Untitled

```
library(readr)
library(dplyr)
library(modeest)
library(ggplot2)
data <- read_csv("titanic.csv")
datam <- read_csv("titanic_MD.csv")
```

Parte 1

1)

El dataset de titanic_MD contiene un total de 12 variables y 183 observaciones. Cada observación representa un pasajero del barco, y las variables son caracteristicas del pasajero.

La columna Sex, correspondiente al genero del pasajero contiene 51 datos faltantes, teniendo es su lugar el signo "?".

La columna Age, correspondiente a la edad del pasajero contiene 25 datos faltantes, registrados como un NA.

La columna SibSp, correspondiente a la cantidad hermanos o cónyuges a bordo que tenia el pasajero contiene 3 datos faltantes, registrados como un NA.

La columna Parch, correspondiente a la cantidad de padres o hijos del pasajero contiene 12 datos faltantes, registrados como un NA.

La columna Fare, correspondiente a la tarifa del pasajero contiene 8 datos faltantes, registrados como un NA.

La columna Embarked, correspondiente al puerto de embarque del pasajero contiene 12 datos faltantes, registrados como un NA.

```
summary(datam)
```

```
##   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
```

```
sum(datam$Sex=="?")
```

```
## [1] 51
```

```
sum(is.na(datam$Embarked))
```

```
## [1] 12
```

2)

Para la columna de Sex, se utilizará imputación de la moda sectorizada, ya que es una variable categórica.

Para la columna age se realizará una imputación de la mediana para determinar los valores faltantes, ya que este valor se ubic al centro de todod los datos, y podrá representar los datos faltantes con un sesgo menor.

En la columna SibSp se utilizará una imputación de la moda, ya que se trata de valores discretos y de poco niveles.

En la columna Parch también se utilizará una imputación de la moda, ya que se trata de valores discretos y de poco niveles.

En la columna fare se utilizará una imputación de la media, ya que estos valores son relativamento continuos (redondeando a dos decimales) y los rangos son muy extenesos. Adicionalmente, al ser tarifas, la media refleja mejor el rumbo de los datos.

Para la columna Embarked se utilizará una imputación de la moda sectorizada, ya que son valores categorico, este es el método que más se ajusta.

3)

El dataset cuanta únicamente con 100 filas completas, es decir que no contengan ningun dato faltante. A continuación se presentan las 100 filas completas con data original.

```
datam1 <- na.omit(datam)
datam2 <- datam1[datam1$Sex!="?",]
datam2
```

```
## # A tibble: 100 x 12
##    PassengerId Survived Pclass Name  Sex     Age SibSp Parch Ticket  Fare Cabin
##          <dbl>    <dbl>  <dbl> <chr> <chr> <dbl> <dbl> <dbl> <chr>  <dbl> <chr>
##  1           4        1      1 Futr~ fema~    35     1     0 113803  53.1 C123
##  2           7        0      1 McCa~ male     54     0     0 17463   51.9 E46
##  3          22        1      2 Bees~ male     34     0     0 248698  13   D56
##  4          55        0      1 Ostb~ male     65     0     1 113509  62.0 B30
##  5          63        0      1 Harr~ male     45     1     0 36973   83.5 C83
##  6          67        1      2 Nye,~ fema~    29     0     0 C.A. ~  10.5 F33
##  7          97        0      1 Gold~ male     71     0     0 PC 17~  34.7 A5
##  8          98        1      1 Gree~ male     23     0     1 PC 17~  63.4 D10 ~
##  9         103        0      1 Whit~ male     21     0     1 35281   77.3 D26
## 10         119        0      1 Baxt~ male     24     0     1 PC 17~ 248.  B58 ~
```

```
## # ... with 90 more rows, and 1 more variable: Embarked <chr>
```

4)

Correlación:

Se generó el codigo para montar una tabla que muestra la correlación entre las distintas variables que contienen datos faltantes. Para ello se incluyeron únicamente las variables númericas, es decir Age, SibSP, Parch y Fare. Cabe mencionar que para visualizar la correlación en forma e tabla, se generan valores redundantes al mostrar la corralacion de todas las combinaciones. Adicionalmente se observa en la diagonal una correlación de 1 obtenida de correlacionar las variables con sí mismas.

```
a0 <- cor(datam$Age, datam$Age, use = "pairwise.complete.obs")
a1 <- cor(datam$Age, datam$SibSp, use = "pairwise.complete.obs")
a2 <- cor(datam$Age, datam$Parch, use = "pairwise.complete.obs")
a3 <- cor(datam$Age, datam$Fare, use = "pairwise.complete.obs")
a4 <- cor(datam$SibSp, datam$Age, use = "pairwise.complete.obs")
a41 <- cor(datam$SibSp, datam$SibSp, use = "pairwise.complete.obs")
a5 <- cor(datam$SibSp, datam$Parch, use = "pairwise.complete.obs")
a6 <- cor(datam$SibSp, datam$Fare, use = "pairwise.complete.obs")
a7 <- cor(datam$Parch, datam$Age, use = "pairwise.complete.obs")
a8 <- cor(datam$Parch, datam$SibSp, use = "pairwise.complete.obs")
a81 <- cor(datam$Parch, datam$Parch, use = "pairwise.complete.obs")
a9 <- cor(datam$Parch, datam$Fare, use = "pairwise.complete.obs")
a10 <- cor(datam$Fare, datam$Age, use = "pairwise.complete.obs")
a11 <- cor(datam$Fare, datam$SibSp, use = "pairwise.complete.obs")
a12 <- cor(datam$Fare, datam$Parch, use = "pairwise.complete.obs")
a13 <- cor(datam$Fare, datam$Fare, use = "pairwise.complete.obs")
age <- c(a0,a1,a2,a3)
sibsp <- c(a4,a41,a5,a6)
parch <- c(a7,a8,a81,a9)
fare <- c(a10,a11,a12,a13)
correlaciones <- data.frame(rbind(age,sibsp,parch,fare))
names(correlaciones) <- c("age","sibsp","parch","fare")
correlaciones
```

```
##               age       sibsp      parch       fare
## age    1.00000000 -0.08795149 -0.2795476 -0.1309791
## sibsp -0.08795149  1.00000000  0.2551521  0.2990607
## parch -0.27954756  0.25515208  1.0000000  0.3814448
## fare  -0.13097906  0.29906071  0.3814448  1.0000000
```

Genear los datos faltantes:

Para generar los datos faltantes, se utilizaron siete metodos en total. Sin emmbargo solamente 6 de ellos fueron aplicables a las variables númericas, y uno a las variables categóricas.

Como primer punto, se creó la siguiente función para generar automáticamente los datos faltantes para las cuatro variables numéricas modificando únicamente los parametros al ejecutar la funacion.

Los metodos utilizaos en esta función son:

- Imputación de la media
- Imputación de la moda
- Imputacion de la mediana
- Regrión lineal simple (Para la regresión se utilizó la tarifa para determinar la edad, y se utilizó la edad para las otras tres variables)

```
missing_d <- function(col1,col2){
  #imputacion media
  media <- round(mean(datam[[col1]],na.rm = TRUE),0)
  i_media <- ifelse(is.na(datam[[col1]]),media,datam[[col1]])
  #imputacion moda
  moda <- round(mfv(datam[[col1]],na_rm = TRUE),0)
  i_moda <- ifelse(is.na(datam[[col1]]),moda,datam[[col1]])
  #imputacion mediana
  mediana <- round(median(datam[[col1]],na.rm = TRUE),0)
  i_mediana <- ifelse(is.na(datam[[col1]]),mediana,datam[[col1]])
  #regresion lineal simple
  if(col1=="Age"){
    regre <- lm(Age ~ Fare,data = datam)
  }
  if(col1=="SibSp"){
    regre <- lm(SibSp ~ Age,data = datam)
  }
  if(col1=="Parch"){
    regre <- lm(Parch ~ Age,data = datam)
  }
  if(col1=="Fare"){
    regre <- lm(Fare ~ Age,data = datam)
  }
  regresar <- datam[[col2]]
  regresar <- ifelse(is.na(regresar),round(mean(datam[[col2]],na.rm = TRUE),0),regresar)
  d <- ifelse(is.na(datam[[col1]]),((regresar*regre$coefficients[2])+regre$coefficients[1]),datam[[col1
  simple_regre <- round(d,0)
  nueva <- data.frame(cbind(i_media,i_moda,i_mediana,simple_regre))
  return(nueva)
}
```

