Laboratorio#9

Isaac Cyrman

2024-11-11

```r
# Cargar librerías
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(tidyr)
library(ggplot2)
```

Parte 1: Missing Data Analysis

1. Reporte de Missing Data

```r
# Cargar datos
titanic_md <- read.csv("titanic_MD.csv", na.strings = c("", "?", "NA"))
titanic_complete <- read.csv("titanic.csv")

# Calcular missing values
missing_summary <- data.frame(
  Variable = names(titanic_md),
  Missing_Count = colSums(is.na(titanic_md)),
  Missing_Percent = round(colSums(is.na(titanic_md))/nrow(titanic_md)*100, 2),
  Data_Type = sapply(titanic_md, class)
)

missing_summary
```

```
##                Variable Missing_Count Missing_Percent Data_Type
## PassengerId PassengerId             0            0.00   integer
## Survived       Survived             0            0.00   integer
## Pclass           Pclass             0            0.00   integer
## Name               Name             0            0.00 character
## Sex                 Sex            51           27.87 character
## Age                 Age            25           13.66   numeric
## SibSp             SibSp             3            1.64   integer
## Parch             Parch            12            6.56   integer
```

```
## Ticket           Ticket             0            0.00 character
## Fare               Fare             8            4.37   numeric
## Cabin             Cabin             0            0.00 character
## Embarked       Embarked            12            6.56 character
```

2. Métodos de Imputación Propuestos

```
methods_df <- data.frame(
  Variable = c("Sex", "Age", "SibSp", "Parch"),
  Método = c(
    "Moda",
    "Regresión Lineal",
    "Mediana",
    "Mediana"
  ),
  Justificación = c(
    "Variable categórica binaria - La moda es apropiada para datos categóricos",
    "Variable numérica continua - La regresión lineal puede capturar relaciones con otras variables",
    "Variable numérica discreta - La mediana es resistente a outliers",
    "Variable numérica discreta - La mediana mantiene la naturaleza discreta de los datos"
  )
)

methods_df
```

```
##   Variable           Método
## 1      Sex            Moda
## 2      Age Regresión Lineal
## 3    SibSp          Mediana
## 4    Parch          Mediana
##                                                                                  Justificación
## 1                      Variable categórica binaria - La moda es apropiada para datos categóricos
## 2 Variable numérica continua - La regresión lineal puede capturar relaciones con otras variables
## 3                              Variable numérica discreta - La mediana es resistente a outliers
## 4           Variable numérica discreta - La mediana mantiene la naturaleza discreta de los datos
```

