

Práctica 5. Ejercicios

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

```
df_iam <- read.csv ("~/Práctica 5_ejercicios") #Revisar ruta
```

Mira qué variables tiene, y qué características tiene cada una

```
head(df_iam)
```

Atribuye las categorías “Varón (Sex==1) y”Mujer” (sex==0)

```
# Convertir variables a factor con etiquetas
```

```
df_iam$sex <- factor(df_iam$sex, levels = c(0, 1), labels = c("Mujer", "Varón"))
```

Etiqueta las respuestas dicotómicas (“no”==0) y Sí ==1) para hta , fum , y “Baja” y “Ala” a los valores (0,1) de la variable clas_soc

```
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)
```

Crear la variable edad en años cumplidos (edad) teniendo sabiendo que fech_nac es la fecha de nacimiento y que el estudio finalizó el 31/05/2024

```
#fecha_nac viene definida como character, cambiamos a formato fecha
df_iam$fecha_nac <- as.Date(df_iam$fecha_nac)
fecha_fin <- as.Date("2024-05-31")
df_iam$edad <- (fecha_fin-df_iam$fecha_nac)/365.25

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

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

Crea una nueva variable, IMC y recategorízala en los grupos “bajo” si $IMC < 20$, “normal” si $20 \leq IMC < 30$ y “obesidad” si $30 \leq IMC$

```
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"))
```

Calcula las frecuencias absolutas de las variables `fum` y `clas_soc`

```
frecuencias_fum <- table(df_iam$fum)
frecuencias_clas_soc <- table(df_iam$clas_soc)
print (frecuencias_fum)
print (frecuencias_clas_soc)
```

Calcula las proporciones de fumadores en varones y mujeres utilizando la función `table()` y `prop.table()`

```
# Creamos una tabla de frecuencias cruzadas entre sexo y fumadores
tabla_fum <- table(df_iam$sex, df_iam$fum)
tabla_fum
# Calculamos la proporción de fumadores dentro de cada grupo de sexo
proporciones_fum <- prop.table(tabla_fum, 1)
proporciones_fum
```

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

```
# Multiplicamos las proporciones por 100 para obtener los porcentajes
porcentajes_fum <- proporciones_fum[, 2] * 100
porcentajes_fum

# Agregamos el símbolo % a los resultados
porcentajes_fum <- paste0(round(porcentajes_fum, 2), "%")

# Mostramos el resultado
porcentajes_fum
```

Estudia las mismas proporciones fum by sex por clase social

```
# Creamos una tabla de frecuencias cruzadas entre sexo, clase social y fums
tabla_fum_clas_soc <- table(df_iam$sex, df_iam$clas_soc, df_iam$fum)

# Calculamos la proporción de fumadores dentro de cada grupo de sexo y clase
proporciones_fum_clas_soc <- prop.table(tabla_fum_clas_soc, c(1, 2))

# Multiplicamos las proporciones por 100 para obtener los porcentajes
porcentajes_fum_clas_soc <- proporciones_fum_clas_soc[, , 2] * 100

# Multiplicamos las proporciones por 100 para obtener los porcentajes
porcentajes_fum_clas_soc <- proporciones_fum_clas_soc[, , 2] * 100

# Mostramos el resultado
porcentajes_fum_clas_soc

# Agregamos el símbolo % a los resultados
porcentajes_fum_clas_soc <- apply(porcentajes_fumadores_clas_soc, 1:2,
function(x) paste0(round((x, 2), "%")))
# Mostramos el resultado
porcentajes_fum_clas_soc
```

El ejercicio anterior tiene algunas complicaciones. Al pedir una tabla de 3 dimensiones, estamos construyendo un “array”. En este caso, tenemos una tabla de tres dimensiones `tabla_fum_clas_soc`, generada por `table(df_iamsex, df_iamclas_soc, df_iam$fum)` tiene las siguientes: dimensiones: la primera dimensión es el sexo (`df_iam$sex`), la segunda dimensión es la clase social (`df_iam$clas_soc`) y la La tercera dimensión es si la persona fuma o no (`df_iam$fum`).

Cuando aparece `c(1, 2)` en la función `prop.table`, está señalando sobre qué dimensiones se calculan las proporciones., 1 hace referencia a la primera dimensión, que es el sexo., 2 hace

referencia a la segunda dimensión, que es la clase social. Al usar `c(1, 2)` en `prop.table`, le estamos diciendo a R que calcule las proporciones dentro de cada combinación de sexo y clase social.

El código `[, , 2]` se lee así: el primer `,` (vacío) indica que tomamos todos los valores en la primera dimensión (sexo); el segundo `,` (vacío) indica que tomamos todos los valores en la segunda dimensión (clase social); y el `2` se refiere a la segunda capa de la tercera dimensión, que es donde se encuentran los que fuman (ya que `df_iam$fum` tiene dos valores: `0 = no` y `1 = sí`). En resumen, `[, , 2]` selecciona los datos correspondientes a las personas que fuman (`fum = 1`), manteniendo las dimensiones de sexo y clase social.

Ahora vas a calcular los cuartiles de la variable `imc`, que has creado previamente.

```
cuart_imc <- quantile(df_iam$imc, probs = seq(0, 1, by = 0.25), na.rm = TRUE)
cuart_imc
```

Ahora describe la distribución de la variable `imc`. Sin poder asegurarlo, ¿dirías que se distribuye como una normal?

```
media_imc <- mean(df_iam$imc, na.rm = TRUE)
varianza_imc <- var(df_iam$imc, na.rm=TRUE)
desv_imc <- sd(df_iam$imc, na.rm=TRUE)
mediana_imc <- median(df_iam$imc, na.rm = TRUE)

media_imc
varianza_imc
desv_imc
mediana_imc
```

Ahora calcula dichas medidas de `imc` en varones y en mujeres usando la función `tapply()`

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

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

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

# Calcular la mediana de imc por grupo de sexo
mediana_imc <- tapply(df_iam$imc, df_iam$sex, median, na.rm = TRUE)
```

```
# Mostrar resultados
media_imc
varianza_imc
desviacion_imc
mediana_imc
```

Finalmente deberás calcular asimetría y curtosis de la variable `imc`. Recuerda descargar e instalar el paquete `e1071`. Si no se descarga puede que sea porque no has elegido un repositorio (mira la primera fila del ejecutable). ¿Qué dirías sobre la forma de la distribución de dicha variable?

```
options(repos = c(CRAN = "https://cloud.r-project.org/"))
install.packages("e1071")
library(e1071)
asimetria_imc <- skewness(df_iam$imc, na.rm = TRUE)
curtosis_imc <- kurtosis(df_iam$imc, na.rm = TRUE)

asimetria_imc
curtosis_imc

#La distribución es (muy) levemente asimétrica a la izquierda y platicúrtica
```