Estadística descriptiva con R

Prof. Edlin Guerra Castro

10/8/23

Trabajando con R

El objetivo principal de este laboratorio es introducirlos a la exploración de datos y estadística descriptiva con R y RStudio, las herramientas computacionales que utilizaremos a lo largo del semestre para aprender a aplicar los conceptos más importantes de *Estadística Aplicada I*, pero en especial, para aprender a procesar y analizar datos reales.

El programa R, sus versiones actualizadas y todos los paquetes con funciones, asi como otra información relevante se encuentre en los repositorios de R conocidos como **CRAN** (Comprehensive R Archive Network). Los distintos servidores distribuidos en todo el mundo, conforman el CRAN y son conocidos como los **CRAN mirrors** (de espejo). Para descargar los paquetes requieres antes escoger un **CRAN mirror**, y la función que te permite escogerlo de una lista que aparece en la consola es `chooseCRANmirror`. Alternativamente, se pueden descargar paquetes desde otros repositorios, uno muy popular y que estaremos usando en esta asignatura es GitHub.

RStudio es un entorno de desarrollo integrado (IDE) para **R**. Incluye una consola, editor de comandos y líneas de programación que admite la ejecución directa de código, así como herramientas para graficar, documentar, registrar el historial de comandos ejecutados, acceder a archivos, y muchas cosas más desde la gestión de un espacio de trabajo. **RStudio** hace que el trabajar con **R** sea más poderoso, y a su vez simple. Para que tengan una idea, esta guía se escribió desde **RStudio**.

Las funciones están organizadas en paquetes. El paquete denominado `base` constituye el núcleo de **R** y contiene las funciones básicas del lenguaje. Otro paquete muy importante es `stats` e incluye las funciones estadísticas más importantes y básicas de **R**. Ambos ya vienen preinstalados en **R**. Existen muchos paquetes, a medida que se requiera el uso de alguno específico se irá indicando para que lo descarguen e instalen. Por ahora les adelanto el uso de un set de paquetes agrupados en una familia de paquetes muy usados para ordenar, limpiar, modelar, reproducir, comunicar y graficar datos; este grupo de paquetes se les denomina tidyverse. Para instalarlos pueden escribir en la consola:

```r
#Para instalar ggplot2 (realizar gráficos de alta calidad)
install.packages("ggplot2")

#Para depurar y reordenar bases de datos, instala: tidyr
install.packages("tidyr")

#Para administrar bases de datos: usa dplyr
install.packages("dplyr")

#Para importar datos desde Excel: readxl
install.packages("readxl")

#Para análisis en ecología de comunidades usa vegan
install.packages("vegan")

#para análisis de diversidad usa iNEXT
install.packages("iNEXT")

#para estimar simetría y curtosis
install.packages("moments")

#Para instalar todos los paquetes del Tidyverse, incluyendo ggplot2, tidyr, dplyr, etc:
install.packages("tidyverse")
```

Alternativamente, puedes usar del menú superior la opción *Tools/install packages...*, se desplegará una ventana para que escribas el nombre del paquete a instalar. Los paquetes se instalan una sola vez, siempre que estes en el mismo computador. Para usarlos debes incluirlos en tu sesión de trabajo cada vez que se inicia la sesión. Esto se logra con la función `library`:

```r
library("tidyverse")
```

Hagamos un análisis exploratorio a los datos de los pinguinos de la Antártida del género *Pygoscelis*. Lo primero que debe hacer es instalar el paquete de datos `palmerpenguins` y cárguelo en su sesión. Luego haga lo siguiente:

1. Busque en la pestaña *Help* qué es *palmerpenguin*.
2. Identifique la base de datos `penguins` y cárguela en su ambiente global con `data(penguins)`.
3. Identifique cuántas variables hay, cuál es la naturaleza de cada una de ellas (tipo de variable, escala), así como cuáles pueden ser consideradas causales y cuáles respuesta.
4. Efectue un gráfico de dispersión entre las variables `body_mass_g` y `flipper_lemgth_mm`. Mida el grado de asociación ¿cómo lo haría? Una forma de hacer estas cosas es graficando

la asociación, y estimando la correlación (método que se desarrollará en otro laboratorio, pero acá es pide con fines demostrativos)

```
plot(penguins$body_mass_g, penguins$flipper_length_mm)

#¿qué hace cor()?
cor(penguins$body_mass_g, penguins$flipper_length_mm)

#Si no obtuvo resultado, trate de resulverlo con el argumento "use"
```

Usemos el paquete `ggplot2` para mejorar el gráfico:

```
pp <- ggplot(data = penguins, aes(x = flipper_length_mm,
                                  y = body_mass_g,
                                  colour = species))+
      geom_point()

pp

#Mejoremos con capas
pp +  theme_bw()+
      xlab("Largo de aleta (mm)")+
      ylab("Masa corporal (g)")+
      scale_y_continuous(breaks = seq(2600,6400,400))+
      scale_x_continuous(breaks = seq(170,240,5))

#¿cuál gráfico le gustó más?
```

Llegados a este punto, vamos hacer algo de estadística descriptiva. Esta es la parte que deberán entregar como tarea. Calculen promedio, varianza, desviación estandar, valor mínimo y máximo, cuartiles, simetría y curtosis a la variable masa corporal. Usen para ello las funciones recomendadas y responda las siguientes preguntas:

```
# Para facilitar cálculos, vamos a remover los datos sin registro (identificados con NA),

penguins2 <- penguins %>%
  na.omit()
```

PREGUNTAS

1. Copia el comando que sigue. ¿Qué se calculó?

```
xx<-penguins2$body_mass_g
sum(xx, na.rm = T)/length(xx)
```

2. Busca y aplica una función que ejecute la linea de comando anterior. PISTA: escribe `?mean` en la cónsola y enter para ampliar tu búsqueda.
3. Calcula la mediana de la masa corporal usando la función correspondiente.
4. Calcula la varianza y desviación estándar de la masa corporal usando el comando `var()`
5. ¿Cuál es la diferencia entre estas dos fórmulas? ¿Representan lo mismo?

sum((xx-mean(xx))^2)/length(xx)

sum((xx-mean(xx))^2)/(length(xx)-1)

6. Con base en el valor de la varianza y usando operadores aritméticos, calcula la desviación estándar de la masa corporal. Confirma tu resultado usando la función `sd()`.
7. Explore el rango de la masa corporal identificando mínimos y máximos con la función `min()` y `max()`, respectivamente.
8. Ahora estime los cuartiles de la masa corporal con la función `quantile()`
9. Describa la forma de la distribución de la masa corporal usando la simetría y curtosis con las funciones `skewness()` y `kurtosis()`.
10. Todas estas estimaciones ignoran las posibles diferencias en la masa corporal entre las especies. ¿Qué le dice este gráfico?

```
boxplot(body_mass_g~species, data = penguins2)
```

11. Calcule estos estimadores para cada especie usando el paquete **dplyr** y sus funciones `group_by()` y `summarize()`. Estas líneas de comando lo ayudarán (interprete los resultados):

```
library(dplyr)

penguins %>%
  group_by(species) %>%
  summarise(media = mean(body_mass_g, na.rm =T),
            desviacion = sd(body_mass_g, na.rm =T),
            simetria = skewness(body_mass_g, na.rm =T),
            curtosis = kurtosis(body_mass_g, na.rm =T))
```

12. Usando como guía el libro digital R Graphics Cookbook, genere: (i) una distribución de frecuencias con histograma, (ii) una distribución de frecuencias basada en densidad, (iii) un diagrama de cajas que incluya promedio. En los tres casos la masa corporal debe distinguirse por especie.

Recuerde cargar las respuestas a estas preguntas en el formato de Google Doc generado en el Google Classroom.