable.

Respecto las variables numéricas y mediante el Whitney Test, pudimos comprobar que todos los atributos numéricos y la salida en el conjunto de datos provienen de poblaciones no idénticas. Podemos concluir que la diferencia entre la media de la población de salida y el atributo numérico correspondiente es estadísticamente significativa.

```
heart.split <- initial_split(ds)
heart.train <- training(heart.split)
heart.test <- testing(heart.split)
#Logistic regression having all the predictors.
heart.full <- glm(output~, data = heart.train, family = "binomial")
summary(heart.full)
```

```

## 
## Call:
## glm(formula = output ~ ., family = "binomial", data = heart.train)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.79415 -0.17456  0.02993  0.30144  2.28690
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 2.233e+00 5.313e+00 0.420 0.674289
## sex1        -1.926e+00 7.523e-01 -2.561 0.010446 *
## cp1          1.022e+00 7.179e-01  1.423 0.154701
## cp2          3.428e+00 8.176e-01  4.193 2.75e-05 ***
## cp3          3.702e+00 1.084e+00  3.415 0.000639 ***
## fbs1         6.332e-01 8.201e-01  0.772 0.440067
## restecg1    8.176e-01 5.182e-01  1.578 0.114608
## restecg2   -1.233e+01 1.560e+03 -0.008 0.993697
## exng1        -8.989e-01 5.670e-01 -1.585 0.112895
## slp1         -6.190e-01 1.370e+00 -0.452 0.651415
## slp2         1.100e+00 1.452e+00  0.757 0.448833
## caa1        -2.991e+00 7.493e-01 -3.992 6.55e-05 ***
## caa2        -3.781e+00 1.094e+00 -3.456 0.000548 ***
## caa3        -2.586e+00 1.253e+00 -2.063 0.039117 *
## caa4         1.689e+00 1.849e+00  0.913 0.361037
## thall1       3.908e+00 4.001e+00  0.977 0.328649
## thall2       3.255e+00 3.924e+00  0.829 0.406842
## thall3       1.106e+00 3.912e+00  0.283 0.777307
## age          6.317e-03 3.280e-02  0.193 0.847291
## trtbps     -4.231e-02 1.605e-02 -2.636 0.008400 **
## chol          1.082e-03 6.627e-03  0.163 0.870288
## thalachh    1.215e-02 1.563e-02  0.777 0.436976
## oldpeak     -5.796e-01 3.358e-01 -1.726 0.084294 .
## ---
## Signif. codes:  0 '****' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 312.86 on 225 degrees of freedom
## Residual deviance: 110.02 on 203 degrees of freedom
## AIC: 156.02
##
## Number of Fisher Scoring iterations: 15

```

```

# set engine
heart_model <- logistic_reg() %>%
  set_engine("glm")

# create recipe
heart_recipe <- recipe(output ~., data = heart.train) %>%
  step_rm(fbs) %>%
  step_rm(age) %>%
  step_rm(chol) %>%
  step_zv(all_predictors())

# build work flow
heart_wflow <- workflow() %>%
  add_model(heart_model) %>%
  add_recipe(heart_recipe)

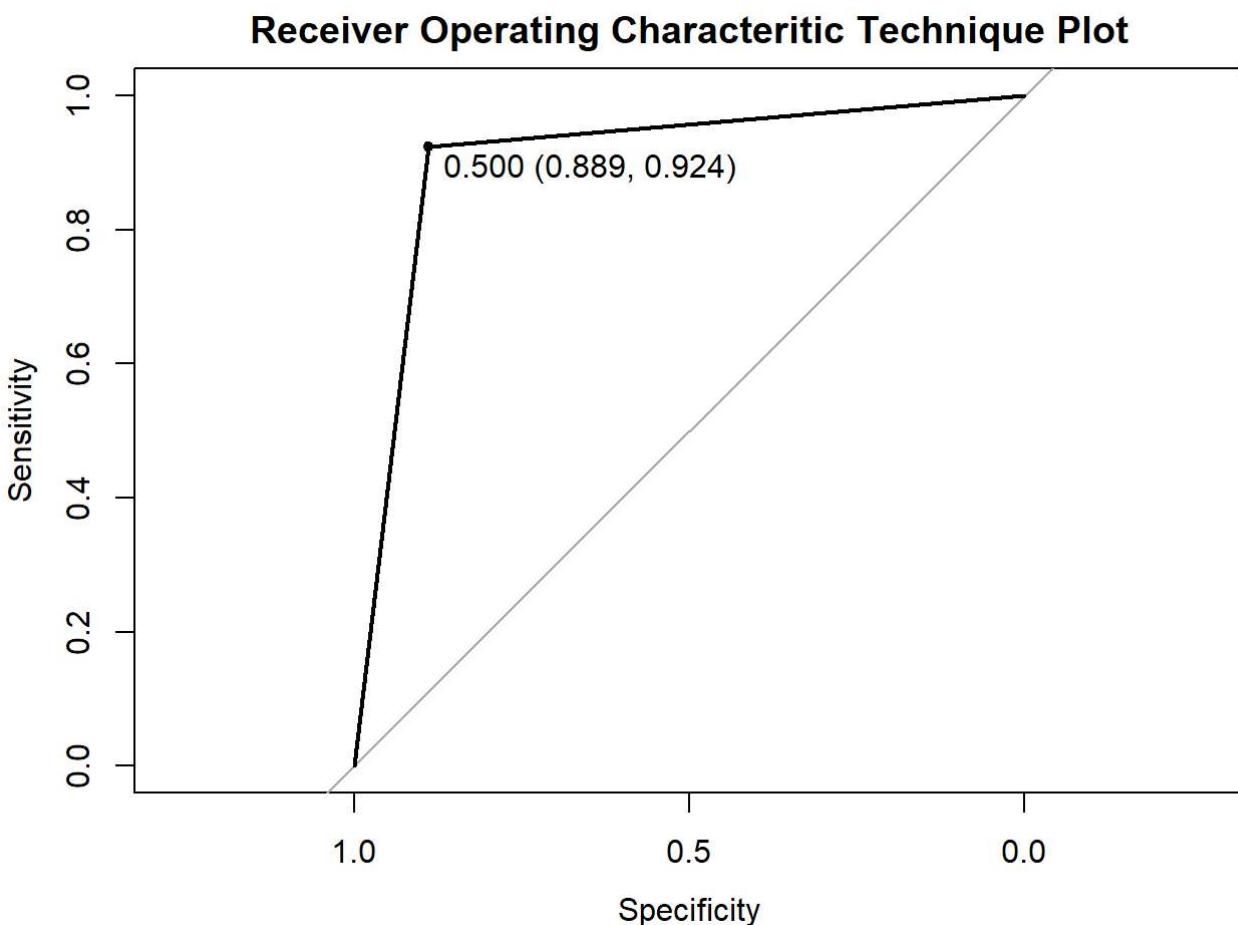
# fit training data through the work flow
heart_fit <- heart_wflow %>%
  fit(data = heart.train)
heart.train.pred = predict(heart_fit, new_data = heart.train)

traincomp <- data.frame(heart.train$output, heart.train.pred)
colnames(traincomp) <- c("train.response", "train.prediction")

heart.roc <- roc(response = ordered(traincomp$train.response), predictor = ordered(traincomp$train.prediction))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
plot(heart.roc, print.thres = "best", main = "Receiver Operating Characteristic Technique Plot")

