



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
FACULTAD DE MATEMÁTICAS
DEPARTAMENTO DE ESTADÍSTICA

Actividad Práctica II

Rubén Soza
Computación Estadística con R

05 de Septiembre de 2019

Instrucciones

Cargue la base de datos `PlayStore.xlsx`, junto con `tidyverse` y las librerías necesarias para utilizar `Shiny`. Las soluciones del Problema 1 y 2 debe realizarlas en un `Rmarkdown`, mientras que para el problema 3 debe entregar un script de R que genere la aplicación Shiny deseada.

```
library(tidyverse)
library(shiny)
library(readxl)
library(kableExtra)
PlayStore <- read_excel(
  "G:/Mi unidad/Universidad/Cursos/Computación Estadística con R/Datasets/PlayStore.xlsx"
```

Problema 1

- 1) Utilizando la ayuda de R, estudie las funciones `as.numeric()` y `str_remove_all()`.
- 2) Combine las dos funciones anteriores para transformar la columna `Descargas` en una variable numérica en R.
- 3) Usted sabe que las `Ganancias` de una aplicación son calculadas como el número de `Descargas` de dicha aplicación multiplicada por el precio de la misma. Seleccionando solo las variables que son de pago, calcule el promedio de `Ganancias` por `Categoría`.
- 4) Reporte sus resultados en una tabla y en una oración reporte los valores mínimos y máximos obtenidos junto con sus respectivas aplicaciones.

Solución

- 1) Basta correr el siguiente código:

```
?as.numeric  
?str_remove_all
```

- 2) Utilizaremos `str_remove_all()` para remover la “,” y el " aprox." de la columna Descargas. Luego, la función `as.numeric()` transformara el string resultante en un número. Esto se puede ver a continuación:

```
PlayStore %>%  
  mutate(Descargas = str_remove_all(Descargas, "[,aprox.]")) %>%  
  mutate(Descargas = as.numeric(Descargas)) -> PlayStore
```

- 3) El siguiente código realiza lo pedido:

```
PlayStore %>%  
  filter(Tipo == "Pago") %>%  
  mutate(Ganancias = Descargas*Precio) %>%  
  group_by(Categoría) %>%  
  summarise("Ganancia_Promedio" = mean(Ganancias)) -> Tabla_Ganancias
```

- 4) La tabla con los resultados totales se muestra a continuación.

```
Tabla_Ganancias %>%  
  arrange(desc(Ganancia_Promedio)) %>%  
  kable(align = c("l","c")) %>%  
  kable_styling(position = "center", full_width = F)
```

Categoría	Ganancia_Promedio
ESTILO DE VIDA	3339678.916
FINANCIERAS	1482920.561
FOTOGRAFÍA	1054382.511
EDUCACIÓN	734161.667
JUEGO	511261.201
ENTRETENIMIENTO	299000.000
COMIDA Y BEBIDA	249500.000
PADRES	249500.000
FAMILIA	219962.319
CLIMA	197480.833
DEPORTES	163986.907
NEGOCIO	157480.327
MÉDICO	134171.211
PRODUCTIVIDAD	133018.123
PERSONALIZACIÓN	121759.781
SALUD Y BELLEZA	60714.075
COMUNICACIÓN	35020.630
VIAJES	23260.440
HERRAMIENTAS	22909.153
COMPRAS	15074.500
ARTE Y DISEÑO	10613.333
MAPAS Y NAVEGACIÓN	10447.250
NOTICIAS	3222.500
LIBROS Y REFERENCIAS	2788.027
SOCIAL	1980.000
DATOS	1712.300
REPRODUCTORES DE VIDEO	990.000
VEHICULOS	324.250
EVENTOS	109.990
LIBRERIAS	99.000

Además, para obtener el mínimo y el máximo utilizamos los siguientes comandos:

```
Tabla_Ganancias %>%
  filter(Ganancia_Promedio == min(Ganancia_Promedio)) -> Minimo
Tabla_Ganancias %>%
  filter(Ganancia_Promedio == max(Ganancia_Promedio)) -> Maximo
```

Por ende, la ganancia mínima en promedio es 99 y viene de la categoría LIBRERIAS. Por otro lado, la ganancia en promedio máxima es 3339678.91578947 y se corresponde con la categoría ESTILO DE VIDA.

Problema 2

- 1) Genere un gráfico boxplot de la **Valoración** para cada **Categoría**. Agregue a su gráfico las siguientes líneas de código. Comente lo obtenido.

```
# boxplot +  
coord_flip() +  
theme(legend.position = "none")
```

- 2) Genere un gráfico de barras que ilustre la **Clasificación** de las aplicaciones. Considere en su solución el siguiente código y comente su utilidad.

