

PROGRAMACION para CIENCIA DE DATOS

EXAMEN FINAL INTEGRADOR

```
# LUNA, Gustavo David
```

```
library(tidyverse)
```

```
library(RColorBrewer)
```

```
library(ggplot2)
```

```
➡ — Attaching core tidyverse packages — tidyverse 2.0.0 —
✓ dplyr      1.1.4      ✓ readr      2.1.5
✓ forcats    1.0.0      ✓ stringr    1.5.1
✓ lubridate   1.9.3      ✓ tibble     3.2.1
✓ purrr       1.0.2      ✓ tidyr      1.3.1
— Conflicts — tidyverse_conflicts() —
✗ purrr::%>%() masks base::%>%()
✗ dplyr::filter() masks stats::filter()
✗ dplyr::lag() masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
separador<- "=====
```

```
#####
## 1. Importe y lectura del csv.
#####
```

```
ruta<-"/content/celulares_PySpark.csv"
```

```
df<-read.csv(ruta)
df
```

```
➡ Mostrar el resultado oculto
```

```
#####
## 2. Cree funciones para ayudarle a optimizar su trabajo.
#####
```

```
valores_unicos <- function(df, columna) {
  # Extraer la columna
  columna_datos <- df[[columna]]

  # Obtener los valores únicos y contarlos
  conteo_valores <- table(columna_datos)

  # Mostrar el conteo de valores
  print(conteo_valores)
}

# Definir la función para cambiar el nombre de una columna
cambiar_nombre_columna <- function(dataframe, nombre_columna_antiguo, nombre_columna_nuevo) {
  if (nombre_columna_antiguo %in% names(dataframe)) {
    names(dataframe)[names(dataframe) == nombre_columna_antiguo] <- nombre_columna_nuevo
    return(dataframe)
  } else {
    print("El nombre de la columna antiguo no existe en el dataframe.")
    return(NULL)
  }
}
```

```
#####
## 3. Filtrar los teléfonos que tienen soporte para NFC.
#####
```

```
# Llamamos a la función para cambiar el nombre de la columna
df <- cambiar_nombre_columna(df, "X5G", "5G")

# Convertir la columna 'batería' a tipo string
df$bateria <- as.character(df$bateria)

# Filtrar los teléfonos
df_2 <- subset(df, NFC == "true")
#df_2
```

```
#####
## 4. Calcular el precio mínimo, promedio y máximo de los teléfonos con soporte para NFC.
#####
```

```
# Calcular el precio promedio de los teléfonos con soporte para NFC:
```

```
price_min<- min(df_2$precio)
price_min_round<-round(price_min,digits = 2)
avg_price_nfc <- mean(df_2$precio)
avg_price_nfc_round<-round(avg_price_nfc,digits = 2)
price_max<- max(df_2$precio)
price_max_round<-round(price_max,digits = 2)

paste(separador)
paste("El precio mínimo de los celulares con soporte para NFC es: ",price_min_round," USD")
paste("El precio promedio de los celulares con soporte para NFC es: ",avg_price_nfc_round," USD")
paste("El precio máximo de los celulares con soporte para NFC es: ",price_max_round," USD")
paste(separador)
```

```
➡ '====='
```

```
'El precio mínimo de los celulares con soporte para NFC es: 107.49 USD'
```

```
'El precio promedio de los celulares con soporte para NFC es: 582.79 USD'
```

```
'El precio máximo de los celulares con soporte para NFC es: 107.49 USD'
```

```
'====='
```

```
#####
## 5. Graficar el TOP 10 de celulares con mayor precio.
#####
```

