Fundamentos de Análisis de Datos

Unidad 5

Análisis Exploratorio de Datos (EDA)

Referencias

- Maldonado, S. (2022). *Analytics y Big Data: ciencia de los Datos aplicada al mundo de los negocios:* (1 ed.). RIL editores. https://elibro.net/es/ereader/bibliotecautpl/225562?page=8 (apartado 2.1, pág 41)
- Diego, I. M. D. & Fernández Isabel, A. (2020). *Ciencia de datos para la ciberseguridad:* (1 ed.). RA-MA Editorial. https://elibro.net/es/ereader/bibliotecautpl/222714?page=129 (apartado Análisis exploratorio de datos, pág 114)

Agenda

- Introducción al EDA
  - Conceptos
  - Objetivos
  - Pasos
  - Herramientas
- Comprensión del conjunto de datos
  - Detección y tratamiento de valores faltantes
  - Identificación de tipos de variables
- Análisis univariado de variables categóricas
- Análisis univariado de variables numéricas
  - Medidas de tendencia central y de dispersión
  - Gráficas de distribución, dispersión, y correlación
  - Detección y tratamiento de outliers

Introducción al Análisis Exploratorio de Datos (EDA)

El Análisis Exploratorio de Datos (AED), conocido como Exploratory Data Analysis (EDA) en inglés, es un enfoque crucial en el análisis de datos que busca comprender y resumir las características principales de un conjunto de datos

El EDA es una etapa crítica en cualquier proyecto de análisis de datos o de ciencia de datos. Proporciona una comprensión profunda de los datos, lo que es esencial para tomar decisiones informadas sobre el procesamiento y modelado de los datos

Introducción al EDA

OBJETIVOS

- **Comprender la estructura del conjunto de datos**: Identificar las variables y sus tipos (numéricas, categóricas, fechas, etc.).
- **Resumir las características principales**: Usar estadísticas descriptivas y visualizaciones.
- **Detectar anomalías y valores faltantes**: Identificar y tratar outliers y valores ausentes.
- **Descubrir patrones y relaciones**: Identificar correlaciones y tendencias entre variables.
- **Preparar los datos para el modelado**: Realizar transformaciones y limpieza de datos.

Introducción al EDA

PASOS

1. Carga y visualización inicial de los datos

Importar datos desde diferentes fuentes / Visualizar una muestra de los datos para familiarizarse con ellos.

2. Detección y corrección de valores faltantes

Identificar la presencia de valores faltantes. / Decidir cómo manejarlos (eliminación, imputación, etc.)

3. Estadísticas descriptivas

Calcular medidas de tendencia central / Calcular medidas de dispersión / Obtener resúmenes estadísticos generales.

4. Detección de valores atípicos

Usando el Rango Intercuartil (IQR) / Usando Diagramas de Caja (Boxplots) / Usando el Test de Grubbs

5. Visualización de datos

Histogramas y diagramas de densidad / Diagramas de cajas (boxplots) / Diagramas de dispersión (scatter plots) / Gráficos de barras / Matrices de correlación

6. Análisis de correlación

Calcular y visualizar matrices de correlación. / Identificar relaciones lineales y no lineales entre variables.

7. Filtrado y limpieza de datos

Filtrar datos irrelevantes o fuera de rango. / Corregir errores tipográficos e inconsistencias

8. Transformación y enriquecimiento de datos

Aplicar transformaciones logarítmicas, de raíz cuadrada, estandarización y normalización. / Crear variables derivadas si es necesario.

Introducción al EDA

HERRAMIENTAS

- **R y Python**: Lenguajes de programación muy utilizados para análisis de datos y muy efectivos para realizar EDA.
- **Bibliotecas de R**: dplyr, tidyr, ggplot2, corrplot, data.table.
- **Bibliotecas de Python**: pandas, numpy, matplotlib, seaborn, scipy.

Introducción al EDA

RESUMEN

El **EDA** es el primer paso clave en un análisis de datos. Su objetivo es **comprender qué hay en el conjunto de datos**, **detectar errores o valores atípicos**, **explorar relaciones entre variables** y **preparar los datos para el modelado posterior**. No se trata solo de ver los datos, sino de analizarlos críticamente.

Comprensión del conjunto de datos

Alcance

- Aquí se identifican los tipos de variables (numéricas, categóricas), sus nombres, si hay valores faltantes, etc.

```r
# Cargar datos
df <- read.csv("titanic.csv")

dim(df)             # Cantidad de observaciones y variables
colnames(df)        # Nombres de las variables
head(df)            # Ver las primeras filas
str(df)             # Ver la estructura de los datos
summary(df)         # Estadísticas fundamentales de cada variable
class(df$Age)       # Consulta la clase lógica de una variable
unique(df$Pclass)   # Obtener dominio de valores de una variables

# Estadisticas valores nulos (NA)
colSums(is.na(df))  # Cantidad valores NA
colMeans(is.na(df)) # Media valores NA

# Estadisticas valores faltantes (nulos y vacios)
sapply(df, function(x) sum(is.na(x) | !nzchar(x)))    # Cantidad valores faltantes
sapply(df, function(x) mean(is.na(x) | !nzchar(x)))   # Media valores faltantes

# Inspeccionar observaciones con valores faltantes
df[is.na(df$Embarked) | !nzchar(df$Embarked),]
```

