

Patrones de las aplicaciones exitosas en Google Play Store

Enrique Esis

15 de junio de 2024

Table of Contents

Definición del problema/objetivo de investigación.....	3
Objetivos Específicos	3
Introducción	3
Descripción de las variables de interes:.....	3
Funciones a programar:	4
Preprocesamiento de Datos.....	4
Carga de paquetes.....	4
Carga de datos	4
Identificar los tipos de variables.....	5
Selección y eliminación de variables	9
Eliminar filas con valores nulos y NA.....	9
Verificar los cambios efectuados	9
Análisis Exploratorio de Datos	11
Graficos variables cuantitativas (BOXPLOT).....	11
Graficos variables cualitativas (BARPLOT y PIECHART)	14
Medidas estadísticas de variables cuantitativas	17
Matriz de correlación de variables cuantitativas.....	18
Gráfico de dispersión entre variables cuantitativas	18
Gráficos multivariantes	22
Modelado de Datos.....	26
Modelo de Random Forest.....	26
Interpretación de Resultados.....	30
Distribuciones y Estadísticas Descriptivas:	30
Correlaciones entre Variables:.....	30
Análisis de Variables Categóricas:	30
Modelado Predictivo.....	31
Modelo de Random Forest:.....	31

Conclusión	31
Código.....	31

Definición del problema/objetivo de investigación.

Desarrollar un análisis integral del conjunto de datos de aplicaciones de la Play Store de Google para entender su estructura, limpiar y preparar los datos, realizar un análisis exploratorio exhaustivo y construir modelos predictivos que permitan predecir la calificación de las aplicaciones, utilizando técnicas de Random Forest y árboles de decisión.

Objetivos Específicos

- 1) Obtener un resumen detallado y rápido del conjunto de datos para entender su estructura y contenido.
- 2) Seleccionar las variables relevantes para el análisis y eliminar las irrelevantes.
- 3) Limpiar el conjunto de datos eliminando filas con valores nulos para asegurar la integridad del análisis.
- 4) Realizar análisis visuales y estadísticos para entender mejor las distribuciones y relaciones entre variables.
- 5) Desarrollar modelos predictivos utilizando Random Forest y árboles de decisión para predecir la Calificación de las aplicaciones.

Introducción

En el acelerado universo de las aplicaciones móviles, la Google Play Store representa un inmenso repositorio que aloja millones de aplicaciones, abarcando desde juegos y herramientas de productividad hasta aplicaciones de estilo de vida y educación. Este proyecto se centra en analizar un conjunto de datos de aplicaciones de la Google Play Store para extraer información valiosa sobre las tendencias predominantes y los factores determinantes del éxito en este dinámico mercado. El conjunto de datos seleccionado contiene información detallada sobre diversas métricas de las aplicaciones, tales como el nombre, la categoría, la calificación, el número de instalaciones, y el precio, entre otros.

Descripción de las variables de interés:

- **Categoría:** Esta variable clasifica la aplicación móvil en una categoría específica, como redes sociales, juegos, productividad, etc.
- **Clasificación.de.Contenido:** Indica la calificación o etiqueta de contenido asignada a la aplicación, como "Para todos", "Solo para adultos", "Adolescentes", etc.
- **Calificación:** Esta variable indica la puntuación promedio otorgada por los usuarios a la aplicación. Por lo general, se basa en una escala de 1 a 5 estrellas, donde 5 representa la mejor calificación.

- **Número.de.Calificaciones:** Indica la cantidad total de calificaciones que ha recibido la aplicación. Cuanto mayor sea este número, más representativa será la calificación promedio.
- **Instalaciones.Máximas:** Es el número total de veces que la aplicación ha sido instalada en dispositivos móviles.
- **Admite.Anuncios:** Esta variable es binaria y especifica si la aplicación admite anuncios publicitarios dentro de su interfaz. Puede ser “Sí” o “No”.
- **Compras.dentro.de.la.Aplicación:** Esta variable es binaria y especifica si la aplicación ofrece compras dentro de la aplicación. Puede ser “Sí” o “No”.
- **Precio:** Indica el costo de la aplicación para el usuario. Puede ser un valor numérico que representa el precio en la moneda local, o “Gratis” si la aplicación no tiene costo.

Funciones a programar:

1) Función para generar boxblot multivariables

Preprocesamiento de Datos

Carga de paquetes

```
library(tidyverse)      # Carga el paquete para manipulación de datos
library(readxl)         # Carga el paquete para leer archivos Excel
library(ggplot2)        # Carga el paquete para gráficos
library(dplyr)           # Carga el paquete para manipulación de datos
library(corrplot)       # Carga el paquete para visualizar la matriz de
correlación
library(visdat)         # Carga el paquete para visualizar datos
faltantes
library(readr)          # Carga el paquete para leer archivos csv
library(skimr)          # Carga el paquete para resúmenes estadísticos
rápidos
library(data.tree)      # Carga el paquete para manipulación de árboles
de datos
library(DiagrammeR)     # Carga el paquete para crear diagramas de redes
library(tinytex)        # Carga el paquete para compilar documentos LaTeX
library(RColorBrewer)   # Carga el paquete para paletas de colores para
gráficos
library(randomForest)   # Carga el paquete para algoritmos de bosques
aleatorios

## Warning: package 'randomForest' was built under R version 4.4.1
```

Carga de datos

Carga el archivo desde ubicación de github

```
#enlace <-
'https://raw.githubusercontent.com/EnriqueEsis/EESIS/main/Google-
Playstore_T_R.csv'
#df <- read.csv(enlace, sep = ",")

# Ruta de ubicación de windows
ruta_archivo <- 'C:\\Users\\EEsis\\Downloads\\Datasets\\Google-
Playstore_T_R.csv'

# Cargar el archivo CSV desde ubicación de windows
df <- read.csv(ruta_archivo, fileEncoding = "Latin1")
```

Identificar los tipos de variables

Mostrar resumen rápido y completo del conjunto de datos

```
skim(df)
```

Data summary

Name	df
Number of rows	31451
Number of columns	23

Column type frequency:



character	18
numeric	5

Group variables	None
-----------------	------

Variable type: character

skim_variable	n_missing	complete_rate	m	m	em	n_unique	whitespace
			n	x	pty		ace
ID.de.la.Aplicación	94	1	5	1	0	3135	0
				3		7	
				9			
Categoría	94	1	5	1	0	10	0
				7			
Instalaciones	94	1	8	8	0	2	0
Gratis	94	1	4	5	0	2	0
Moneda	94	1	0	3	1	3	0
Tamaño	94	1	3	1	0	839	0
				8			

skim_variable	n_missing	complete_rate	in	max	empty	n_unique	whitespace
Android.Mínimo	94	1	0	18	34	40	0
ID.del.Desarrollador	94	1	2	106	0	21229	0
Sitio.Web.del.Desarrollador	94	1	0	223	8261	17023	0
Correo.Electrónico.del.Desarrollador	94	1	0	111	4	22092	0
Fecha.de.Lanzamiento	94	1	0	12	732	3913	0
Última.Actualización	94	1	12	12	0	2867	0
Clasificación.de.Contenido	94	1	4	15	0	6	0
Política.de.Privacidad	94	1	0	233	4066	20446	0
Admite.Anuncios	94	1	4	5	0	2	0
Compras.dentro.de.la.Aplicación	94	1	4	5	0	2	0
Elección.de.los.Editores	94	1	4	5	0	2	0
Tiempo.de.Extracción	94	1	19	19	0	24718	0

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
Calificación	94	1	3.97	0.62	1.1	3.7	4.1	4.4	5.00	
Número.de.Calificaciones	94	1	1133.70	561.65	400.0	656.0	1004.0	1518.0	2500.00	

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
Instalaciones.Mínimas	94	1	1260 10.1 4	9862 9.81	100 000. 0	100 000. 0	100 000. 0	100 000. 0	5000 00.0 0	
Instalaciones.Máximas	94	1	2452 63.3 3	1292 50.8 7	105 482. 0	145 735. 0	203 759. 0	307 287. 0	6765 07.0 0	
Precio	94	1	0.01	0.24	0.0	0.0	0.0	0.0	19.9 9	

head(df)

```
##                               ID.de.la.Aplicación                Categoría
Calificación
## 1                               com.bisgumah.barbie             Entertainment
3.8
## 2                               com.impark.isemdijital             Education
2.4
## 3                               com.qamar.ide.web                Productivity
4.1
## 4                               fi.dna.omadna                    Tools
3.9
## 5                               com.bilinguae.deutsch.vokabular    Education
4.7
## 6 com.digital.bangla.sokol_duyar_fojilot_o_amol Books & Reference
4.7
##  Número.de.Calificaciones  Instalaciones  Instalaciones.Mínimas
## 1                               736          500,000+             5e+05
## 2                               853          100,000+             1e+05
## 3                               1476         100,000+             1e+05
## 4                               440          100,000+             1e+05
## 5                               520          100,000+             1e+05
## 6                               570          100,000+             1e+05
##  Instalaciones.Máximas  Gratis  Precio  Moneda                Tamaño
Android.Mínimo
## 1                646456    True      0      USD                20M
5.0 and up
## 2                119488    True      0      USD  Varies with device
4.1 and up
```

