

Missing data and Feature Engineering

Laboratorio 8, Raul Castellanos 20180052

#Parte 1

```
library(dplyr)
library(readr)
library(tidyr)
library(stringr)
library(modeest)
library(DataExplorer)
library(DT)
library(gdata)
timd <- read_csv("titanic_MD.csv")
ti <- read_csv("titanic.csv")
```

Inciso 1: Reporte detallado de missing data para todas las columnas

```
summary(timd)
```

```
## PassengerId      Survived      Pclass         Name
## Min.   : 2.0      Min.   :0.0000   Min.   :1.000   Length:183
## 1st Qu.:263.5     1st Qu.:0.0000   1st Qu.:1.000   Class  :character
## Median :457.0     Median :1.0000   Median :1.000   Mode   :character
## Mean   :455.4     Mean   :0.6721   Mean    :1.191
## 3rd Qu.:676.0     3rd Qu.:1.0000   3rd Qu.:1.000
## Max.   :890.0     Max.   :1.0000   Max.    :3.000
##
##      Sex          Age          SibSp          Parch
## Length:183      Min.   : 0.92   Min.   :0.0000   Min.   :0.000
## Class :character 1st Qu.:24.00   1st Qu.:0.0000   1st Qu.:0.000
## Mode  :character Median :35.50   Median :0.0000   Median :0.000
##                      Mean  :35.69   Mean  :0.4611   Mean  :0.462
##                      3rd Qu.:48.00   3rd Qu.:1.0000   3rd Qu.:1.000
##                      Max.   :80.00   Max.   :3.0000   Max.   :4.000
##                      NA's   :25      NA's    :3        NA's    :12
##      Ticket      Fare          Cabin          Embarked
## Length:183      Min.   : 0.00   Length:183      Length:183
## Class :character 1st Qu.: 29.70   Class :character Class :character
## Mode  :character Median : 56.93   Mode  :character Mode  :character
##                      Mean   : 78.96
##                      3rd Qu.: 90.54
##                      Max.   :512.33
##                      NA's   :8
```

Columnas con missing data

- Sex

- Age
- SibSp
- Parch
- Fare
- Embarked

Inciso 2: Para cada columna especificar que tipo de modelo se utilizará y qué valores se le darán a todos los missing values

- Para la columna de Sex: filling values ya que en la columna de name están las abreviaciones como: Mr, Mrs, Miss, Master, Sir, Mme, Dr, Major, Countess
- Para la columna de Age: Predicción con Regresión lineal con la relación de las columnas Age y Survived
- Para la columna de SibSp: llenar los valores NA por imputación de la media
- Para la columna de Parch: llenar los valores NA por imputación de la moda
- Para la columna de Fare: llenar los valores NA por imputación de la media
- Para la columna de Embarked: filling values top to down

Inciso 3: Reporte de qué filas están completas

```
timd$Sex2 <- ifelse(timd$Sex == "female", 0,
                    ifelse(timd$Sex == "male", 1, NA))
paste0("Hay ", nrow(na.omit(timd)), " filas completas")
```

```
## [1] "Hay 100 filas completas"
```

Inciso 4: Utilizar los siguientes métodos para cada columna que contiene missing values:

- * Pairwise deletion
- * Imputación general (media, moda y mediana)
- * Imputación sectorizada
- * Modelo de regresión lineal simple
- * Eliminación de outliers: Standard deviation approach
- * Eliminación de outliers: Percentile approach

```
timd2 <- read_csv("titanic_MD.csv")
```

Funciones Generales

- Imputación, Media, Mediana y Moda

```
Imputacion_MMM <- function(df, columna){
  if(columna == 6){
    df$media <- NA
    df$mediana <- NA
    df$moda <- NA
    df$media <- ifelse(is.na(df$Age), round(mean(df$Age, na.rm = TRUE),0), df$Age)
    df$mediana <- ifelse(is.na(df$Age), round(median(df$Age, na.rm = TRUE),0), df$Age)
```

```

df$moda <- ifelse(is.na(df$Age), round(mfv1(df$Age, na_rm = TRUE),0), df$Age)
MMM <- cbind.data.frame(df$media, df$mediana, df$moda)
}
else if(columna == 7){
  df$media <- NA
  df$mediana <- NA
  df$moda <- NA
  df$media <- ifelse(is.na(df$SibSp), round(mean(df$SibSp, na_rm = TRUE),0), df$SibSp)
  df$mediana <- ifelse(is.na(df$SibSp), round(median(df$SibSp, na_rm = TRUE),0), df$SibSp)
  df$moda <- ifelse(is.na(df$SibSp), round(mfv1(df$SibSp, na_rm = TRUE),0), df$SibSp)
  MMM <- cbind.data.frame(df$media, df$mediana, df$moda)
}
else if(columna == 8){
  df$media <- NA
  df$mediana <- NA
  df$moda <- NA
  df$media <- ifelse(is.na(df$Parch), round(mean(df$Parch, na_rm = TRUE),0), df$Parch)
  df$mediana <- ifelse(is.na(df$Parch), round(median(df$Parch, na_rm = TRUE),0), df$Parch)
  df$moda <- ifelse(is.na(df$Parch), round(mfv1(df$Parch, na_rm = TRUE),0), df$Parch)
  MMM <- cbind.data.frame(df$media, df$mediana, df$moda)
}
else if(columna == 10){
  df$media <- NA
  df$mediana <- NA
  df$moda <- NA
  df$media <- ifelse(is.na(df$Fare), round(mean(df$Fare, na_rm = TRUE),0), df$Fare)
  df$mediana <- ifelse(is.na(df$Fare), round(median(df$Fare, na_rm = TRUE),0), df$Fare)
  df$moda <- ifelse(is.na(df$Fare), round(mfv1(df$Fare, na_rm = TRUE),0), df$Fare)
  MMM <- cbind.data.frame(df$media, df$mediana, df$moda)
}
return(as.data.frame(MMM))
}

```

- Imputacion sectorizada

```

Imputacion_Sectorizada <- function(df, columna){
  if(columna == 6){
    df$IS <- NA
    g <- df %>% filter(!is.na(df$Age)) %>% group_by(Age) %>% summarise(freq = n(), .groups = 'drop') %>%
    a <- as.numeric(g[1,1])
    df$IS <- ifelse(is.na(df$Age), a, df$Age)
    ISM <- df$IS
  }
  else if(columna == 7){
    df$IS <- NA
    g <- df %>% filter(!is.na(df$SibSp)) %>% group_by(SibSp) %>% summarise(freq = n(), .groups = 'drop') %>%
    a <- as.numeric(g[1,1])
    df$IS <- ifelse(is.na(df$SibSp), a, df$SibSp)
    ISM <- df$IS
  }
  else if(columna == 8){
    df$IS <- NA
    g <- df %>% filter(!is.na(df$Parch)) %>% group_by(Parch) %>% summarise(freq = n(), .groups = 'drop') %>%
    a <- as.numeric(g[1,1])
  }
}

```

```

df$IS <- ifelse(is.na(df$Parch), a, df$Parch)
ISM <- df$IS
}
else if(columna == 10){
df$IS <- NA
g <- df %>% filter(!is.na(df$Fare)) %>% group_by(Fare) %>% summarise(freq = n(), .groups = 'drop')
a <- as.numeric(g[1,1])
df$IS <- ifelse(is.na(df$Fare), a, df$Fare)
ISM <- df$IS
}
return(as.data.frame(ISM))
}

```

- Regresion Lineal

```

RegressionLineal <- function(df, columna, columna2){
  if(columna == 6 & columna2 == 2){
    df$RLlineal <- NA
    a <- lm(Age ~ Survived, df)
    df$RLlineal <- ifelse(is.na(df$Age), round((a$coefficients[1] + a$coefficients[2]*df$Survived),0), df$Age)
    RL <- df$RLlineal
  }
  else if(columna == 6 & columna2 == 3){
    df$RLlineal <- NA
    a <- lm(Age ~ Pclass, df)
    df$RLlineal <- ifelse(is.na(df$Age), round((a$coefficients[1] + a$coefficients[2]*df$Pclass),0), df$Age)
    RL <- df$RLlineal
  }
  else if(columna == 7 & columna2 == 2){
    df$RLlineal <- NA
    a <- lm(SibSp ~ Survived, df)
    df$RLlineal <- ifelse(is.na(df$SibSp), round((a$coefficients[1] + a$coefficients[2]*df$Survived),0), df$SibSp)
    RL <- df$RLlineal
  }
  else if(columna == 7 & columna2 == 3){
    df$RLlineal <- NA
    a <- lm(SibSp ~ Pclass, df)
    df$RLlineal <- ifelse(is.na(df$SibSp), round((a$coefficients[1] + a$coefficients[2]*df$Pclass),0), df$SibSp)
    RL <- df$RLlineal
  }
  else if(columna == 8 & columna2 == 2){
    df$RLlineal <- NA
    a <- lm(Parch ~ Survived, df)
    df$RLlineal <- ifelse(is.na(df$Parch), round((a$coefficients[1] + a$coefficients[2]*df$Survived),0), df$Parch)
    RL <- df$RLlineal
  }
  else if(columna == 8 & columna2 == 3){
    df$RLlineal <- NA
    a <- lm(Parch ~ Pclass, df)
    df$RLlineal <- ifelse(is.na(df$Parch), round((a$coefficients[1] + a$coefficients[2]*df$Pclass),0), df$Parch)
    RL <- df$RLlineal
  }
  else if(columna == 10 & columna2 == 2){
    df$RLlineal <- NA

