Tipología y Ciclo de vida del Dato: Práctica 2: Limpieza y validación de los datos

Autores: Inmaculada Pizarro Moreno y Enrique Fernández Morales

Enero 2023

Contents

Descripción del dataset

En el siguiente ejercicio nos disponemos a analizar el conjunto de datos sugerido para la PRA2 de tipología y ciencia de vida del dato sobre ataques de corazón y variables asociadas a cada muestra.

El conjunto de datos objeto de análisis se ha obtenido a partir de este enlace en Kaggle: Kaggle dataset sobre ataques de corazón

y está constituido por 14 características (columnas) que presentan 303 pacientes (filas o registros). Entre los campos de este conjunto de datos, encontramos los siguientes:

- **age** Edad del paciente
- **sex** Sexo del paciente
  - Value 0: mujer
  - Value 1: hombre
- **cp** Tipo de dolor en el pecho
  - Value 0: angina típica
  - Value 1: angina atípica
  - Value 2: dolor no de angina
  - Value 3: asintomático
- **trtbps** Presión arterial en reposo (mm Hg)
- **chol** Colestoral en mg/dl obtenido a través del sensor BMI
- **fbs** Azúcar en sangre en ayunas > 120 mg/dl
  - Value 0: falso
  - Value 1: verdadero
- **restecg** Resultados electrocardiográficos en reposo
  - Value 0: normal
  - Value 1: tener anomalías en la onda ST-T (inversiones de la onda T y/o elevación o depresión del ST $>$ 0,05 mV)
  - Value 2: que muestra hipertrofia ventricular izquierda probable o definitiva según los criterios de Estes
- **thalachh** Frecuencia cardíaca máxima alcanzada
- **exng** Angina inducida por el ejercicio
  - Value 0: no
  - Value 1: sí
- **oldpeak** Pico Anterior
- **slp** La pendiente del segmento ST de ejercicio máximo
  - Value 0: pendiente descendente
  - Value 1: plano
  - Value 1: pendiente ascendente
- **caa** Número de vasos principales (0-3)
- **thall** Thalassemia: trastorno genético de la sangre que se caracteriza por una tasa de hemoglobina más baja de lo normal. Resultado de la prueba de esfuerzo con talio ~ (0,3)
  - Value 0: nada
  - Value 1: defecto fijo
  - Value 2: normal
  - Value 3: defecto reversible
- **output** Posibilidad de infarto
  - Value 0: menos posibilidades de infarto
  - Value 1: más posibilidades de infarto

Importancia y objetivos de los análisis

Mediante el análisis de este dataset sobre pacientes de corazón nos planteamos saber si alguna de las pruebas realizadas en ellos podría ser determinante en que la posibilidad de infarto sea mayor o menor para el paciente.

Este tipo de análisis es muy común en el sector de la salud y presenta grandes retos, teniendo beneficios enormes en la investigación de todo tipo de enfermedades y tratamientos. Como ejemplo tenemos este mismo caso que trataremos en esta practica, donde el conocimiento que podemos extraer a la hora de tratar este tipo de datos puede ayudarnos a predecir infartos y salvar vidas.

Nos resulta también interesante saber qué variables se correlacionan entre ellas, de modo que podamos de algún modo no sólo saber la importancia de cada variable en el resultado sino la dependencia entre ellas, ya que podría ser útil a la hora de atacar el problema por algún camino médico, por ejemplo, el colesterol con respecto a los problemas de presión arterial o viceversa. Además, también nos interesa realizar otro tipo de pruebas estadísticas, como por ejemplo el contraste de hipótesis para comprobar las diferencias estadísticas significativas entre grupos de datos, y como por ejemplo realizar un modelo de regresión lineal para analizar la relación entre algunas variables.

Estos análisis nos permiten inferir propiedades sobre el resto de la población de pacientes de corazón.

Limpieza de los datos

El primer paso es cargar los datos para trabajar con ellos haciendo uso de la función read.csv de la librería readr.

```
pacientes <- read.csv('dataset/heart.csv', header = TRUE, sep = ',')
```

Vemos qué clase tiene cada variable usando la función sapply que nos permite aplicar la función class sobre todo data frame.

```
# Tipo de dato asignado a cada campo
sapply(pacientes, function(x) class(x))
```

```
##       age       sex        cp    trtbps      chol       fbs   restecg  thalachh
## "integer" "integer" "integer" "integer" "integer" "integer" "integer" "integer"
##      exng   oldpeak       slp       caa     thall    output
## "integer" "numeric" "integer" "integer" "integer" "integer"
```

Comprobemos de nuevo las características y estadísticas de los datos originales.

```
glimpse(pacientes)
```

```
## Rows: 303
## Columns: 14
## $ age      <int> 63, 37, 41, 56, 57, 57, 56, 44, 52, 57, 54, 48, 49, 64, 58, 5~
## $ sex      <int> 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 1~
## $ cp       <int> 3, 2, 1, 1, 0, 0, 1, 1, 2, 2, 0, 2, 1, 3, 3, 2, 2, 3, 0, 3, 0~
## $ trtbps   <int> 145, 130, 130, 120, 120, 140, 140, 120, 172, 150, 140, 130, 1~
## $ chol     <int> 233, 250, 204, 236, 354, 192, 294, 263, 199, 168, 239, 275, 2~
## $ fbs      <int> 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0~
## $ restecg  <int> 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1~
## $ thalachh <int> 150, 187, 172, 178, 163, 148, 153, 173, 162, 174, 160, 139, 1~
## $ exng     <int> 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0~
## $ oldpeak  <dbl> 2.3, 3.5, 1.4, 0.8, 0.6, 0.4, 1.3, 0.0, 0.5, 1.6, 1.2, 0.2, 0~
## $ slp      <int> 0, 0, 2, 2, 2, 1, 1, 2, 2, 2, 2, 2, 2, 1, 2, 1, 2, 0, 2, 2, 1~
## $ caa      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0~
## $ thall    <int> 1, 2, 2, 2, 2, 1, 2, 3, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3~
## $ output   <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~
```

Nos ha clasificado automáticamente todas las variables como integer o numeric. Sabemos por su descripción y valores posibles que varias de las variables son discretas y deberían ser factorizadas. Para ello usamos la función lapply y le pasamos la función factor a subset del dataframe de variables discretas.

```
#Para factorizar selecciono las columnas que creo se deben factorizar
col_disc = colnames(select(pacientes,contains(c("sex","cp","fbs","restecg","exng","slp",
                                                "caa","thall","output"))))
#Aplico la funcion factor y reviso la clase después
pacientes[col_disc] <- lapply(pacientes[col_disc], factor)
sapply(pacientes, class)
```

```
##       age       sex        cp    trtbps      chol       fbs   restecg  thalachh
## "integer"  "factor"  "factor" "integer" "integer"  "factor"  "factor" "integer"
##      exng   oldpeak       slp       caa     thall    output
##  "factor" "numeric"  "factor"  "factor"  "factor"  "factor"
```