Comprensión del conjunto de datos

Tratamiento de valores faltantes

- Buscar variables que contengan valores NULOS o cadenas vacías
- Determinar la proporción y cantidad de valores faltantes
- Inspeccionar los valores faltantes
- Determinar el tratamiento a aplicar para dichos valores:
  - Eliminación de filas
  - Eliminación de columnas
  - Imputación de valores nulos
  - Relleno adelante o atrás

Comprensión del conjunto de datos

Tratamiento de valores faltantes

```r
library(dplyr)

# Opción 1: Eliminación de filas
df1 <- df %>% filter(!is.na(Embarked) & nzchar(Embarked))

# Opción 2: Eliminación de columnas
df2 <- df %>% select(-Cabin)

# Opción 3: Imputación de valors nulos (media, mediana, u otro)
# Ej: asignar la mediana del resto de observaciones
mediana_Age <- median(df$Age,na.rm = TRUE)
df3 <- df %>% mutate(Age = ifelse(is.na(Age),mediana_Age,Age))

# Opción 4: Imputación con valore múltiples
# Ej: asignar la media de la edad según la clase a la que pertenece
df4 <- df %>%
  mutate (Age = case_when(
    is.na(Age) & Pclass == 1 ~ mean(Age[Pclass == 1],na.rm = TRUE),
    is.na(Age) & Pclass == 2 ~ mean(Age[Pclass == 2],na.rm = TRUE),
    is.na(Age) & Pclass == 3 ~ mean(Age[Pclass == 3],na.rm = TRUE),
    TRUE ~ Age
  ))
```

Comprensión del conjunto de datos

Identificación de tipos de variables

- Como parte del EDA es muy importante identificar los tipos de variables del dataset, en el contexto estadístico. Que pueden ser:
  - **Numéricas/cuantitativas**: discretas o continuas
  - **Categóricas**: nominales u ordinales
  - **Bimodales**: Que pueden ser tratadas como numéricas o categóricas dependiendo del tipo de análisis a realizar.
  - **Variables de texto**: no relevantes desde el contexto estadístico
- En función del tipo de variables sabremos como tratarlas en el contexto de EDA, y para otros tipos de análisis.

Comprensión del conjunto de datos

Identificación de tipos de variables

```r
names(df)
sapply(df, function(x) mode(x))
str(df)
unique(df$Parch)
unique(df$Ticket)
unique(df$Cabin)
unique(df$Embarked)

# TIPOS DE VARIABLES EN DATASET TITANIC

# Numericas/cuantitativas:
#                   Age, Fare
# Categóricas:
#                   Survived, Pclass, Sex, Embarked
# Bimodales:
#                   SibSp, Parch
# Variables texto:
#                   PassengerId, Name, Ticket, Cabin
```

Análisis univariado V. Categóricas

Sobrevista

- Se analiza una sola variable categórica para ver cómo se distribuyen sus categorías
- **Importante:** En R, las variables categóricas deben convertirse en **factores** antes de usarlas en gráficos o análisis estadísticos:
  - Los factores permiten que R entienda que la variable representa categorías, no texto libre.
  - Mejora la interpretación en modelos estadísticos y visualizaciones.
  - Permite controles precisos sobre el orden y agrupamiento de los niveles.
  - Si una variable no se va a usar en modelos de análisis y visualización, no es necesario factorizar

Análisis univariado V. Categóricas

¿Cómo factorizar variables categóricas?

```r
str(df)
```

```
$ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
$ Survived   : int  0 1 1 1 0 0 0 0 1 1 ...
$ Pclass     : int  3 1 3 1 3 3 1 3 3 2 ...
$ Name       : chr  "Braund, Mr. Owen Harris"
Laina" "Futrelle, Mrs. Jacques Heath (Lily May
$ Sex        : chr  "male" "female" "female"
$ Age        : num  22 38 26 35 35 NA 54 2 27
$ SibSp      : int  1 1 0 1 0 0 0 3 0 1 ...
$ Parch      : int  0 0 0 0 0 0 0 1 2 0 ...
$ Ticket     : chr  "A/5 21171" "PC 17599" "S
```

```r
# Convertir variables categóricas a factores
df$Sex <- as.factor(df$Sex)
df$Pclass <- as.factor(df$Pclass)

str(df)
sapply(df, function(x) mode(x))

# o

df <- df %>%
  mutate(Sex = as.factor(Sex),
         Pclass = as.factor(Pclass))
```

```
$ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
$ Survived   : int  0 1 1 1 0 0 0 0 1 1 ...
$ Pclass     : Factor w/ 3 levels "1","2","3": 3 1 3 1
$ Name       : chr  "Braund, Mr. Owen Harris" "Cumings,
Laina" "Futrelle, Mrs. Jacques Heath (Lily May Peel)" ...
$ Sex        : Factor w/ 2 levels "female","male": 2 1
$ Age        : num  22 38 26 35 35 NA 54 2 27 14 ...
$ SibSp      : int  1 1 0 1 0 0 0 3 0 1 ...
$ Parch      : int  0 0 0 0 0 0 0 1 2 0 ...
$ Ticket     : chr  "A/5 21171" "PC 17599" "STON/O2. 31
```