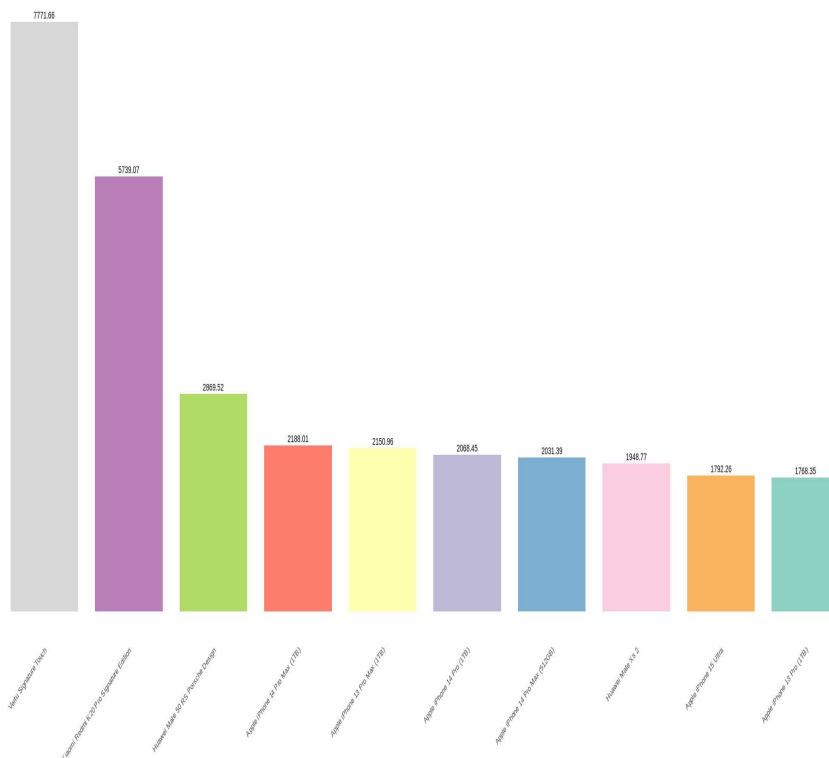
```
# Filtrar los celulares que tienen NFC igual a true
celulares_con_nfc <- df %>% filter(NFC == "true")

# Obtener el top 10 de celulares con mejor rating y mayor precio, teniendo en cuenta NFC
top_10_celulares_max_precio <- celulares_con_nfc %>%
  arrange(desc(precio)) %>%
  distinct(precio, .keep_all = "true") %>%
  head(10)
```

```
options(repr.plot.width = 18, repr.plot.height = 10)
# Crear el gráfico de barras con ggplot2
ggplot(top_10_celulares_max_precio, aes(x = reorder(modelo,-precio), y = precio,fill = modelo)) +
  geom_bar(stat = "identity",width = 0.8) +
  geom_text(aes(label = precio), vjust = -0.5,size=3) + # Añadir etiquetas de los valores
labs(title = "Top 10 celulares con mayor precio",
      x = "Modelo del Celular",
      y = "Precio") +
theme(
  panel.background = element_rect(fill = "white"), # Cambiar el fondo a blanco
  plot.background = element_rect(fill = "white"), # Cambiar el fondo del gráfico a blanco
  axis.text.x = element_text(angle = 45, hjust = 1,size=8),
  axis.title.y = element_blank(), # Eliminar el título del eje y
  axis.text.y = element_blank(), # Eliminar las etiquetas del eje y
  axis.ticks.y = element_blank(), # Eliminar las marcas del eje y
  axis.line.y = element_blank(), # Eliminar la línea del eje y
  axis.title.x = element_blank(), # Eliminar el título del eje x
  axis.ticks.x = element_blank(), # Eliminar las marcas del eje x
  axis.line.x = element_blank(), # Eliminar la línea del eje x
  legend.position = "none",
  plot.title = element_text(hjust = 0.5) # Centrar el título
) +
scale_fill_brewer(palette = "Set3") # Usar una paleta predefinida de RColorBrewer
```



