Análisis de datos: Examen de Diseño Experimental

Arenas Tamara, Medina Nuria, Noriega Berenice, Picasso David, Ruiz Braulio, Vázquez Mariana

2022-11-17

Cargando las librerias a usar

R tiene un amplio catálogo de librerias que pueden extender las funciones a emplear en el ambiente de trabajo. La forma básica de usarlas es instalando el paquete externo ejecutando una vez el comando `install.packages()` y luego llamando el paquete al ambiente de trabajo cada vez que sea necesario empleando al incio de un script la función `library()`. Sin embargo, el paquete `pacman` con su función `p_load()` permite en un solo comando instalar las librerías que no se tenga y cargarlas al ambiente de trabajo.

Esta es la forma clásica:

```
install.packages("tidyverse")
library(tidyverse)
```

Vamos a optar por este uso:

```
if (!require("pacman")) install.packages("pacman")
pacman::p_load(tidyverse, nortest, ggplot2,car)
```

Cargando datos en RStudio

Se pueden usar funciones que inician con "read" para cargar datos, por ejemplo `read.csv()` o la función `read_xlsx()` del paquete `readxl`.

Con estas funciones R suele cargar el objeto como tipo de dato "data.frame", que es un formato donde es posible combinar en un solo objeto datos de otros tipos (character, integer, numeric, logical, factor). Se puede comprobar la estructura usando la función `str()`.

```
BD <- read.csv(file = "../Data/data.csv")
str(BD)
```

```
## 'data.frame':    24 obs. of  9 variables:
##  $ ID            : chr  "E_AM_S1_M" "E_AM_S2_NA" "E_AM_S3_H" "E_AM_S4_H" ...
##  $ Sexo_Aplicador: chr  "M" "M" "M" "M" ...
##  $ Sexo          : chr  "M" NA "H" "H" ...
##  $ Chiste_1      : int  0 NA 0 0 NA NA NA 1 0 0 ...
##  $ Chiste_2      : int  1 NA 0 0 NA NA NA 1 0 0 ...
##  $ Chiste_3      : int  0 NA 1 0 NA NA NA 1 0 0 ...
##  $ Chiste_4      : int  1 NA 1 1 NA NA NA 1 1 1 ...
##  $ Chiste_5      : int  0 NA 1 0 NA NA NA 1 0 0 ...
##  $ Veces_Risa    : int  2 NA 3 1 NA NA NA 5 1 1 ...
```

Eliminamos participantes con datos faltantes:

```
BD <- na.omit(BD)
```

Obteniendo descriptivos

Para comenzar a explorar los datos de una variable es conveniente calcular las medidas de estadística descriptiva. Se puede hacer esto empleando la función `group_by()` para resumir los datos por grupo usando también `summarise()`

Por ejemplo, para el conjunto de datos cargado observemos los descriptivos para la variable de edad:

```
BD %>% group_by(Sexo_Aplicador) %>% summarise(
  media= mean(Veces_Risa),
  mediana= median(Veces_Risa),
  ds= sd(Veces_Risa),
  varianza= var(Veces_Risa),
  minimo= min(Veces_Risa),
  maximo= max(Veces_Risa),
  muestra= n(),
  error_estandar=ds/sqrt(muestra),
  i_confianza_low= media-2*error_estandar,
  i_confianza_up= media+2*error_estandar
)
```

```
## # A tibble: 2 x 11
##   Sexo_Aplic~1 media mediana    ds varia~2 minimo maximo muestra error~3 i_con~4
##   <chr>        <dbl>   <dbl> <dbl>   <dbl>  <int>  <int>   <int>   <dbl>   <dbl>
## 1 H             2.75       3  1.16    1.36      1      5       8   0.412   1.93
## 2 M             1.75       1  1.58    2.5       0      5       8   0.559   0.632
## # ... with 1 more variable: i_confianza_up <dbl>, and abbreviated variable
## #   names 1: Sexo_Aplicador, 2: varianza, 3: error_estandar, 4: i_confianza_low
```

Generamos una variable que identifique a cada participante (ID)

```
BD<- BD %>% mutate(ID_Card=1:16)
```

Para futuros ejercicios guardamos la base generada:

```
write.csv(BD, file = "../Output/BD.csv")
```

Normalidad y homocedasticidad de los datos

Empleamos el test de Shapiro-Wilk ya que el tamaño de la muestra de cada grupo es menor a 50.

```
# Muestra menor a 50 participantes
shapiro.test(BD$Veces_Risa[BD$Sexo_Aplicador=="M"])
```