Análisis univariado V. Categóricas

Herramientas

- **Tablas de frecuencia**: muestran el conteo de cada categoría.
- **Gráficos de barras**: ayudan a visualizar la distribución de variables categóricas
- **Gráficos de barras apiladas**: para ver la relación entre dos variables categóricas.
- **Diagramas de caja (combinado con una variable numérica)**: permite comparar distribuciones de una variable numérica a través de las categorías de una variable categórica
- **Tablas de contingencia**: muestra la relación entre 2 variables categóricas.
- **Gráficos de Mosaico**: útiles para visualizar tablas de contingencia

Análisis univariado V. Categóricas

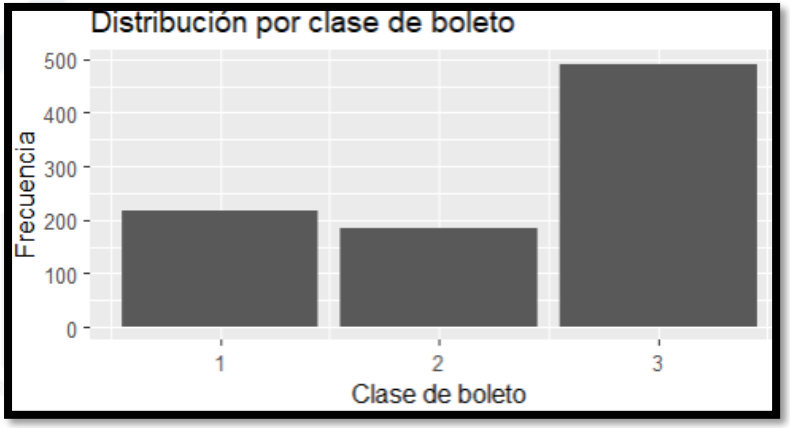
Ejemplo

```
82  # TABLAS DE FRECUENCIA
83
84  table(df$Pclass)
85
86
87  table(df$Sex)
88
89
90  prop.table(table(df$Pclass))
91
92
93  # GRAFICO DE BARRAS
94
95  install.packages("ggplot2")
96  library(ggplot2)
97
98  ggplot(df, aes(x = Pclass)) +
99    geom_bar() +
100   labs(title = "Distribución por clase de boleto",
101        x = "Clase de boleto",
102        y = "Frecuencia")
```

```
  1   2   3
216 184 491
```

```
female   male
   314    577
```

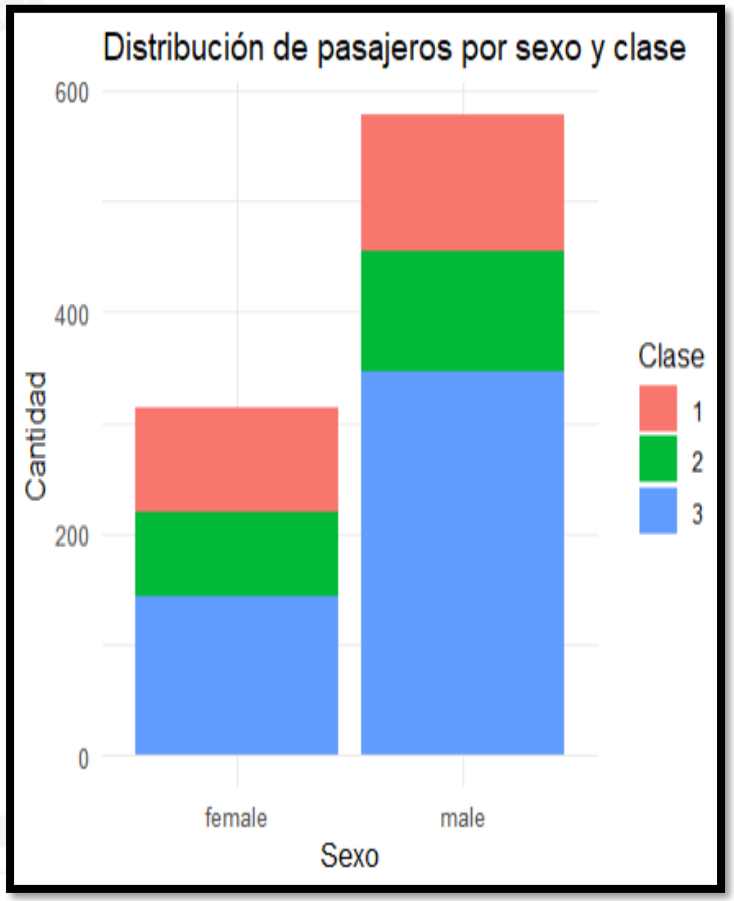
```
        1         2         3
0.2424242 0.2065095 0.5510662
```



Análisis univariado V. Categóricas

Ejemplo

```
124  # GRÁFICO DE BARRAS APILADAS
125
126  ggplot(df, aes(x = Sex, fill = Pclass)) +
127    geom_bar() +
128    labs(
129      title = "Distribución de pasajeros por sexo y clase",
130      x = "Sexo",
131      y = "Cantidad",
132      fill = "Clase"
133    ) +
134    theme_minimal()
```