```

```

a <- lm(Fare ~ Survived, df)
df$RLlineal <- ifelse(is.na(df$Fare), round((a$coefficients[1] + a$coefficients[2]*df$Survived),0),
RL <- df$RLlineal
}
else if(columna == 10 & columna2 == 3){
df$RLlineal <- NA
a <- lm(Age ~ Fare, df)
df$RLlineal <- ifelse(is.na(df$Fare), round((a$coefficients[1] + a$coefficients[2]*df$Pclass),0), df$
RL <- df$RLlineal
}
return(as.data.frame(RL))
}

```

- Standard Deviation Approach

```

StandardDev_Approach <- function(df, columna){
if(columna == 6){
x <- 3
Lower <- mean(df$Age, na.rm = TRUE) - (sd(df$Age, na.rm = TRUE)*x)
Upper <- mean(df$Age, na.rm = TRUE) + (sd(df$Age, na.rm = TRUE)*x)
Sd <- df[(!is.na(df$Age)) & (df$Age >= Lower) & (df$Age <= Upper),]
SDA <- Sd$Age
}
else if(columna == 7){
x <- 3
Lower <- mean(df$SibSp, na.rm = TRUE) - (sd(df$SibSp, na.rm = TRUE)*x)
Upper <- mean(df$SibSp, na.rm = TRUE) + (sd(df$SibSp, na.rm = TRUE)*x)
Sd <- df[(!is.na(df$SibSp)) & (df$SibSp >= Lower) & (df$SibSp <= Upper),]
SDA <- Sd$SibSp
}
else if(columna == 8){
x <- 3
Lower <- mean(df$Parch, na.rm = TRUE) - (sd(df$Parch, na.rm = TRUE)*x)
Upper <- mean(df$Parch, na.rm = TRUE) + (sd(df$Parch, na.rm = TRUE)*x)
Sd <- df[(!is.na(df$Parch)) & (df$Parch >= Lower) & (df$Parch <= Upper),]
SDA <- Sd$Parch
}
else if(columna == 10){
x <- 3
Lower <- mean(df$Fare, na.rm = TRUE) - (sd(df$Fare, na.rm = TRUE)*x)
Upper <- mean(df$Fare, na.rm = TRUE) + (sd(df$Fare, na.rm = TRUE)*x)
Sd <- df[(!is.na(df$Fare)) & (df$Fare >= Lower) & (df$Fare <= Upper),]
SDA <- Sd$Fare
}
return(as.data.frame(SDA))
}

```

- Percentile Approach

```

Percentile_Approach <- function(df, columna){
if(columna == 6){
Lower <- quantile(df$Age, na.rm = TRUE, probs = 0.1)
Upper <- quantile(df$Age, na.rm = TRUE, probs = 0.9)
Pa <- df[(!is.na(df$Age)) & (df$Age >= Lower) & (df$Age <= Upper),]
PA <- Pa$Age
}

```

```

}
else if(columna == 7){
  Lower <- quantile(df$SibSp, na.rm = TRUE, probs = 0.1)
  Upper <- quantile(df$SibSp, na.rm = TRUE, probs = 0.9)
  Pa <- df[(!is.na(df$SibSp)) & (df$SibSp >= Lower) & (df$SibSp <= Upper),]
  PA <- Pa$SibSp
}
else if(columna == 8){
  Lower <- quantile(df$Parch, na.rm = TRUE, probs = 0.1)
  Upper <- quantile(df$Parch, na.rm = TRUE, probs = 0.9)
  Pa <- df[(!is.na(df$Parch)) & (df$Parch >= Lower) & (df$Parch <= Upper),]
  PA <- Pa$Parch
}
else if(columna == 10){
  Lower <- quantile(df$Fare, na.rm = TRUE, probs = 0.1)
  Upper <- quantile(df$Fare, na.rm = TRUE, probs = 0.9)
  Pa <- df[(!is.na(df$Fare)) & (df$Fare >= Lower) & (df$Fare <= Upper),]
  PA <- Pa$Fare
}
return(as.data.frame(PA))
}

```

Columna: Sex

- Los siguientes métodos no aplican para esta columna ya que esta columna tiene valores categóricos:
 - Pairwise deletion
 - Imutación por media y mediana
 - Regresión Lineal
 - Eliminación de outliers: Standard deviation approach
 - Eliminación de outliers: Percentile approach
- Imputación por moda

```

timd2$Sex2 <- ifelse(timd2$Sex == "female", 0,
                    ifelse(timd2$Sex == "male", 1, NA))
timd2$Sex <- ifelse(is.na(timd2$Sex2), mfv1(timd2$Sex2, na_rm = TRUE), timd2$Sex2)
timd2[1:10, 5]

```

```

## # A tibble: 10 x 1
##   Sex
##   <dbl>
## 1     1
## 2     0
## 3     1
## 4     0
## 5     0
## 6     1
## 7     1
## 8     1
## 9     0
## 10    1

```

```
# Cero = female y Uno = male
```

- Filling Values con informacion adicional (columna Name)

```
timd2$Sex <- ifelse(str_detect(timd2$Name, "Mrs|Miss|Mme|Mlle|Countess"), "female", "male")
Sex <- as.data.frame(timd2$Sex)
Sex[1:10,]
```

```
## [1] "female" "female" "male" "female" "female" "male" "male" "male"
## [9] "female" "male"
```

Columna: Age

- Deletion Pairwise

```
cor(timd2$Age, timd2$Survived, use = "pairwise.complete.obs")
```

```
## [1] -0.2577034
```

- Imputación Media, Mediana y Moda

```
as.data.frame(Imputacion_MMM(timd2, 6)[1:10,])
```

```
##      df$media df$mediana df$moda
## 1         38         38      38
## 2         35         35      35
## 3         54         54      54
## 4         36         36      24
## 5         58         58      58
## 6         34         34      34
## 7         36         36      24
## 8         19         19      19
## 9         49         49      49
## 10        65         65      65
```

- Imputación sectorizada

```
as.data.frame(Imputacion_Sectorizada(timd2, 6)[1:10,])
```

```
##      Imputacion_Sectorizada(timd2, 6)[1:10, ]
## 1                                         38
## 2                                         35
## 3                                         54
## 4                                         24
## 5                                         58
## 6                                         34
## 7                                         24
## 8                                         19
## 9                                         49
## 10                                        65
```

- Regresión Lineal

```
as.data.frame(RegresionLineal(timd2, 6, 3)[1:10,])
```

```
##      RegresionLineal(timd2, 6, 3)[1:10, ]
## 1                                         38
## 2                                         35
## 3                                         54
## 4                                         16
## 5                                         58
## 6                                         34
## 7                                         37
```

```
## 8 19
## 9 49
## 10 65
```

- Eliminación de outliers: Standard deviation approach

```
as.data.frame(StandardDev_Approach(timd2, 6)[1:10,])
```

```
## StandardDev_Approach(timd2, 6)[1:10, ]
## 1 38
## 2 35
## 3 54
## 4 58
## 5 34
## 6 19
## 7 49
## 8 65
## 9 45
## 10 29
```

- Eliminación de outliers: Percentile approach

```
as.data.frame(Percentile_Approach(timd2, 6)[1:10,])
```

```
## Percentile_Approach(timd2, 6)[1:10, ]
## 1 38
## 2 35
## 3 54
## 4 34
## 5 19
## 6 49
## 7 45
## 8 29
## 9 25
## 10 46
```

Columna: SibSp

- Deletion Pairwise

```
cor(timd2$SibSp, timd2$Pclass, use = "pairwise.complete.obs")
```

```
## [1] -0.102294
```

- Imputación Media, Mediana y Moda

```
as.data.frame(Imputacion_MMM(timd2, 7)[1:10,])
```

```
## df$media df$mediana df$moda
## 1 1 1 1
## 2 1 1 1
## 3 0 0 0
## 4 1 1 1
## 5 0 0 0
## 6 0 0 0
## 7 0 0 0
## 8 3 3 3
## 9 1 1 1
## 10 0 0 0
```