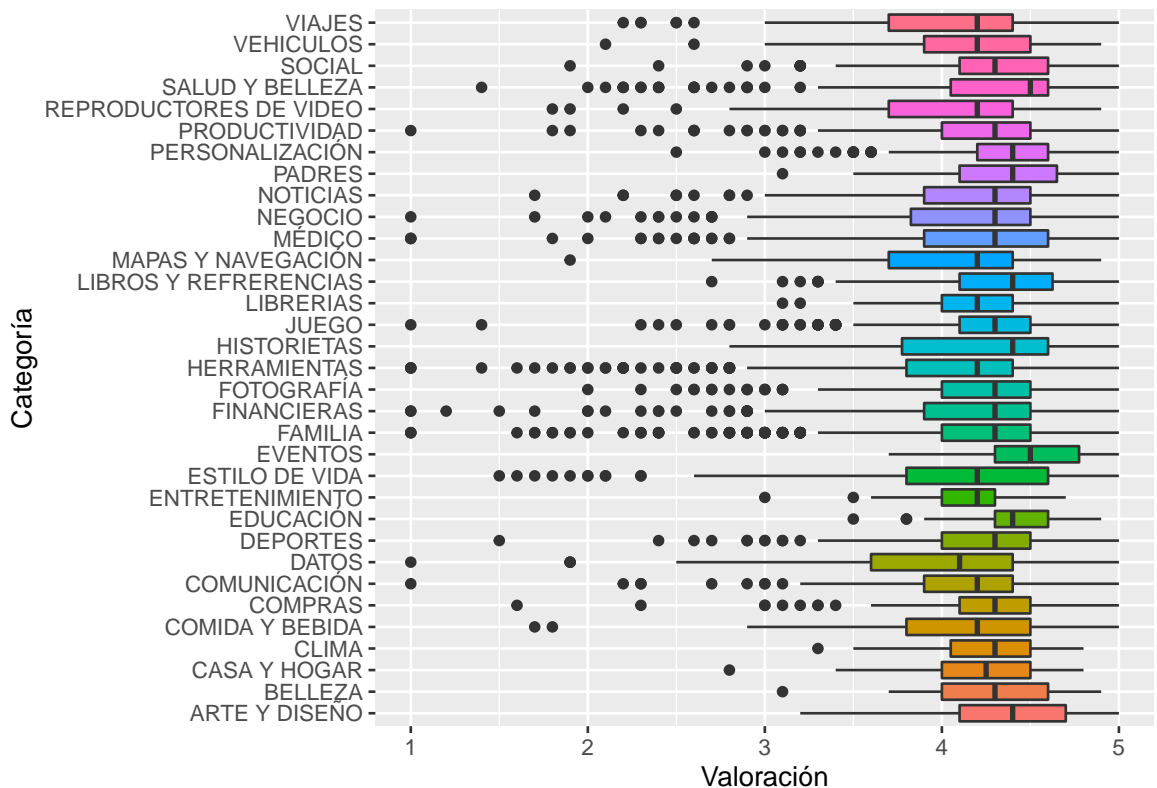
```
PlayStore %>%  
  mutate(Clasificación = factor(Clasificación,  
                                levels = c("Todo público", "Adolescentes",  
                                           "Mayores de 17 años", "Mayores de 10 años",  
                                           "Sólo adultos", "Sin clasificación")))
```

- 3) Genere un indicador de 3 categorías sobre la variable **Comentarios**. Sobre cada indicador, realice un gráfico de dispersión de la **Valoración** vs **Tamaño** que además aporte información del precio de la aplicación.

Solución

- 1) A continuación se presenta un código que realiza lo pedido:

```
ggplot(data = PlayStore) +  
  geom_boxplot(aes(x = Categoría, y = Valoración, fill = Categoría)) +  
  coord_flip() +  
  theme(legend.position = "none")
```