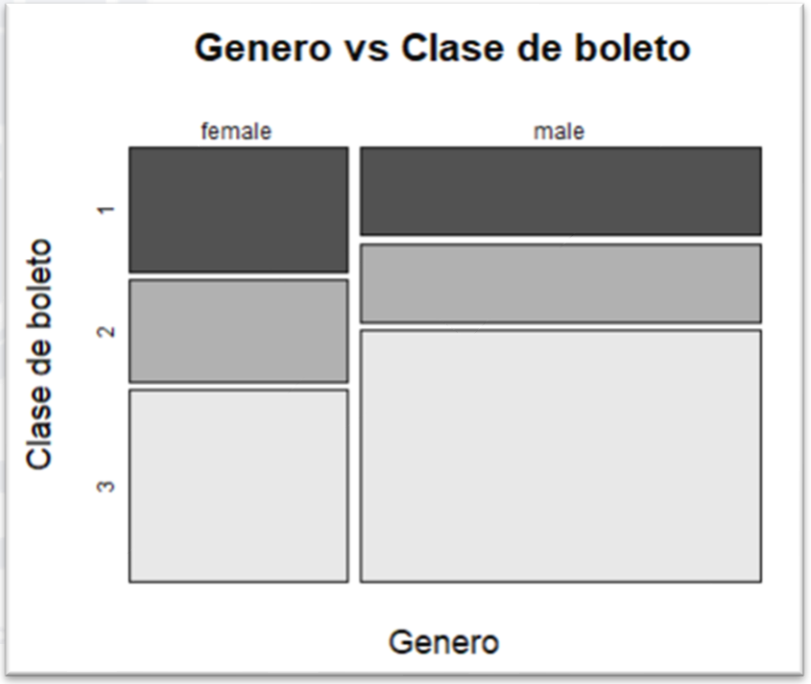


Análisis univariado V. Categóricas

Ejemplo

```r
# TABLAS DE CONTINGENCIA

table(df$Sex, df$Pclass)



# GRAFICO DE MOSAICO

tabla_contingencia <- table(df$Sex, df$Pclass)

mosaicplot(tabla_contingencia,
           color = TRUE,
           main = "Genero vs Clase de boleto",
           xlab = "Genero",
           ylab = "Clase de boleto")
```

```
        1   2   3
female 94  76 144
male  122 108 347
```



Análisis univariado V. Numéricas

Sobrevista y herramientas

- Se estudia una variable numérica para ver su comportamiento
- Para EDA de variables numéricas, se puede usar:
  - **Medidas de tendencia central y dispersión**: media, mediana, moda, desviación típica, varianza, rango
  - **Histogramas**: que muestra la distribución de una variable numérica
  - **Diagrama de dispersión (Scatter Plot)**: para mostrar la relación entre dos variables numéricas.
  - **Matriz de Correlación**: muestra las relaciones entre todas las variables numéricas del conjunto de datos.
  - **Diagramas de caja (bloxplots)**: para visualizar la dispersión, la mediana, y detectar outliers.
  - **Análisis de outliers**: obtenidos mediante consultas directas o utilizando boxplots

Análisis univariado V. Numéricas

Medidas de tendencia central y de dispersión

- **Media**: Promedio aritmético de un conjunto de valores
- **Mediana:** valor central de un conjunto de datos ordenados
- **Moda**: valor que aparece con mayor frecuencia en un conjunto de datos.
- **Desviación típica**: mide la dispersión de un conjunto de datos en relación con su media. Indica cuánto se desvían, en promedio, los valores respecto a la media
- **Varianza**: media de los cuadrados de las desviaciones respecto a la media
- **Rango**: la diferencia entre el valor máximo y el valor mínimo de un conjunto de datos.
- **Rango intercuartil (IQR)**: la diferencia entre el tercer cuartil (Q3) y el primer cuartil (Q1). Mide la dispersión del 50% central de los datos

Análisis univariado V. Numéricas

Ejemplo

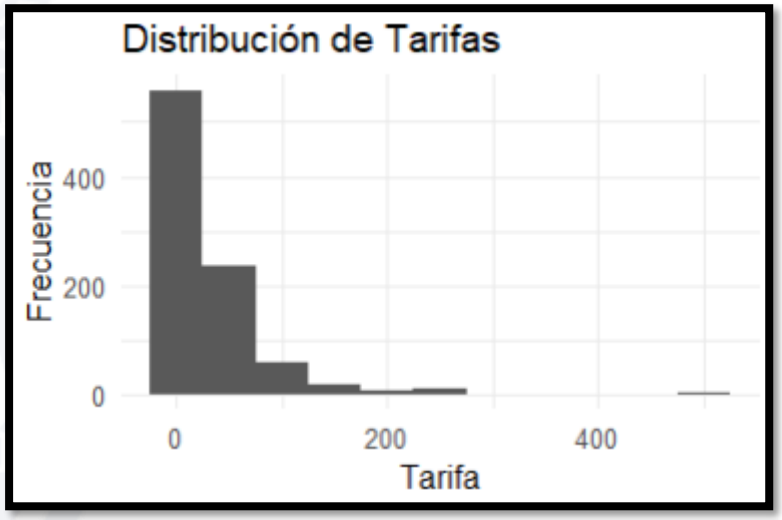
```r
# MEDIDAS DE TENDENCIA CENTRAL Y DISPERSIÓN

mean(df$Age, na.rm = TRUE)                                  # Media

median(df$Age, na.rm = TRUE)                                # Mediana

sd(df$Age, na.rm = TRUE)                                    # Desviación estándar

var(df$Age, na.rm = TRUE)                                   # Varianza

max(df$Age, na.rm = TRUE) - min(df$Age, na.rm = TRUE)       # Rango

diff(range(df$Age, na.rm = TRUE))                           # Rango

summary(df$Age)                                             # Resumen estadístico
```