- Imputación sectorizada

```
as.data.frame(Imputacion_Sectorizada(timd2, 7)[1:10,])
```

```
##      Imputacion_Sectorizada(timd2, 7)[1:10, ]
## 1      1
## 2      1
## 3      0
## 4      1
## 5      0
## 6      0
## 7      0
## 8      3
## 9      1
## 10     0
```

- Regresión Lineal

```
as.data.frame(RegresionLineal(timd2, 7, 3)[1:10,])
```

```
##      RegresionLineal(timd2, 7, 3)[1:10, ]
## 1      1
## 2      1
## 3      0
## 4      1
## 5      0
## 6      0
## 7      0
## 8      3
## 9      1
## 10     0
```

- Eliminación de outliers: Standard deviation approach

```
as.data.frame(StandardDev_Approach(timd2, 7)[1:10,])
```

```
##      StandardDev_Approach(timd2, 7)[1:10, ]
## 1      1
## 2      1
## 3      0
## 4      1
## 5      0
## 6      0
## 7      1
## 8      0
## 9      1
## 10     0
```

- Eliminación de outliers: Percentile approach

```
as.data.frame(Percentile_Approach(timd2, 7)[1:10,])
```

```
##      Percentile_Approach(timd2, 7)[1:10, ]
## 1      1
## 2      1
## 3      0
## 4      1
## 5      0
```

```
## 6 0
## 7 1
## 8 0
## 9 1
## 10 0
```

Columna: Parch

- Deletion Pairwise

```
cor(timd2$Parch, timd2$Pclass, use = "pairwise.complete.obs")
```

```
## [1] 0.04196947
```

- Imputación Media, Mediana y Moda

```
as.data.frame(Imputacion_MMM(timd2, 8)[1:10,])
```

```
##      df$media df$mediana df$moda
## 1          0          0          0
## 2          0          0          0
## 3          0          0          0
## 4          0          0          0
## 5          0          0          0
## 6          0          0          0
## 7          0          0          0
## 8          2          2          2
## 9          0          0          0
## 10         1          1          1
```

- Imputación sectorizada

```
as.data.frame(Imputacion_Sectorizada(timd2, 8)[1:10,])
```

```
##      Imputacion_Sectorizada(timd2, 8)[1:10, ]
## 1          0
## 2          0
## 3          0
## 4          0
## 5          0
## 6          0
## 7          0
## 8          2
## 9          0
## 10         1
```

- Regresión Lineal

```
as.data.frame(RegresionLineal(timd2, 8, 3)[1:10,])
```

```
##      RegresionLineal(timd2, 8, 3)[1:10, ]
## 1          0
## 2          0
## 3          0
## 4          1
## 5          0
## 6          0
## 7          0
## 8          2
```

```
## 9 0
## 10 1
```

- Eliminación de outliers: Standard deviation approach

```
as.data.frame(StandardDev_Approach(timd2, 8)[1:10,])
```

```
## StandardDev_Approach(timd2, 8)[1:10, ]
## 1 0
## 2 0
## 3 0
## 4 0
## 5 0
## 6 0
## 7 2
## 8 1
## 9 0
## 10 0
```

- Eliminación de outliers: Percentile approach

```
as.data.frame(Percentile_Approach(timd2, 8)[1:10,])
```

```
## Percentile_Approach(timd2, 8)[1:10, ]
## 1 0
## 2 0
## 3 0
## 4 0
## 5 0
## 6 0
## 7 2
## 8 1
## 9 0
## 10 0
```

Columna: Fare

- Deletion Pairwise

```
cor(timd2$Fare, timd2$Pclass, use = "pairwise.complete.obs")
```

```
## [1] -0.3044376
```

- Imputación Media, Mediana y Moda

```
as.data.frame(Imputacion_MMM(timd2, 10)[1:10,])
```

```
## df$media df$mediana df$moda
## 1 71.2833 71.2833 71.2833
## 2 53.1000 53.1000 53.1000
## 3 51.8625 51.8625 51.8625
## 4 16.7000 16.7000 16.7000
## 5 26.5500 26.5500 26.5500
## 6 13.0000 13.0000 13.0000
## 7 35.5000 35.5000 35.5000
## 8 263.0000 263.0000 263.0000
## 9 76.7292 76.7292 76.7292
## 10 61.9792 61.9792 61.9792
```

- Imputación sectorizada

```
as.data.frame(Imputacion_Sectorizada(timd2, 10)[1:10,])
```

```
##      Imputacion_Sectorizada(timd2, 10)[1:10, ]
## 1      71.2833
## 2      53.1000
## 3      51.8625
## 4      16.7000
## 5      26.5500
## 6      13.0000
## 7      35.5000
## 8      263.0000
## 9      76.7292
## 10     61.9792
```

- Regresión Lineal

```
as.data.frame(RegresionLineal(timd2, 10, 3)[1:10,])
```

```
##      RegresionLineal(timd2, 10, 3)[1:10, ]
## 1      71.2833
## 2      53.1000
## 3      51.8625
## 4      16.7000
## 5      26.5500
## 6      13.0000
## 7      35.5000
## 8      263.0000
## 9      76.7292
## 10     61.9792
```

- Eliminación de outliers: Standard deviation approach

```
as.data.frame(StandardDev_Approach(timd2, 10)[1:10,])
```

```
##      StandardDev_Approach(timd2, 10)[1:10, ]
## 1      71.2833
## 2      53.1000
## 3      51.8625
## 4      16.7000
## 5      26.5500
## 6      13.0000
## 7      35.5000
## 8      263.0000
## 9      76.7292
## 10     61.9792
```

- Eliminación de outliers: Percentile approach

```
as.data.frame(Percentile_Approach(timd2, 10)[1:10,])
```

```
##      Percentile_Approach(timd2, 10)[1:10, ]
## 1      71.2833
## 2      53.1000
## 3      51.8625
## 4      16.7000
## 5      26.5500
```

```
## 6 35.5000
## 7 76.7292
## 8 61.9792
## 9 83.4750
## 10 61.1750
```

Columna: Embarked

- Los siguientes métodos no aplican para esta columna ya que esta columna tiene valores categóricos:
 - Pairwise deletion
 - Imutación por media y mediana
 - Regresión Lineal
 - Eliminación de outliers: Standard deviation approach
 - Eliminación de outliers: Percentile approach
- Imputación por moda

```
timd2$Embarked2 <- ifelse(timd2$Embarked == "C", 1,
                          ifelse(timd2$Embarked == "Q", 2,
                                ifelse(timd2$Embarked == "S", 3, NA)))
timd2$Embarked <- ifelse(is.na(timd2$Embarked), mfv1(timd2$Embarked2, na_rm = TRUE), timd2$Embarked)
timd2[1:10, 12]
```

```
## # A tibble: 10 x 1
##   Embarked
##   <chr>
## 1 C
## 2 S
## 3 S
## 4 S
## 5 S
## 6 S
## 7 S
## 8 S
## 9 C
## 10 C
```

- Filling Values Top to Down

```
timd2 <- timd2 %>% fill(Embarked)
timd2[1:10, 12]
```

```
## # A tibble: 10 x 1
##   Embarked
##   <chr>
## 1 C
## 2 S
## 3 S
## 4 S
## 5 S
## 6 S
## 7 S
## 8 S
## 9 C
## 10 C
```

Inciso 5: Al comparar los métodos del inciso 4 contra “titanic.csv”, ¿Qué método (para cada columna) se acerca más a la realidad y por qué?

Sex

```
timd2$Sex2 <- ifelse(timd2$Sex == "female", 0,
                    ifelse(timd2$Sex == "male", 1, NA))
timd2$Sex3 <- ifelse(is.na(timd2$Sex2), mfv1(timd2$Sex2, na_rm = TRUE), timd2$Sex2)
Sexo_Moda <- timd2 %>% select(Sex3)
Sexo_Moda$Sex3 <- ifelse(Sexo_Moda$Sex3 == 0, "female", "male")

timd2$Sex <- ifelse(str_detect(timd2$Name, "Mrs|Miss|Mme|Mlle|Countess"), "female", "male")
Sexo_FillValues <- timd2 %>% select(Sex)

Sex <- ti %>% select(Sex)

Sexo <- cbindX(Sex, Sexo_Moda, Sexo_FillValues)

colnames(Sexo) <- c("Original", "Imp_Moda", "Fill_Values")

Sexo[1:10,]
```

```
##      Original Imp_Moda Fill_Values
## 1      female      female      female
## 2      female      female      female
## 3       male       male       male
## 4      female      female      female
## 5      female      female      female
## 6       male       male       male
## 7       male       male       male
## 8       male       male       male
## 9      female      female      female
## 10      male       male       male
```

Podemos observar que los dos metodos devolvieron un resultado igual al de la data original, el que más se acerca a la realidad sería el de Filling Values, ya que con la información del nombre se puede obtener un resultado muy cercano a la realidad.