```



```
print(auc(heart.roc))
```

```

## Area under the curve: 0.9063

#Perform 5-fold cross validation
set.seed(470)
folds <- vfold_cv(heart.train, v=5)

heart_fit_rs <- heart_wflow %>%
  fit_resamples(folds)

## → A | error: factor restecg has new levels 2

##
There were issues with some computations A: x1

There were issues with some computations A: x1

metrics <- data.frame(collect_metrics(heart_fit_rs, summarize = FALSE))

metrics <- metrics %>%
  select(-.config)
colnames(metrics) <- c("Fold", "Metric", "Estimator", "Estimate")
metrics

##   Fold Metric Estimator Estimate
## 1 Fold1 accuracy    binary 0.7826087
## 2 Fold1  roc_auc    binary 0.8712121
## 3 Fold2 accuracy    binary 0.8888889
## 4 Fold2  roc_auc    binary 0.9700000
## 5 Fold3 accuracy    binary 0.8222222
## 6 Fold3  roc_auc    binary 0.8769841
## 7 Fold4 accuracy    binary 0.7777778
## 8 Fold4  roc_auc    binary 0.8952569

# heart.test.pred = predict(heart_fit, new_data=heart.test, type="response")
#
# y_pred_num <- ifelse(heart.test.pred > 0.5, 1, 0)
# y_pred <- factor(y_pred_num, levels=c(0, 1))
# y_act <- testData$target
#
# # Result : Prediction Accuracy (Proportion of predicted target that matches with actual target)
# mean(y_pred == y_act)

# #Generate predictions on testing data
# heart_disease_pred <- predict(heart_fit, new_data = heart.test) %>%
#   bind_cols(heart.test %>% select(output))
#
# test_accuracy <- accuracy(heart_disease_pred, truth = heart_disease, estimate = .pred_class)
# test_specificity <- spec(heart_disease_pred, truth = heart_disease, estimate = .pred_class)
# test_sensitivity <- sens(heart_disease_pred, truth = heart_disease, estimate = .pred_class)
#
# test.values <- data.frame(test_accuracy$.estimate, test_sensitivity$.estimate, test_specificity$.estimate)
# colnames(test.values) <- c("Test set Accuracy", "Test set Sensitivity", "Test set Specificity")
# test.values

```

7. Código

Hay que adjuntar el código, preferiblemente en R, con el que se ha realizado la limpieza, análisis y representación de los datos. Si lo preferís, también podéis trabajar en Python.

Disponible en este mismo fichero.

8. Vídeo

Realizar un breve vídeo explicativo de la práctica (máximo 10 minutos), donde ambos integrantes del equipo expliquen con sus propias palabras el desarrollo de la práctica, basándose en las preguntas del enunciado para justificar y explicar el código desarrollado. Este vídeo se deberá entregar a través de un enlace al Google Drive de la UOC ([https://drive.google.com/...](https://drive.google.com/)), ([https://drive.google.com/...](https://drive.google.com/)), junto con enlace al repositorio Git entregado.

Se puede visualizar en cualquiera de estas dos alternativas:

- Github: <https://github.com/dcanete/heart> (<https://github.com/dcanete/heart>)
- Drive: <https://drive.google.com/open?id=1gko9Vw2D3FQP5aewnnfp-hPio-AZ70KZ> (<https://drive.google.com/open?id=1gko9Vw2D3FQP5aewnnfp-hPio-AZ70KZ>)