## 3	192935	True	0	USD	6.0M
4.1 and up					
## 4	174403	True	0	USD	60M
6.0 and up					
## 5	130045	True	0	USD	6.2M
5.1 and up					
## 6	172665	True	0	USD	3.5M
4.1 and up					
##	ID.del.Desarrollador	Sitio.Web.del.Desarrollador			
## 1	bisgumah	https://appsabsadstxt.blogspot.com/			
## 2	Ä°MPARK	http://www.impark.com.tr			
## 3	Alif software				
## 4	DNA oyj	https://www.dna.fi			
## 5	AltairApps	https://www.bilinguae.com			
## 6	BD Apps Station				
##	Correo.Electrónico.del.Desarrollador	Fecha.de.Lanzamiento			
## 1	bisgumah418@gmail.com	Apr 21, 2019			
## 2	yonetim@impark.com.tr	Mar 15, 2019			
## 3	alifsoftware6@gmail.com	Sep 7, 2018			
## 4	app.development@dna.fi	May 23, 2019			
## 5	info@bilinguae.com	May 16, 2017			
## 6	shahenmustakim008@gmail.com	Feb 20, 2019			
##	Última.Actualización	Clasificación.de.Contenido			
## 1	Nov 13, 2020	Everyone			
## 2	Nov 02, 2020	Everyone			
## 3	Jun 05, 2021	Everyone			
## 4	May 27, 2021	Everyone			
## 5	Jun 14, 2021	Everyone			
## 6	Mar 21, 2020	Everyone			
##					
Política.de.Privacidad					
## 1	https://docs.google.com/document/d/1kUDGpRA8fFBpy6x_L3tuC-C0po_ONPcSu1qRKVv3gNg/edit?usp=sharing				
## 2	http://impark.com.tr/gizlilik_sozlesmesi.pdf				
## 3					
## 4	https://corporate.dna.fi/tietosuoja				
## 5					
## 6	https://www.bilinguae.com/privacy-policy				
##					
https://appsstationbd.blogspot.com/p/privacy-policy.html					
##	Admite.Anuncios	Compras.dentro.de.la.Aplicación			
Elección.de.los.Editores					
## 1	True	False			
False					
## 2	False	False			
False					
## 3	False	True			
False					


```
## 4          True          False
False
## 5          True          True
False
## 6          True          False
False
##  Tiempo.de.Extracción
## 1  2021-06-15 20:19:36
## 2  2021-06-15 20:19:48
## 3  2021-06-15 20:19:54
## 4  2021-06-15 20:20:10
## 5  2021-06-15 20:20:16
## 6  2021-06-15 20:20:17
```

Selección y eliminación de variables

Se seleccionan las variables cuantitativas, cualitativas y se procede a eliminar el resto de las variables:

```
# Nombre de Las columnas que deseas conservar
columnas_a_conservar <- c("Admite.Anuncios",
                          "Calificación",
                          "Categoría",
                          "Precio",
                          "Clasificación.de.Contenido",
                          "Instalaciones.Máximas",
                          "Compras.dentro.de.la.Aplicación",
                          "Número.de.Calificaciones")

# Seleccionar solo las columnas especificadas
df <- df[, columnas_a_conservar]

view(df)
```

Eliminar filas con valores nulos y NA

```
# Eliminar filas con NA en las columnas "Calificación"
df <- df[!is.na(df$Calificación), ]

# Eliminar filas con valores NA
df <- na.omit(df)
```

Verificar los cambios efectuados

Mostrar resumen rápido y completo del conjunto de datos para verificar los cambios realizados

```
#df <- iris # Para comprobar Los gráficos con otro dataset

skim(df)
```

Name	df
Number of rows	31357
Number of columns	8

character	4
numeric	4

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
Admite.Anuncios	0	1	4	5	0	2	0
Categoría	0	1	5	17	0	10	0
Clasificación.de.Contenido	0	1	4	15	0	6	0
Compras.dentro.de.la.Aplicación	0	1	4	5	0	2	0

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
Calificación	0	1	3.97	0.62	1.1	3.7	4.1	4.4	5.00	
Precio	0	1	0.01	0.24	0.0	0.0	0.0	0.0	19.99	
Instalaciones.Máximas	0	1	245263.33	129250.87	105482.0	145735.0	203759.0	307287.0	676507.0	

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
Número.de.Calificaciones	0	1	1133.70	561.65	400.0	656.0	1004.0	1518.0	2500.00	— ■ ■ ■ ■ ■

```
head(df)
```

```
##   Admite.Anuncios Calificación Categoría Precio
## 1             True           3.8 Entertainment    0
## 2             False           2.4 Education      0
## 3             False           4.1 Productivity    0
## 4             True           3.9 Tools          0
## 5             True           4.7 Education      0
## 6             True           4.7 Books & Reference 0
##   Clasificación.de.Contenido Instalaciones.Máximas
## 1                      Everyone           646456
## 2                      Everyone           119488
## 3                      Everyone           192935
## 4                      Everyone           174403
## 5                      Everyone           130045
## 6                      Everyone           172665
##   Compras.dentro.de.la.Aplicación Número.de.Calificaciones
## 1                      False              736
## 2                      False              853
## 3                      True             1476
## 4                      False              440
## 5                      True              520
## 6                      False              570
```

Análisis Exploratorio de Datos

Graficos variables cuantitativas (BOXPLOT)

```
numeric_df <- dplyr::select_if(df, is.numeric)
```

```
for (columna in names(numeric_df)) {
```

```
  # Calcular Los valores estadísticos
  mediana <- median(numeric_df[[columna]])
  q1 <- quantile(numeric_df[[columna]], 0.25, na.rm = TRUE)
  q3 <- quantile(numeric_df[[columna]], 0.75, na.rm = TRUE)
  min_val <- min(numeric_df[[columna]])
  max_val <- max(numeric_df[[columna]])
```