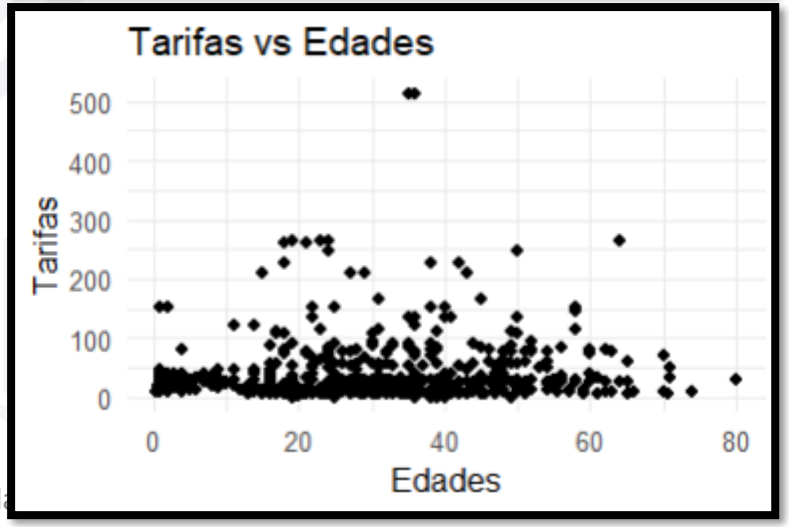
```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
   0.42   20.12   28.00   29.70   38.00   80.00     177
```

Análisis univariado V. Numéricas

Ejemplo

```
174  # HISTOGRAMAS
175
176  library(ggplot2)
177
178  ggplot(df, aes(x=Fare)) +
179    geom_histogram(binwidth = 50) +
180    labs(title = "Distribución de Tarifas",
181         x = "Tarifa",
182         y = "Frecuencia") +
183    theme_minimal()
184
185
186  # DIAGRAMAS DE DISPERSIÓN
187
188  ggplot(df, aes(x = Age, y = Fare)) +
189    geom_point() +
190    labs(title = "Tarifas vs Edades",
191         x = "Edades",
192         y = "Tarifas") +
193    theme_minimal()
```



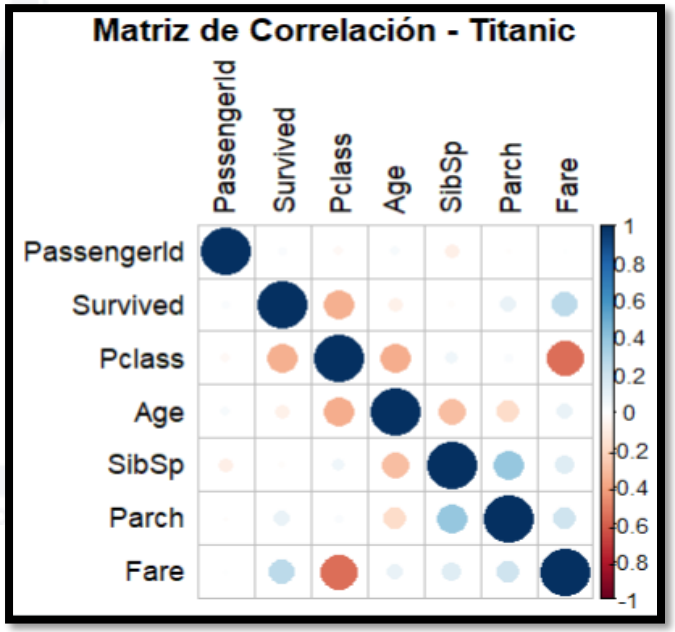


Análisis univariado V. Numéricas

Ejemplo

```r
# MATRIZ DE CORRELACIÓN

install.packages("corrplot")
library(corrplot)
library(dplyr)

# Seleccionar solo variables numéricas
df_num <- df %>%
  select(where(is.numeric))

# Ver estructura de variables numéricas seleccionadas
str(df_num)

# Calcular la matriz de correlación (omitiendo NA)
cor_matrix <- cor(df_num, use = "complete.obs")

# Mostrar la matriz en consola
print(cor_matrix)

# Graficar la matriz de correlación
corrplot(cor_matrix, method = "circle",
         tl.col = "black",
         title = "Matriz de Correlación - Titanic",
         mar = c(0,0,2,0))
```

```
             PassengerId     Survived       Pclass
PassengerId   1.00000000  0.02934016 -0.03534911  0.036
Survived      0.02934016  1.00000000 -0.35965268 -0.077
Pclass       -0.03534911 -0.35965268  1.00000000 -0.369
Age           0.03684720 -0.07722109 -0.36922602  1.000
SibSp        -0.08239772 -0.01735836  0.06724737 -0.308
Parch        -0.01161741  0.09331701  0.02568307 -0.189
Fare          0.00959178  0.26818862 -0.55418247  0.096
```



Análisis univariado V. Numéricas

Análisis de outliers (valores atípicos)

- Es muy importante identificar y tratar los outliers adecuadamente para evitar que influyan negativamente en los resultados de un análisis.
- Le método comúnmente usado detectar outliers está basado en el Rango Intercuartil (IQR), tomando como base los límites de dicho IQR (Q1, Q3).
- Para detectar valores atípicos, buscamos aquellos muy alejados del "centro" del conjunto de datos. Usamos el IQR para definir qué tan lejos es "demasiado lejos".
- Los valores que se encuentran por debajo de $Q1-1.5\times IQR$ o por encima de $Q3+1.5\times IQR$ son considerados outliers.