Selección de los datos de interés

La mayoría de los atributos parecen necesarios para analizar su impacto en la probabilidad de sufrir un infarto. Excepto quizás el número de vasos principales, que podríamos eliminar del conjunto de datos.

```
pacientes <- pacientes[,-12]
```

Antes de continuar procedemos a traducir los nombres de las columnas para simplificar la comprensión de la práctica, y volvemos a ver el resumen estadístico con la traducción y las variables factorizadas.

```
names(pacientes) <- c("edad","sexo","dolor_pecho","presion_arterial","colesterol",
                      "azucar","electro","frecuencia_cardiaca","angina_ejercicio",
                      "pico_anterior","pendiente","prueba_esfuerzo",
                      "posibilidad_infarto")
col_disc <- c("sexo",   "dolor_pecho",  "azucar","electro", "angina_ejercicio",
              "pendiente", "prueba_esfuerzo", "posibilidad_infarto")
str(pacientes)
```

```
## 'data.frame':    303 obs. of  13 variables:
##  $ edad               : int  63 37 41 56 57 57 56 44 52 57 ...
##  $ sexo               : Factor w/ 2 levels "0","1": 2 2 1 2 1 2 1 2 2 2 ...
##  $ dolor_pecho        : Factor w/ 4 levels "0","1","2","3": 4 3 2 2 1 1 2 2 3 3 ...
##  $ presion_arterial   : int  145 130 130 120 120 140 140 120 172 150 ...
##  $ colesterol         : int  233 250 204 236 354 192 294 263 199 168 ...
##  $ azucar             : Factor w/ 2 levels "0","1": 2 1 1 1 1 1 1 1 2 1 ...
##  $ electro            : Factor w/ 3 levels "0","1","2": 1 2 1 2 2 2 1 2 2 2 ...
##  $ frecuencia_cardiaca: int  150 187 172 178 163 148 153 173 162 174 ...
##  $ angina_ejercicio   : Factor w/ 2 levels "0","1": 1 1 1 1 2 1 1 1 1 1 ...
##  $ pico_anterior      : num  2.3 3.5 1.4 0.8 0.6 0.4 1.3 0 0.5 1.6 ...
##  $ pendiente          : Factor w/ 3 levels "0","1","2": 1 1 3 3 3 2 2 3 3 3 ...
##  $ prueba_esfuerzo    : Factor w/ 4 levels "0","1","2","3": 2 3 3 3 3 2 3 4 4 3 ...
##  $ posibilidad_infarto: Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
```

Tenemos finalmente 12 atributos y 303 observaciones, las 8 variables factorizadas que son discretas son sexo, dolor_pecho, azucar, electro, angina_ejercicio, pendiente, prueba_esfuerzo y posibilidad de infarto, el resto son continuas.

Ceros y elementos vacíos

Procedemos a comprobar la existencia de valores ausentes en el dataset.

```
print('Porcentajes de valores ausentes en cada variable ordenado descendentemente')
```

```
## [1] "Porcentajes de valores ausentes en cada variable ordenado descendentemente"
```

```
sort(colMeans(is.na(pacientes) | pacientes==""), decreasing = TRUE)
```

```
##                edad                sexo         dolor_pecho    presion_arterial
##                   0                   0                   0                   0
##          colesterol              azucar             electro frecuencia_cardiaca
##                   0                   0                   0                   0
##    angina_ejercicio       pico_anterior           pendiente     prueba_esfuerzo
##                   0                   0                   0                   0
## posibilidad_infarto
##                   0
```

```
cat(ifelse(any(!complete.cases(pacientes)),"SÍ","NO"),"hay valores ausentes")
```

```
## NO hay valores ausentes
```

Podemos comprobar como no había valores ausentes. En el caso de haber encontrado que una variable pocos registros con valores ausentes, hubiera bastado con utilizar técnicas de sustitución por un valor por defecto o bien el valor k-vecino más próximo (kNN imputation). En cambio, de haber tenido una variable muchos valores vacíos se podría haber descartado esta sin llegar a perder mucha información.

Valores extremos

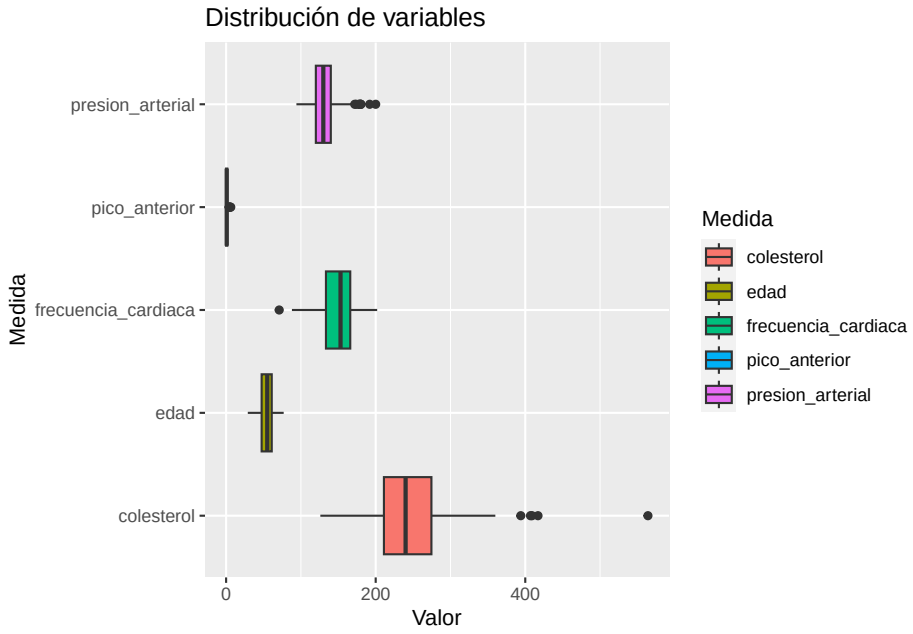
Veamos cuales son los valores outliers para cada variable continua:

```
pacientes_cat <-  pacientes %>% select(col_disc)
col_numericas_cont <- colnames(select(pacientes,!contains(col_disc)))
pacientes_num_cont <- pacientes %>% select(all_of(col_numericas_cont))
pacientes_table <- as.data.frame(apply(pacientes_num_cont, 2, summary))
resultado <- lapply(pacientes_num_cont,function(x) boxplot.stats(x)$out)
resultado
```

```
## $edad
## integer(0)
##
## $presion_arterial
## [1] 172 178 180 180 200 174 192 178 180
##
## $colesterol
## [1] 417 564 394 407 409
##
## $frecuencia_cardiaca
## [1] 71
##
## $pico_anterior
## [1] 4.2 6.2 5.6 4.2 4.4
```

También lo podemos visualizar en boxplots. Donde vemos claramente la representación de los outliers anteriores como puntos fuera de los bigotes del boxplot en presion_arterial (6), pico_anterior (4), frecuencia_cariaca (1), colesterol (4). No hay outliers en edad.

```
pacientes %>%
  pivot_longer(cols = col_numericas_cont,
               names_to = "Medida",
               values_to = "Valor") %>%
  ggplot() +
  geom_boxplot(aes(x = Medida, y = `Valor`, fill = Medida)) +
  ggtitle("Distribución de variables") +
  coord_flip()
```