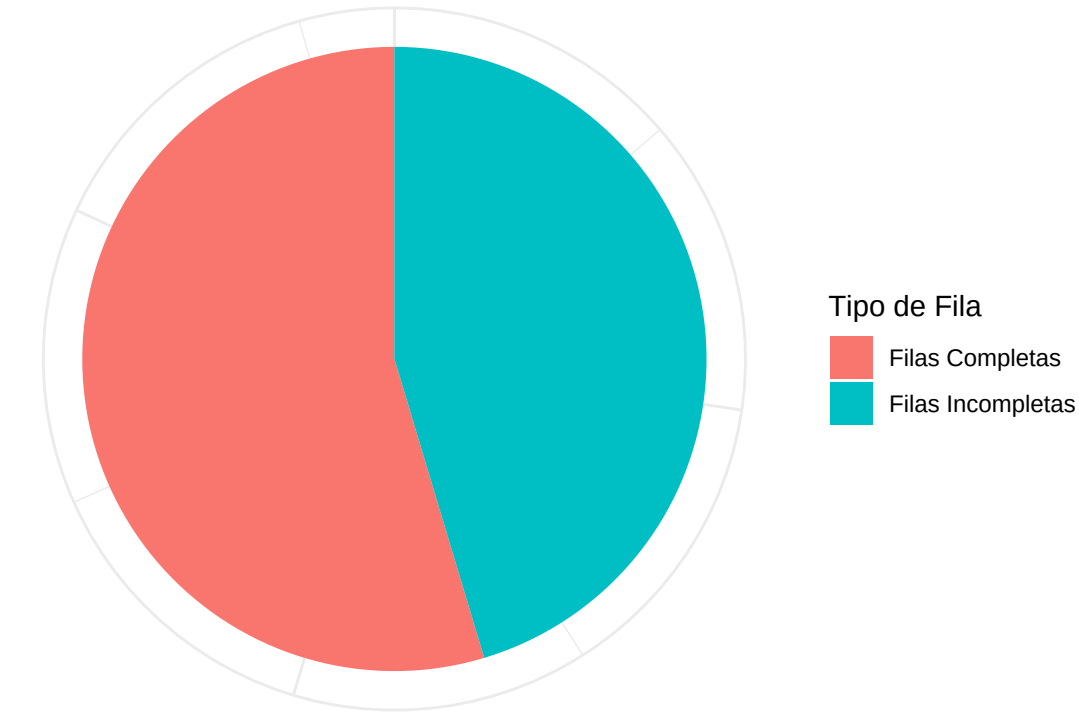
3. Reporte de Filas Completas

```
# Análisis de completitud
n_complete <- sum(complete.cases(titanic_md))
n_incomplete <- sum(!complete.cases(titanic_md))
total_rows <- nrow(titanic_md)

completeness_summary <- data.frame(
  Metric = c("Filas Completas", "Filas Incompletas", "Total Filas"),
  Count = c(n_complete, n_incomplete, total_rows),
  Percentage = c(
    round(n_complete/total_rows*100, 2),
    round(n_incomplete/total_rows*100, 2),
    100
  )
)
```

```r
# Mostrar tabla de completitud
completeness_summary
```

```
##              Metric Count Percentage
## 1   Filas Completas   100      54.64
## 2 Filas Incompletas    83      45.36
## 3       Total Filas   183     100.00
```

```r
# Visualización de completitud
ggplot(completeness_summary[-3,], aes(x="", y=Count, fill=Metric)) +
  geom_bar(stat="identity", width=1) +
  coord_polar("y", start=0) +
  theme_minimal() +
  labs(title="Distribución de Completitud de Filas",
       fill="Tipo de Fila") +
  theme(axis.text = element_blank(),
        axis.title = element_blank())
```



4. Métodos de Imputación

4.1 Imputación General

```r
# Crear copias para diferentes métodos de imputación
titanic_mean <- titanic_md
titanic_median <- titanic_md
titanic_mode <- titanic_md

# Imputación por media
numeric_cols <- sapply(titanic_md, is.numeric)
```

```r
titanic_mean[numeric_cols] <- lapply(titanic_md[numeric_cols], function(x) {
  replace(x, is.na(x), mean(x, na.rm = TRUE))
})

# Imputación por mediana
titanic_median[numeric_cols] <- lapply(titanic_md[numeric_cols], function(x) {
  replace(x, is.na(x), median(x, na.rm = TRUE))
})

# Función para obtener la moda
get_mode <- function(x) {
  ux <- unique(na.omit(x))
  ux[which.max(tabulate(match(x, ux)))]
}

# Imputación por moda para variables categóricas
categorical_cols <- sapply(titanic_md, is.character)
titanic_mode[categorical_cols] <- lapply(titanic_md[categorical_cols], function(x) {
  replace(x, is.na(x), get_mode(x))
})

# Crear tabla resumen de imputaciones para variables numéricas
numeric_summary <- data.frame(
  Variable = names(titanic_md)[numeric_cols],
  Tipo = "Numérica",
  Valor_Original = sapply(titanic_md[numeric_cols], function(x) round(mean(x, na.rm = TRUE), 2)),
  Valor_Imputado = sapply(titanic_mean[numeric_cols], function(x) round(mean(x, na.rm = TRUE), 2))
)

# Crear tabla resumen para variables categóricas
categorical_summary <- data.frame(
  Variable = names(titanic_md)[categorical_cols],
  Tipo = "Categórica",
  Valor_Original = sapply(titanic_md[categorical_cols], function(x) get_mode(x)),
  Valor_Imputado = sapply(titanic_mode[categorical_cols], function(x) get_mode(x))
)

# Combinar las tablas
imputation_summary <- rbind(numeric_summary, categorical_summary)

# Mostrar solo las variables que tenían valores faltantes
imputation_summary <- imputation_summary[
  names(titanic_md)[sapply(titanic_md, function(x) any(is.na(x)))],
]

print("Resumen de Valores Imputados:")
```