Análisis univariado V. Numéricas

Detección de outliers

```r
# Mostrar cuartiles
quantile(df$Fare, na.rm = TRUE)

# Calcular IQR
Q1 <- quantile(df$Fare,0.25, na.rm = TRUE)
Q3 <- quantile(df$Fare,0.75, na.rm = TRUE)
IQR <- Q3 - Q1

# Calcular límites
limite_inf <- Q1 - 1.5 * IQR
limite_sup <- Q3 + 1.5 * IQR

# Detectar outliers usando los límites inferior y superior
outliers <- df$Fare[df$Fare < limite_inf | df$Fare > limite_sup]
print(outliers)

# Cantidad absoluta y relativa de outliers
sum(df$Fare < limite_inf | df$Fare > limite_sup)
mean(df$Fare < limite_inf | df$Fare > limite_sup)

# Observaciones con valores atípicos
df_out <- df %>% filter(Fare < limite_inf | Fare > limite_sup)

# Visualización con boxplot
boxplot(df$Fare, main = "Boxplot de Fare",
        ylab = "Precio del pasaje")
boxplot.stats(df$Fare)$out
```

Análisis univariado V. Numéricas

Detección de outliers

```r
# Mostrar cuartiles
quantile(df$Fare, na.rm = TRUE)

# Calcular IQR
Q1 <- quantile(df$Fare,0.25, na.rm = TRUE)
Q3 <- quantile(df$Fare,0.75, na.rm = TRUE)
IQR <- Q3 - Q1

# Calcular límites
limite_inf <- Q1 - 1.5 * IQR
limite_sup <- Q3 + 1.5 * IQR

# Detectar outliers usando los límites inferior y superior
outliers <- df$Fare[df$Fare < limite_inf | df$Fare > limite_sup]
print(outliers)

# Cantidad absoluta y relativa de outliers
sum(df$Fare < limite_inf | df$Fare > limite_sup)
mean(df$Fare < limite_inf | df$Fare > limite_sup)

# Observaciones con valores atípicos
df_out <- df %>% filter(Fare < limite_inf | Fare > limite_sup)

# Visualización con boxplot
boxplot(df$Fare, main = "Boxplot de Fare",
        ylab = "Precio del pasaje")
boxplot.stats(df$Fare)$out
```

```
     0%     25%     50%     75%    100%
 0.0000  7.9104 14.4542 31.0000 512.3292
```

23.09

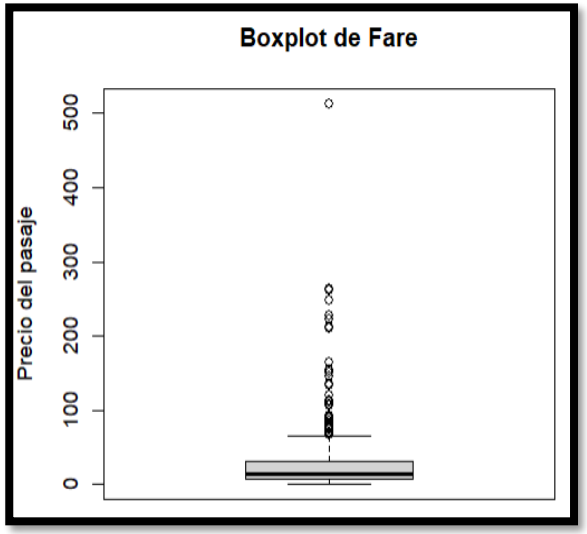
-26.72

65.63

```
  [1]  71.2833 263.0000 146.
 [16]  69.5500  69.5500 146.
 [31]  77.9583  78.8500  91.
 [46] 133.6500  66.6000 134.
 [61] 263.0000  81.8583  89.
 [76] 227.5250  79.6500 110.
 [91]  76.7292 211.3375 110.
[106]  79.2000  69.5500 120.
```

116

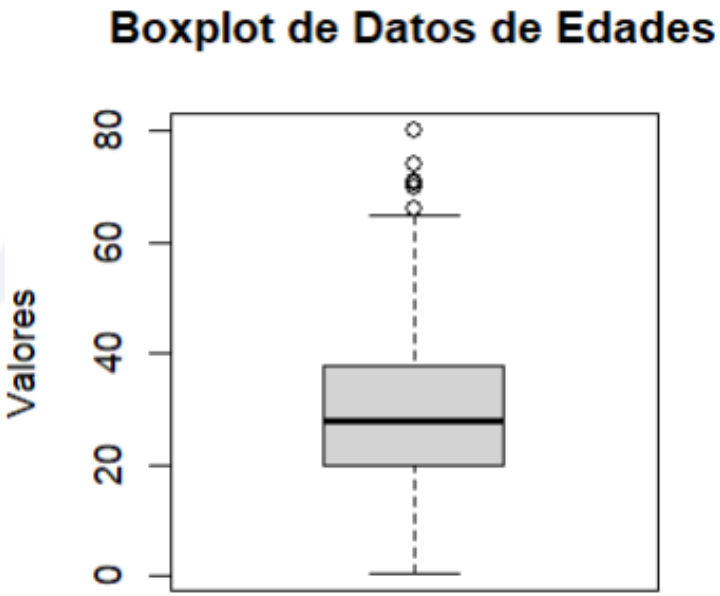
0.1302



Análisis univariado V. Numéricas

Detección de outliers

- Se puede visualizar y analizar outliers usando diagramas de caja (boxplots)
- Un boxplot muestra la distribución de una variable y destaca los valores atípicos.
- La caja es el IQR
- La línea dentro de la caja es la mediana
- Los bigotes se extienden a los limites superior e inferior de valores normales.
- Los puntos fuera de la los bigotes son considerados outliers



