

Práctica 10. Prueba repaso.

Contenidos

Introducción	1
Ejercicio 1	1
Ejercicio 2	2
Ejercicio 3	2
Ejercicio 4	3
Ejercicio 5	4
Ejercicio 6	5
Ejercicio 7	6
Ejercicio 8	8
Ejercicio 9	10

Introducción

En esta práctica vamos a realizar un simulacro en el que estudiemos cómo importar una base de datos, cómo hacer descriptivos básicos y cómo calcular intervalos de confianza de una media y de una proporción. Se articulará en torno a una serie de preguntas y, finalmente habrá que compilar un documento html. El presente documento es la solución a la práctica, que habrá que construir en Rmarkdown

Ejercicio 1

La primera tarea consiste en importar la base de datos `df_iam2.csv`. Es importante saber en qué directorio estamos trabajando

```
getwd ()
```

```
[1] "C:/Users/jesus.martin/OneDrive - Madrid Digital/Docente/Introducción a la AP/Curso 2024."
```

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

```
head (df_prueba)
```

	fech_nac	sex	alt	peso	imc	hta	fum	DM	colesterol	clas_soc	iam1
1	1982-08-06	Mujer	152.0	54.0	23.37258	No	No	0	209	Media	No
2	1982-09-23	Varón	175.8	88.8	28.73262	No	No	0	185	Baja	No
3	1939-03-04	Mujer	155.8	66.8	27.51956	Sí	No	0	184	Media	No
4	1936-01-15	Varón	183.3	85.3	25.38774	Sí	No	0	162	Baja	No
5	1940-03-23	Varón	168.1	61.1	21.62249	No	No	0	140	Alta	No
6	1962-11-10	Mujer	152.8	49.8	21.32960	No	Sí	0	210	Media	Sí

Ejercicio 2

Debes crear la variable edad a partir de la fecha de nacimiento, sabiendo que la fecha de fin de estudio fue el 31/12/2023.

Además debes recodificar la variable imc en tres categorías: Bajo si $imc < 20$, normal si imc entre 20 y 30 y alto si $imc > 30$. Llama a la nueva variable `imc_r`

```
#fech_nac viene definida como character, cambiamos a formato fecha
df_prueba$fech_nac <- as.Date(df_prueba$fech_nac)
fecha_fin <- as.Date("2023-12-31")
df_prueba$edad <- (fecha_fin-df_prueba$fech_nac)/365.25
df_prueba$edad <- as.numeric (round (df_prueba$edad,0))

#Categorizamos imc
df_prueba$imc_r <- cut(df_prueba$imc,
                      breaks = c(-Inf, 20, 30, Inf),
                      labels = c("Bajo", "Normal", "Obesidad"))
```

Ejercicio 3

Describe las proporciones de hombres y mujeres en cada una de las categorías de la variable `iam1`. Usa las funciones `table` y `prop.table`

```
#Primero obtenemos las frecuencias absolutas

sex_by_iam <- table(df_prueba$sex, df_prueba$iam1)
sex_by_iam
```

	No	Sí
Mujer	244	42
Varón	211	39

```
#Ahora las proporciones por fila
prop_sex_by_iam <- prop.table(sex_by_iam, margin = 1) * 100
prop_sex_by_iam
```

	No	Sí
Mujer	85.31469	14.68531
Varón	84.40000	15.60000

Ejercicio 4

Describe la variable edad, encuentra su media y su desviación estándar, así como el recorrido intercuartil. Da tu opinión sobre si se puede distribuir como una normal o no.

Puedes usar la función `describe` del paquete `psych`

Después describe la distribución de la variable `edad` en hombres y mujeres utilizando la función `by`

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

library(psych)
```

Warning: package 'psych' was built under R version 4.4.2

```
#Calcular estadísticas básicas para 'edad'
stats_edad <- describe (df_prueba$edad)

stats_edad
```

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
X1	1	536	62.65	17.02	63	62.5	22.24	35	93	58	0.05	-1.22	0.74

