Práctica 5. Estadística descriptiva 1

Jesús Martín Fernández

Contenidos

1. Introducción

La estadística descriptiva es una rama de la estadística que se encarga de resumir y organizar los datos obtenidos de una muestra de manera clara y comprensible, utilizando medidas como la media, la mediana, la moda, y la desviación estándar, además de representaciones gráficas como histogramas o diagramas de dispersión. Es crucial en cualquier estudio porque permite obtener una visión general de las características de la muestra, lo que facilita la interpretación inicial de los datos. Al conocer bien nuestra muestra, podemos evaluar si es representativa de la población objetivo, lo que impacta en la validez externa del estudio, es decir, en la posibilidad de generalizar los resultados. Además, la estadística descriptiva nos ayuda a valorar la calidad de nuestros datos, detectando posibles sesgos, errores o valores atípicos que podrían afectar el análisis posterior.

Por otra parte, las variables cualitativas y cuantitativas requieren diferentes medidas de resumen según su naturaleza. Las **cualitativas nominales** se clasifican en categorías sin orden y se describen con frecuencias y porcentajes (también la moda), mientras que las **ordinales**, al tener jerarquía, también pueden resumirse con la mediana y percentiles. Las **cuantitativas discretas** se cuentan y se resumen con medidas como la media, la moda y también percentiles, que permiten describir su distribución. Con cierta precauciones también se pueden estudiar sus medidas de dispersión. Las **cuantitativas continuas** se miden en un rango infinito y se describen con medidas como la media, mediana, desviación estándar y percentiles para reflejar tanto la tendencia central como la dispersión.

1.1. Creando un entorno de trabajo para el análisis descriptivo.

Vamos a crear el directorio de trabajo y comprobar que estamos en él

```
setwd ("Práctica 5") #La ruta o pathway es diferente para cada uno.
getwd ()
```

```
[1] "/Práctica 5"
```

Ahora recuperamos la base de datos iam (df_iam Aula Virtual, Práctica 5), es un archivo.csv

```
df_iam <- read.csv ("df_iam.csv")
```

Vamos a ver cómo está construido el archivo

```
head(df_iam)
```

```
    fech_nac sex   alt peso hta fum colesterol clas_soc iam
1 1983-08-06   0 157.2 54.2   1   1        192        1   1
2 1983-09-23   1 172.5 70.5   0   1        176        1   0
3 1940-03-03   0 167.4 63.4   1   0        226        0   0
4 1937-01-14   1 173.3 96.3   1   0        188        0   0
5 1941-03-23   1 172.7 62.7   0   0        178        1   1
6 1963-11-10   0 154.3 51.3   1   0        157        1   0
```

Vamos a atribuir valores a las categorías de las variables que lo permiten

```
# Convertir variables a factor con etiquetas

df_iam$sex <- factor(df_iam$sex,
          levels = c(0, 1), labels = c("Mujer", "Varón"))
df_iam$hta <- factor(df_iam$hta, levels = c(0, 1), labels = c("No", "Sí"))
df_iam$fum <- factor(df_iam$fum, levels = c(0, 1), labels = c("No", "Sí"))
df_iam$clas_soc <- factor(df_iam$clas_soc, levels = c(0, 1),
                          labels = c("Baja", "Alta"))
df_iam$iam <- factor(df_iam$iam, levels = c(0, 1), labels = c("No", "Sí"))



# Mostrar las primeras filas del dataframe con los factores etiquetados
head(df_iam)
```

```
    fech_nac   sex   alt peso hta fum colesterol clas_soc iam
1 1983-08-06 Mujer 157.2 54.2  Sí  Sí        192     Alta  Sí
2 1983-09-23 Varón 172.5 70.5  No  Sí        176     Alta  No
3 1940-03-03 Mujer 167.4 63.4  Sí  No        226     Baja  No
4 1937-01-14 Varón 173.3 96.3  Sí  No        188     Baja  No
5 1941-03-23 Varón 172.7 62.7  No  No        178     Alta  Sí
6 1963-11-10 Mujer 154.3 51.3  Sí  No        157     Alta  No
```