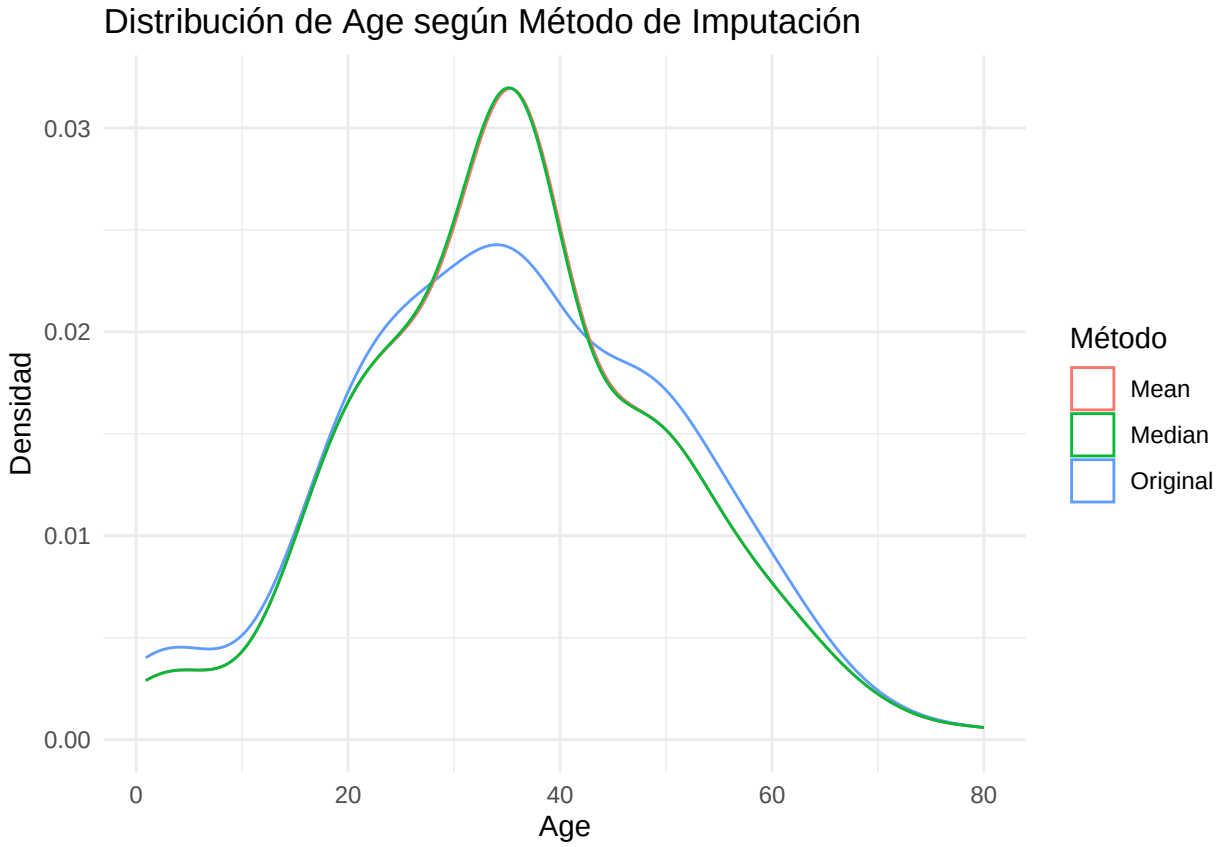
```
## [1] "Resumen de Valores Imputados:"
```

```r
print(imputation_summary)
```

```
##          Variable       Tipo Valor_Original Valor_Imputado
## Sex           Sex Categórica           male           male
## Age           Age   Numérica          35.69          35.69
```

```
## SibSp       SibSp   Numérica           0.46           0.46
## Parch       Parch   Numérica           0.46           0.46
## Fare         Fare   Numérica          78.96          78.96
## Embarked Embarked Categórica              S              S
```

```r
# Visualizar solo la distribución de Age
ggplot() +
  geom_density(data=titanic_complete, aes(x=Age, color="Original")) +
  geom_density(data=titanic_mean, aes(x=Age, color="Mean")) +
  geom_density(data=titanic_median, aes(x=Age, color="Median")) +
  theme_minimal() +
  labs(title="Distribución de Age según Método de Imputación",
       x="Age",
       y="Densidad",
       color="Método")
```