```
# Utilizando la función by() para calcular estadísticas descriptivas por sexo
stats_edad_sex <- by(df_prueba$edad, df_prueba$sex, describe)

stats_edad_sex
```

```
df_prueba$sex: Mujer
```

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
X1	1	286	62.71	16.57	62.5	62.62	20.02	35	93	58	0.04	-1.16	0.98

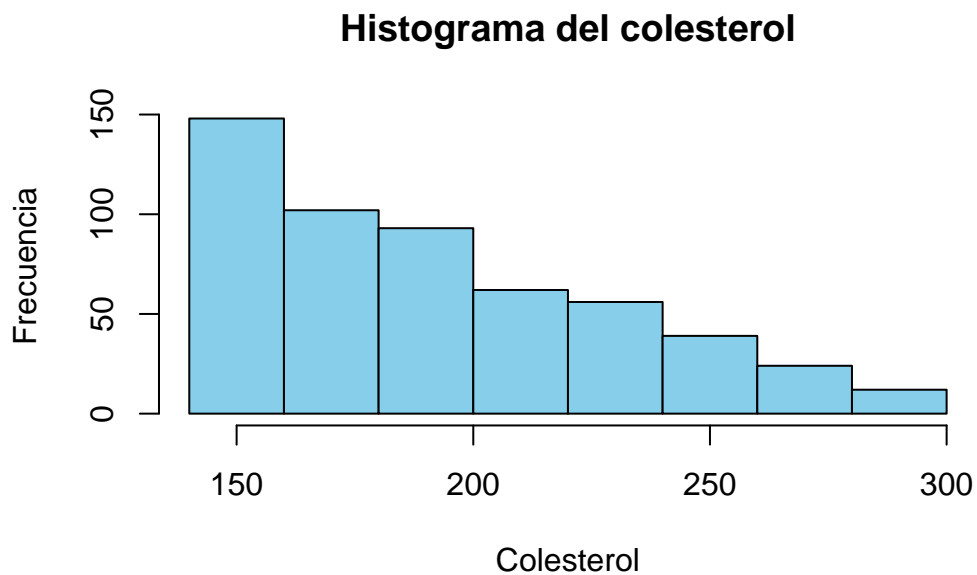
```
df_prueba$sex: Varón
```

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
X1	1	250	62.58	17.55	63	62.37	22.24	35	93	58	0.05	-1.29	1.11

Ejercicio 5

Haz un histograma con la variable `colesterol`. Después haz un boxplot de la distribución del colesterol en personas sin/con `iam`. Las funciones a utilizar son `hist` y `boxplot`. ¿Te parece que la distribución del colesterol en pacientes sin/con `iam` pueda asemejarse a una normal? ¿Hay diferencias esperables entre los niveles de colesterol en pacientes sin /con `iam`?

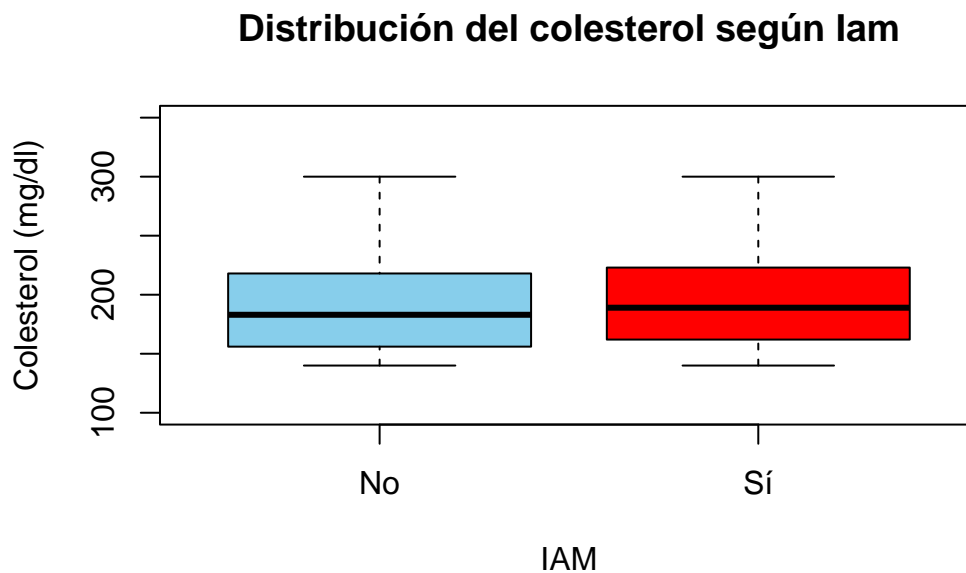
```
hist(df_prueba$colesterol,  
     breaks = 10,  
     main = "Histograma del colesterol",  
     xlab = "Colesterol",  
     ylab = "Frecuencia",  
     col = "skyblue")
```