Crear la variable edad en años cumplidos (edad) teniendo sabiendo que `fech_nac` es la fecha de nacimiento y que el estudio finalizó el 31/12/2023

```r
#fech_nac viene definida como character, cambiamos a formato fecha
df_iam$fech_nac <- as.Date(df_iam$fech_nac)
fecha_fin <- as.Date("2023-12-31")
df_iam$edad <- (fecha_fin-df_iam$fech_nac)/365.25

df_iam$edad <- as.numeric (round (df_iam$edad,0))

print (df_iam$edad [1:10])
```

```
 [1] 40 40 84 87 83 60 79 74 48 84
```

Crea una nueva variable, IMC y recategorizala en los grupos “bajo” si IMC <20, “normal” si 20<=IMC<30 y “obesidad” si 30<=IMC

```r
df_iam$imc <- df_iam$peso / (df_iam$alt / 100)^2
df_iam$imc <- round(df_iam$imc, 2) #redondeamos a 2 decimales
df_iam$imc_r <- cut(df_iam$imc,
                        breaks = c(-Inf, 20, 30, Inf),
                        labels = c("Bajo", "Normal", "Obesidad"))

#En esta orden los valores no son incluidos en la categoría inferior.

# Verificamos la nueva columna
head (df_iam$imc_r)
```

2. Análisis descriptivo variables cualitativas

En primer lugar calcularemos las frecuencias absolutas de las variables sex y hta.

Para calcular las frecuencias absolutas de una variable categórica, usamos la función `table()` en **R**.

```r
frecuencias_sex <- table(df_iam$sex)
frecuencias_hta <- table(df_iam$hta)
print (frecuencias_sex)
```

```
Mujer Varón
  471   465
```

```r
print (frecuencias_hta)
```

```
 No  Sí
519 417
```

En segundo lugar vamos a calcular las proporciones de varones y de hipertensos con la función `prop.table()`

```r
proporciones_sex <- prop.table(frecuencias_sex) * 100
proporciones_hta <- prop.table(frecuencias_hta) * 100
# Multiplicamos por 100 para porcentaje


print(proporciones_sex)
```

```
   Mujer    Varón
50.32051 49.67949
```

```r
print (proporciones_hta)
```

```
      No       Sí
55.44872 44.55128
```

Se pueden redondear a dos cifras las proporciones y añadirles el signo `%`, con las funciones `round()` y `paste()`

```r
proporciones_sex <- round(proporciones_sex, 2)
proporciones_sex <- paste(proporciones_sex,"%")

proporciones_hta <- round(proporciones_hta, 2)
proporciones_hta <- paste(proporciones_hta,"%")

print (proporciones_sex)
```

```
[1] "50.32 %" "49.68 %"
```

```r
print (proporciones_hta)
```

```
[1] "55.45 %" "44.55 %"
```

Vamos a averiguar las frecuencias absolutas de hta en varones y mujeres

```r
hta_by_sex <- table(df_iam$hta, df_iam$sex)
#primero filas y después columnas

print(hta_by_sex)
```

```
    Mujer Varón
  No   250   269
  Sí   221   196
```

Y ahora las proporciones

```r
proporciones_filas <- prop.table(hta_by_sex, margin = 1) * 100
#margin=1, se puede sustituir simplemente por 1
print (proporciones_filas)
```

```
       Mujer    Varón
  No 48.16956 51.83044
  Sí 52.99760 47.00240
```

```r
proporciones_columnas <- prop.table(hta_by_sex, margin = 2) * 100
print (proporciones_columnas)
```

```
       Mujer    Varón
  No 53.07856 57.84946
  Sí 46.92144 42.15054
```

Redondeamos los valores de las proporciones y les acompañamos del símbolo `%`

```r
proporciones_filas <- round(proporciones_filas, 2)
proporciones_filas <- paste(proporciones_filas,"%")

proporciones_columnas <- round(proporciones_columnas, 2)
proporciones_columnas <- paste(proporciones_columnas,"%")

print (proporciones_filas)
```

```
[1] "48.17 %" "53 %"    "51.83 %" "47 %"
```

```r
print (proporciones_columnas)
```

```
[1] "53.08 %" "46.92 %" "57.85 %" "42.15 %"
```

Ahora vamos a estudiar las frecuencias y proporciones de la hta por sexo en cada clase social. La función `table()` permite hacerlo, pero construye un array.

```r
# Crear la tabla de contingencia entre 'hta', 'sex' y 'clas_soc'
hta_by_sex_clas_soc <- table(df_iam$hta, df_iam$sex, df_iam$clas_soc)

# Imprimir la tabla de contingencia
print("Tabla de contingencia entre hta, sex y clas_soc:")
```

```
[1] "Tabla de contingencia entre hta, sex y clas_soc:"
```

```r
print(hta_by_sex_clas_soc)
```

```
, ,  = Baja


    Mujer Varón
  No   114   116
  Sí   136   112

, ,  = Alta


    Mujer Varón
  No   136   153
  Sí    85    84
```