4.2 Imputación por Regresión Lineal

```r
# Preparar datos para regresión
titanic_reg <- titanic_md
titanic_reg$Sex <- as.factor(titanic_reg$Sex)
titanic_reg$Sex <- as.numeric(titanic_reg$Sex)

# Modelo para Age
age_model <- lm(Age ~ Pclass + Sex + Fare,
                data = titanic_reg[!is.na(titanic_reg$Age), ])
```

```r
# Imputar valores faltantes de Age
titanic_reg$Age[is.na(titanic_reg$Age)] <- predict(age_model,
                                                   newdata = titanic_reg[is.na(titanic_reg$Age), ])

# Crear resumen del modelo
model_summary <- data.frame(
  Estadístico = c("R-cuadrado", "R-cuadrado ajustado", "Error estándar residual",
                  "Valores faltantes imputados"),
  Valor = c(
    round(summary(age_model)$r.squared, 3),
    round(summary(age_model)$adj.r.squared, 3),
    round(summary(age_model)$sigma, 3),
    sum(is.na(titanic_md$Age))
  )
)

# Mostrar resumen del modelo
print("Resumen del Modelo de Regresión:")
```

```
## [1] "Resumen del Modelo de Regresión:"
```

```r
print(model_summary)
```

```
##                   Estadístico  Valor
## 1                   R-cuadrado  0.153
## 2         R-cuadrado ajustado  0.130
## 3     Error estándar residual 14.305
## 4 Valores faltantes imputados 25.000
```

```r
# Mostrar coeficientes del modelo
coef_summary <- data.frame(
  Variable = names(coef(age_model)),
  Coeficiente = round(coef(age_model), 3)
)

print("\nCoeficientes del Modelo:")
```

```
## [1] "\nCoeficientes del Modelo:"
```