Los valores de colesterol y presión arterial son claramente muy altos, por lo que es posible que sean errores. Aun así, ya que estamos viendo la relación de estos valores altos o fuera de la norma y su influencia en la probabilidad de infarto, no los eliminaremos de la muestra original. En consecuencia vamos a crear otro dataframe sin estos outliers para eventualmente probar su efecto en los siguientes análisis.

```
#Upper level colesterol
upc_manual <- 400
#Upper level presion arterial
upp_manual <- 150
#Eliminacion registros con outliers en colesterol y presión
pacientes_sinout <- filter(pacientes, pacientes$colesterol <= upc_manual)
pacientes_sinout <- filter(pacientes, pacientes$presion_arterial <= upp_manual)
```

Normalización

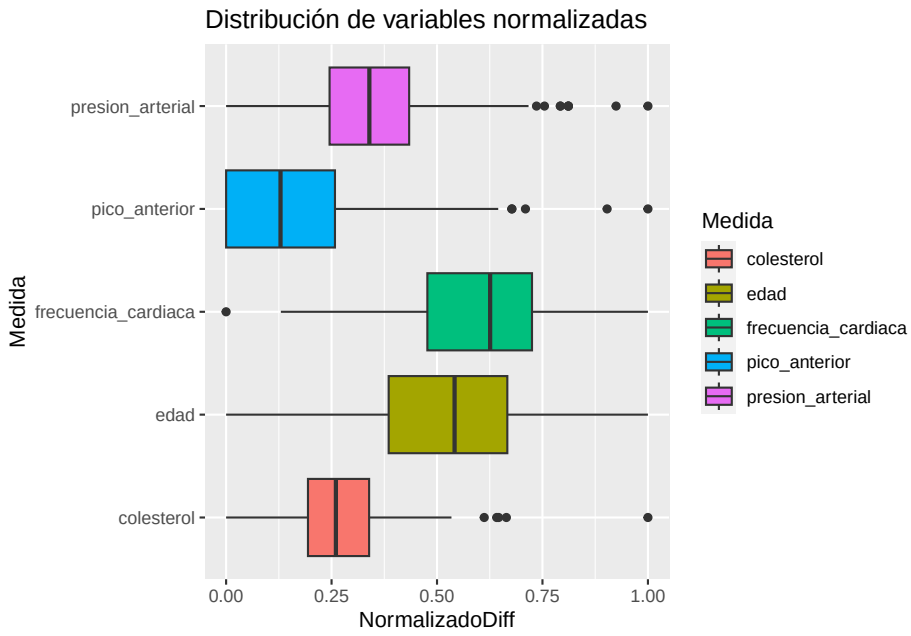
Podríamos normalizar y pasar los valores a la misma escala para mejorar el rendimiento de un posible modelo o visualizar los boxplot más fácilmente, pero perderíamos interpretación de los datos. Así que decidimos no normalizar las variables continuas. Un método hubiera sido por el máximo como se ve a continuación.

```
#Para normalizar sólo puedo usar las columnas numéricas que filtro
# Definimos la función de normalización por el máximo
```

```r
nor <-function(x) { (x -min(x))/(max(x)-min(x))}
# Guardamos un nuevo dataset normalizado de variables numéricas para usar en kmeans
pacientes_num_cont_nor <- as.data.frame(lapply(pacientes_num_cont, nor))
```

Veamos como queda la distribución de las variables numéricas del dataset normalizado. Cuando se usan los boxplot en conjunto para identificar outliers la visualización es mejor si están todas las variables normalizadas.

```r
pacientes_num_cont_nor %>%
  pivot_longer(cols = col_numericas_cont,
               names_to = "Medida",
               values_to = "NormalizadoDiff") %>%
  ggplot() +
  geom_boxplot(aes(x = Medida, y = `NormalizadoDiff`, fill = Medida)) +
  ggtitle("Distribución de variables normalizadas") +
  coord_flip()
```



Discretización de variable edad

Categorizamos la variable edad en rangos para poder hacer el estudio estadístico por grupos con ella y vemos su distribución usando la función table.

```r
pacientes["rango_edad"] <- cut(pacientes$edad, breaks = c(0,45,55,65,100),
                               labels = c("Grupo1:<=45", "Grupo2:46-55",
                                          "Grupo3:56-65","Grupo3:>65"))
table(pacientes$rango_edad)
```

```
##
##  Grupo1:<=45 Grupo2:46-55 Grupo3:56-65   Grupo3:>65
##           64           88          118           33
```

Los hemos categorizado en los siguientes grupos: Grupo1:<=45, Grupo2:46-55, Grupo3:56-65 y Grupo3:>65.

Exportación de los datos preprocesados

Una vez que hemos acometido sobre el conjunto de datos inicial los procedimientos de integración, validación y limpieza anteriores, procedemos a guardar estos en un nuevo fichero denominado heart_clean.csv.csv:

```
write.csv(pacientes, "dataset/heart_clean.csv")
```

Análisis de los datos

Selección de los grupos de datos a analizar

Vamos a seleccionar los grupos dentro de nuestro conjunto de datos que pueden resultar interesantes para analizar y/o comparar. Para esto, utilizaremos la nueva variable discretizada: rango de edad y también usaremos el sexo.

```
# Agrupación por rango de edad: "Grupo1:<=45", "Grupo2:46-55", "Grupo3:56-65",
#"Grupo3:>65"
pacientes.grupoh45 <- pacientes[pacientes$rango_edad == "Grupo1:<=45",]
pacientes.grupoh55 <- pacientes[pacientes$rango_edad == "Grupo2:46-55",]
pacientes.grupoh65 <- pacientes[pacientes$rango_edad == "Grupo3:56-65",]
pacientes.grupom65 <- pacientes[pacientes$rango_edad == "Grupo3:>65",]
# Agrupación por sexo
pacientes.hombres <- pacientes[pacientes$sexo == 1,]
pacientes.mujeres <- pacientes[pacientes$sexo == 0,]
```