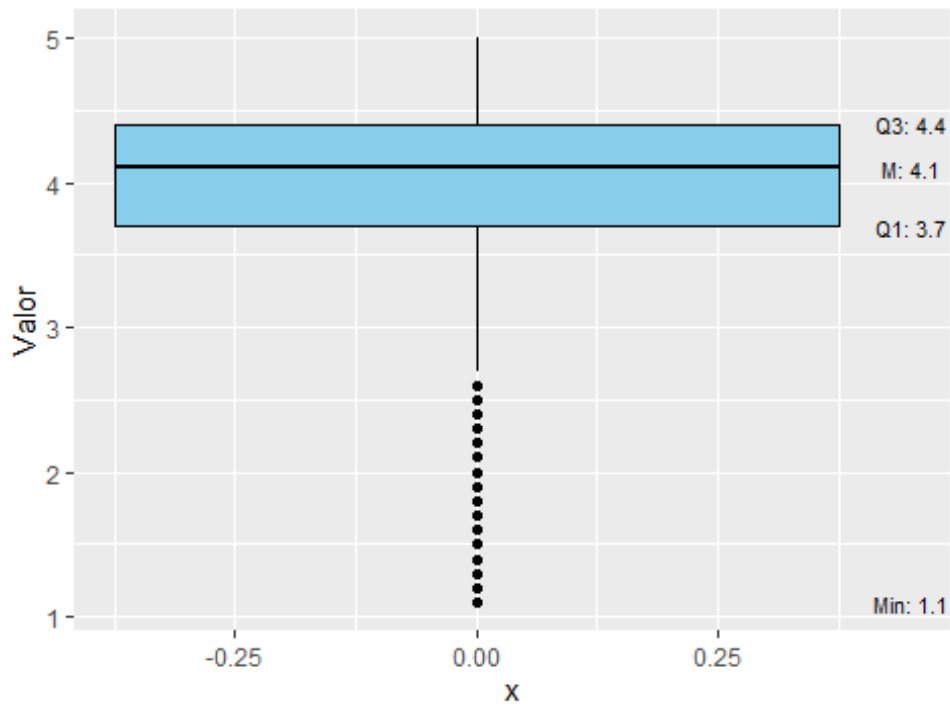
```

# Boxplot con etiquetas numéricas usando annotate
boxplot <- ggplot(numeric_df, aes(y = !!sym(columna))) +
  geom_boxplot(fill = "skyblue", color = "black") +
  annotate("text", x = 0.45, y = mediana, label = paste("M:",
round(mediana, 2)), color = "black", size = 3) +
  annotate("text", x = 0.45, y = q1, label = paste("Q1:", round(q1,
2)), color = "black", size = 3) +
  annotate("text", x = 0.45, y = q3, label = paste("Q3:", round(q3,
2)), color = "black", size = 3) +
  annotate("text", x = 0.45, y = min_val, label = paste("Min:",
round(min_val, 2)), color = "black", size = 3) +
  labs(title = paste("Boxplot de", columna), y = "Valor")

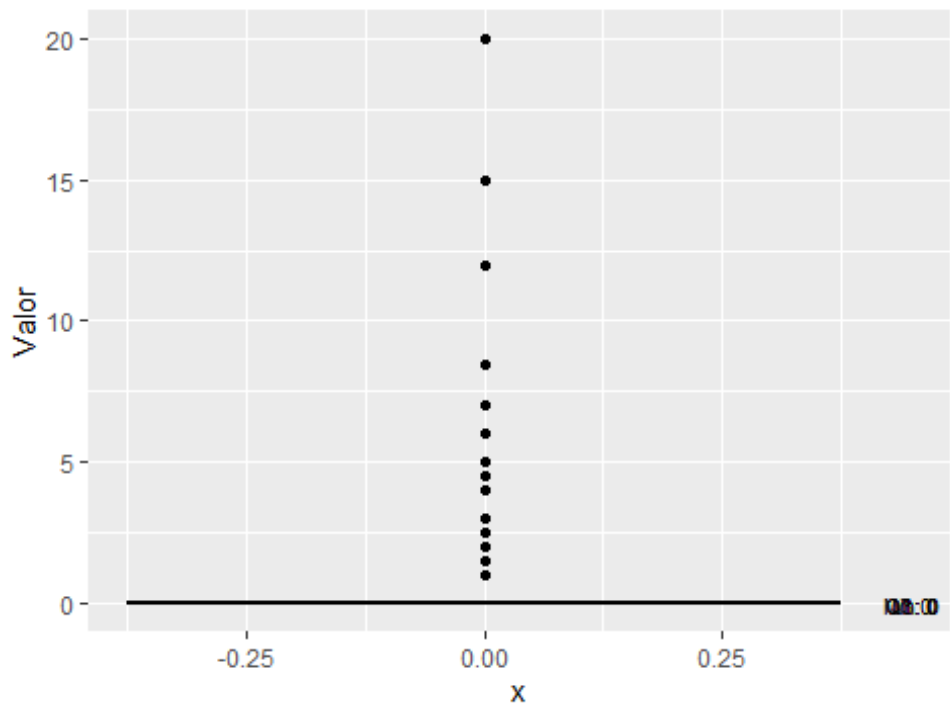
# Mostrar el gráfico
print(boxplot)
}

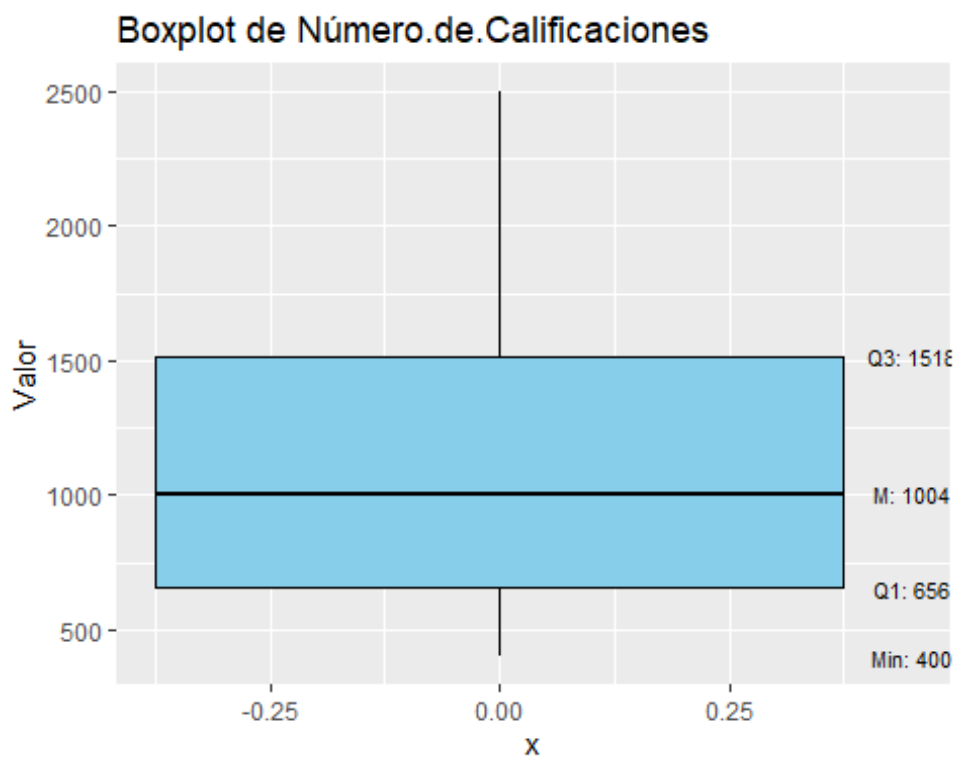
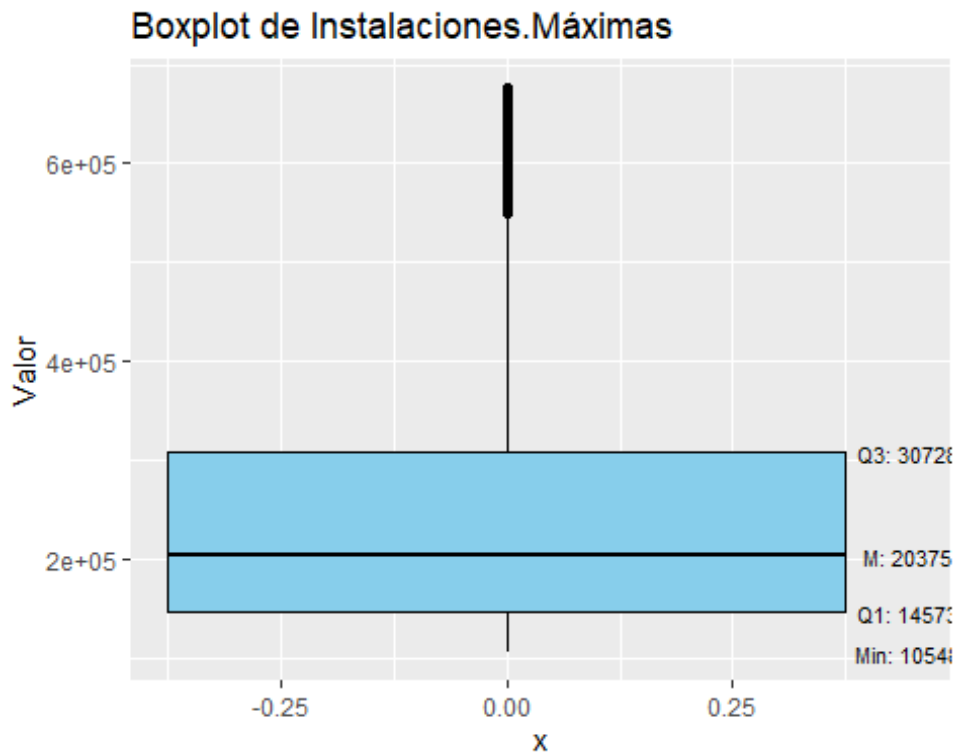
```

Boxplot de Calificación



Boxplot de Precio





Graficos variables cualitativas (BARPLOT y PIECHART)

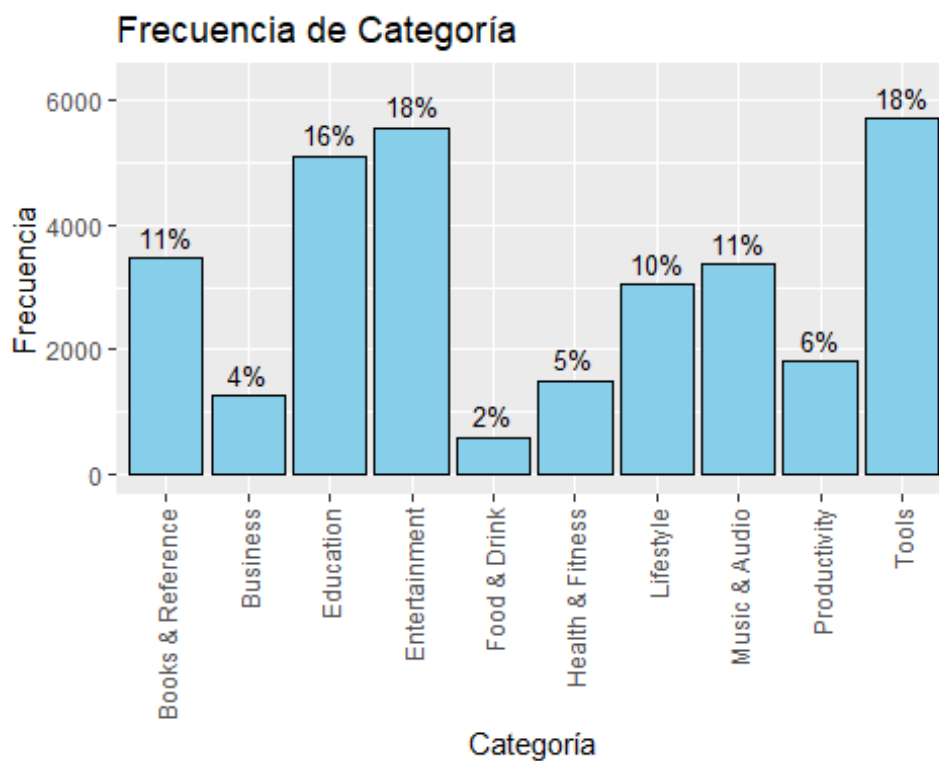
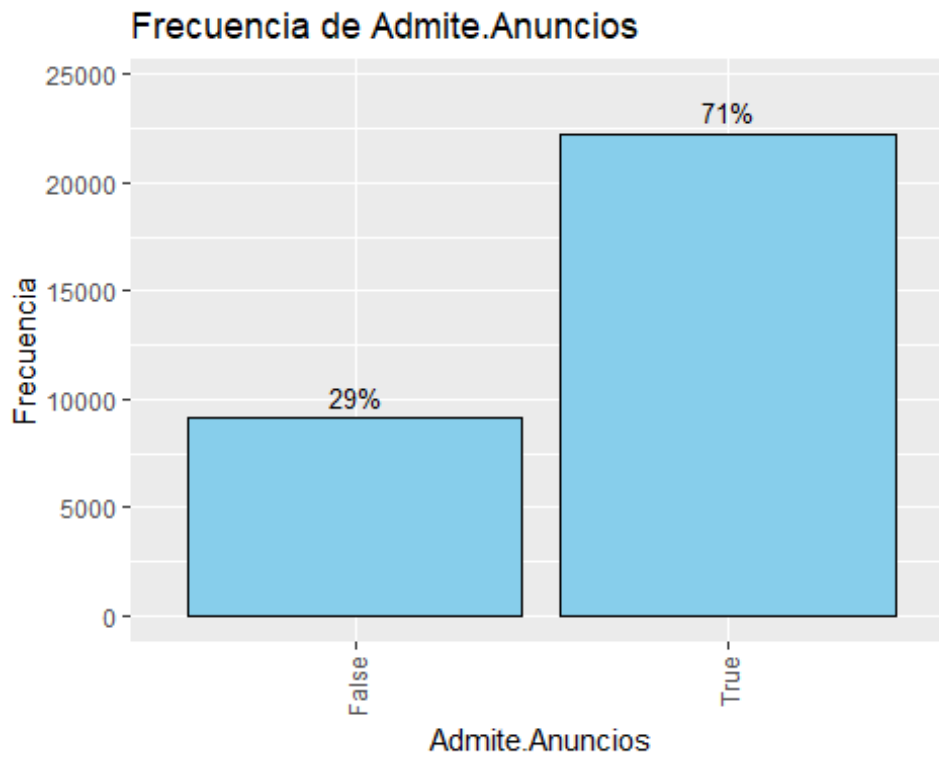
```
# Seleccionar solo las columnas categóricas
columnas_chr <- names(df)[sapply(df, is.character)]
# Iterar a través de las columnas categóricas y crear los gráficos
```