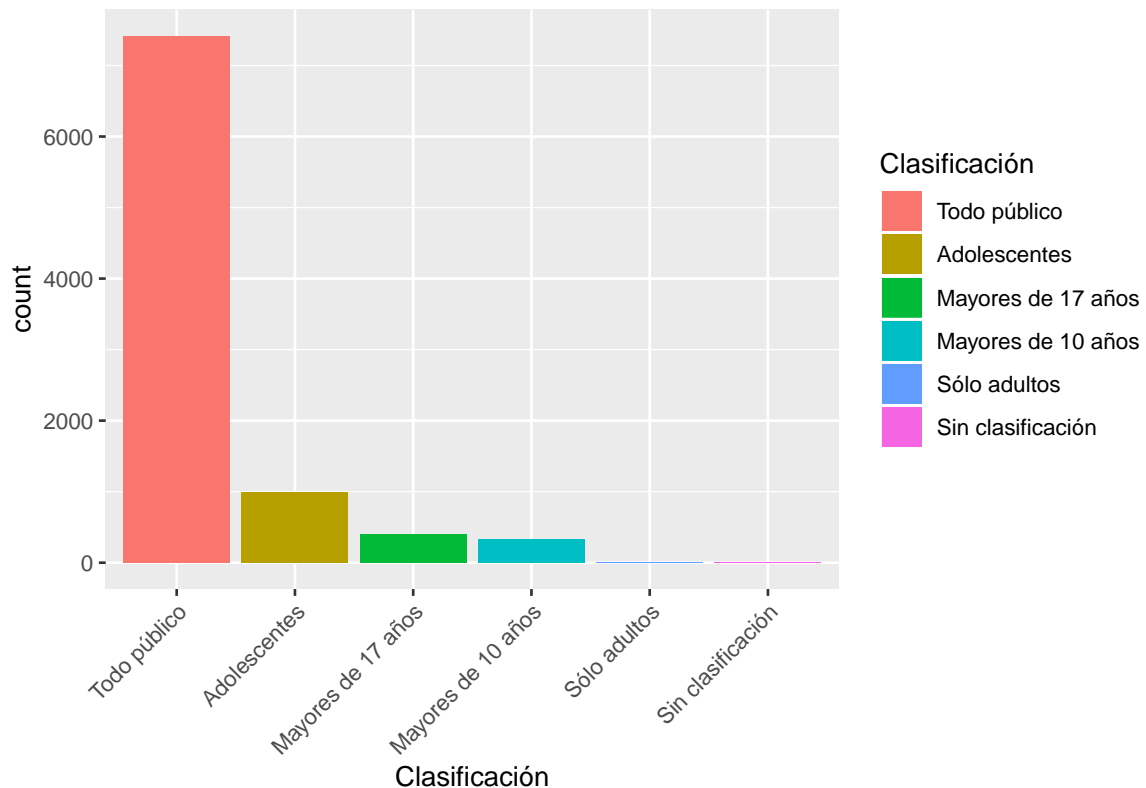
Los comandos sugeridos sirven para invertir el gráfico y borrar la leyenda. Esto último es necesario pues la información mostrada por ella es redundante con el gráfico.

- 2) En primer lugar, reescribimos nuestra base de datos utilizando el código sugerido:

```
PlayStore %>%
  mutate(Clasificación = factor(Clasificación,
                                levels = c("Todo público", "Adolescentes",
                                             "Mayores de 17 años", "Mayores de 10 años",
                                             "Sólo adultos", "Sin clasificación"))) -> PlayStore
```

Luego, generamos el gráfico de barras solicitado:

```
ggplot(data = PlayStore) +
  geom_bar(aes(x = Clasificación, fill = Clasificación)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



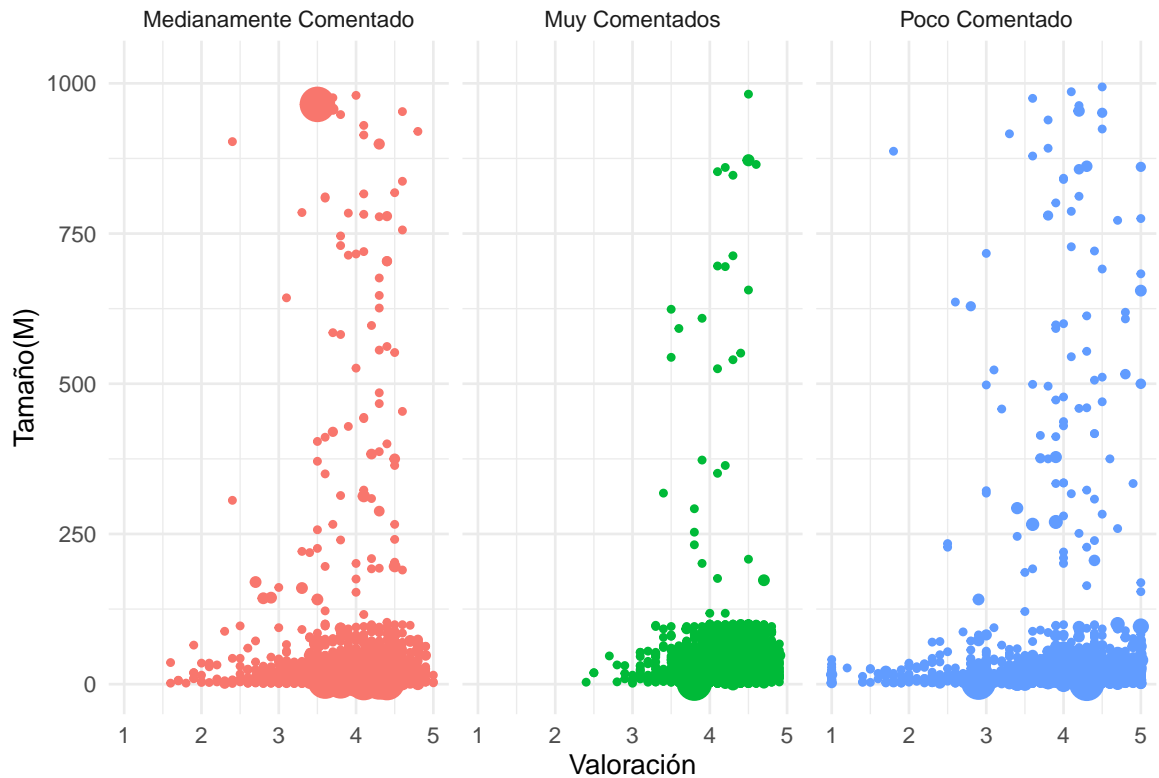
El código previamente mostrado permite ordenar de mayor a menor las categorías, para así mostrar un gráfico de barras ordenado. Esta forma de presentar el gráfico de barras es considerada la más útil a la hora de transmitir una historia determinada.

- 3) En primer lugar, generamos las 3 categorías utilizando la función `case_when()`:

```
PlayStore %>%
  mutate(Comentarios_cat = case_when(
    Comentarios <= 100 ~ "Poco Comentado",
    Comentarios > 100 & Comentarios <= 3500 ~ "Medianamente Comentado",
    TRUE ~ "Muy Comentados")) -> PlayStore
```

Ahora generamos gráficos de Valoración vs Tamaño:

```
ggplot(data = PlayStore) +
  geom_point(aes(x = Valoración, y = `Tamaño(M)`,
                 size = Precio, col = Comentarios_cat)) +
  theme_minimal() +
  theme(legend.position = "none") +
  facet_wrap(vars(Comentarios_cat))
```



Problema 3

Genere una aplicación en Shiny que cumpla las siguientes características:

- Permita al usuario seleccionar una **Categoría**, con la cual se reportará una tabla de la base de datos filtrada y un gráfico de dispersión de **Descargas** vs **Comentarios** considerando solo los elementos de la categoría seleccionada.
- Permita al usuario modificar el color y tamaño de los puntos.
- Permita al usuario modificar el título del gráfico.

Para lo anterior, puede ser útil el siguiente código:

```
library(plotly)
library(shiny)
ui <- fluidPage(
  sidebarLayout(
    sidebarPanel(
      selectInput(inputId = "categoría", label = "Seleccione la categoría",
        choices = c("ARTE Y DISEÑO", "BELLEZA", "CASA Y HOGAR",
          "CLIMA", "COMIDA Y BEBIDA", "COMPRAS",
          "COMUNICACIÓN", "DATOS", "DEPORTES",
          "EDUCACIÓN", "ENTRETENIMIENTO", "ESTILO DE VIDA",
          "EVENTOS", "FAMILIA", "FINANCIERAS",
```

```

        "FOTOGRAFÍA", "HERRAMIENTAS", "HISTORIETAS",
        "JUEGO", "LIBRERIAS", "LIBROS Y REFERENCIAS",
        "MAPAS Y NAVEGACIÓN", "MÉDICO", "NEGOCIO",
        "NOTICIAS", "PADRES", "PERSONALIZACIÓN",
        "PRODUCTIVIDAD", "REPRODUCTORES DE VIDEO",
        "SALUD Y BELLEZA", "SOCIAL",
        "VEHICULOS", "VIAJES", selected = "DEPORTES")
    ),
    textInput("_ _ _"),
    sliderInput("_ _ _"),
    "_ _ _"Input(inputId = "color", label = "Color de Puntos",
                  choices = c("red", "blue", "green"),
                  selected = "red")
  ),
  mainPanel( plotlyOutput("dispersion"), dataTableOutput("tabla"))
)
)

server <- function(input,output){

  # Cargar Base de Datos
  PlayStore <- read_excel("G:/Mi unidad/Universidad/Cursos/Computación Estadística con R")

  # Arreglar Descargas
  PlayStore %>%
    mutate(Descargas = str_remove_all(Descargas, "[,aprox.]")) %>%
    mutate(Descargas = as.numeric(Descargas)) -> PlayStore

  output$"_ _ _" <- renderDataTable({
    PlayStore %>%
      filter( Categoría == input$"_ _ _") -> PlayStore_filtrada
    "_ _ _"
  })

  output$dispersión <- renderPlotly({
    PlayStore %>%
      filter( Categoría == input$"_ _ _") -> PlayStore_filtrada
    g <- ggplot(data = PlayStore_filtrada) +
      geom_point(aes(x = Descargas, y = Comentarios),
                  size = input$"_ _ _", col = input$"_ _ _") +
      ggtitle(input$"_ _ _") +
      theme_light()
    ggplotly(g)
  })
}

```



```
}  
  
shinyApp(ui = ui,server = server)
```