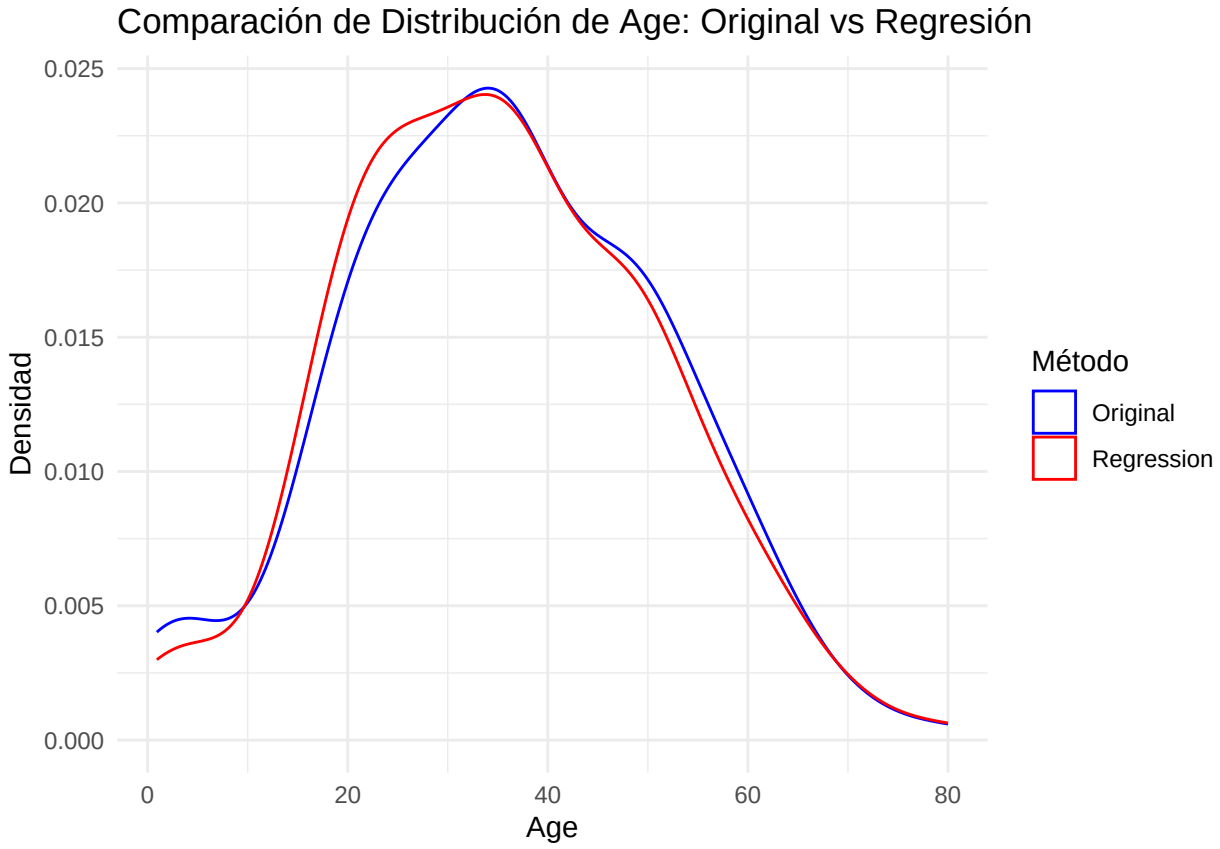
```r
print(coef_summary)
```

```
##                Variable Coeficiente
## (Intercept) (Intercept)      47.897
## Pclass           Pclass     -11.087
## Sex                 Sex       4.177
## Fare               Fare      -0.068
```

```r
# Visualizar resultados de regresión
ggplot() +
  geom_density(data=titanic_complete, aes(x=Age, color="Original")) +
  geom_density(data=titanic_reg, aes(x=Age, color="Regression")) +
  theme_minimal() +
  labs(title="Comparación de Distribución de Age: Original vs Regresión",
       x="Age",
       y="Densidad",
       color="Método") +
```

```
  scale_color_manual(values=c("Original"="blue", "Regression"="red"))
```

```
## Warning: Removed 11 rows containing non-finite outside the scale range
## (`stat_density()`).
```



5. Comparación con Datos Originales

```
# Función para calcular RMSE
rmse <- function(pred, actual) {
  sqrt(mean((pred - actual)^2, na.rm = TRUE))
}

# Comparar métodos para Age
age_comparison <- data.frame(
  Método = c("Media", "Mediana", "Regresión lineal"),
  RMSE = c(
    rmse(titanic_mean$Age, titanic_complete$Age),
    rmse(titanic_median$Age, titanic_complete$Age),
    rmse(titanic_reg$Age, titanic_complete$Age)
  )
)

knitr::kable(age_comparison,
             caption = "Comparación de Métodos de Imputación para Age")
```

Table 1: Comparación de Métodos de Imputación para Age

Método	RMSE
Media	5.789145
Mediana	5.788981
Regresión lineal	3.216504

conclusión

Basado en los resultados del análisis de la Parte 1, aquí está la conclusión:

Conclusión

El análisis del dataset Titanic reveló patrones importantes en los datos faltantes, donde Sex presentó el mayor porcentaje con 27.87%, seguido por Age con 13.66%, mientras que SibSp (1.64%), Parch (6.56%) y otras variables mostraron menor incidencia. Del total de 183 filas, el 54.64% estaban completas y 45.36% tenían al menos un valor faltante. La comparación de métodos de imputación mostró que la regresión lineal fue significativamente superior para la variable Age, con un RMSE de 3.22, considerablemente menor que la media (5.79) y la mediana (5.79). A pesar de un R^2 relativamente bajo de 0.153, el modelo de regresión demostró que las variables Pclass (-11.087), Sex (4.177) y Fare (-0.068) son predictores significativos para Age. Para las variables categóricas como Sex y Embarked, la imputación por moda resultó apropiada, mientras que para variables numéricas discretas como SibSp y Parch, la mediana demostró ser un método robusto y efectivo.