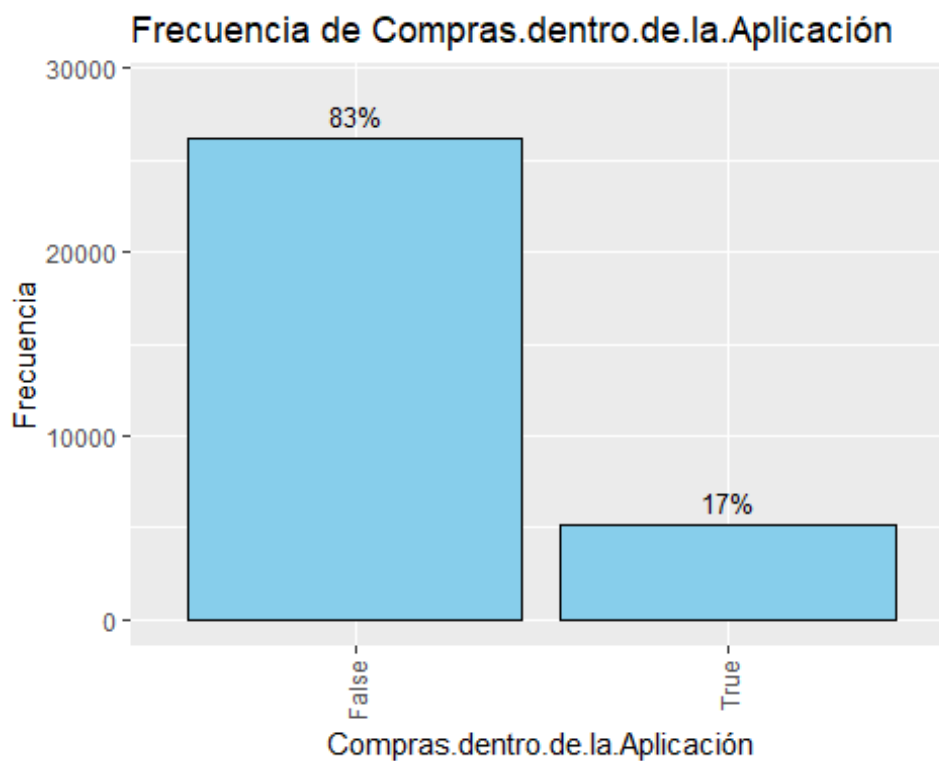
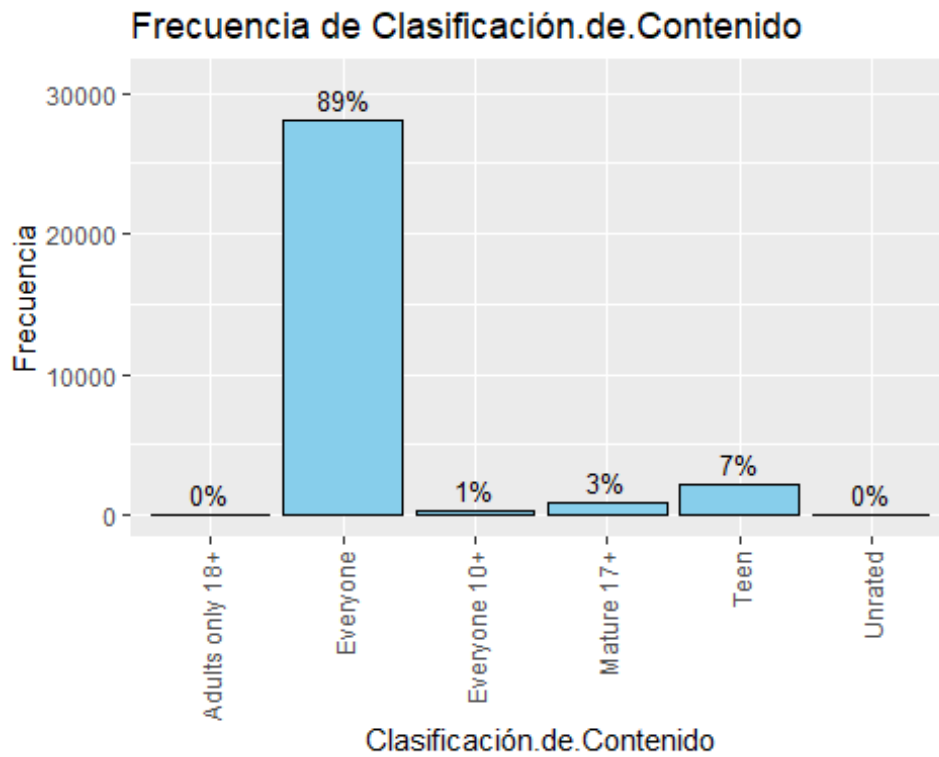
```

for (column in columnas_chr) {

  # Crear un gráfico de barras
  barplot <- ggplot(data = df, aes(x = !!as.name(column))) +
    geom_bar(fill = "skyblue", color = "black") +
    geom_text(stat = "count", aes(label = paste0(round(after_stat(count)
/ sum(after_stat(count)) * 100), "%")),
              position = position_stack(vjust = 1), vjust = -0.5, color =
"black", size = 3.5) +
    labs(title = paste("Frecuencia de", column),
          x = column,
          y = "Frecuencia")+
    theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1)) +
  # Rotar las etiquetas del eje x
  expand_limits(y = max(table(df[[column]])) * 1.1) # Ajustar los
límites del eje y para que las etiquetas no se salgan del gráfico
  print(barplot)
}

```





Medidas estadísticas de variables cuantitativas

```
summary(numeric_df)
```

```
## Calificación Precio Instalaciones.Máximas
## Min. :1.100 Min. : 0.000000 Min. :105482
## 1st Qu.:3.700 1st Qu.: 0.000000 1st Qu.:145735
## Median :4.100 Median : 0.000000 Median :203759
## Mean :3.965 Mean : 0.007014 Mean :245263
## 3rd Qu.:4.400 3rd Qu.: 0.000000 3rd Qu.:307287
## Max. :5.000 Max. :19.990000 Max. :676507
## Número.de.Calificaciones
## Min. : 400
## 1st Qu.: 656
## Median :1004
## Mean :1134
## 3rd Qu.:1518
## Max. :2500
```

Matriz de correlación de variables cuantitativas

Matriz de correlación de variables cuantitativas

```
correlation_matrix <- cor(numeric_df)
correlation_matrix
```

```
##
## Instalaciones.Máximas Calificación Precio
## Calificación 1.000000000 0.009501618 -
0.096382305
## Precio 0.009501618 1.000000000 -
0.009850668
## Instalaciones.Máximas -0.096382305 -0.009850668
1.000000000
## Número.de.Calificaciones 0.116875488 0.021971408
0.292684396
##
## Número.de.Calificaciones
## Calificación 0.11687549
## Precio 0.02197141
## Instalaciones.Máximas 0.29268440
## Número.de.Calificaciones 1.00000000
```

Gráfico de dispersión entre variables cuantitativas

Gráfico de dispersión

Crear una lista de todas las combinaciones posibles de variables
combinations <- **combn**(names(numeric_df), 2)