Age

```
paste("La correlación entre la edad y sobrevivir es de: ", cor(timd2$Age, timd2$Survived, use = "pairwise.complete.obs"))

## [1] "La correlación entre la edad y sobrevivir es de: -0.25770337396344"

Age_MediaMedianaModa <- Imputacion_MMM(timd2, 6)
Age_ImpSectorizada <- Imputacion_Sectorizada(timd2, 6)
Age_Regresion <- RegresionLineal(timd2, 6, 3)
Age_StDevApproach <- StandardDev_Approach(timd2, 6)
Age_Percentile <- Percentile_Approach(timd2, 6)
Age_Original <- ti %>% select(Age)

Age <- cbindX(Age_Original, Age_MediaMedianaModa, Age_ImpSectorizada, Age_Regresion,
              Age_StDevApproach, Age_Percentile)
colnames(Age) <- c("Original", "Media", "Mediana", "Moda", "Imp_SecModa", "R_Lineal", "Sd_App", "Perc_App")
Age[1:10,]
```

```
##      Original Media Mediana Moda Imp_SecModa R_Lineal Sd_App Perc_App
```

## 1	38	38	38	38	38	38	38	38
## 2	35	35	35	35	35	35	35	35
## 3	54	54	54	54	54	54	54	54
## 4	4	36	36	24	24	16	58	34
## 5	58	58	58	58	58	58	34	19
## 6	34	34	34	34	34	34	19	49
## 7	28	36	36	24	24	37	49	45
## 8	19	19	19	19	19	19	65	29
## 9	49	49	49	49	49	49	45	25
## 10	65	65	65	65	65	65	29	46

Como podemos observar en la tabla (fila 4) el método que más se le acerca es el método de regresión lineal, correlación x,y Age,Survived, ya que es el que se le acerca más a la edad original.

SibSp

```
paste("La correlación entre el número de hermanos o esposas en el barco y la clase del pasajero es de: ")
```

```
## [1] "La correlación entre el número de hermanos o esposas en el barco y la clase del pasajero es de: "
```

```
SibSp_MediaMedianaModa <- Imputacion_MMM(timd2, 7)
SibSp_ImpSectorizada <- Imputacion_Sectorizada(timd2, 7)
SibSp_Regresion <- RegresionLineal(timd2, 7, 3)
SibSp_StDevApproach <- StandardDev_Approach(timd2, 7)
SibSp_Percentile <- Percentile_Approach(timd2, 7)
SibSp_Original <- ti %>% select(SibSp)
```

```
SibSp <- cbindX(SibSp_Original, SibSp_MediaMedianaModa, SibSp_ImpSectorizada, SibSp_Regresion,
                SibSp_StDevApproach, SibSp_Percentile)
colnames(SibSp) <- c("Original", "Media", "Mediana", "Moda", "Imp_SecModa", "R_Lineal", "Sd_App", "Perc_App")
SibSp[1:10,]
```

##	Original	Media	Mediana	Moda	Imp_SecModa	R_Lineal	Sd_App	Perc_App
## 1	1	1	1	1	1	1	1	1
## 2	1	1	1	1	1	1	1	1
## 3	0	0	0	0	0	0	0	0
## 4	1	1	1	1	1	1	1	1
## 5	0	0	0	0	0	0	0	0
## 6	0	0	0	0	0	0	0	0
## 7	0	0	0	0	0	0	1	1
## 8	3	3	3	3	3	3	0	0
## 9	1	1	1	1	1	1	1	1
## 10	0	0	0	0	0	0	0	0

El modelo que más se le acercaría a esta columna es el de imputación por moda, ya que los valores que podría tomar esta columna podrían ser muy parecidos y repetitivos.

Parch

```
paste("La correlación entre tener padres o hijos en el barco y la clase del pasajero es de: ", cor(timd2[, Parch, Survived]))
```

```
## [1] "La correlación entre tener padres o hijos en el barco y la clase del pasajero es de: 0.0419694"
```

```
Parch_MediaMedianaModa <- Imputacion_MMM(timd2, 8)
Parch_ImpSectorizada <- Imputacion_Sectorizada(timd2, 8)
Parch_Regresion <- RegresionLineal(timd2, 8, 3)
Parch_StDevApproach <- StandardDev_Approach(timd2, 8)
```

```
Parch_Percentile <- Percentile_Approach(timd2, 8)
Parch_Original <- ti %>% select(Parch)

Parch <- cbindX(Parch_Original, Parch_MediaMedianaModa, Parch_ImpSectorizada, Parch_Regresion,
               Parch_StDevApproach, Parch_Percentile)
colnames(Parch) <- c("Original", "Media", "Mediana", "Moda", "Imp_SecModa", "R_Lineal", "Sd_App", "Perc_
Parch[1:10,]
```

```
##      Original Media Mediana Moda Imp_SecModa R_Lineal Sd_App Perc_App
## 1         0      0      0      0          0          0      0      0
## 2         0      0      0      0          0          0      0      0
## 3         0      0      0      0          0          0      0      0
## 4         1      0      0      0          0          1      0      0
## 5         0      0      0      0          0          0      0      0
## 6         0      0      0      0          0          0      0      0
## 7         0      0      0      0          0          0      2      2
## 8         2      2      2      2          2          2      1      1
## 9         0      0      0      0          0          0      0      0
## 10        1      1      1      1          1          1      0      0
```

El método que más se acerca a la realidad para esta columna es el método de regresión lineal, ya que en la fila 4 y fila 9 la regresión pudo calcular el valor que tomaría por medio de la correlación entre Parch y Pclass, ya que tienen una correlación cercana a uno y los datos calculados se ajustan más a la edad real según la variable de clase de pasajero.

Fare

```
paste("La correlación entre la tarifa de pago y la clase del pasajero es de: ", cor(timd2$Fare, timd2$P
```

```
## [1] "La correlación entre la tarifa de pago y la clase del pasajero es de: -0.304437608095598"
```

```
Fare_MediaMedianaModa <- Imputacion_MMM(timd2, 10)
Fare_ImpSectorizada <- Imputacion_Sectorizada(timd2, 10)
Fare_Regresion <- RegresionLineal(timd2, 10, 3)
Fare_StDevApproach <- StandardDev_Approach(timd2, 10)
Fare_Percentile <- Percentile_Approach(timd2, 10)
Fare_Original <- ti %>% select(Fare)

Fare <- cbindX(Fare_Original, Fare_MediaMedianaModa, Fare_ImpSectorizada, Fare_Regresion,
               Fare_StDevApproach, Fare_Percentile)
colnames(Fare) <- c("Original", "Media", "Mediana", "Moda", "Imp_SecModa", "R_Lineal", "Sd_App", "Perc_
Fare[1:10,]
```

```
##      Original      Media      Mediana      Moda Imp_SecModa R_Lineal      Sd_App Perc_App
## 1    71.2833    71.2833    71.2833    71.2833      71.2833    71.2833    71.2833    71.2833
## 2    53.1000    53.1000    53.1000    53.1000      53.1000    53.1000    53.1000    53.1000
## 3    51.8625    51.8625    51.8625    51.8625      51.8625    51.8625    51.8625    51.8625
## 4    16.7000    16.7000    16.7000    16.7000      16.7000    16.7000    16.7000    16.7000
## 5    26.5500    26.5500    26.5500    26.5500      26.5500    26.5500    26.5500    26.5500
## 6    13.0000    13.0000    13.0000    13.0000      13.0000    13.0000    13.0000    35.5000
## 7    35.5000    35.5000    35.5000    35.5000      35.5000    35.5000    35.5000    76.7292
## 8   263.0000   263.0000   263.0000   263.0000     263.0000   263.0000   263.0000    61.9792
## 9    76.7292    76.7292    76.7292    76.7292      76.7292    76.7292    76.7292    83.4750
## 10   61.9792    61.9792    61.9792    61.9792      61.9792    61.9792    61.9792    61.1750
```

Para esta columna, sería mejor utilizar el Standard Deviation Approach, ya que se puede observar que hay

distintas tarifas a los pasajeros, entonces sería mejor omitir los NA con este método y escoger la data en algún rango debido a que asignar alguna tarifa con los otros métodos no sería tan bueno ya que sesgar los datos y tener mala información sobre las tarifas.