Análisis univariado V. Numéricas

Tratamiento de outliers

- Los valores atípicos puede incidir significativamente en los resultados del análisis. valores
- No hay una regla fija del % de outliers que son significativos para un análisis. Pero en general en distribuciones normales, si más de 5% de lo datos son atípicos, pueden incidir significativamente en los resultados.
- Opciones para tratar valores atípicos:
  - **Imputación de outliers** (con la media, mediana, limite superior, limite inferior)
  - **Transformación de datos** (logarítmo natural, raíz cuadrada)
  - **Eliminación**
- Si los outliers corresponden a errores en los datos, se los debe eliminar.
- Si lo que se busca es analizar comportamiento atípicos (Ej lavado de dinero), entonces los outliers son relevantes para el análisis

Análisis univariado V. Numéricas

Tratamiento de outliers

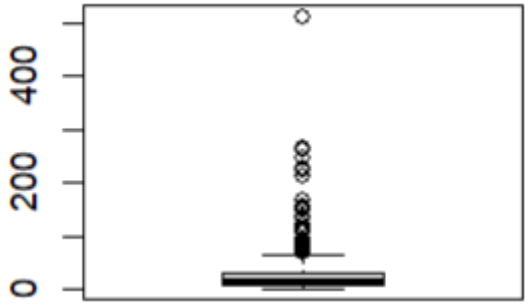
```
259  # SIN TRATAMIENTO
260
261  sd(df$Fare)
262  quantile(df$Fare)
263  boxplot(df$Fare)
264
265  # OPCIÓN 1: Imputar atípicos con mediana
266
267  df_op1 <- df %>%
268    mutate(Fare = ifelse(Fare < limite_inf | Fare > limite_sup,
269                         median(Fare),Fare))
270  sd(df_op1$Fare)
271  quantile(df_op1$Fare)
272  boxplot(df_op1$Fare)
273
274  # OPCIÓN 2: Imputar atipicos con limites
275
276  df_op2 <- df %>%
277    mutate(Fare = ifelse(Fare < limite_inf, 0,
278                         ifelse(Fare > limite_sup,
279                                limite_sup, Fare)))
280  sd(df_op2$Fare)
281  quantile(df_op2$Fare)
282  boxplot(df_op2$Fare)
```

Análisis univariado V. Numéricas

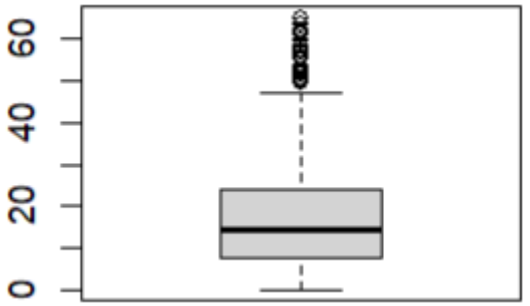
Tratamiento de outliers

```r
# SIN TRATAMIENTO

sd(df$Fare)
quantile(df$Fare)
boxplot(df$Fare)

# OPCIÓN 1: Imputar atípicos con mediana

df_op1 <- df %>%
  mutate(Fare = ifelse(Fare < limite_inf | Fare > limite_sup,
                       median(Fare),Fare))
sd(df_op1$Fare)
quantile(df_op1$Fare)
boxplot(df_op1$Fare)

# OPCIÓN 2: Imputar atipicos con limites

df_op2 <- df %>%
  mutate(Fare = ifelse(Fare < limite_inf, 0,
                       ifelse(Fare > limite_sup,
                              limite_sup, Fare)))
sd(df_op2$Fare)
quantile(df_op2$Fare)
boxplot(df_op2$Fare)
```