Crear un gráfico de dispersión para cada par de variables

```
for (i in 1:ncol(combinations)) {
  var1 <- combinations[1, i]
  var2 <- combinations[2, i]

  # Crear el gráfico de dispersión
  gg <- ggplot(numeric_df, aes(x = !!sym(var2), y = !!sym(var1))) +
    geom_point() +
    geom_smooth(method = "lm", se = FALSE, color = "blue") + # Agregar
```

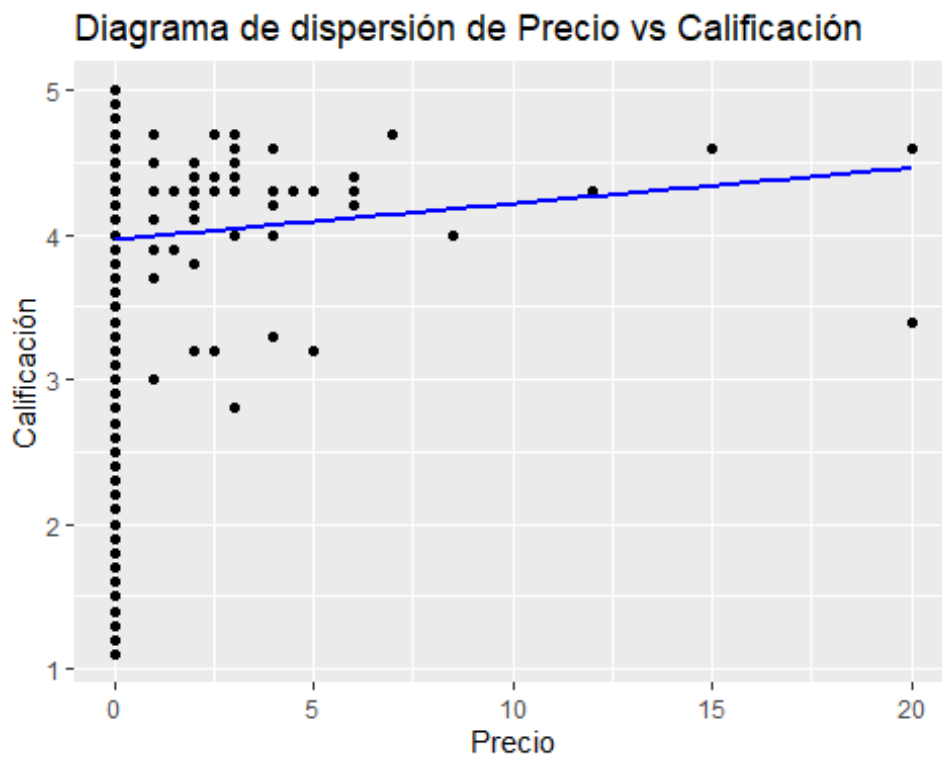
```

línea de regresión lineal
  labs(title = paste("Diagrama de dispersión de", var2, "vs", var1))

# Mostrar el gráfico
print(gg)
}

## `geom_smooth()` using formula = 'y ~ x'

```

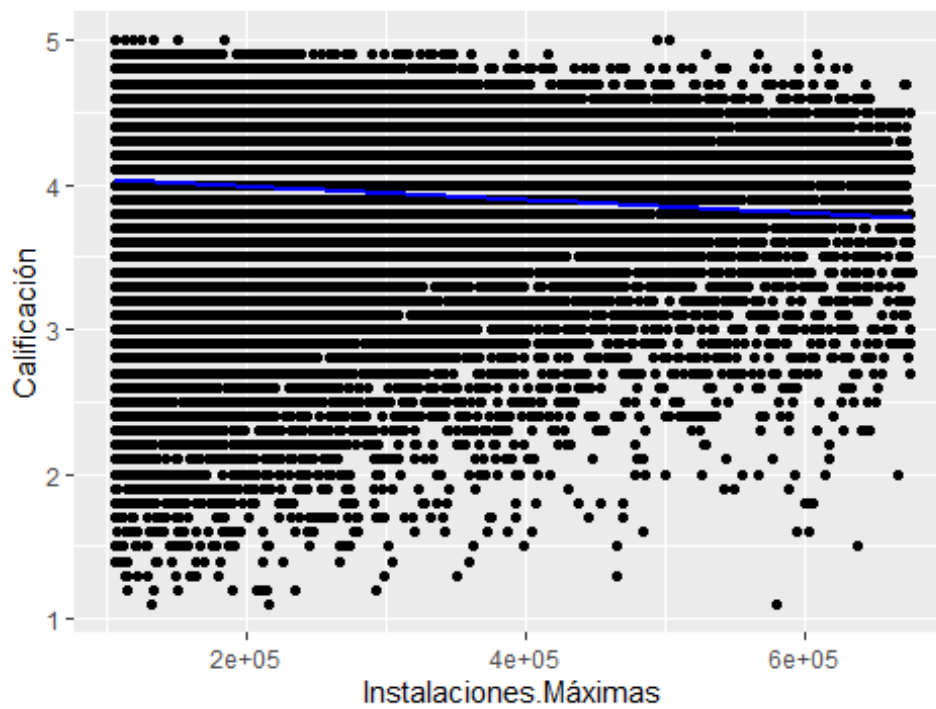


```

## `geom_smooth()` using formula = 'y ~ x'

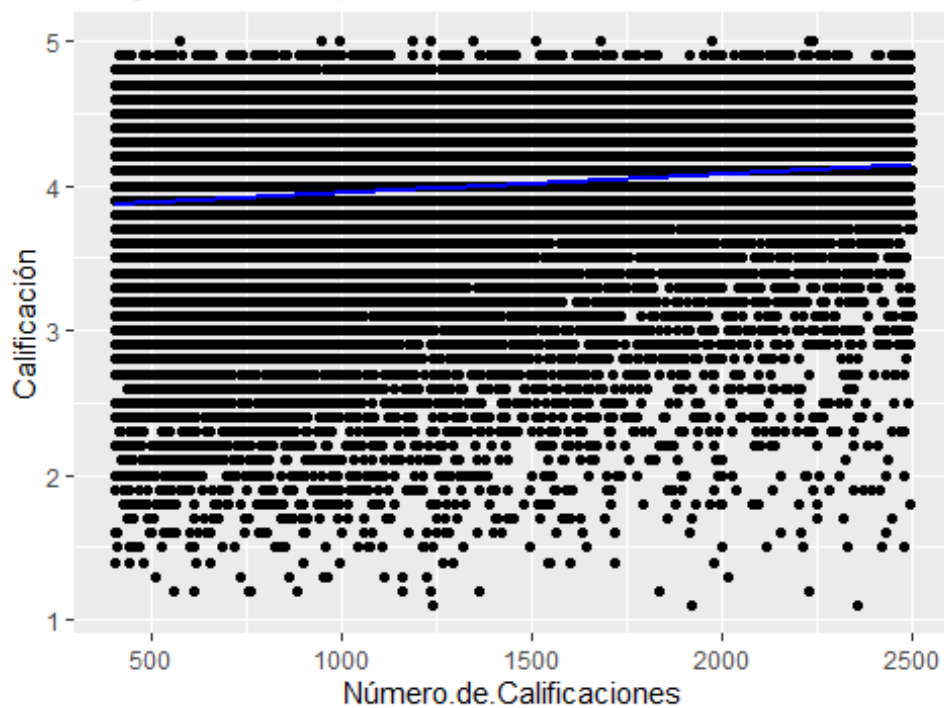
```

Diagrama de dispersión de Instalaciones.Máximas vs C

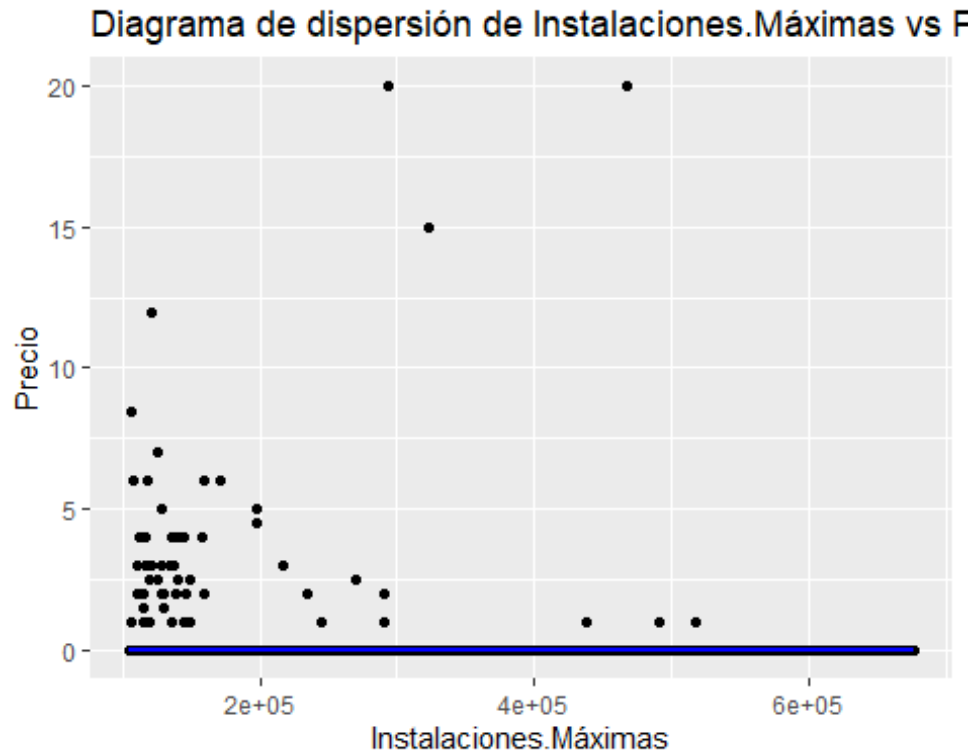


```
## `geom_smooth()` using formula = 'y ~ x'
```