Análisis estadístico descriptivo

Tal y como hemos tenido que hacer con anterioridad para realizar el análisis de ouliters en la fase de limpieza de datos, vemos a continuación los datos estadísticos descriptivos de nuestro dataset de pacientes resultante así como los resúmenes estadísticos de los grupos de dataset que hemos creado para poder compararlos.

```
summary(pacientes)
```

```
##       edad       sexo    dolor_pecho presion_arterial   colesterol    azucar
##  Min.   :29.00   0: 96   0:143       Min.   : 94.0    Min.   :126.0   0:258
##  1st Qu.:47.50   1:207   1: 50       1st Qu.:120.0    1st Qu.:211.0   1: 45
##  Median :55.00           2: 87       Median :130.0    Median :240.0
##  Mean   :54.37           3: 23       Mean   :131.6    Mean   :246.3
##  3rd Qu.:61.00                       3rd Qu.:140.0    3rd Qu.:274.5
##  Max.   :77.00                       Max.   :200.0    Max.   :564.0
##  electro frecuencia_cardiaca angina_ejercicio pico_anterior  pendiente
##  0:147   Min.   : 71.0       0:204            Min.   :0.00   0: 21
##  1:152   1st Qu.:133.5       1: 99            1st Qu.:0.00   1:140
##  2:  4   Median :153.0                        Median :0.80   2:142
##          Mean   :149.6                        Mean   :1.04
##          3rd Qu.:166.0                        3rd Qu.:1.60
##          Max.   :202.0                        Max.   :6.20
##  prueba_esfuerzo posibilidad_infarto        rango_edad
##  0:  2           0:138               Grupo1:<=45 : 64
##  1: 18           1:165               Grupo2:46-55: 88
##  2:166                               Grupo3:56-65:118
##  3:117                               Grupo3:>65  : 33
##
##
```

La función summary nos ha dado todos los datos estadísticos necesarios como mínimo, máximo, media, mediana, cuartiles para las variables numéricas continuas y la distribución de las categóricas. Las visualizaciónes ya realizadas en la primera fase con boxplot nos daban la misma información de manera visual.

Como ya comentamos hay outliers, ausencia de distribución normal y también una distribución desequilibrada en las variables categóricas.

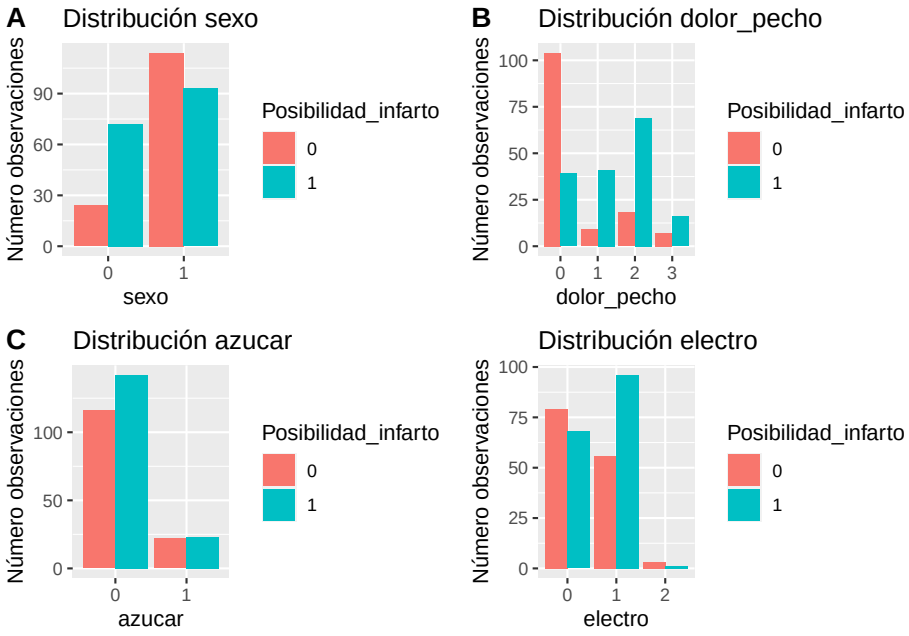
Un análisis visual bivariable usando los rangos de edad y sexo como el que mostramos aquí nos da también de forma más fácil de interpretar el análisis estadístico descriptivo de algunas variables por parejas.

```
#hacemos de forma automática las tablas para todas las variables categóricas
tbcat <- sapply(pacientes_cat, inherits, "numeric")
tbcat
```

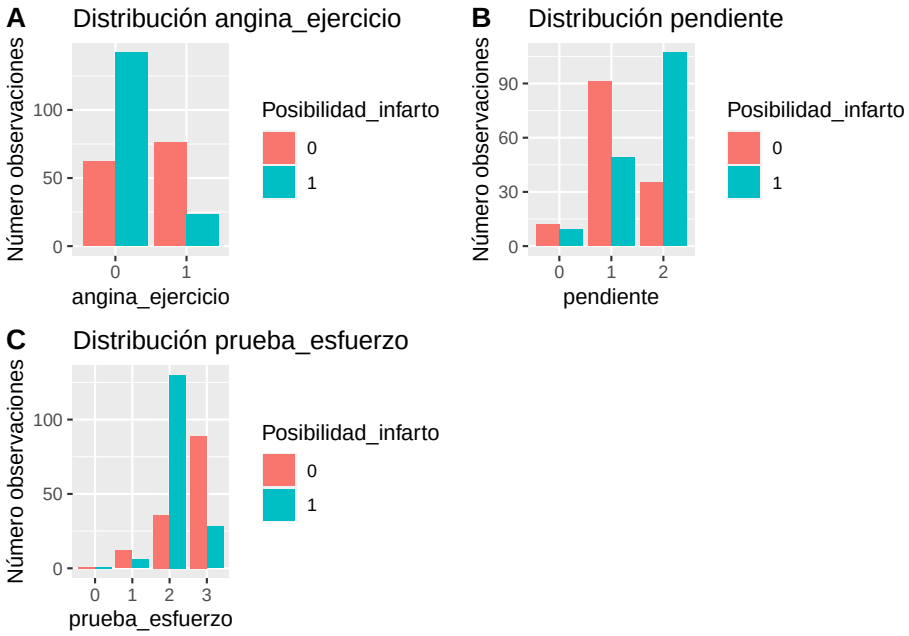
```
##                sexo         dolor_pecho              azucar             electro
##               FALSE               FALSE               FALSE               FALSE
##    angina_ejercicio           pendiente     prueba_esfuerzo posibilidad_infarto
##               FALSE               FALSE               FALSE               FALSE
```

```
#creamos lista con todas las tablas de contingencia para cada variable con
#la objetivo
contg <- lapply(pacientes_cat[!tbcat][-8], table, pacientes_cat[!tbcat][[8]])
#parseamos cada elemento de la lista (una tabla con una variable y la variable
#objetivo) para poder hacer el plot
Distribucion <- function(y) {
    contg[y]
    col <- names(contg[y])
    list_to_df <- as.data.frame(contg[y])
    names(list_to_df) <- c(col,"Posibilidad_infarto","Freq")
    col2 <- "Posibilidad_infarto"
    col3 <- "Freq"
    gp <- ggplot(list_to_df, aes_string(fill=col2, y=col3, x=col)) +
      geom_bar(position='dodge', stat='identity')  +
      labs(title = paste("Distribución",col), x= col, y= "Número observaciones")
  return(gp)
}
figure <- ggarrange(Distribucion(1), Distribucion(2), Distribucion(3),
                    Distribucion(4), Distribucion(5), Distribucion(6),
                    Distribucion(7),
                    labels = c("A", "B", "C"),
                    ncol = 2, nrow = 2)
figure
```

```
## $`1`
```




```
##
## $`2`
```




```
##
## attr(,"class")
## [1] "list"      "ggarrange"
```

Como nos han mostrado las gráficas anteriores, en estas podemos comprobar las siguientes conclusiones:

- Las mujeres tienen proporcionalmente mas riesgo de sufrir un infarto que los hombres.
- Los pacientes con dolores en el pecho de tipo angina típica muestran poca probabilidad de sufrir un infarto, y los dolores que no son de angina o las anginas atipicas muestran un riesgo alto de posible infarto.
- El azúcar en sangre en ayunas no nos indica mucha diferencia en el riesgo de infarto.