Si queremos saber la suma de las proporciones de hta por sex en cada grupo de clas_soc hay que hacer los siguiente

```
proporciones_hta_sex_clas_soc <- prop.table(hta_by_sex_clas_soc,
                                    margin = c(2, 3))*100
print (proporciones_hta_sex_clas_soc)
```

```
, ,  = Baja


       Mujer    Varón
  No 45.60000 50.87719
  Sí 54.40000 49.12281

, ,  = Alta


       Mujer    Varón
  No 61.53846 64.55696
  Sí 38.46154 35.44304
```

Redondeamos los valores de las proporciones y les acompañamos del símbolo %

```
proporciones_hta_sex_clas_soc <- round(proporciones_hta_sex_clas_soc, 2)
proporciones_hta_sex_clas_soc <- paste0(proporciones_hta_sex_clas_soc, "%")
print (proporciones_hta_sex_clas_soc)
```

```
[1] "45.6%"  "54.4%"  "50.88%" "49.12%" "61.54%" "38.46%" "64.56%" "35.44%"
```

Podemos ver una presentación más "amigable"

```r
hta_sex_clas_soc <- table(df_iam$hta, df_iam$sex, df_iam$clas_soc)

# Calcular las proporciones de hta por sex en cada grupo de clas_soc
proporciones_hta_sex_clas_soc <- prop.table(hta_sex_clas_soc,
                                            margin = c(2, 3)) * 100

# Convertir la tabla de proporciones a un dataframe
df_proporciones <- as.data.frame(proporciones_hta_sex_clas_soc)

# Renombrar las columnas para mayor claridad
colnames(df_proporciones) <- c("HTA", "Sex", "Clas_Soc", "Proporcion")

# Redondear las proporciones a 2 decimales y añadir el símbolo %
df_proporciones$Proporcion <- paste0(round(df_proporciones$Proporcion, 2), "%")

print(df_proporciones)
```

```
  HTA   Sex Clas_Soc Proporcion
1  No Mujer     Baja      45.6%
2  Sí Mujer     Baja      54.4%
3  No Varón     Baja     50.88%
4  Sí Varón     Baja     49.12%
5  No Mujer     Alta     61.54%
6  Sí Mujer     Alta     38.46%
7  No Varón     Alta     64.56%
8  Sí Varón     Alta     35.44%
```

3. Análisis descriptivo variables cuantitativas

3.1 Medidas resumen de una distribución

En primer lugar vamos a calcular el rango , el máximo y el mínimo de las variables `alt` y `peso` , ya mencionamos las funciones en prácticas anteriores. Calcula también los quintiles

```r
# Rango
rango_alt <- range(df_iam$alt, na.rm = TRUE)
rango_peso <- range(df_iam$peso, na.rm = TRUE)

#sabemos que no hay valores faltantes, pero eso no es lo habitual

# Diferencia entre máximo y mínimo (rango)
rango_alt_valor <- diff(rango_alt)
rango_peso_valor <- diff(rango_peso)

# Máximo
max_alt <- max(df_iam$alt, na.rm = TRUE)
max_peso <- max(df_iam$peso, na.rm = TRUE)

# Mínimo
min_alt <- min(df_iam$alt, na.rm = TRUE)
min_peso <- min(df_iam$peso, na.rm = TRUE)

# Quintiles
quintiles_alt <- quantile(df_iam$alt,
                          probs = seq(0, 1, by = 0.20),
                          na.rm = TRUE)
quintiles_peso <- quantile(df_iam$peso,
                           probs = seq(0, 1, by = 0.20),
                           na.rm = TRUE)

#Con esta función puedes calcular deciles, cuartiles, ...by = 0.10, by= 0.25,...

# Resultados
rango_alt_valor
```

```
[1] 54
```

```r
rango_peso_valor
```

```
[1] 77
```

```r
max_alt
```

```
[1] 199
```

```
max_peso
```

```
[1] 122
```

```
min_alt
```

```
[1] 145
```

```
min_peso
```

```
[1] 45
```

```
quintiles_alt
```

```
   0%   20%   40%   60%   80%  100% 
145.0 156.2 163.6 170.0 177.0 199.0 
```

```
quintiles_peso
```

```
   0%   20%   40%   60%   80%  100% 
 45.0  60.8  70.6  77.7  87.9 122.0 
```