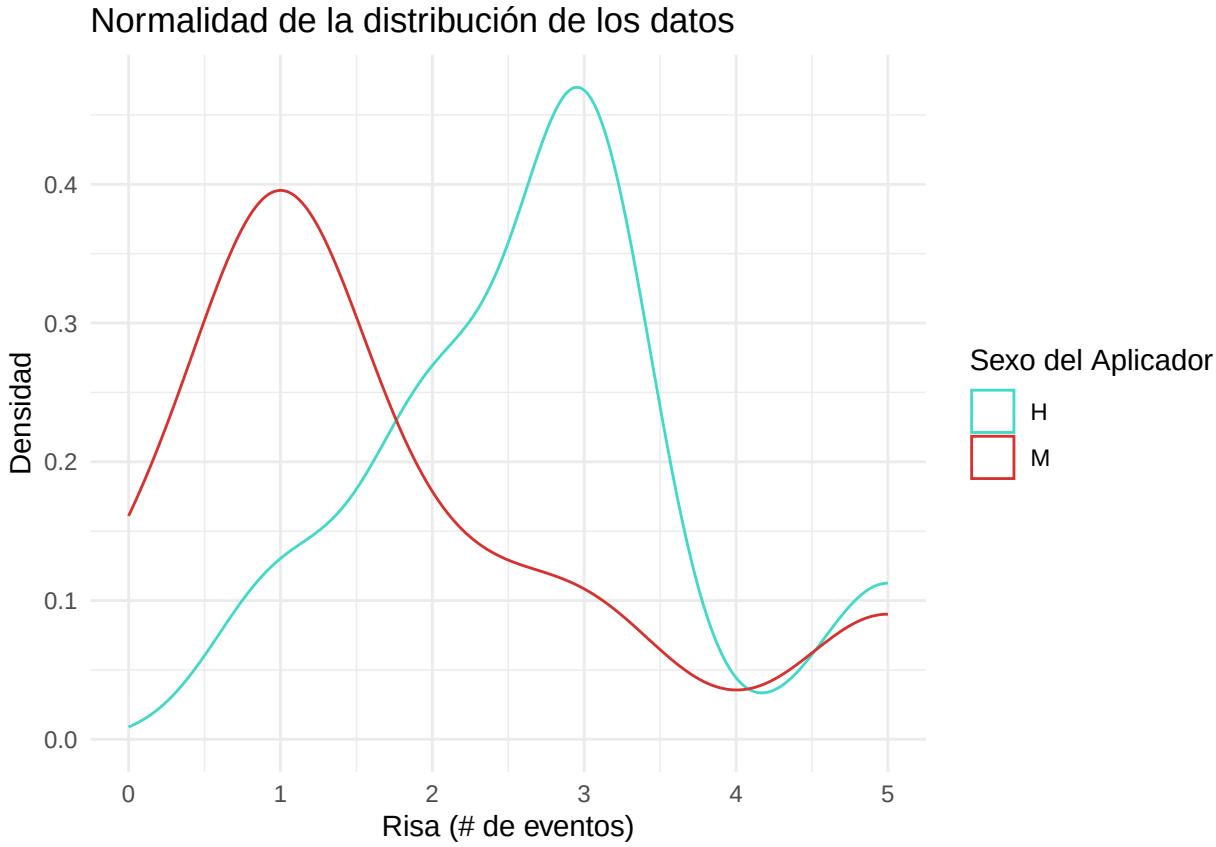
```
##
##  Shapiro-Wilk normality test
##
## data:  BD$Veces_Risa[BD$Sexo_Aplicador == "M"]
## W = 0.83955, p-value = 0.07452
```

```
shapiro.test(BD$Veces_Risa[BD$Sexo_Aplicador=="H"])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  BD$Veces_Risa[BD$Sexo_Aplicador == "H"]
## W = 0.89239, p-value = 0.2463
```

Tanto para el grupo A (expositor mujer) como el grupo B (expositor hombre), se obtiene un p-value mayor de 0.05, por lo cual se concluye que los datos de edad cumplen el supuesto de normalidad. Se puede comprobar visualmente que hay una distribución similar a la de la curva normal:

```
ggplot(data= BD, aes(x=Veces_Risa, color=Sexo_Aplicador)) +
  geom_density() +
  scale_color_manual(values = c("#42D9C8","#D63230"))+
       theme_minimal()+
      labs(title = 'Normalidad de la distribución de los datos',
       color = 'Sexo del Aplicador',
       x = 'Risa (# de eventos)',
       y='Densidad')
```



Dado que los datos cumplen el supuesto de normalidad, se pone a prueba el supuesto de homocedasticidad con la prueba de Bartlett

```
bartlett.test(BD$Veces_Risa ~ BD$Sexo_Aplicador)
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  BD$Veces_Risa by BD$Sexo_Aplicador
## Bartlett's K-squared = 0.60033, df = 1, p-value = 0.4385
```

Como el p-value es un valor mayor de 0.05, aceptamos la hipótesis nula (H0). Esto nos indica que nuestras dos muestras presentan varianzas iguales. Es decir: no se encuentran diferencias significativas entre las varianzas de los dos grupos.

Se observa este hallazgo visualmente:

```r
ggplot(data = BD, aes(x = Sexo_Aplicador, y = Veces_Risa)) +
  geom_jitter(aes(color = Sexo_Aplicador), size = 1, alpha = 0.5) +
  geom_boxplot(aes(color = Sexo_Aplicador), alpha = 0.5) +
  labs(
    title = 'Homocedasticidad de la distribución de datos',
       x = 'Sexo del Aplicador',
       y = 'Risa (# de eventos)',
       caption="Bartlett, p=.43"
    ) +
  theme_bw() +
  theme(legend.position = "none") +
  theme_minimal()
```