Posteriormente se utilizó la función para formar los conjuntos de posibles columnas completas para cada una de las variales (segun el metodo utilizado).

```
missing_age <- missing_d("Age","Fare")
missing_SibSp <- missing_d("SibSp","Age")
missing_Parch <- missing_d("Parch","Age")
missing_Fare <- missing_d("Fare","Age")
```

Edad:

```
head(missing_age)
```

```
##   i_media i_moda i_mediana simple_regre
## 1      38     38        38           38
## 2      35     35        35           35
## 3      54     54        54           54
## 4      36     24        36           38
## 5      58     58        58           58
## 6      34     34        34           34
```

Hermanos y cónyuges:

```
head(missing_SibSp)
```

```
##   i_media i_moda i_mediana simple_regre
```

```
## 1       1      1         1            1
## 2       1      1         1            1
## 3       0      0         0            0
## 4       1      1         1            1
## 5       0      0         0            0
## 6       0      0         0            0
```

Padres e hijos:

```
head(missing_Parch)
```

```
##   i_media i_moda i_mediana simple_regre
## 1       0      0         0            0
## 2       0      0         0            0
## 3       0      0         0            0
## 4       0      0         0            0
## 5       0      0         0            0
## 6       0      0         0            0
```

Tarifa:

```
head(missing_Fare)
```

```
##   i_media  i_moda i_mediana simple_regre
## 1 71.2833 71.2833   71.2833           71
## 2 53.1000 53.1000   53.1000           53
## 3 51.8625 51.8625   51.8625           52
## 4 16.7000 16.7000   16.7000           17
## 5 26.5500 26.5500   26.5500           27
## 6 13.0000 13.0000   13.0000           13
```

Para los metodos de los Outliers se realizó una función que incluye outliers con desviación estandar, utilizando los datos que se encuentren a menos de dos desviaciones estandar de la media, y outliers con percentiles, los cuales incluyen los datos entre el percentil 10 y el 90.

```
outliers <- function(col1){
  #outliers desvest.
  e1 <- na.omit(datam[[col1]])
  desv <- sd(e1)
  med <- mean(e1)
  #Se aplicó el factor de dos desviaciones estandar de distancia repecto a la media
  Outliers_desv <- ifelse(e1 > med+(2*desv),round(med+(2*desv),0),ifelse(e1< med-(2*desv),round(med-(2*
  #outliers percentil
  f1 <- na.omit(datam[[col1]])
  #Se aplicó el factor de percentiles 10 y 90
  p90 <- quantile(f1,0.9)
  p10 <- quantile(f1,0.1)
  Outliers_perc <- ifelse(f1 > p90,p90,ifelse(f1< p10,p10,f1))
  outs <- data.frame(cbind(Outliers_desv,Outliers_perc))
  return(outs)
}
```

Utilizando esta función se generaron los siguiente datos para cada una de las variables:

```
outliers_age <- outliers("Age")
outliers_sibsp <- outliers("SibSp")
outliers_parch <- outliers("Parch")
outliers_fare <- outliers("Fare")
```

Edad:

```
head(outliers_age)
```

```
##   Outliers_desv Outliers_perc
## 1            38            38
## 2            35            35
## 3            54            54
## 4            58            56
## 5            34            34
## 6            19            19
```