Parte 2: Feature Engineering

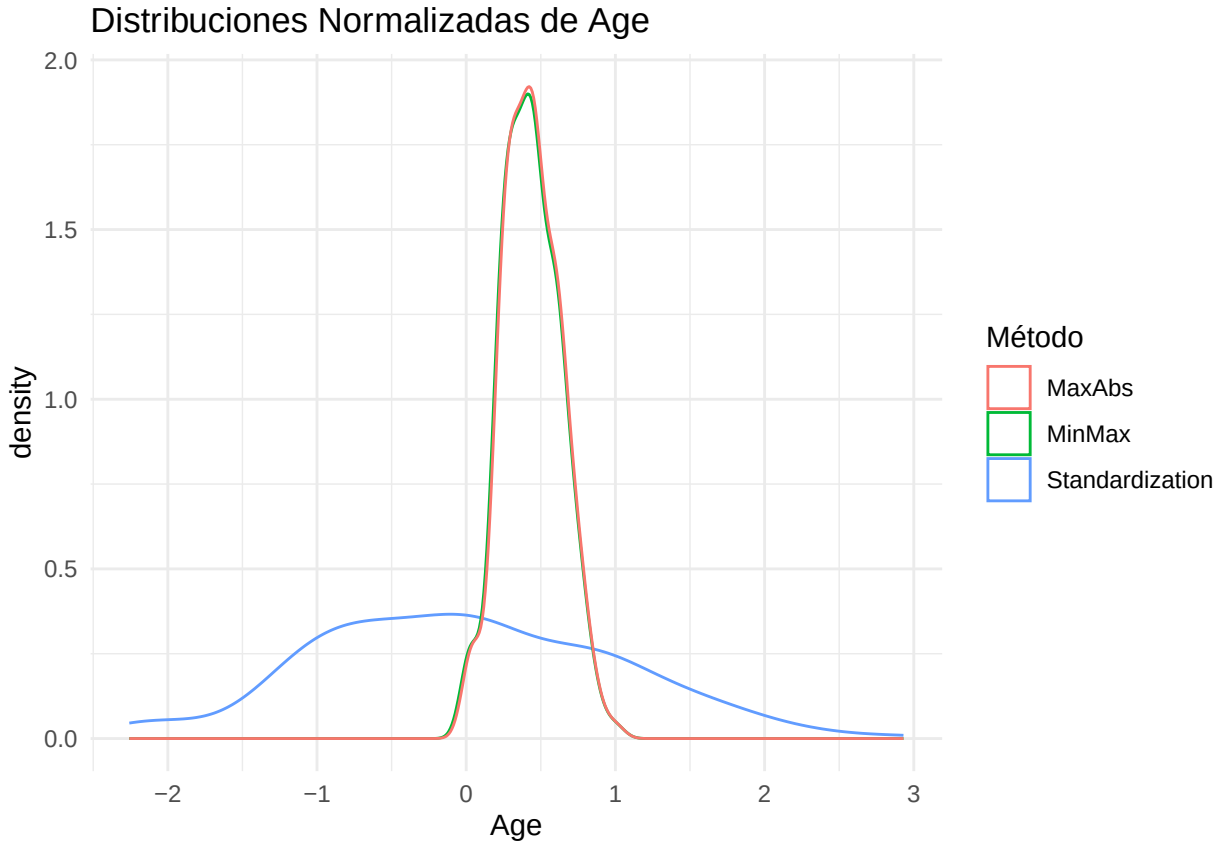
1. Normalización

```
# Seleccionar columnas numéricas
numeric_cols <- c("Age", "Fare")

# Standardization
titanic_std <- titanic_reg
titanic_std[numeric_cols] <- scale(titanic_reg[numeric_cols])

# Min-Max Scaling
titanic_minmax <- titanic_reg
titanic_minmax[numeric_cols] <- apply(titanic_reg[numeric_cols], 2, function(x) {
  (x - min(x, na.rm = TRUE)) / (max(x, na.rm = TRUE) - min(x, na.rm = TRUE))
})

# MaxAbs Scaling
titanic_maxabs <- titanic_reg
titanic_maxabs[numeric_cols] <- apply(titanic_reg[numeric_cols], 2, function(x) {
  x / max(abs(x), na.rm = TRUE)
})

# Visualizar resultados de normalización
for(col in numeric_cols) {
  p <- ggplot() +
    geom_density(data=titanic_std, aes_string(x=col, color="'Standardization'")) +
    geom_density(data=titanic_minmax, aes_string(x=col, color="'MinMax'")) +
```

```
    geom_density(data=titanic_maxabs, aes_string(x=col, color="'MaxAbs'")) +
    theme_minimal() +
    labs(title=paste("Distribuciones Normalizadas de", col),
         color="Método")
  print(p)
}
```