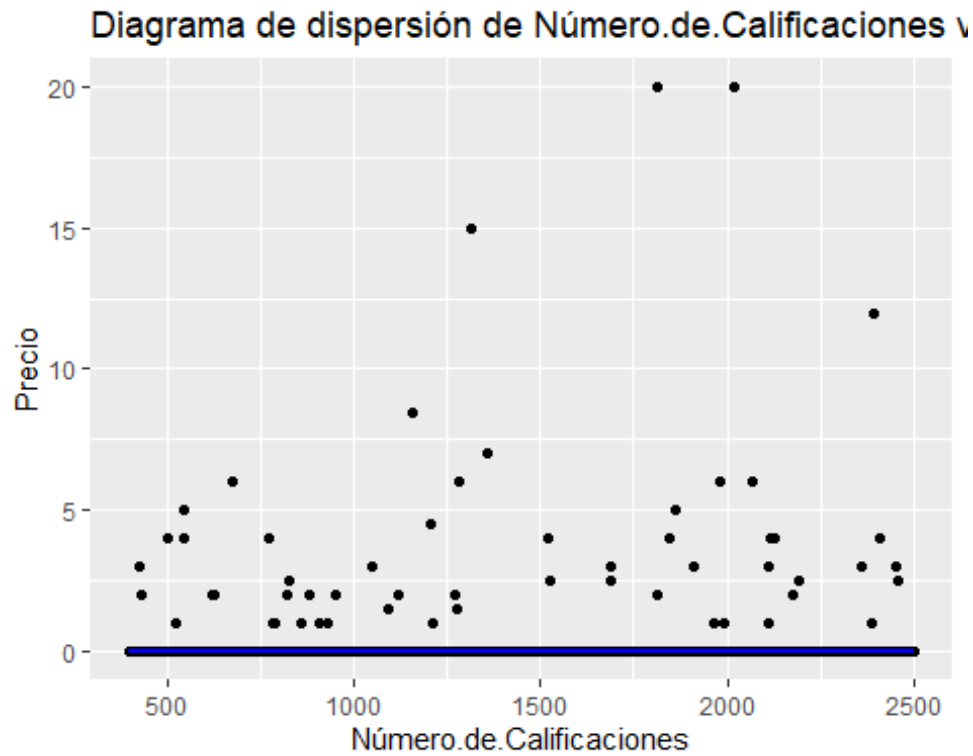
Diagrama de dispersión de Número.de.Calificaciones vs



```
## `geom_smooth()` using formula = 'y ~ x'
```

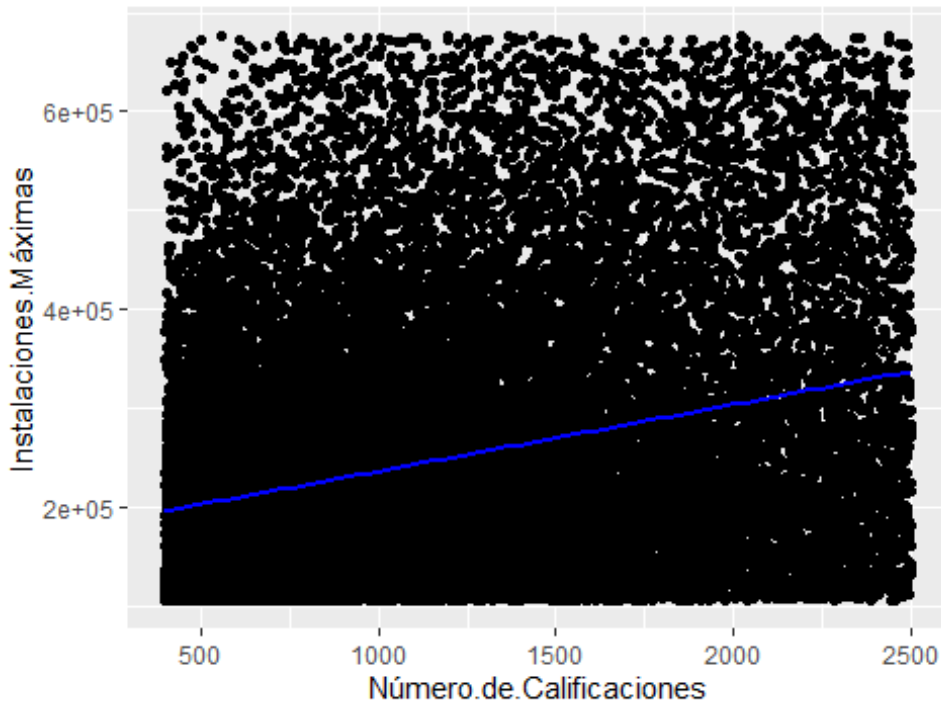


```
## `geom_smooth()` using formula = 'y ~ x'
```



```
## `geom_smooth()` using formula = 'y ~ x'
```