Hermanos y cónyuges:

```
head(outliers_sibsp)
```

```
##   Outliers_desv Outliers_perc
## 1             1             1
## 2             1             1
## 3             0             0
## 4             1             1
## 5             0             0
## 6             0             0
```

Padres e hijos:

```
head(outliers_parch)
```

```
##   Outliers_desv Outliers_perc
## 1             0             0
## 2             0             0
## 3             0             0
## 4             0             0
## 5             0             0
## 6             0             0
```

Tarifa:

```
head(outliers_fare)
```

```
##   Outliers_desv Outliers_perc
## 1       71.2833      71.28330
## 2       53.1000      53.10000
## 3       51.8625      51.86250
## 4       16.7000      16.70000
## 5       26.5500      26.55000
## 6       13.0000      13.31668
```

Para las variables categóricas, el metodo utilizado fue imputación sectorizada. Se creó la siguiente función para generar los datos faltantes para las columnas Sex y Embarked.

```
imp_sect <- function(Col1){
  #imputacion sectorizada
  moda <- mfv(datam[[Col1]],na_rm = TRUE)
  i_moda <- ifelse(is.na(datam[[Col1]]) | datam[[Col1]]=="?",moda,datam[[Col1]])
  sect <- data.frame(i_moda)
  return(sect)
}
```

Por medio de a función se generaron los siguientes datos:

```
imp_sect_sex <- imp_sect("Sex")
imp_sect_embarked <- imp_sect("Embarked")
```

Sex

```
head(imp_sect_sex)
```

```
##   i_moda
## 1   male
## 2 female
## 3   male
## 4 female
## 5 female
## 6   male
```

Embarked

```
head(imp_sect_embarked)
```

```
##   i_moda
## 1      C
## 2      S
## 3      S
## 4      S
## 5      S
## 6      S
```

5)

Teniendo los valores completos para las columnas con datos faltantes, se procedió a comprarar los datos generados con los datos reales.

Como primer punto se creó una función para unir las columnas generadas por los cuatro metodos numéricos con los datos reales, para podes luego compararlos.

```
unir <- function(datos,col1){
  data <- read_csv("titanic.csv")
  a <- datos
  real <- data[[col1]]
  c <- data.frame(cbind(a,real))
  return(c)
}
```

Se utilizó la función para generar los datasets para cada variable conteniendo los metodos de completacion y los valores reales.

```
m_age <- unir(missing_age,"Age")
m_sibsp <- unir(missing_SibSp,"SibSp")
m_parch <- unir(missing_Parch,"Parch")
m_fare <- unir(missing_Fare,"Fare")
m_sex <- unir(imp_sect_sex,"Sex")
m_embarked <- unir(imp_sect_embarked,"Embarked")
```

Con estos datos, se buscó determinar cual fue el mejor metodo de completación para cada variable. Pare ello se creó una función que devuelva la suma de errores cuadrados para las variables numéricas, y una que devuelva el total de errores para las variables categóricas.

```
errores <- function(tabla){
  media <- (tabla$real - tabla$i_media)^2
  moda <- (tabla$real - tabla$i_moda)^2
  mediana <- (tabla$real - tabla$i_mediana)^2
  regre_ <- (tabla$real - tabla$simple_regre)^2
  er_media <- sum(media)
  er_moda <- sum(moda)
  er_mediana <- sum(mediana)
  er_regre <- sum(regre_)
  errores_cuadrados <- data.frame(cbind(er_media,er_moda,er_mediana,er_regre))
  return(errores_cuadrados)
}

errores_cat <- function(tabla){
  err <- ifelse(tabla$i_moda == tabla$real,0,1)
  error <- sum(err)
  return(error)
}
```

Utilizando estas funciones, se obtuvieron los errores obtenidos por cada metodo utilizado para cada una de las variables. (Errores cuadrados para las variables numéricas y cantidad de errores para las categóricas)

```
errores_sex <- errores_cat(m_sex)
errores_embarked <- errores_cat(m_embarked)
errores_age <- errores(m_age)
errores_sibSp <- errores(m_sibsp)
errores_parch <- errores(m_parch)
errores_fare <- errores(m_fare)
```