Embarked

```
timd2$Embarked2 <- ifelse(timd2$Embarked == "C", 1,
                          ifelse(timd2$Embarked == "Q", 2,
                                ifelse(timd2$Embarked == "S", 3, NA)))
timd2$Embarked3 <- ifelse(is.na(timd2$Embarked2), mfv1(timd2$Embarked2, na_rm = TRUE), timd2$Embarked2)

timd2$Embarked4 <- ifelse(timd2$Embarked3 == 1, "C",
                          ifelse(timd2$Embarked3 == 2, "Q", "S"))

Embarked_Moda <- timd2 %>% select(Embarked4)

Embarked_Fill <- timd2 %>% fill(Embarked)
Embarked_Fill$Embarked <- ifelse(Embarked_Fill$Embarked == "3", "S", Embarked_Fill$Embarked)
Embarked_Fill <- Embarked_Fill %>% select(Embarked)

Embarked_Original <- ti %>% select(Embarked)

Embarked <- cbindX(Embarked_Original, Embarked_Moda, Embarked_Fill)
colnames(Embarked) <- c("Original", "Moda", "Fill_Values")

Embarked[1:10,]
```

```
##      Original Moda Fill_Values
## 1          C    C            C
## 2          S    S            S
## 3          S    S            S
## 4          S    S            S
## 5          S    S            S
## 6          S    S            S
## 7          S    S            S
## 8          S    S            S
## 9          C    C            C
## 10         C    C            C
```

Para esta columna el método que más se le acercaría a la realidad sería el de imputación por moda, ya que al tener sólo 3 categorías, elegir el más repetitivo para rellenar un NA es mejor que llenar los datos top to down porque la información estadística podría ser mejor.

Inciso 6: Conclusiones

El método de filling values with information available (utilizado en la columna de Sex) es un buen método para poder rellenar los missing values según la información que creamos necesaria utilizar para rellenar estos.

El método de imputación de media, mediana, moda y moda sectorizada, podría ser una buena opción para rellenar los missing values en las columnas que tengan valores cercanos o en secuencia ya que al ser imputaciones estadísticas estas podrían sesgar los datos, por ejemplo en las columnas de: Sex, Embarked, SibSp toman valores muy cercanos y sería más conveniente llenar los NA con estos métodos en el caso de no tener información adicional.

El modelo de regresión lineal, puede ser bastante conveniente de utilizar cuándo las variables a utilizar tienen una correlación casi perfecta o cercana a uno, por ejemplo la columna de hijos y padres en el barco (Parch)

tiene una correlación muy buena con la de clase, la cuál permite calcular una mejor aproximación de los valores.

Los modelos de eliminación de outliers por medio de Standar Deviation Approach o Percentile Approach, son útiles cuando la variable que contenga missing data tenga datos muy diferentes, que podrían ser aleatorios o asignados discriminadamente, por ejemplo con la columna de fare, a tener datos muy variados sería mejor remover los NA y trabajar con los datos que estos outliers los recomiendan según los rangos.

Parte 2

Inciso 1: Luego del pre-procesamiento de la data con Missing Values, normalice las columnas numéricas por los métodos:

a. Standarization

b. MinMaxScaling

c. MaxAbsScaler

```
ZValue <- function(df, columna){
  if(columna == 6){
    Z <- df %>% mutate(Z_Age = (Age - mean(Age, na.rm = TRUE))/sd(Age, na.rm = TRUE))
    Z <- Z %>% select(Z_Age)
    return(as.data.frame(Z))
  }

  else if(columna == 10){
    Z <- df %>% mutate(Z_Fare = (Fare - mean(Fare, na.rm = TRUE))/sd(Fare, na.rm = TRUE))
    Z <- Z %>% select(Z_Fare)
    return(as.data.frame(Z))
  }
}

MinMax <- function(df, columna){
  if(columna == 6){
    MM <- df %>% mutate(MM_Age = (Age - min(Age, na.rm = TRUE))/(max(Age, na.rm = TRUE)-min(Age, na.rm = TRUE)))
    MM <- MM %>% select(MM_Age)
    return(as.data.frame(MM))
  }

  else if(columna == 10){
    MM <- df %>% mutate(MM_Fare = (Fare - min(Fare, na.rm = TRUE))/(max(Fare, na.rm = TRUE)-min(Fare, na.rm = TRUE)))
    MM <- MM %>% select(MM_Fare)
    return(as.data.frame(MM))
  }
}

MaxAbs <- function(df, columna){
  if(columna == 6){
    Max <- max(df$Age, na.rm = TRUE)
    Min <- min(df$Age, na.rm = TRUE)
    MA <- df %>% mutate(MA_Age = (Age - mean(c(Max,Min))) / (Max - mean(c(Max,Min))))
    MMA <- MA %>% select(MA_Age)
    return(as.data.frame(MMA))
  }
}
```

```

if(columna == 10){
  Max <- max(df$Fare, na.rm = TRUE)
  Min <- min(df$Fare, na.rm = TRUE)
  MA <- df %>% mutate(MA_Fare = (Fare - mean(c(Max,Min))) / (Max - mean(c(Max,Min))))
  MMA<- MA %>% select(MA_Fare)
  return(as.data.frame(MMA))
}
}

```

Sólo se podrán normalizar las columnas numéricas cuyos valores sean muy dispersos y que no sigan una buena distribución Columnas que se le aplicarán los métodos: Age y Fare

Age

```

Z_Age <- ZValue(timd2, 6)
MinMax_Age <- MinMax(timd2, 6)
MaxAbs_Age <- MaxAbs(timd2, 6)

Age_Normalized <- cbindX(Z_Age, MinMax_Age, MaxAbs_Age)
Age_Normalized[1:10,]

```

```

##           Z_Age      MM_Age      MA_Age
## 1    0.14752825  0.4688923 -0.06221548
## 2   -0.04427709  0.4309560 -0.13808801
## 3    1.17049005  0.6712190  0.34243804
## 4           NA           NA           NA
## 5    1.42623050  0.7218007  0.44360142
## 6   -0.10821220  0.4183106 -0.16337886
## 7           NA           NA           NA
## 8   -1.06723889  0.2286292 -0.54274153
## 9    0.85081449  0.6079919  0.21598381
## 10   1.87377629  0.8103187  0.62063733

```

Fare

```

Z_Fare <- ZValue(timd2, 10)
MinMax_Fare <- MinMax(timd2, 10)
MaxAbs_Fare <- MaxAbs(timd2, 10)

Fare_Normalized <- cbindX(Z_Fare, MinMax_Fare, MaxAbs_Fare)
Fare_Normalized[1:10,]

```

```

##           Z_Fare      MM_Fare      MA_Fare
## 1   -0.09965282  0.13913574 -0.72172853
## 2   -0.33571886  0.10364430 -0.79271141
## 3   -0.35178479  0.10122886 -0.79754228
## 4   -0.80828455  0.03259623 -0.93480754
## 5   -0.68040620  0.05182215 -0.89635570
## 6   -0.85632007  0.02537431 -0.94925138
## 7   -0.56421216  0.06929139 -0.86141723
## 8    2.38932341  0.51334181  0.02668362
## 9   -0.02895102  0.14976542 -0.70046915
## 10  -0.22044399  0.12097534 -0.75804932

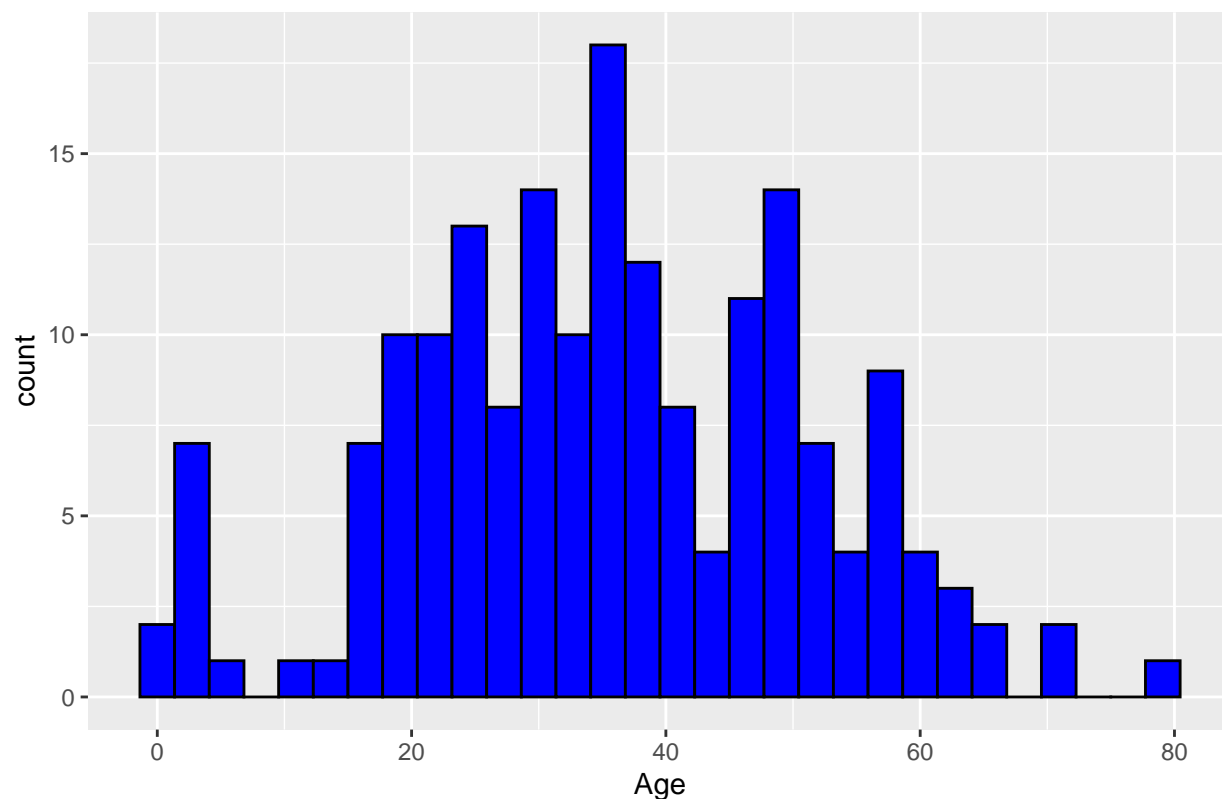
```