Ahora obtén la media y la mediana de ambas variables. También conoces las funciones. Sin poder asegurarlo, ¿dirías que ambas variables se distribuyen como una normal?

```
media_alt <- mean(df_iam$alt, na.rm = TRUE)
media_peso <- mean(df_iam$peso, na.rm = TRUE)

#sabemos que no hay valores faltantes, pero eso no es lo habitual

media_alt
```

```
[1] 167.0214
```

```
media_peso
```

```
[1] 74.46624
```

```r
mediana_alt <- median(df_iam$alt, na.rm = TRUE)
mediana_peso <- median(df_iam$peso, na.rm = TRUE)

mediana_alt
```

```
[1] 166.8
```

```r
mediana_peso
```

```
[1] 74.3
```

Calcula la varianza y la desviación típica de las distribuciones de `alt` y `peso`

```r
varianza_alt <- var(df_iam$alt, na.rm=TRUE)
varianza_peso <- var(df_iam$peso, na.rm=TRUE)

desv_alt <- sd(df_iam$alt, na.rm=TRUE)
desv_peso <- sd(df_iam$peso, na.rm=TRUE)

varianza_alt
```

```
[1] 132.624
```

```r
varianza_peso
```

```
[1] 229.3202
```

```r
desv_alt
```

```
[1] 11.51625
```

```r
desv_peso
```

```
[1] 15.14332
```

Ahora calcula la medida de `alt` y `peso` en varones y en mujeres

```r
# Media de alt para varones
media_alt_V <- mean(df_iam$alt[df_iam$sex == "Varón"], na.rm = TRUE)

# Media de alt para mujeres
media_alt_M <- mean(df_iam$alt[df_iam$sex == "Mujer"], na.rm = TRUE)

# Media de peso para varones
media_peso_V <- mean(df_iam$peso[df_iam$sex == "Varón"], na.rm = TRUE)

# Media de peso para mujeres
media_peso_M <- mean(df_iam$peso[df_iam$sex == "Mujer"], na.rm = TRUE)

# Mostrar los resultados
media_alt_V
```

```
[1] 174.657
```

```r
media_alt_M
```

```
[1] 159.483
```

```r
media_peso_V
```

```
[1] 82.00624
```

```r
media_peso_M
```

```
[1] 67.02229
```

Una forma más conveniente de estudiar las variables respecto a las caracteriísticas de una variable factor es con la función `tapply()`.

La función `tapply()` es muy útil para realizar cálculos de estadísticas descriptivas para subconjuntos de datos definidos por un factor. Es especialmente conveniente cuando necesitas aplicar la misma función a diferentes grupos dentro de un dataset, permitiendo análisis comparativos entre esos grupos. Mira un ejemplo que nos permite valorar la distribución de `alt` y `peso` en varones y en mujeres.

```r
# Calcular la media de alt por grupo de sexo
media_alt <- tapply(df_iam$alt, df_iam$sex, mean, na.rm = TRUE)

# Calcular la varianza de alt por grupo de sexo
varianza_alt <- tapply(df_iam$alt, df_iam$sex, var, na.rm = TRUE)

# Calcular la desviación estándar de alt por grupo de sexo
desviacion_alt <- tapply(df_iam$alt, df_iam$sex, sd, na.rm = TRUE)

# Calcular la media de peso por grupo de sexo
media_peso <- tapply(df_iam$peso, df_iam$sex, mean, na.rm = TRUE)

# Calcular la varianza de peso por grupo de sexo
varianza_peso <- tapply(df_iam$peso, df_iam$sex, var, na.rm = TRUE)

# Calcular la desviación estándar de peso por grupo de sexo
desviacion_peso <- tapply(df_iam$peso, df_iam$sex, sd, na.rm = TRUE)

# Mostrar resultados
media_alt
```

```
  Mujer   Varón
159.483 174.657
```

```r
varianza_alt
```

```
   Mujer    Varón
63.64120 86.67228
```

```r
desviacion_alt
```

```
   Mujer    Varón
7.977543 9.309795
```

```r
media_peso
```

```
   Mujer    Varón
67.02229 82.00624
```

```
varianza_peso
```

```
   Mujer    Varón
160.8621 185.9355
```

```
desviacion_peso
```

```
   Mujer    Varón
12.68314 13.63582
```