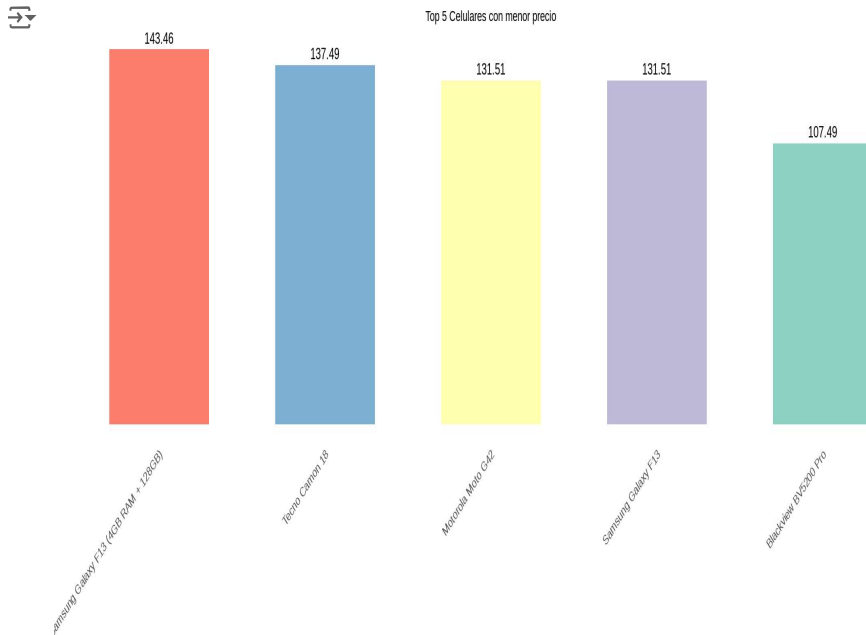
Top 10 celulares con mayor precio



```
#####
## 6. Graficar el TOP 5 de celulares con menor precio
#####
```

```
# Obtener el top 10 de celulares con menor precio, teniendo en cuenta NFC
top_10_celulares_min_precio <- df_2 %>%
  arrange(desc(precio)) %>%
  #distinct(precio, .keep_all = "true") %>%
  tail(5)
options(repr.plot.width = 18, repr.plot.height = 8)
# Crear el gráfico de barras con ggplot2 usando una paleta de RColorBrewer
ggplot(top_10_celulares_min_precio, aes(x = reorder(modelo, -precio), y = precio, fill = modelo)) +
  geom_bar(stat = "identity", width = 0.6) +
  geom_text(aes(label = precio), vjust = -0.5, size = 5) +
labs(title = "Top 5 Celulares con menor precio",
      x = "Modelo del Celular",
      y = "Precio") +
theme(
  panel.background = element_rect(fill = "white"), # Cambiar el fondo a blanco
  plot.background = element_rect(fill = "white"), # Cambiar el fondo del gráfico a blanco
  axis.text.x = element_text(angle = 45, hjust = 1, size = 12),
  axis.title.y = element_blank(),
  axis.text.y = element_blank(),
  axis.ticks.y = element_blank(),
  axis.line.y = element_blank(),
```

```
axis.title.x = element_blank(),
axis.ticks.x = element_blank(),
axis.line.x = element_blank(),
legend.position = "none",
plot.title = element_text(hjust = 0.5) # Centrar el título
) +
scale_fill_brewer(palette = "Set3") # Usar una paleta predefinida de RColorBrewer
```



```
#####
## 7. Graficar la distribución de los tipos de memorias.
#####
```