- Los resultados electrocardiográficos en reposo de tener anomalías en la onda ST-T indican posiblidad de infarto.
- Como las anginas producidas por el ejercicio son poco problables de posible infarto, pero las anginas sin origen en el ejercicio si que indican un posible infarto.
- La pendiente del segmento ST de ejercicio máximo plano indica que es poco problable un infarto, pero cuando es pendiente ascendente indica una alta posibilidad de infarto.
- El resultado de la prueba de esfuerzo con talio ~ (0,3) siendo defecto reversible indica poca posibilidad de infarto, pero cuando es normal indica un alta posibilidad de infarto.

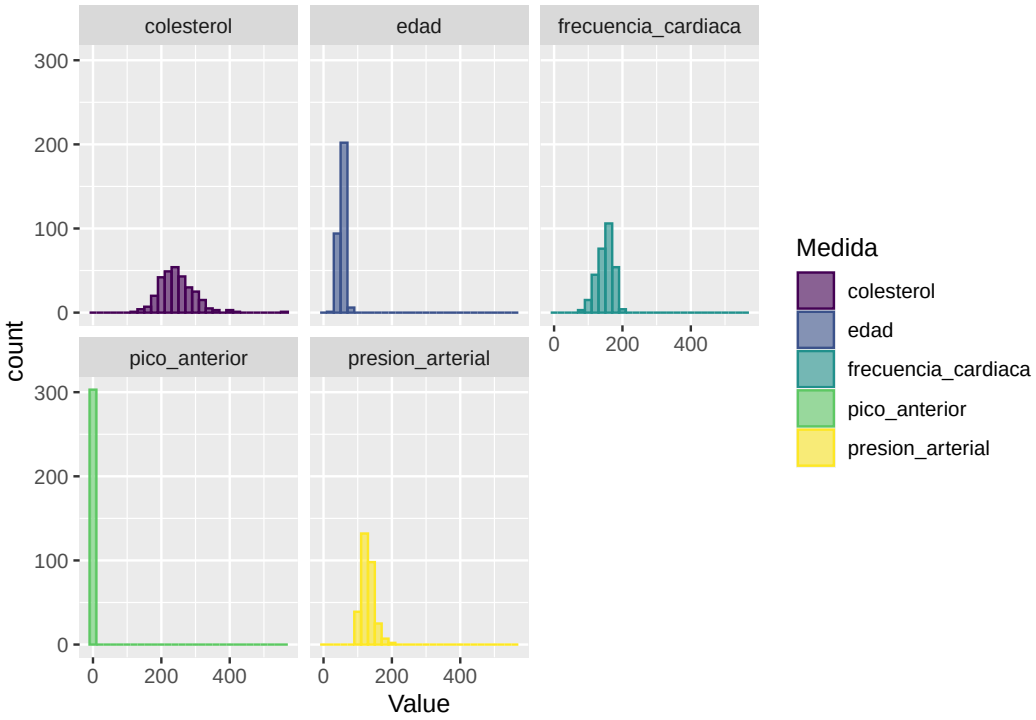
Análisis estadístico inferencial

En este tipo de análisis pretenderemos modelar o inferir cómo es esa población de pacientes de corazón, asumiendo un grado de error en las estimaciones por el hecho de disponer de una muestra reducida de los datos.

Comprobación de la normalidad y homogeneidad de la varianza

Vamos a ver la distribución de los valores de las variables continuas.

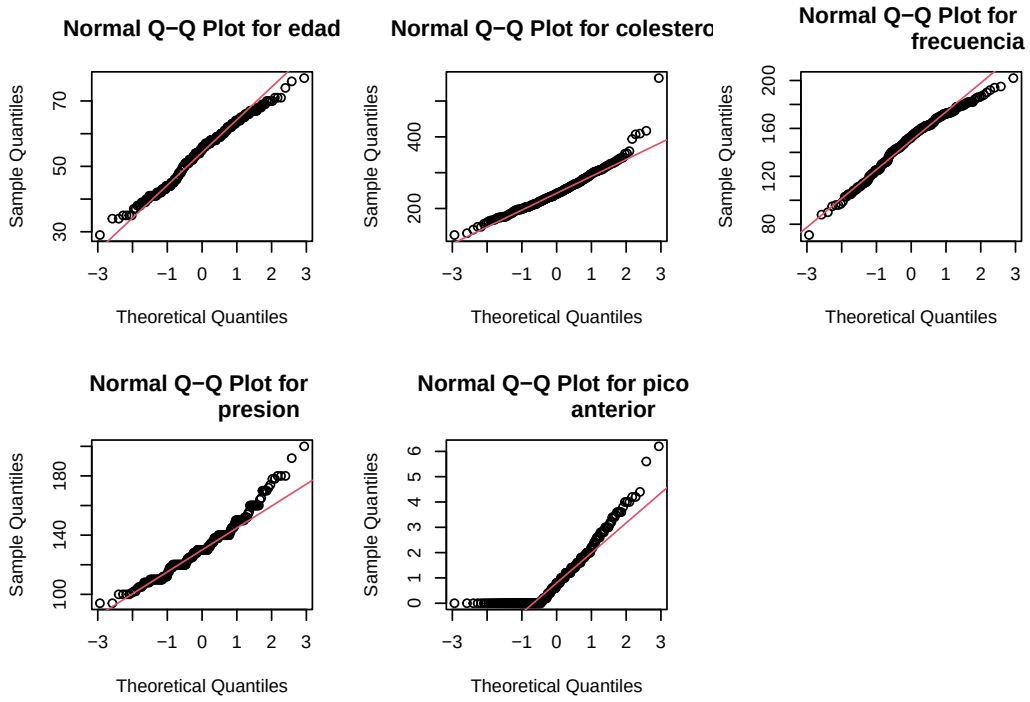
```
#creamos visualizaciones de histogramas
pacientes_num_cont %>%
  pivot_longer(cols = col_numericas_cont,
               names_to = "Medida",
               values_to = "Value") %>%
ggplot( aes(x=Value, color=Medida, fill=Medida)) +
    geom_histogram(alpha=0.6, binwidth = 20) +
    scale_fill_viridis(discrete=TRUE) +
    scale_color_viridis(discrete=TRUE) +

    facet_wrap(~Medida)
```



Por los histogramas vemos que nuestros valores no tienen una distribución normal. De las variables continuas, excepto colesteros, las otras cuatro variables presentan la cola hacia la derecha que denota una asimetría positiva.