Diagrama de dispersión de Número.de.Calificacione

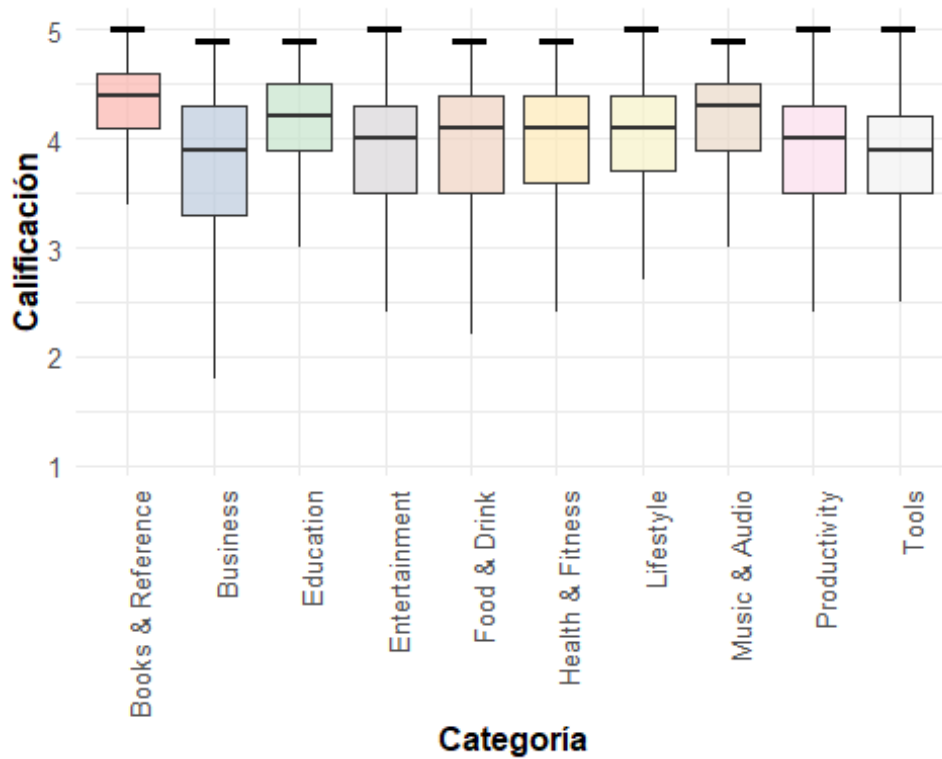


Gráficos multivariables

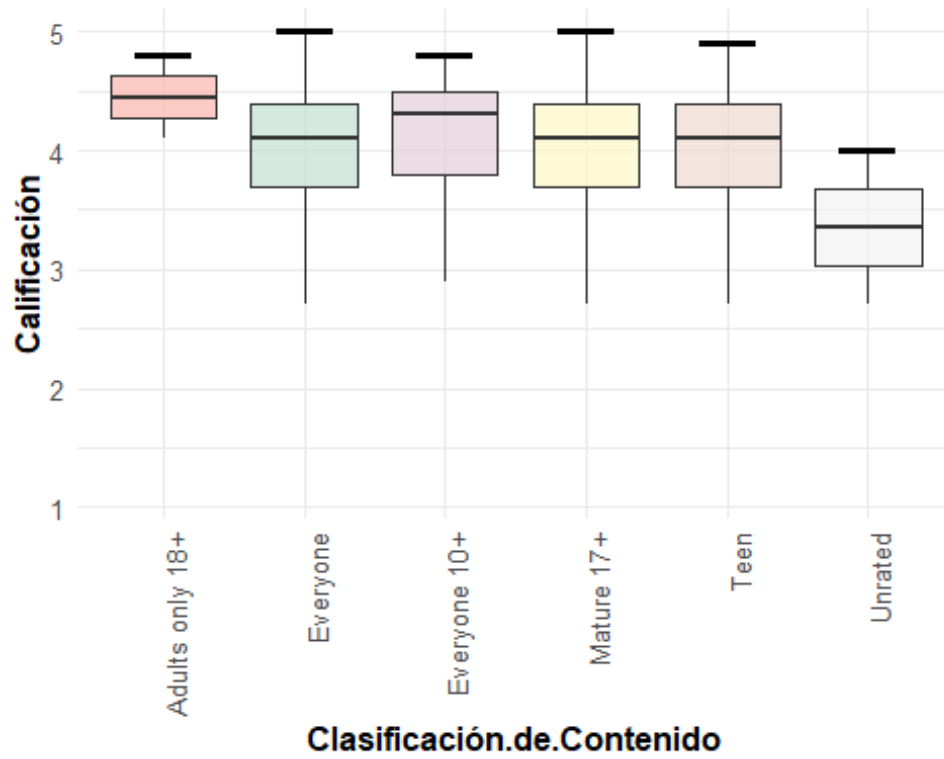
```
# función para crear el boxplot
crear_boxplot <- function(data, columna_x, columna_y) {
  # Creamos el gráfico base
  plot <- ggplot(data, aes(x = !!sym(columna_x), y = !!sym(columna_y),
    fill = !!sym(columna_x))) +
    geom_boxplot(outlier.shape = NA, alpha = 0.7) +
    stat_summary(fun = function(x) quantile(x, 0.75), geom = "errorbar",
    width = 0.4, color = "black", fun.max = max, linewidth = 1.2) +
    labs(x = columna_x, y = columna_y) +
    scale_fill_manual(values = colorRampPalette(brewer.pal(9,
    "Pastel1"))(length(unique(data[[columna_x]])))) +
    theme_minimal() +
    theme(
      axis.text.x = element_text(angle = 90, vjust = 1, hjust = 1, size =
10),
      axis.text.y = element_text(size = 10),
      axis.title = element_text(size = 12, face = "bold"),
      legend.position = "none"
    )

  return(plot)
}
# Categoría y Calificación
grafico <- crear_boxplot(data = df, columna_x = "Categoría", columna_y =
```

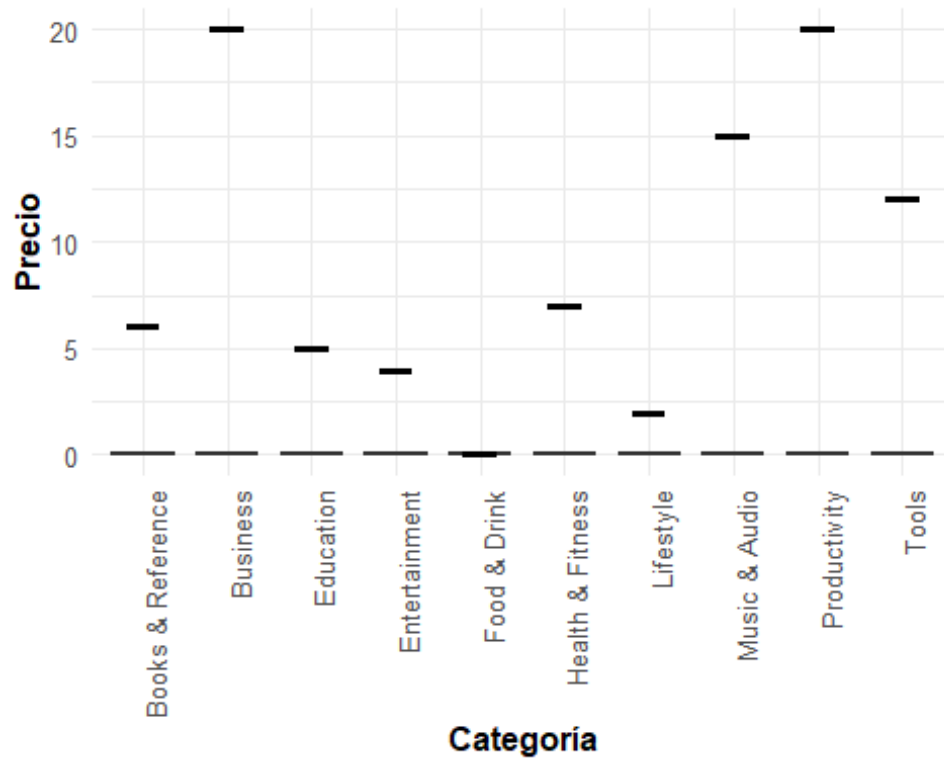
```
"Calificación")  
print(grafico)
```



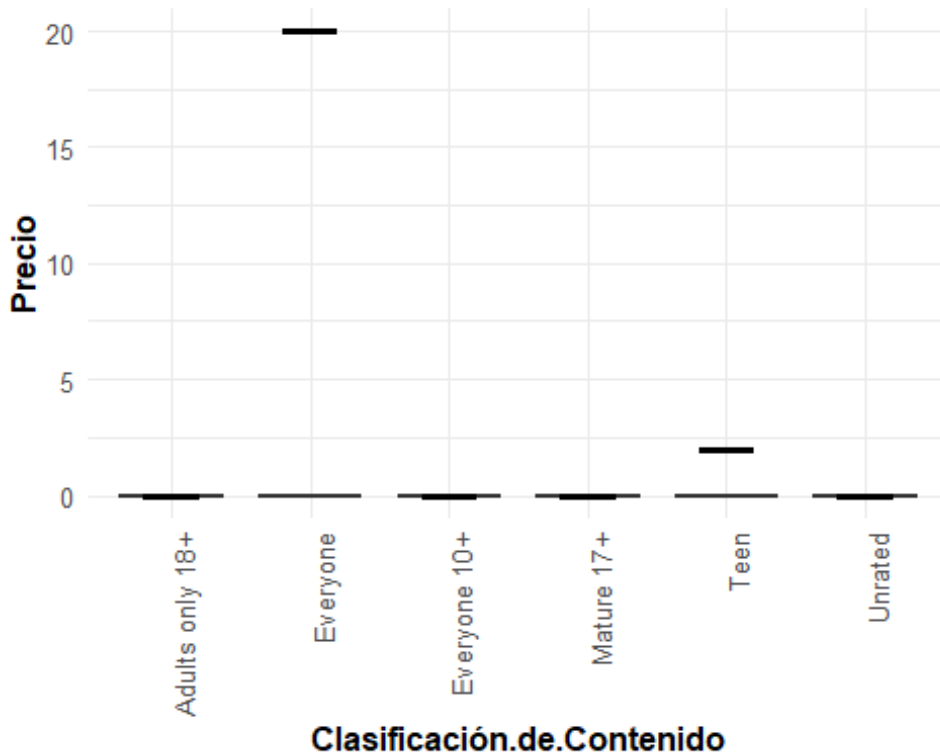
```
# Clasificación.de.Contenido y Calificación  
grafico <- crear_boxplot(data = df, columna_x =  
"Clasificación.de.Contenido", columna_y = "Calificación")  
print(grafico)
```



```
# Categoría y Precio
grafico <- crear_boxplot(data = df, columna_x = "Categoría", columna_y =
"Precio")
print(grafico)
```

```
# Clasificación.de.Contenido y Precio
grafico <- crear_boxplot(data = df, columna_x =
"Clasificación.de.Contenido", columna_y = "Precio")
print(grafico)
```



Modelado de Datos

Modelo de Random Forest

```
# Seleccionar variables de interés
df_selected <- df[, c("Calificación", "Categoría",
"Clasificación.de.Contenido", "Admite.Anuncios")]

# Verificar y manejar valores faltantes si los hay
sum(is.na(df_selected)) # Verificar valores faltantes

## [1] 0

df_selected <- na.omit(df_selected) # Eliminar filas con valores
faltantes, si es necesario

# Convertir variables categóricas a factores si es necesario
df_selected$Categoría <- as.factor(df_selected$Categoría)
df_selected$Clasificación.de.Contenido <-
as.factor(df_selected$Clasificación.de.Contenido)
df_selected$Admite.Anuncios <- as.factor(df_selected$Admite.Anuncios)

# Entrenar el modelo de Random Forest
rf_model <- randomForest(Calificación ~ Categoría +
Clasificación.de.Contenido + Admite.Anuncios, data = df_selected)
```

```

# Ver resumen del modelo
print(rf_model)

##
## Call:
## randomForest(formula = Calificación ~ Categoría +
Clasificación.de.Contenido +      Admite.Anuncios, data = df_selected)
##
##           Type of random forest: regression
##           Number of trees: 500
## No. of variables tried at each split: 1
##
##           Mean of squared residuals: 0.338422
##           % Var explained: 12.08

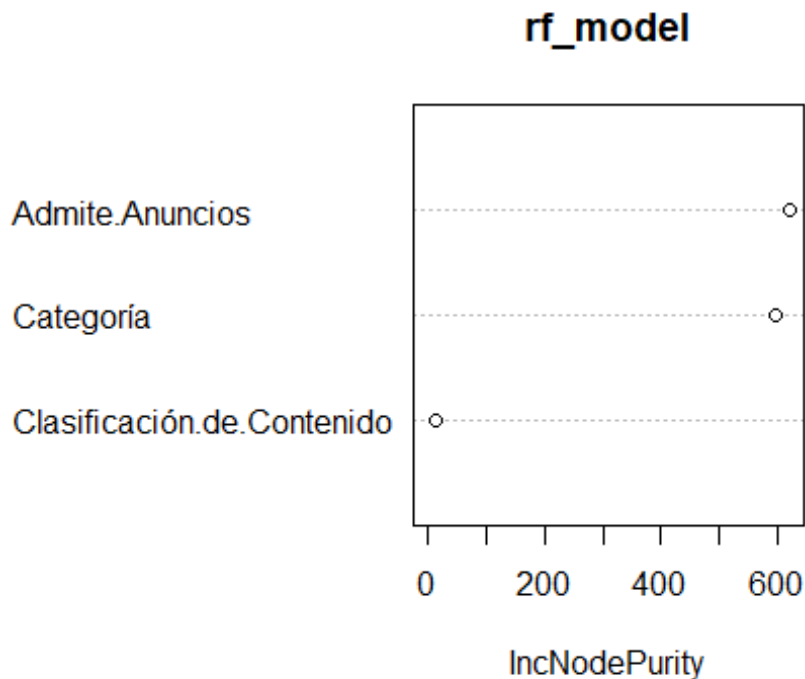
# Obtener la importancia de Las variables
importancia_variables <- importance(rf_model)

# Imprimir los valores de importancia
print(importancia_variables)

##
##           IncNodePurity
## Categoría              595.77001
## Clasificación.de.Contenido    14.58391
## Admite.Anuncios             620.36678

# Ver importancia de Las variables
varImpPlot(rf_model)