Otra forma de hacer los mismo, la función `by`

```
# Calcular la media de peso por grupo de sexo
media_peso <- by(df_iam$peso, df_iam$sex, function(x) mean(x, na.rm = TRUE))

# Calcular la varianza de peso por grupo de sexo
varianza_peso <- by(df_iam$peso, df_iam$sex, function(x) var(x, na.rm = TRUE))

# Calcular la desviación estándar de peso por grupo de sexo
desviacion_peso <- by(df_iam$peso, df_iam$sex, function(x) sd(x, na.rm = TRUE))
```

El paquete `psych` tiene una función , denominada `describe` que ofrece la información sobre toda la distribución de una variable contínua. Vamos a ver un ejemplo

```
#install.packages("psych")

library(psych)
```

```
Warning: package 'psych' was built under R version 4.4.1
```

```
# Calcular estadísticas básicas para 'alt' y 'peso'
stats_alt <- describe (df_iam$alt)
stats_peso <- describe(df_iam$peso)



# Mostrar los resultados
stats_alt
```

```
   vars   n   mean    sd median trimmed   mad min max range skew kurtosis   se
X1    1 936 167.02 11.52  166.8  166.75 12.31 145 199    54  0.2    -0.52 0.38
```

```
stats_peso
```

```
   vars   n  mean    sd median trimmed   mad min max range skew kurtosis   se
X1    1 936 74.47 15.14   74.3   74.21 15.86  45 122    77 0.18    -0.39 0.49
```

3.2 Medidas sobre la forma de una distribución

Las medidas que evalúan la forma de una distribuión son la asimetría ("skewness") y el puntamiento ("curtosis"). La **asimetría** indica el grado y la dirección de la falta de simetría de una distribución. Un valor de asimetría cercano a 0 sugiere una distribución simétrica, un valor positivo indica que la distribución tiene una cola más larga o pesada a la derecha (asimetría positiva), y un valor negativo refleja que la cola es más larga o pesada a la izquierda (asimetría negativa). El **apuntamiento** (curtosis), por otro lado, mide la "altura" o el grado de concentración de los datos en el centro de la distribución. Una curtosis normalizada de 0 indica una distribución similar a la normal (mesocúrtica), un valor positivo sugiere que la distribución es más apuntada y tiene colas más pesadas (leptocúrtica), mientras que un valor negativo indica una distribución más plana y con colas menos pronunciadas (platicúrtica).

Vamos a calcula los valores de asimetría y curtosis de las distribuciones de las variables `alt` y `peso`.

Existen dos formas, una es implementar la fórmular de la asimetría y la curtosis y luego aplicar la función a las distribuciones, y una más sencilla, que utilizaremos aquí. Vamos a descargar el paquete `e1071`, que también tiene funciones directas para calcular la asimetría y curtosis.(hay que obtener

```
options(repos = c(CRAN = "https://cloud.r-project.org/"))
install.packages("e1071")
```

```
package 'e1071' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
    C:\Users\jesus.martin\AppData\Local\Temp\Rtmp4ehZOD\downloaded_packages
```

```
library(e1071)
```

```
Warning: package 'e1071' was built under R version 4.4.1
```

```r
asimetria_alt <- skewness(df_iam$alt, na.rm = TRUE)
asimetria_peso <- skewness(df_iam$peso, na.rm = TRUE)

curtosis_alt <- kurtosis(df_iam$alt, na.rm = TRUE)
curtosis_peso <- kurtosis(df_iam$peso, na.rm = TRUE)

asimetria_alt
```

```
[1] 0.2001601
```

```r
asimetria_peso
```

```
[1] 0.1752089
```

```r
curtosis_alt
```

```
[1] -0.51884
```

```r
curtosis_peso
```

```
[1] -0.3863402
```

Calcula la asimetría y curtosis de `alt` y `peso` en las mujeres

```r
asimetria_alt_M <- skewness(df_iam$alt[df_iam$sex == "Mujer"], na.rm = TRUE)
curtosis_alt_M <- kurtosis(df_iam$alt[df_iam$sex == "Mujer"], na.rm = TRUE)

asimetria_alt_M
```

```
[1] 0.1690368
```

```r
curtosis_alt_M
```

```
[1] -0.6301795
```