Con la función `qqnorm` podemos hacer un Q-Q plot para ver si la variable tiene una distribución normal. Si se aleja de la linea roja, como es el caso, querrá decir que no lo es.

```
op <- par(mfrow=c(2,3))
normal1 <- qqnorm(pacientes$edad, main = "Normal Q-Q Plot for edad");qqline(
  pacientes$edad, col = 2)
normal2 <- qqnorm(pacientes$colesterol, main = "Normal Q-Q Plot for colesterol");
qqline(pacientes$colesterol, col = 2)
normal3 <- qqnorm(pacientes$frecuencia_cardiaca, main = "Normal Q-Q Plot for
                  frecuencia");qqline(pacientes$frecuencia_cardiaca, col = 2)
normal4 <- qqnorm(pacientes$presion_arterial, main = "Normal Q-Q Plot for
                  presion");qqline(pacientes$presion_arterial, col = 2)
normal5 <- qqnorm(pacientes$pico_anterior, main = "Normal Q-Q Plot for pico
                  anterior");qqline(pacientes$pico_anterior, col = 2)
par(op)
```



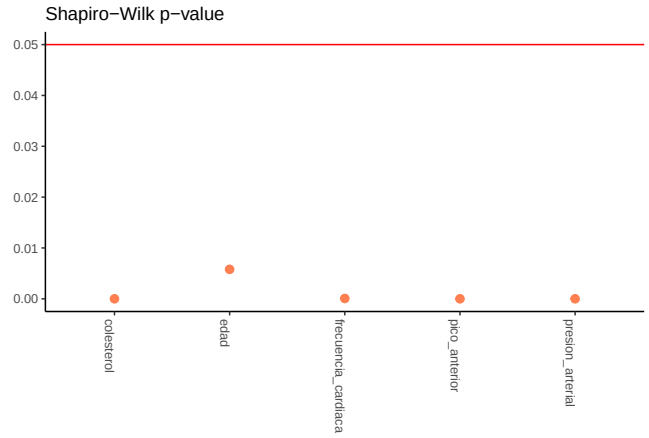
También vamos a **comprobar la normalidad** con los test estadísticos de **Shapiro-Wilk** considerado uno de los métodos más potentes para contrastar la normalidad. Asumiremos como **hipótesis nula que la población está distribuida normalmente**, si el p-valor es menor al nivel de significancia $\alpha A = 0{,}05$, generalmente , entonces la hipótesis nula es rechazada y se concluye que los datos no cuentan con una distribución normal.

Representamos en un plot los valores p-value y la línea roja correspondiente al valor límite αA .

```
shapiro_list <- lapply(pacientes_num_cont, shapiro.test)
shdf <- ldply (sapply(shapiro_list, '[[', 'p.value'), data.frame)
shdf %>% arrange(desc(X..i..))
```

```
##                   .id       X..i..
## 1               edad 5.798359e-03
## 2 frecuencia_cardiaca 6.620819e-05
## 3    presion_arterial 1.458097e-06
## 4          colesterol 5.364848e-09
## 5       pico_anterior 8.183378e-17
```

```
names(shdf) <- c("Variable","Valor")
shdf$id <- "ShapiroPValue"
pcpv = ggplot(shdf, aes(x=Variable, y=Valor)) +
    geom_line(size=1, colour = "coral") +
    geom_point(size=2.5, colour = "coral") + geom_hline(aes(yintercept = 0.05),
                                                        colour="red") +
    ylab("") + xlab("") + ggtitle("Shapiro-Wilk p-value") +
    theme_classic()  + theme(axis.text.x=element_text(angle = -90, hjust = 0))
print(pcpv)
```



Veamos si el test de **Anderson-Darling** nos da el mismo resultado, esta vez mostramos una lista de las variables que no cumplen la hipótesis nula y por tanto no tienen distribución normal.

```
alpha = 0.05
col.names = colnames(pacientes)
for (i in 1:ncol(pacientes)) {
  if (i == 1) cat("Variables que no siguen una distribución normal:\n")
  if (is.integer(pacientes[,i]) | is.numeric(pacientes[,i])) {
    p_val = ad.test(pacientes[,i])$p.value
    if (p_val < alpha) {
      cat(col.names[i]," con p-value:", p_val)
      # Format output
      if (i < ncol(pacientes) - 1) cat(", ")
      if (i %% 3 == 0) cat("\n")
    }
  }
  }
```

```
## Variables que no siguen una distribución normal:
## edad  con p-value: 0.0006570291, presion_arterial  con p-value: 1.672518e-06, colesterol  con p-value:
```

Ahora vamos a estudiar la homogeneidad de varianzas o **homocedasticidad** mediante la aplicación del test **Fligner-Killeen** ya que como hemos comprobado los datos no cumplen con la condición de normalidad. Sino se hubiera podido usar el test de Levene. En ambas pruebas, la **hipótesis nula asume igualdad de varianzas en los diferentes grupos de datos**, por lo que p-valores inferiores al nivel de significancia indicarán heterocedasticidad.

Vamos a analizar la homocedasticidad con todas las variables en los grupos de posibilidad_infarto diferentes.

```r
alpha = 0.05
col.names = colnames(pacientes)
for (i in 1:ncol(pacientes)) {
  if (i == 1) cat("Variables que cumplen heterocedasticidad para
                  posible infarto:\n")
  if (is.integer(pacientes[,i]) | is.numeric(pacientes[,i])) {
    p_val = fligner.test(pacientes[,i] ~ posibilidad_infarto,
                         data = pacientes)$p.value
    if (p_val < alpha) {
      cat("\n",col.names[i]," con p-value:", p_val)
      # Format output
      if (i < ncol(pacientes) - 1) cat(", ")
      if (i %% 3 == 0) cat("\n")
    }
  }
  }
```

```
## Variables que cumplen heterocedasticidad para
##                   posible infarto:
##
##  edad  con p-value: 0.006898429,
##  frecuencia_cardiaca  con p-value: 0.02023875,
##  pico_anterior  con p-value: 1.777207e-08,
```

Para edad, frecuencia cardiaca, pico anterior vemos que no obtenemos un p-valor superior a 0,05, y no podemos aceptar la hipótesis de que las varianzas de ambas muestras son homogéneas.