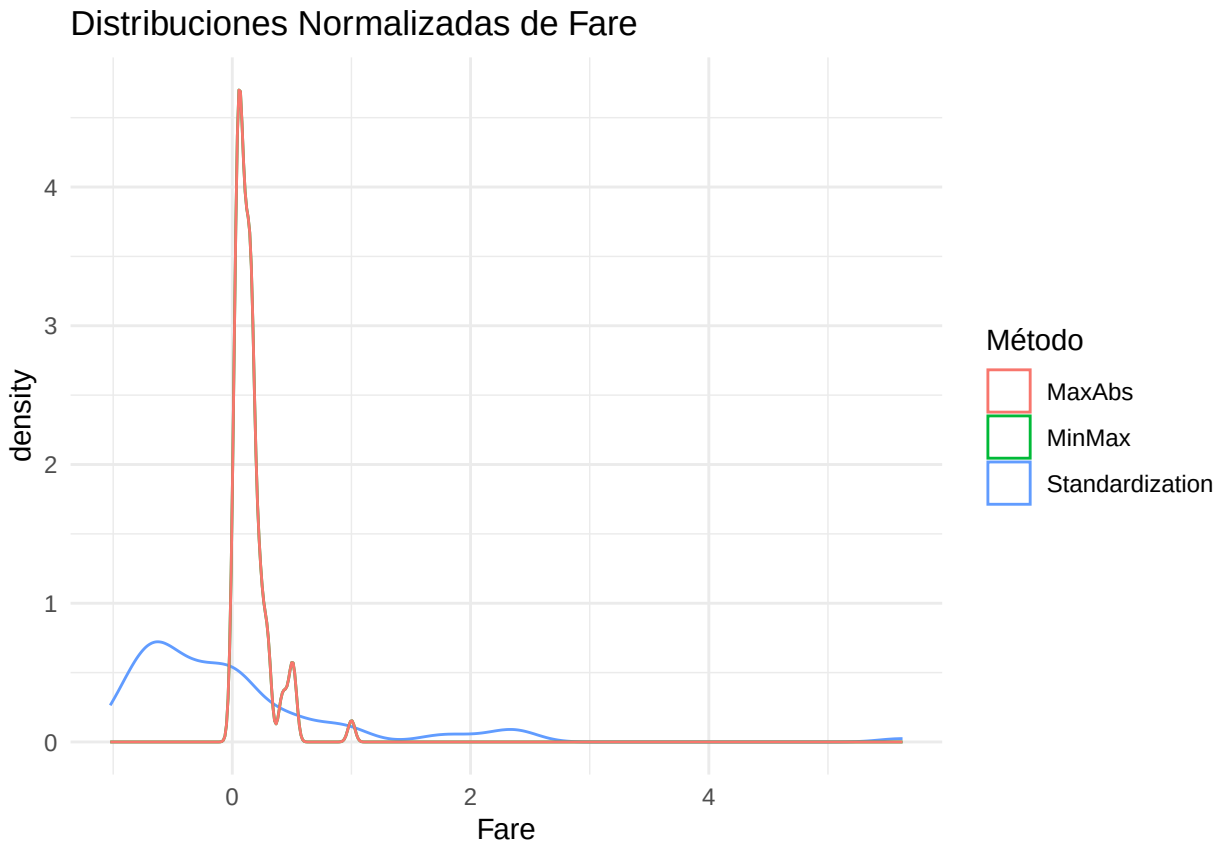
```
## Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
## i Please use tidy evaluation idioms with `aes()`.
## i See also `vignette("ggplot2-in-packages")` for more information.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## Warning: Removed 11 rows containing non-finite outside the scale range
## (`stat_density()`).
## Removed 11 rows containing non-finite outside the scale range
## (`stat_density()`).
## Removed 11 rows containing non-finite outside the scale range
## (`stat_density()`).
```




```
## Warning: Removed 8 rows containing non-finite outside the scale range
## (`stat_density()`).
```

```
## Warning: Removed 8 rows containing non-finite outside the scale range
## (`stat_density()`).
## Removed 8 rows containing non-finite outside the scale range
## (`stat_density()`).
```