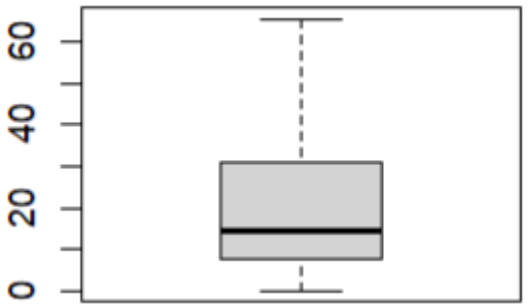
sd = 49.69343



sd = 12.71302



sd = 20.48162



Análisis univariado V. Numéricas

Tratamiento de outliers

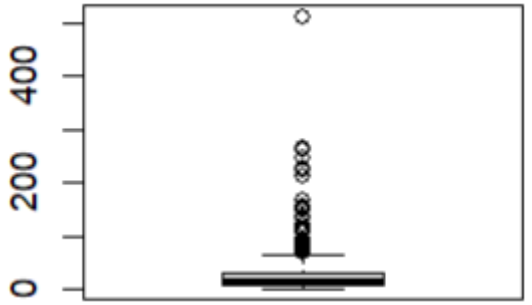
```r
# SIN TRATAMIENTO

sd(df$Fare)
quantile(df$Fare)
boxplot(df$Fare)

# OPCIÓN 3: Transforma datos por Raiz cuadrada

df_op3 <- df %>%
  mutate(Fare = sqrt(Fare))
sd(df_op3$Fare)
quantile(df_op3$Fare)
boxplot(df_op3$Fare)


# OPCIÓN 4: Transformar datos por logaritmo

df_op4 <- df %>%
  mutate(Fare = log(Fare + 1))
sd(df_op4$Fare)
quantile(df_op4$Fare)
boxplot(df_op4$Fare)
```

Análisis univariado V. Numéricas

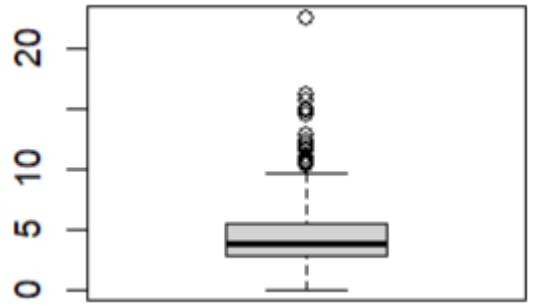
Tratamiento de outliers

```
285  # SIN TRATAMIENTO
286
287  sd(df$Fare)
288  quantile(df$Fare)
289  boxplot(df$Fare)
290
291  # OPCIÓN 3: Transforma datos por Raiz cuadrada
292
293  df_op3 <- df %>%
294    mutate(Fare = sqrt(Fare))
295  sd(df_op3$Fare)
296  quantile(df_op3$Fare)
297  boxplot(df_op3$Fare)
298
299
300  # OPCIÓN 4: Transformar datos por logaritmo
301
302  df_op4 <- df %>%
303    mutate(Fare = log(Fare + 1))
304  sd(df_op4$Fare)
305  quantile(df_op4$Fare)
306  boxplot(df_op4$Fare)
```

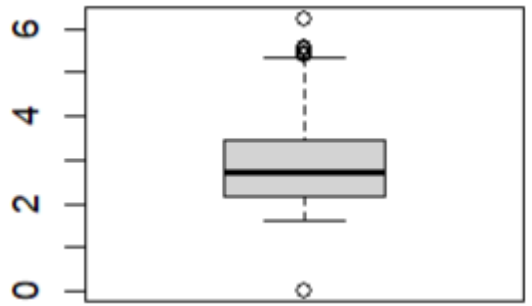
sd = 49.69343



sd = 2.946119



sd = 0.969048

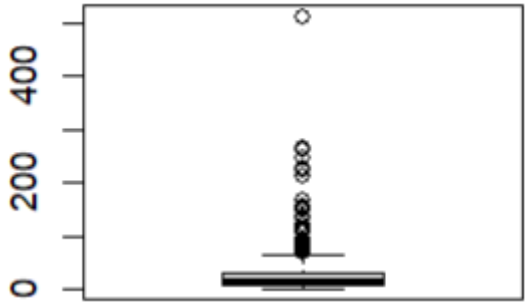


Análisis univariado V. Numéricas

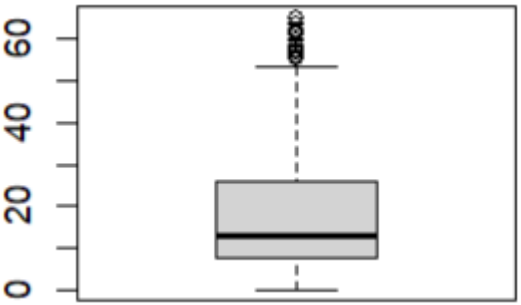
Tratamiento de outliers

```
309 # SIN TRATAMIENTO
310
311 sd(df$Fare)
312 quantile(df$Fare)
313 boxplot(df$Fare)
314
315 # OPCIÓN 5: Eliminar atípicos
316
317 df_op5 <- df %>%
318   filter(df$Fare >= limite_inf & df$Fare <= limite_sup)
319 sd(df_op5$Fare)
320 quantile(df_op5$Fare)
321 boxplot(df_op5$Fare)
```

sd = 49.69343



sd = 13.57809



Análisis univariado V. Numéricas

Tratamiento de outliers

- **Imputación con media o mediana**: Cuando outlier es el resultado de error o ruido, no de un valor real importante.
  - **Mediana**: si la distribución está muy sesgada
  - **Media**: si la distribución es simétrica o no hay sesgo fuerte
- **Imputación con límites**: Queremos reducir el impacto de los outliers sin eliminarlos ni cambiarlos totalmente. El valor extremo es válido, pero demasiado influyente. Queremos mantener cierta dispersión sin alterar la estructura de los datos.
- **Transformación con raíz cuadrada**: La variable tiene sesgo moderado. Los valores extremos son válidos, pero necesitamos reducir la variabilidad. No funciona con negativos.
- **Transformación con logaritmo natural**: La variable tiene alto sesgo (muy dispersa, con valores extremos grandes). El rango es amplio y necesitamos comprimir la escala. No funciona con valores menores o iguales a 0.