```



```

library(rpart)

## Warning: package 'rpart' was built under R version 4.4.1

library(rpart.plot)

## Warning: package 'rpart.plot' was built under R version 4.4.1

# Ajustar el modelo de árbol de decisión
tree_model <- rpart(Calificación ~ Categoría + Clasificación.de.Contenido
+ Admite.Anuncios, data = df_selected)

# Imprimir detalles del árbol
printcp(tree_model)

##
## Regression tree:
## rpart(formula = Calificación ~ Categoría + Clasificación.de.Contenido
+
##      Admite.Anuncios, data = df_selected)
##
## Variables actually used in tree construction:
## [1] Admite.Anuncios Categoría
##
## Root node error: 12070/31357 = 0.38494
##
## n= 31357
##
##      CP nsplit rel error  xerror      xstd
## 1 0.063638      0  1.00000 1.00005 0.0109078
## 2 0.040884      1  0.93636 0.93650 0.0098811
## 3 0.015058      2  0.89548 0.89573 0.0096584
## 4 0.010000      3  0.88042 0.88126 0.0095730

summary(tree_model)

## Call:
## rpart(formula = Calificación ~ Categoría + Clasificación.de.Contenido
+
##      Admite.Anuncios, data = df_selected)
##      n= 31357
##
##      CP nsplit rel error    xerror      xstd
## 1 0.06363752      0 1.0000000 1.0000475 0.010907769
## 2 0.04088407      1 0.9363625 0.9365003 0.009881079
## 3 0.01505812      2 0.8954784 0.8957282 0.009658368
## 4 0.01000000      3 0.8804203 0.8812642 0.009573045
##
## Variable importance
## Admite.Anuncios      Categoría
##           51           49

```

```

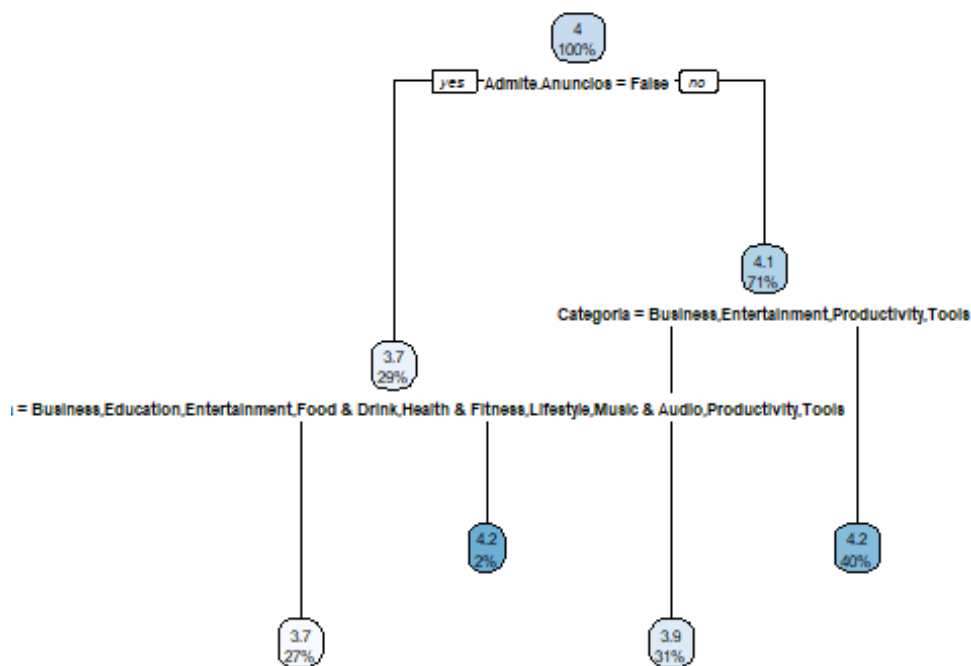
##
## Node number 1: 31357 observations,    complexity param=0.06363752
##   mean=3.965102, MSE=0.3849367
##   left son=2 (9127 obs) right son=3 (22230 obs)
##   Primary splits:
##       Admite.Anuncios           splits as  LR,
improve=0.0636375200, (0 missing)
##       Categoría                 splits as  RLRLLLLRLL,
improve=0.0597264300, (0 missing)
##       Clasificación.de.Contenido splits as  RLRLLL,
improve=0.0004429109, (0 missing)
##   Surrogate splits:
##       Categoría splits as  RLRRRRRRRR, agree=0.732, adj=0.081, (0
split)
##
## Node number 2: 9127 observations,    complexity param=0.01505812
##   mean=3.720839, MSE=0.568469
##   left son=4 (8504 obs) right son=5 (623 obs)
##   Primary splits:
##       Categoría                 splits as  RLLLLLLLLLL,
improve=0.0350315900, (0 missing)
##       Clasificación.de.Contenido splits as  -LRLL-,
improve=0.0004594666, (0 missing)
##
## Node number 3: 22230 observations,    complexity param=0.04088407
##   mean=4.065389, MSE=0.2750297
##   left son=6 (9737 obs) right son=7 (12493 obs)
##   Primary splits:
##       Categoría                 splits as  RLRLRRRRLL,
improve=0.0807158700, (0 missing)
##       Clasificación.de.Contenido splits as  RRRLLL,
improve=0.0009366401, (0 missing)
##   Surrogate splits:
##       Clasificación.de.Contenido splits as  RRRLRR, agree=0.565,
adj=0.007, (0 split)
##
## Node number 4: 8504 observations
##   mean=3.682643, MSE=0.5592237
##
## Node number 5: 623 observations
##   mean=4.242215, MSE=0.4029209
##
## Node number 6: 9737 observations
##   mean=3.896621, MSE=0.3082817
##
## Node number 7: 12493 observations
##   mean=4.196926, MSE=0.2096119

```

```

# Graficar el árbol de decisión
rpart.plot(tree_model)

```



Interpretación de Resultados

Distribuciones y Estadísticas Descriptivas:

Se observó que la variable “Calificación” tiene una distribución centrada alrededor de 4.2, con una dispersión moderada.

Las variables como “Instalaciones Máximas” y “Número de Calificaciones” mostraron amplias variaciones, indicando diferentes niveles de popularidad entre las aplicaciones.

Los gráficos de caja revelaron variaciones significativas en la calificación promedio entre diferentes categorías de aplicaciones y clasificaciones de contenido.

Correlaciones entre Variables:

La matriz de correlación sugiere que las variables numéricas analizadas tienen correlaciones débiles entre sí, lo que implica que cada variable aporta información única al modelo.

Análisis de Variables Categóricas:

Se exploraron las frecuencias de las categorías de aplicaciones y las clasificaciones de contenido a través de gráficos de barras y gráficos circulares, destacando las distribuciones relativas de estas características entre las aplicaciones.

Modelado Predictivo

Modelo de Random Forest:

Se construyó un modelo de Random Forest para predecir las calificaciones de las aplicaciones.

Las variables “Categoría”, “Clasificación de Contenido” y “Admite Anuncios” fueron identificadas como predictores importantes de la calificación de las aplicaciones según la importancia calculada por el modelo.

Este modelo proporciona una buena precisión predictiva y es capaz de manejar relaciones no lineales entre las variables predictoras y la variable objetivo.

Conclusión

La experiencia inicial con R revela un viaje enriquecedor y transformador en el análisis de datos. Este lenguaje no solo simplifica la manipulación y análisis de datos complejos, sino que también abre las puertas a un vasto conjunto de herramientas y técnicas estadísticas. Las librerías desarrolladas por la comunidad activa y colaborativa que respalda R, ofrecen un entorno ideal para el crecimiento continuo en habilidades analíticas y de programación. La amplia disponibilidad de librerías y su naturaleza de código abierto hacen de R una elección poderosa y motivadora para cualquiera interesado en explorar y comprender el mundo a través de los datos.

Código

Se adjunta el código completo como un archivo .rmd, sin embargo el código se detalla paso a paso en el archivo .html adjunto.