```

boxplot(df_prueba$col ~ df_prueba$iam1,
main = "Distribución del colesterol según iam",
xlab = "IAM",
ylab = "Colesterol (mg/dl)",
col = c("skyblue", "red"),
border = c("black", "black"),
ylim = c(100, 350)
)

```



La distribución del colesterol es bastante asimétrica en ambos casos, no parece que se asemeje a una normal y no hay claras diferencias entre los pacientes con/sin IAM

Ejercicio 6

Debes calcular la media y el IC 95% del `imc` en hombres y en mujeres. Usa la función `group.CI` del paquete `Hmisc`.

Explica qué significa ese IC del 99%

```

#install.packages("Rmisc")

library(Rmisc)

```

Warning: package 'Rmisc' was built under R version 4.4.2

Cargando paquete requerido: lattice

Cargando paquete requerido: plyr

Warning: package 'plyr' was built under R version 4.4.1

```
group.CI(imc ~ sex, data = df_prueba, ci = 0.95)
```

	sex	imc.upper	imc.mean	imc.lower
1	Mujer	26.57349	26.09270	25.61191
2	Varón	27.41641	26.98145	26.54650

Si, en este caso, calculas un IC del 95% para la media del IMC en mujeres, el intervalo (25,61-26,09) te especificará el rango de valores entre los que encontrarás el estimador de dicha media en el 95% de las ocasiones si repitieses el proceso muchas veces, con muestras obtenidas por el mismo procedimiento.

Ejercicio 7

Representa gráficamente las estimaciones y los IC del 95% del ejercicio anterior. Usa la función `plotmeans` del paquete `gplots`

```
#install.packages("gplots")  
library(gplots)
```

Warning: package 'gplots' was built under R version 4.4.2

Adjuntando el paquete: 'gplots'

The following object is masked from 'package:stats':

lowess

```

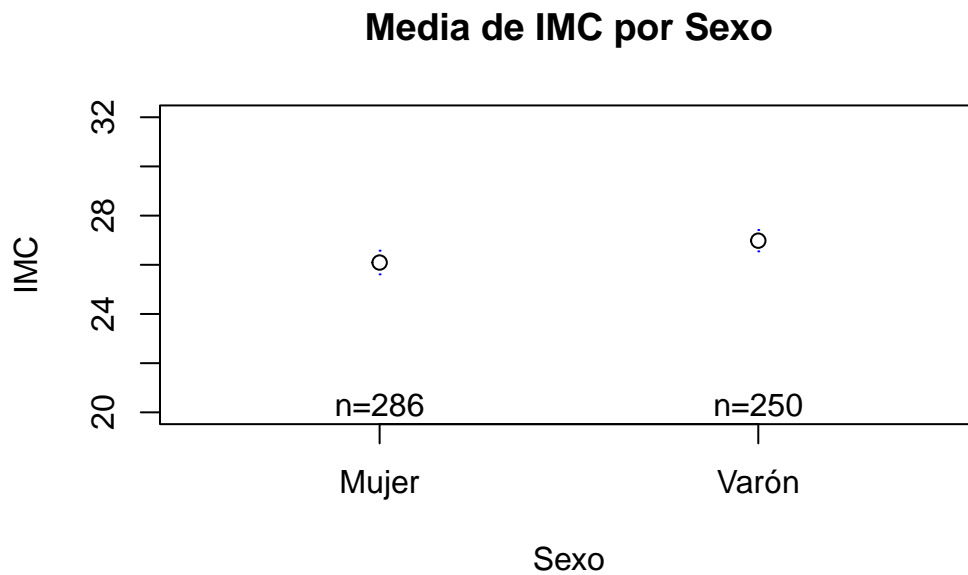
plotmeans(
  imc ~ sex,          # Fórmula
  data = df_prueba,   # Datos
  ci.label = FALSE,   # No mostrar los IC
  n.label = TRUE,     # Mostrar n de la muestra
  connect = FALSE,    # No conectar las medias
  mean.label = FALSE, # No mostrar valores de media
  barcol = "blue",    # Color de las barras
  xlab = "Sexo",      # Etiqueta del eje x
  ylab = "IMC",       # Etiqueta del eje y
  main = "Media de IMC por Sexo", # Título
  ylim = c(20, 32),   # Ajustar los límites de y
)

```

Warning in arrows(x, li, x, pmax(y - gap, li), col = barcol, lwd = lwd, :
zero-length arrow is of indeterminate angle and so skipped

Warning in arrows(x, li, x, pmax(y - gap, li), col = barcol, lwd = lwd, :
zero-length arrow is of indeterminate angle and so skipped

Warning in arrows(x, ui, x, pmin(y + gap, ui), col = barcol, lwd = lwd, :
zero-length arrow is of indeterminate angle and so skipped
Warning in arrows(x, ui, x, pmin(y + gap, ui), col = barcol, lwd = lwd, :
zero-length arrow is of indeterminate angle and so skipped



Ejercicio 8

Calcula la proporción de personas que sufren IAM con los métodos de Wilson y Clopper-Pearson (exacto). Instala el paquete `Hmisc`, y entonces puedes usar su función `binconf`