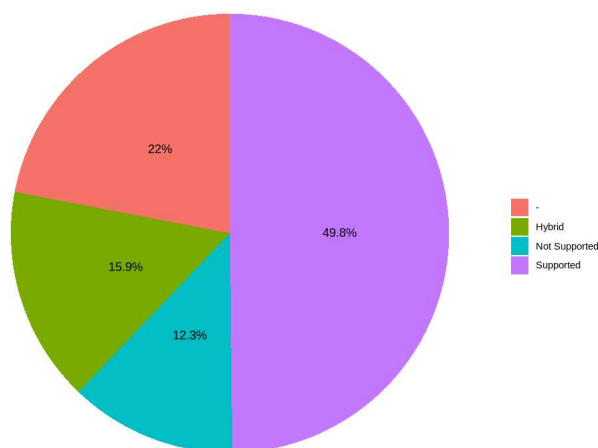
```
# Agrupar y contar los datos por la columna 'memoria'
memoria_counts <- df %>%
  group_by(memoria) %>%
  summarise(count = n()) %>%
  mutate(percentage = count / sum(count) * 100) # Calcular el porcentaje para cada categoría

options(repr.plot.width = 8, repr.plot.height = 8)

# Crear el gráfico de pie con ggplot2
ggplot(memoria_counts, aes(x = "", y = count, fill = memoria)) +
  geom_bar(stat = "identity", width = 1) +
  coord_polar(theta = "y") +
  geom_text(aes(label = paste0(round(percentage, 1), "%")),
            position = position_stack(vjust = 0.5)) + # Añadir etiquetas con el porcentaje
  labs(title = "Distribución de Tipos de Memoria",
        x = NULL,
        y = NULL) +
  theme_void() + # Quitar los ejes y el fondo del gráfico
  theme(legend.title = element_blank(), # Quitar el título de la leyenda
        plot.title = element_text(hjust = 0.5)) # Centrar el título del gráfico
```



Distribución de Tipos de Memoria



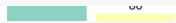
```
#####  
## 8. Graficar el TOP 5 de baterías más utilizadas.  
#####
```

```
# Calcular las frecuencias y filtrar el top 5  
top_5_baterias <- df %>%  
  count(bateria) %>%  
  arrange(desc(n)) %>%  
  top_n(5, n)  
  
# Filtrar el dataframe original para incluir solo las filas con los valores de batería en el top 5  
df_top_5 <- df %>%  
  filter(bateria %in% top_5_baterias$bateria)  
  
# Ordenar el factor de la columna 'batería' de mayor a menor  
df_top_5$bateria <- factor(df_top_5$bateria, levels = top_5_baterias$bateria[order(-top_5_baterias$n)])  
  
# Configuración del tamaño de la figura  
options(repr.plot.width = 7, repr.plot.height = 6)  
  
# Crear el gráfico de barras con ggplot2  
ggplot(df_top_5, aes(x = bateria, fill = bateria)) +  
  geom_bar() +  
  geom_text(stat = 'count', aes(label = ..count..), vjust = -0.5) +  
  labs(title = "Top 5 baterías más comunes", x = "batería", y = "") +  
  #theme_minimal(base_size = 15) + # Cambiar el tema a minimal y establecer tamaño base de fuente  
  theme(  
    panel.background = element_rect(fill = "white"), # Cambiar el fondo a blanco  
    plot.background = element_rect(fill = "white"), # Cambiar el fondo del gráfico a blanco  
    axis.text.x = element_text(angle = 0, hjust = 1, size = 10),  
    axis.title.y = element_blank(), # Eliminar el título del eje y  
    axis.text.y = element_blank(), # Eliminar las etiquetas del eje y  
    axis.ticks.y = element_blank(), # Eliminar las marcas del eje y  
    axis.line.y = element_blank(), # Eliminar la línea del eje y  
    axis.title.x = element_blank(), # Eliminar el título del eje x  
    axis.ticks.x = element_blank(), # Eliminar las marcas del eje x  
    axis.line.x = element_blank(), # Eliminar la línea del eje x  
    legend.position = "none", # Eliminar la leyenda  
    plot.title = element_text(hjust = 0.5) # Centrar el título  
  ) +  
  scale_fill_brewer(palette = "Set3") # Usar la paleta Set3 de RColorBrewer
```

Warning message:
"The dot-dot notation (`.count.`) was deprecated in ggplot2 3.4.0.
Please use `after_stat(count)` instead."



```
#####  
## 9. Realiza un análisis descriptivo de la columna 'precio' y compara los resultados con los del punto 4.  
#####
```



```
# Descriptivos específicos para una columna:  
summary(df_2$precio)  
  
paste("Los resultados coinciden con los del punto 4.")
```

Min. 1st Qu. Median Mean 3rd Qu. Max.
107.5 298.8 418.5 582.8 657.6 7771.7
'Los resultados coinciden con los del punto 4.'

```
#####  
##### 10. Muestra la frecuencia de la columna 'sim'.#####  
#####
```