Inciso 2: Compare los estadísticos que considere más importantes para su conclusión y compare contra la data completa de “titanic.csv”

Comparación por histogramas para la columna de Age

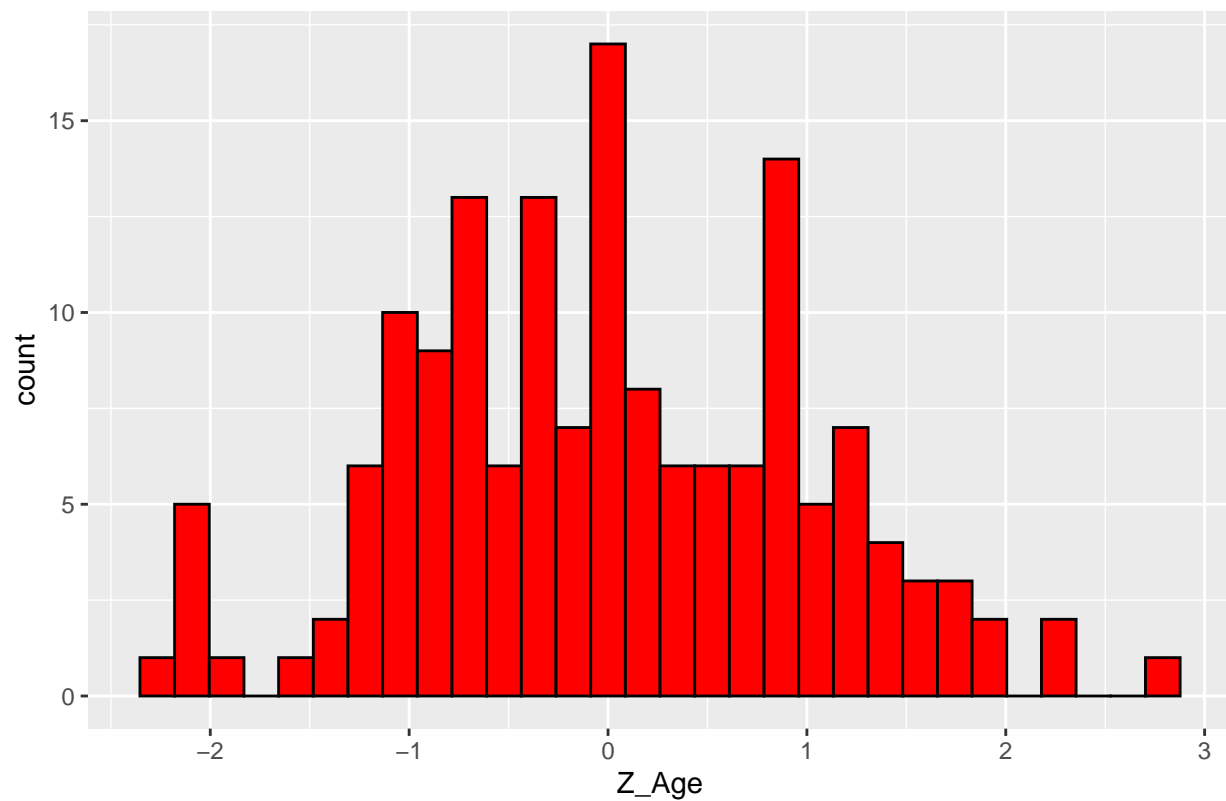
```
library(ggplot2)
ggplot(ti, aes(Age)) +
  geom_histogram(color="black", fill = "blue") +
  labs(title = "Histograma para la variable Age Original")
```

Histograma para la variable Age Original

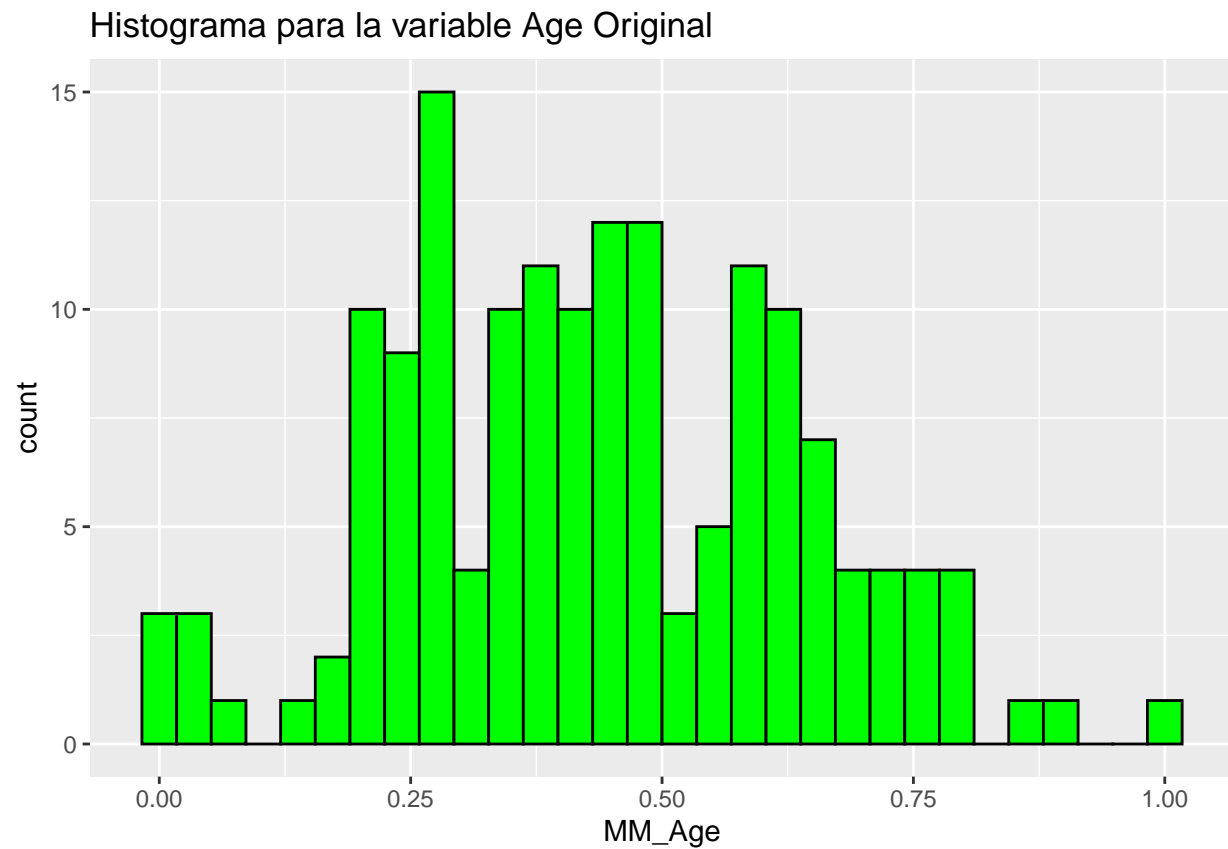


```
ggplot(Age_Normalized, aes(Z_Age)) +
  geom_histogram(color="black", fill = "red") +
  labs(title = "Histograma para la variable Age Original")
```