Pruebas estadísticas

Correlación de variables

En el caso de nuestras variables para analizar la correlación entre ellas, al no pasar los test de normalidad ni homocedasticidad, debemos usar la correlación de Spearman, que aparece como alternativa no paramétrica. Además para variables categóricas con escala ordinal el método Spearman también es el indicado. Obtendremos valores entre -1 y 1 siendo los extremos la correlación negativa o positiva perfectas. Probemos entre presión y frecuencia cardiáca.

```r
#correlación método pearson
cor.test(pacientes$presion_arterial,pacientes$frecuencia_cardiaca)
```

```
##
##  Pearson's product-moment correlation
##
## data:  pacientes$presion_arterial and pacientes$frecuencia_cardiaca
## t = -0.81106, df = 301, p-value = 0.418
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.15854155  0.06632933
## sample estimates:
##         cor
## -0.04669773
```

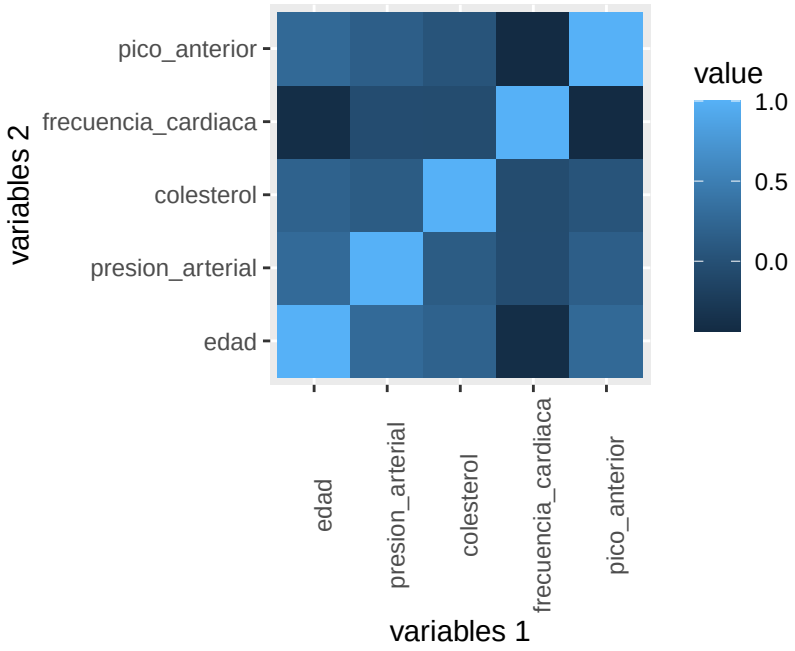
```r
#correlación método spearman
cor.test(pacientes$presion_arterial,pacientes$frecuencia_cardiaca,
         method="spearman")
```

```
##
##  Spearman's rank correlation rho
##
## data:  pacientes$presion_arterial and pacientes$frecuencia_cardiaca
## S = 4823645, p-value = 0.4835
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##         rho
## -0.04040735
```

En ambos casos el p-valor no es significativo y el coeficiente de correlación es negativo y superior a 0.04, siendo más optimista el test pearson, aunque el que debe usarse con estos datos es el de Spearman.

Visualicemos un heatmap a ver si coincide con los datos anteriores.

```r
if(!require('reshape2')) install.packages('reshape2'); library('reshape2')
qplot(x=Var1, y=Var2, data=melt(cor(pacientes_num_cont,
                                    method = "spearman")), fill=value,
      geom="tile",xlab = "variables 1",ylab = "variables 2") +
   theme(axis.text.x = element_text(angle = 90)) +
   coord_fixed()
```



En el gráfico (azul claro) se aprecia una correlación leve entre presión arterial y edad, o colesterol y edad o pico_anterior y edad. Veamos los tests:

```r
#correlación método spearman
cor.test(pacientes$presion_arterial,pacientes$edad, method="spearman")
```

```
##
```

```
##  Spearman's rank correlation rho
##
## data:  pacientes$presion_arterial and pacientes$edad
## S = 3312098, p-value = 4.262e-07
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.2856168
```

```
cor.test(pacientes$colesterol,pacientes$edad, method="spearman")
```

```
##
##  Spearman's rank correlation rho
##
## data:  pacientes$colesterol and pacientes$edad
## S = 3728581, p-value = 0.0006099
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##      rho
## 0.195786
```

```
cor.test(pacientes$pico_anterior,pacientes$edad, method="spearman")
```

```
##
##  Spearman's rank correlation rho
##
## data:  pacientes$pico_anterior and pacientes$edad
## S = 3392424, p-value = 2.159e-06
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.2682912
```

Efectivamente con edad están más correlacionadas aunque muy levemente con coeficiente de correlación 0.26 versus 0.04 anterior.

Contraste de hipótesis

Comparación entre dos grupos de datos

En nuestro dataset de pacientes, puesto que no se cumple la normalidad y homocedasticidad en las variables continuas y como hemos comprobado con los tests paramétricos, para ver que las distribuciones de los grupos de datos que hemos seleccionado anteriormente son las mismas, se deberán aplicar pruebas no paramétricas como **Wilcoxon** (cuando se comparen datos dependientes) o **Mann-Whitney** (cuando los grupos de datos sean independientes).

Vamos a utilizar la función wilcox.test() para realizar las pruebas de Wilcoxon y Mann-Whitney y comparar las distribuciones de todas las variables numéricas continuas y la variable dependiente posibilidad_infarto.

```
alpha = 0.05
col.names = colnames(pacientes)
for (i in 1:ncol(pacientes)) {
  if (i == 1) cat("Variables con las que se ven diferencias estadísticamente
                  significativas para posible infarto:\n")
  if (is.integer(pacientes[,i]) | is.numeric(pacientes[,i])) {
```

```
    p_val = wilcox.test(pacientes[,i] ~ posibilidad_infarto,
                        data = pacientes)$p.value
    if (p_val < alpha) {
      cat("\n",col.names[i]," con p-value:", p_val)
      # Format output
      if (i < ncol(pacientes) - 1) cat(", ")
      if (i %% 3 == 0) cat("\n")
    }
  }
  }
```

```
## Variables con las que se ven diferencias estadísticamente
##                   significativas para posible infarto:
##
##  edad  con p-value: 3.43851e-05,
##  presion_arterial  con p-value: 0.03465245,
##  colesterol  con p-value: 0.03571518,
##  frecuencia_cardiaca  con p-value: 9.796555e-14,
##  pico_anterior  con p-value: 2.406979e-13,
```

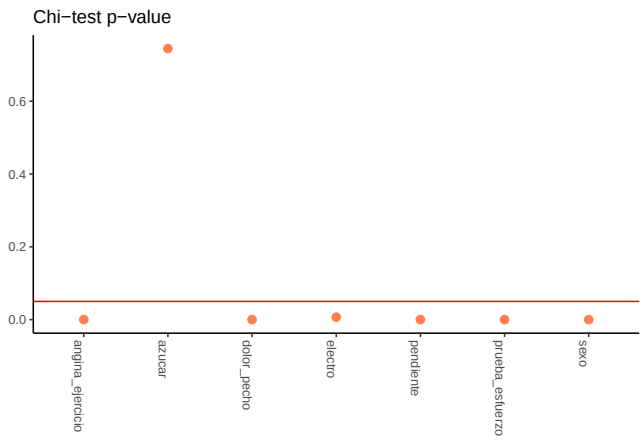
En este caso, sí se observan diferencias estadísticamente significativas en la todas las variables numéricas entre posibilidad mayor o menor de infarto.