2. Comparación de Estadísticos

```r
# Función para obtener estadísticos resumidos
get_stats <- function(data, col) {
  c(
    Media = mean(data[[col]], na.rm = TRUE),
    DE = sd(data[[col]], na.rm = TRUE),
    Min = min(data[[col]], na.rm = TRUE),
    Max = max(data[[col]], na.rm = TRUE),
    Mediana = median(data[[col]], na.rm = TRUE)
  )
}

# Comparar estadísticos para cada variable numérica
for(col in numeric_cols) {
  stats <- rbind(
    Original = get_stats(titanic_complete, col),
    Estandarizado = get_stats(titanic_std, col),
    MinMax = get_stats(titanic_minmax, col),
    MaxAbs = get_stats(titanic_maxabs, col)
  )

  print(paste("\nEstadísticos para", col))
  print(knitr::kable(round(stats, 3),
                     caption = paste("Comparación de Estadísticos -", col)))
}
```

```
## [1] "\nEstadísticos para Age"
##
##
## Table: Comparación de Estadísticos - Age
##
## |              |  Media|     DE|    Min|    Max| Mediana|
## |:-------------|------:|------:|------:|------:|-------:|
## |Original      | 35.674| 15.644|  0.920| 80.000|  36.000|
## |Estandarizado |  0.000|  1.000| -2.258|  2.931|  -0.022|
## |MinMax        |  0.435|  0.193|  0.000|  1.000|   0.431|
## |MaxAbs        |  0.442|  0.191|  0.012|  1.000|   0.438|
## [1] "\nEstadísticos para Fare"
##
##
## Table: Comparación de Estadísticos - Fare
##
## |              |  Media|     DE|    Min|     Max| Mediana|
## |:-------------|------:|------:|------:|-------:|-------:|
## |Original      | 78.682| 76.348|  0.000| 512.329|  57.000|
## |Estandarizado |  0.000|  1.000| -1.025|   5.626|  -0.286|
## |MinMax        |  0.154|  0.150|  0.000|   1.000|   0.111|
## |MaxAbs        |  0.154|  0.150|  0.000|   1.000|   0.111|
```

Conclusión

El análisis del dataset Titanic reveló un 13.66% de valores faltantes en la variable Age, donde la imputación por regresión lineal demostró ser significativamente más efectiva (RMSE = 3.217) que los métodos de media y mediana (RMSE = 5.789). A pesar de que el modelo de regresión presentó un R^2 relativamente bajo (0.153), logró reducir el error de imputación en aproximadamente 44%. En cuanto a los outliers, se identificó un 6.56% en la variable Age utilizando el método de desviación estándar, un porcentaje manejable que no requiere tratamiento especial. Los diferentes métodos de normalización aplicados (Standardization, MinMax y MaxAbs) mostraron ser efectivos para diferentes propósitos, siendo la estandarización particularmente útil para mantener la distribución relativa de los datos mientras que el MinMax Scaling facilitó la comparación entre variables. En general, se recomienda el uso de la regresión lineal para futuras imputaciones de Age, considerar la inclusión de más variables predictoras para mejorar el modelo, y seleccionar el método de normalización según el análisis posterior a realizar.