Histograma para la variable Age Original

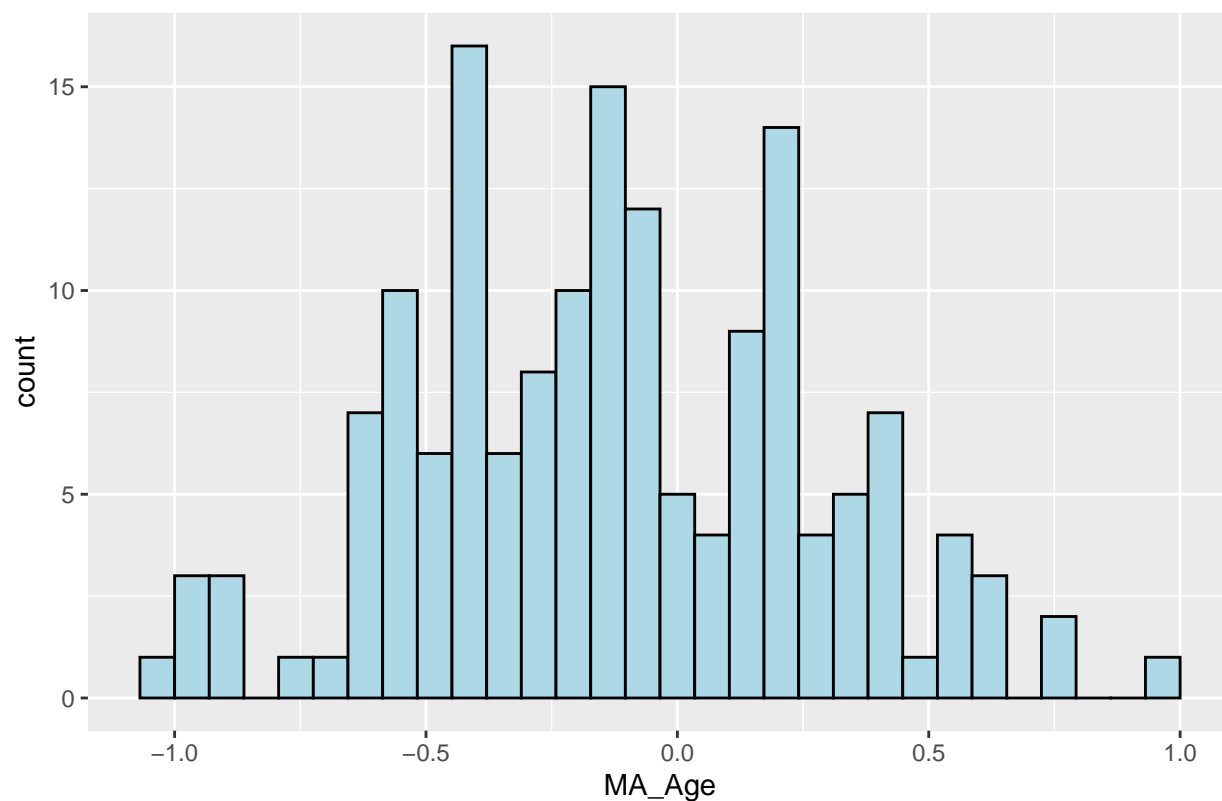


```
ggplot(Age_Normalized, aes(MM_Age)) +  
  geom_histogram(color="black", fill = "green") +  
  labs(title = "Histograma para la variable Age Original")
```



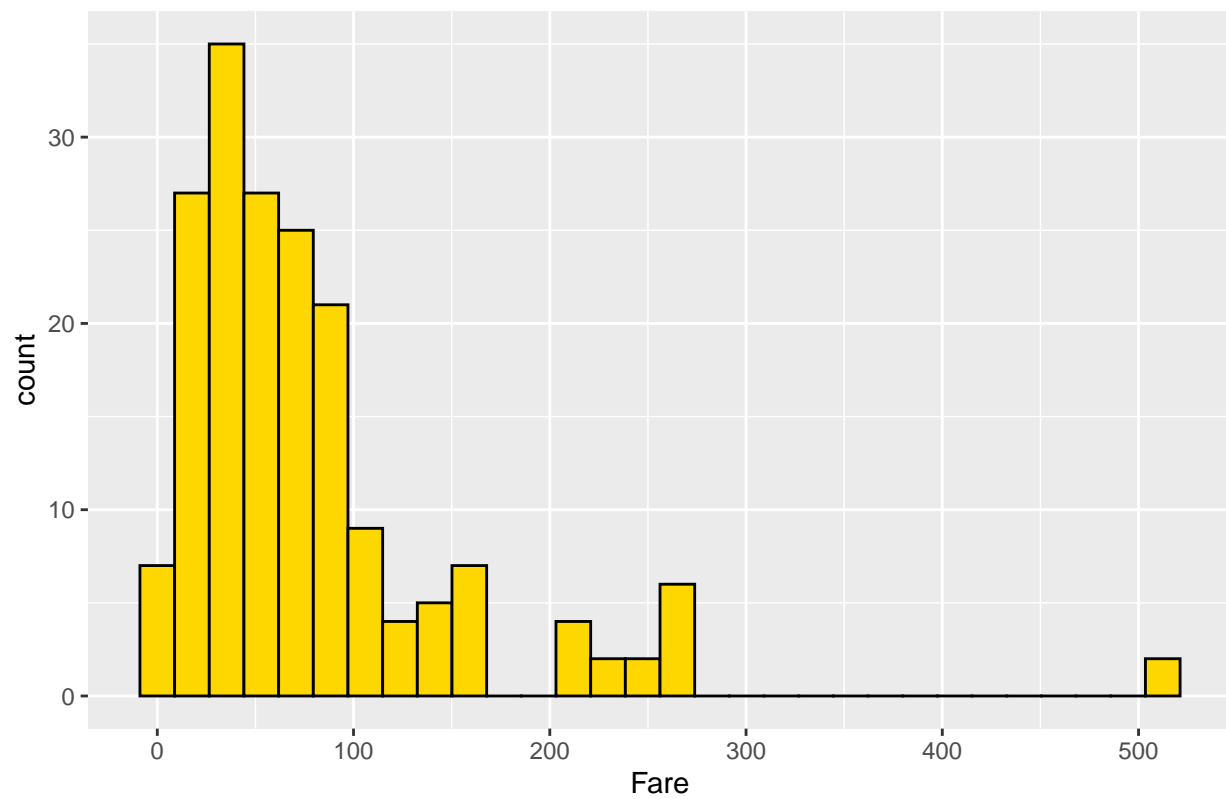
```
ggplot(Age_Normalized, aes(MA_Age)) +  
  geom_histogram(color="black", fill = "lightblue") +  
  labs(title = "Histograma para la variable Age Original")
```

Histograma para la variable Age Original



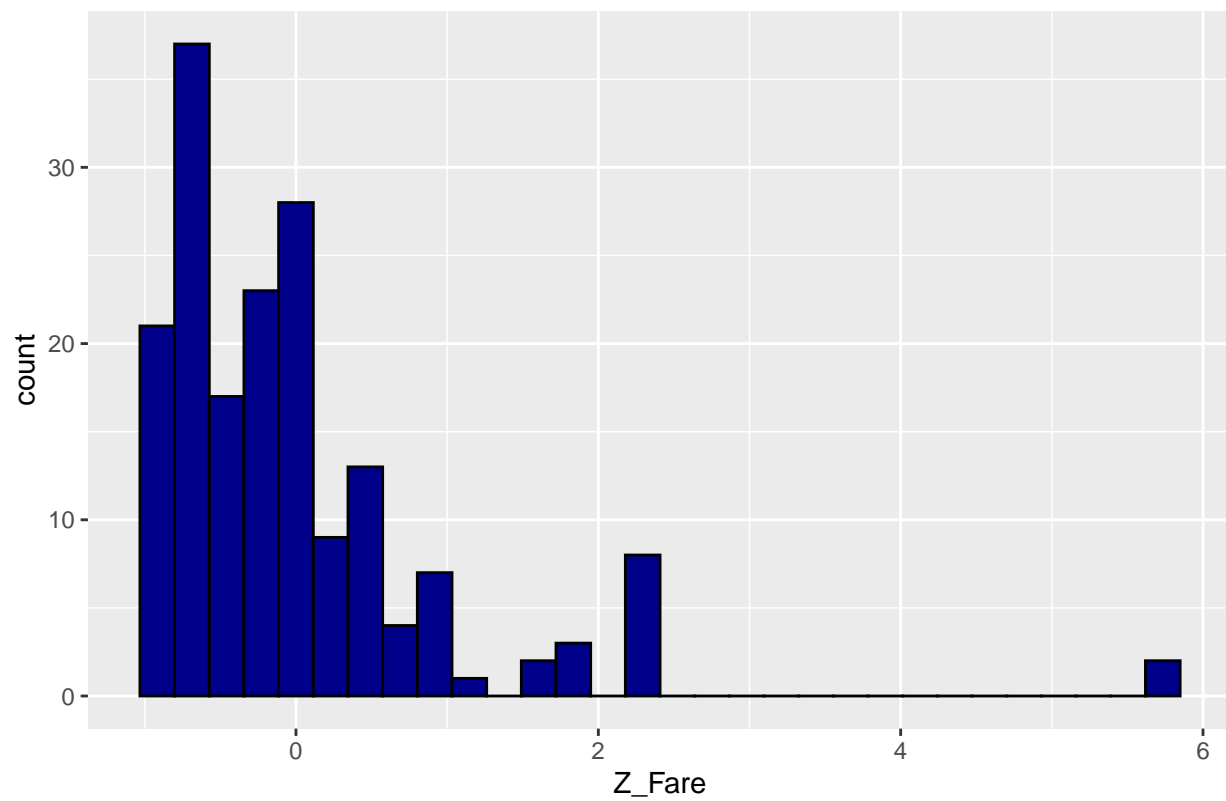
```
library(ggplot2)
ggplot(ti, aes(Fare)) +
  geom_histogram(color="black", fill = "gold") +
  labs(title = "Histograma para la variable Age Original")
```