Puesto que tenemos tantas variables categóricas, vamos a ver si existen diferencias significativas en cada variable categórica y los grupos definidos por la variable categórica posibilidad_infarto. Para ello hay que aplicar el test de **chi^2** en R, mediante la función chisq.test().

```
chiq_list <- lapply(contg, chisq.test)
chdf <- ldply (sapply(chiq_list, '[[', 'p.value'), data.frame)
chdf %>% arrange(desc(X..i..))
```

```
##                .id       X..i..
## 1           azucar 7.444281e-01
## 2          electro 6.660599e-03
## 3             sexo 1.876778e-06
## 4        pendiente 4.830682e-11
## 5 angina_ejercicio 7.454409e-14
## 6      dolor_pecho 1.334304e-17
## 7  prueba_esfuerzo 2.233351e-18
```

```
names(chdf) <- c("Variable","Valor")
chdf$id <- "ChiTestPValue"
pcpv2 = ggplot(chdf, aes(x=Variable, y=Valor)) +
    geom_line(size=1, colour = "coral") +
    geom_point(size=2.5, colour = "coral") + geom_hline(aes(yintercept = 0.05),
                                                        colour="red") +
    ylab("") + xlab("") + ggtitle("Chi-test p-value") +
    theme_classic()  + theme(axis.text.x=element_text(angle = -90, hjust = 0))
print(pcpv2)
```



Como podemos comprobar, en este caso sólo rechazamos **la hipótesis 0, es decir que no hay diferencias estadísticas significativas para los diferentes grupos**, para el azúcar.

Regresión

Tras haber visto la correlación entre algunas variables numéricas, vamos a analizar con un modelo de regresión lineal la relación entre alguna de ellas: presión arterial y edad, y entre edad y frecuencia cardiáca, así como entre frecuencia cardiaca y presión arterial.

```
m1 <- lm(presion_arterial~edad,data=pacientes)
m2 <- lm(frecuencia_cardiaca~edad,data=pacientes)
m3 <- lm(presion_arterial~frecuencia_cardiaca,data=pacientes)
```

Siendo el coeficiente de determinación (R-squared) una medida de calidad del modelo que toma valores entre 0 y 1, se comprueba cómo se correlacionan muy débilmente las variables seleccionadas, y en el caso de presion_arterial y frecuencia_cardiaca la correlación es negativa.

Podemos usar la regresión logística para analizar la regresión usando la variable dicotómica posibilidad_infarto en función de las demás variables predicatoras.

```
m4<- glm(posibilidad_infarto ~ colesterol+presion_arterial,data=pacientes,
         family="binomial")
```

Este modelo nos proporciona un AIC (criterio de información de Akaike) de 415.8. Probemos introduciendo la edad.

```
m5 <- glm(posibilidad_infarto ~ colesterol+presion_arterial+edad,data=pacientes,
         family="binomial")
summary(m5)
```

```
##
## Call:
## glm(formula = posibilidad_infarto ~ colesterol + presion_arterial +
##     edad, family = "binomial", data = pacientes)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7357  -1.1502   0.7949   1.0571   1.5904
##
## Coefficients:
```

```
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)       4.354296   1.132266   3.846  0.00012 ***
## colesterol       -0.001378   0.002353  -0.585  0.55825
## presion_arterial -0.010324   0.007106  -1.453  0.14626
## edad             -0.045344   0.014281  -3.175  0.00150 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 417.64  on 302  degrees of freedom
## Residual deviance: 399.25  on 299  degrees of freedom
## AIC: 407.25
##
## Number of Fisher Scoring iterations: 4
```

Este modelo nos proporciona un AIC (criterio de información de Akaike) de 407.25. Por lo que el modelo anterior (m4) es mejor.

Vamos a usarlo para intentar predecir la variable posibilidad_infarto en nuevos datos.

```
nuevo_paciente <- data.frame(
colesterol = 220,
presion_arterial = 100
)
# Predecir el precio
predict(m4, nuevo_paciente)
```

```
##         1
## 0.7598769
```

Parece que el nuevo paciente está más cerca de tener más posibilidades de infarto que de tener menos con un resultado de 0.76.

Conclusiones

En esta práctica hemos analizado el dataset de Kaggle dataset sobre ataques de corazón, donde nos hemos planteado si alguno de los valores de pruebas realizadas en ellos podría ser determinante en que la posibilidad de infarto sea mayor o menor para el paciente.

Para ello primero hemos ralizado tareas de limpieza de datos, tales como convertir las variables a sus respectivos tipos, filtrar que variables nos interesa utilizar e incluso limpiar el dataset de valores nulos. En el caso de los valores extremos hemos decidido diverger el dataset en uno donde los incluye, para asi poder estudiar todos los casos, y otro donde los excluye, para poder trabajar con datos mas generalizados. Ademas, hemos comprobado como quedarían los valores numéricos si se normalizaban y hemos discretizado la variable edad, para poder trabajar con ella de forma más eficiente.

A continuación hemos realizado un análisis estadístico descriptivo, donde las conclusiones han sido las siguientes: Hay mayor posibilidad de infarto en mujeres y en pacientes con dolores que no son de angina o las anginas atípica con respecto a los otros. El azúcar no tiene importancia significativa en el riesgo de infarto. Otros indicios de posibilidad de infarto alta son: tener anomalías en la onda ST-T en el electrocardiograma, las anginas de origen distinto al ejercicio, una pendiente ascendiente y una prueba de esfuerzo con talio ~ (0,3) normal.

Tambien hemos realizado un análisis estadístico inferencial, y hemos podido comprobar que ninguna de las variables numéricas ha pasado el test de distribución normal ni homocedasticidad, por ello los test analíticos realizados han sido no paramétricos.

Para finalizar, hemos realizado tres tipos de pruebas estadísticas al conjunto de datos para poder conocer nuestras muestras, saber cuanta información proveen con respecto a la población.

Las pruebas han sido buscar correlación de variables, el contraste de distintas hipótesis (diferencias estadísticamente significativas) y generar un modelo de regresión para predecir si un paciente tiene riesgo a tener un infarto. Las conclusiones que hemos sacado en base a los resultados obtenidos han sido las siguientes:

- El análisis de correlación nos ha permitido averiguar que no hay correlación fuerte entre las variables ni con la variable objetivo.

- El contraste de hipótesis nos ha permitido conocer que, menos la variable de nivel de azucar en sangre, con todas las variables rechazaríamos la hipótesis nula. Dicho de otro modo, estas variables ejercen una mayor influencia sobre la posibilidad de infarto.

- El modelo de regresión logística obtenido parece no tener calidad aunque hemos podido utilizarlo para predecir la posibilidad de infarto en un nuevo paciente.

Contribuciones	Firma
Investigación previa	Enrique, Inma
Redacción de las respuestas	Enrique, Inma
Desarrollo del código	Enrique, Inma
Participación en el vídeo	Enrique, Inma

Bibliografía

- Descripción de variables dataset: https://www.kaggle.com/datasets/rashikrahmanpritom/heart-attack-analysis-prediction-dataset/discussion/329925
- Test for homogeneity of variances: https://biostats.w.uib.no/test-for-homogeneity-of-variances-levenes-test/
- Data science concepts you need to know! Part 1: https://towardsdatascience.com/introduction-to-statistics-e9d72d818745
- Introducción a la limpieza y análisis de los datos: Calvo M, Subirats L, Pérez D (2019). Introducción a la limpieza y análisis de los datos. Editorial UOC.