```
#install.packages("Hmisc")  
  
library(Hmisc)
```

Warning: package 'Hmisc' was built under R version 4.4.1

Adjuntando el paquete: 'Hmisc'

The following objects are masked from 'package:plyr':

`is.discrete`, `summarize`

The following object is masked from 'package:psych':

`describe`

The following objects are masked from 'package:base':

format.pval, units

```
# Contar las observaciones de "iam" y "No iam"
iam_c <- table(df_prueba$iam1)

# Extraer las proporciones
n_si <- iam_c["Sí"] # número de "Sí"
n_no <- iam_c["No"] # número de "No"
n_total <- sum(iam_c) # total

# Calcular los intervalos de confianza para "Sí" usando el método de Wilson
ci_si <- binconf(n_si, n_total, method = "wilson")

# Calcular los intervalos de confianza para "No" usando el método de Wilson
ci_no <- binconf(n_no, n_total, method = "wilson")

# Calcular los intervalos de confianza para "Sí" usando el método exacto
ci_e_si <- binconf(n_si, n_total, method = "exact")

# Calcular los intervalos de confianza para "No" usando el método exacto
ci_e_no <- binconf(n_no, n_total, method = "exact")

# Mostrar resultados
ci_si
```

PointEst	Lower	Upper
0.1511194	0.1232869	0.1839171

ci_no

PointEst	Lower	Upper
0.8488806	0.8160829	0.8767131

```
ci_e_si
```

PointEst	Lower	Upper
0.1511194	0.121858	0.184289

```
ci_e_no
```

PointEst	Lower	Upper
0.8488806	0.815711	0.878142

Ejercicio 9

Ahora representa los IC anteriores con la función `plot`

```
# Calcular el intervalo de confianza para cada categoría
resultado_si <- prop.test(n_si, n_total, correct = FALSE)
resultado_no <- prop.test(n_no, n_total, correct = FALSE)

# Graficar las proporciones con IC
# Ajustar los márgenes del gráfico
par(mar = c(5, 5, 2, 5))
# Aumentar el margen inferior (5) y lateral izquierdo (5)
plot(c(resultado_si$estimate, resultado_no$estimate),
     xaxt = "n",          # Evitar etiquetas automáticas en x
     ylim = c(0.05, 0.95), # Ajustar el límite de y
     pch = 19,            # Tipo de punto
     col = "darkblue",    # Color de los puntos
     xlab = "IAM",        # Etiqueta del eje x
     ylab = "Proporción", # Etiqueta del eje y
     main = "Proporciones de IAM", # Título
     cex = 1,            # Tamaño de los puntos
     xlim = c(0.5, 2.5)  # Ajustar límite de eje x
)

# Añadir los nombres de las categorías en el eje x
axis(1, at = 1:2, labels = c("Sí", "No"))

# Añadir los intervalos de confianza para "Sí"
arrows(
  x0 = 1,
```

```

y0 = resultado_si$conf.int[1],
y1 = resultado_si$conf.int[2],
angle = 90,
code = 3,
length = 0.1,
col = "blue"
)

# Añadir los intervalos de confianza para "No"
arrows(
  x0 = 2,
  y0 = resultado_no$conf.int[1],
  y1 = resultado_no$conf.int[2],
  angle = 90,
  code = 3,
  length = 0.1,
  col = "blue"
)

```