Histograma para la variable Age Original



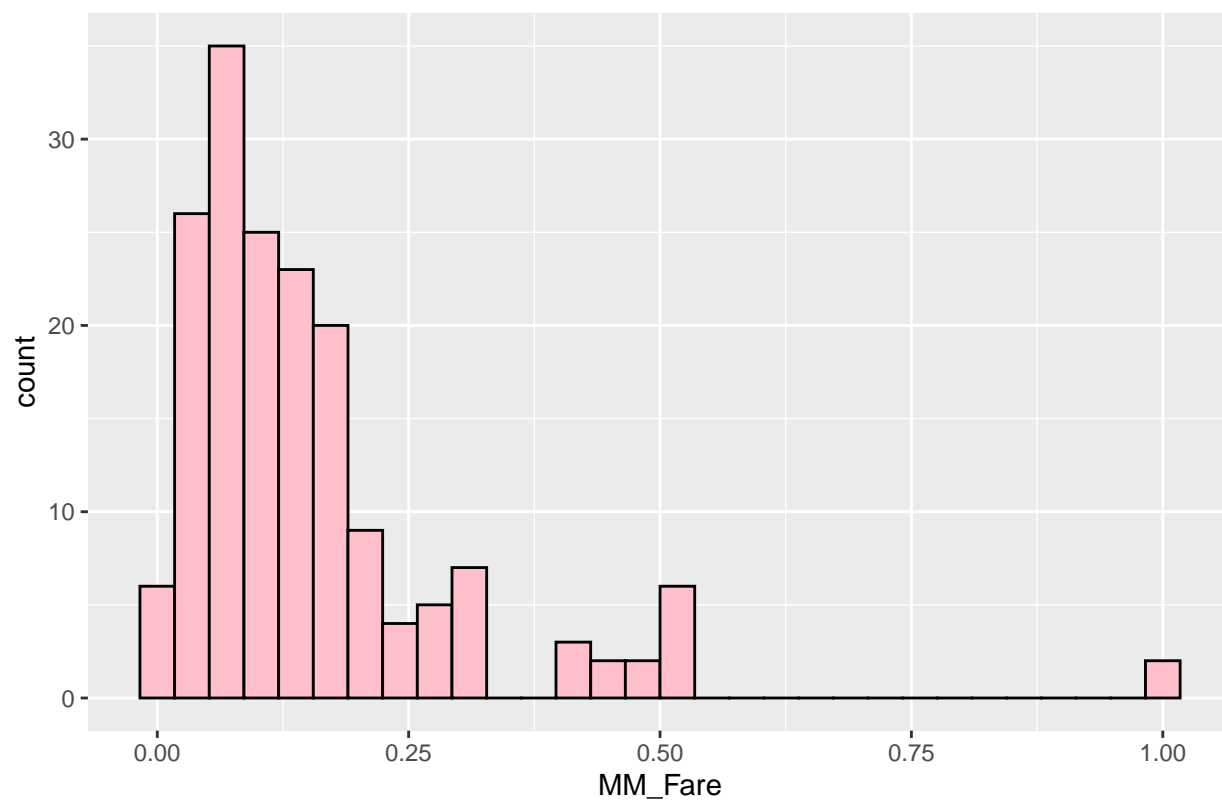
```
ggplot(Fare_Normalized, aes(Z_Fare)) +  
  geom_histogram(color="black", fill = "darkblue") +  
  labs(title = "Histograma para la variable Age Original")
```


Histograma para la variable Age Original



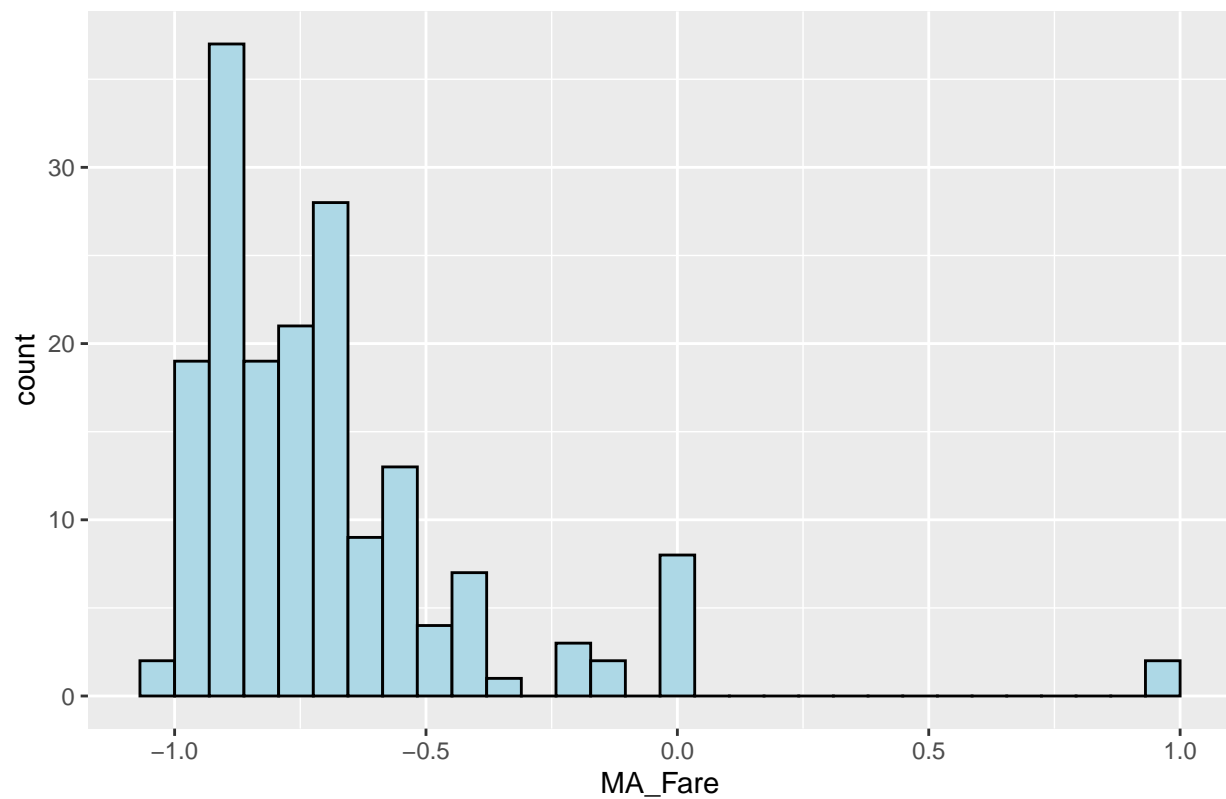
```
ggplot(Fare_Normalized, aes(MM_Fare)) +  
  geom_histogram(color="black", fill = "pink") +  
  labs(title = "Histograma para la variable Age Original")
```

Histograma para la variable Age Original



```
ggplot(Fare_Normalized, aes(MA_Fare)) +  
  geom_histogram(color="black", fill = "lightblue") +  
  labs(title = "Histograma para la variable Age Original")
```

Histograma para la variable Age Original



Como podemos observar en los histogramas los metodos de MinMaxScaling y MaxAbsScaling se acercan mucho al histograma original de la data, con esto podría concluir que utilizar uno los dos métodos sería lo mejor para poder normalizar la data con base a sus datos estadísticos de máximos y mínimos.