Edad

```
errores_age
```

```
##   er_media er_moda er_mediana er_regre
## 1   6137.5  9473.5     6137.5 6448.756
```

Hermanos y cónyuges

```
errores_sibSp
```

```
##   er_media er_moda er_mediana er_regre
## 1        2       2          2        2
```

Padres e hijos

```
errores_parch
```

```
##   er_media er_moda er_mediana er_regre
## 1       12      12         12       11
```

Tarifa

```
errores_fare
```

```
##   er_media  er_moda er_mediana er_regre
## 1 28543.29 44874.77   30172.76 31545.62
```

Sexo

```
errores_sex
```

```
## [1] 24
```

Embarque

```
errores_embarked
```

```
## [1] 6
```

También se creó una función para determinar que metodo de outliers es más cercano a la realidad. En este caso se comparan los metodos de desviación estandar contra percentiles.

```
outliers_diff <- function(tabla,col1){
  #outliers desviacion estandar
  media_gen <- mean(tabla$Outliers_desv)
  media_real <- mean(data[[col1]])
  desv_gen <- sd(tabla$Outliers_desv)
  desv_real <- sd(data[[col1]])
  media_diff <- abs(media_real - media_gen)
  desv_diff <- abs(desv_real - desv_gen)
  desviacion <- cbind(media_diff,desv_diff)
  #outliers percentiles
  media_gen <- mean(tabla$Outliers_perc)
  media_real <- mean(data[[col1]])
  desv_gen <- sd(tabla$Outliers_perc)
  desv_real <- sd(data[[col1]])
  media_diff <- abs(media_real - media_gen)
  desv_diff <- abs(desv_real - desv_gen)
  percentiles <- cbind(media_diff,desv_diff)
  metodo <- c("Desviacion Estandar","Percentiles")
  diferencias <- data.frame(cbind(metodo,rbind(desviacion,percentiles)))
  return(diferencias)
}
```

Utilizando la función anterior se generaron tablas que muestran la diferencia entre la media y la desviacion estandar reales versus las que sufrieron la eliminacion de outliers con cada metodo. Cada tabla representa una variable distinta.

```
out_diff_age <- outliers_diff(outliers_age,"Age")
out_diff_sibsp <- outliers_diff(outliers_sibsp,"SibSp")
out_diff_parch <- outliers_diff(outliers_parch,"Parch")
out_diff_fare <- outliers_diff(outliers_fare,"Fare")
```

Edad

```
out_diff_age
```

```
##                metodo         media_diff         desv_diff
## 1 Desviacion Estandar 0.0573376219132626 0.411047754008255
## 2         Percentiles 0.0705104793525635  2.67447256227628
```

Hermanos y cónyuges

```
out_diff_sibsp
```

```
##                metodo         media_diff          desv_diff
## 1 Desviacion Estandar 0.0200364298724955 0.0535219167030838
## 2         Percentiles 0.0700364298724955  0.154064256197974
```

Padres e hijos

```
out_diff_parch
```

```
##                metodo         media_diff         desv_diff
## 1 Desviacion Estandar 0.0251174384047551 0.042010085250058
## 2         Percentiles 0.0251174384047551 0.042010085250058
```

Tarifa

```
out_diff_fare
```

```
##                metodo       media_diff        desv_diff
## 1 Desviacion Estandar 4.10299228103045 16.1246692121588
## 2         Percentiles 10.1578006238876 31.2981165130299
```

6)

Conclusiones

Metodos de Completación:

- Para la columna Age el mejor método para completar los datos faltantes fueron tanto la imputación de la media como la mediana, ya que en este caso ambas eran el mismo valor (36), ya que obtuvieron el menor valor de suma de errores cuadrados.
- Para la columna SibSp sucedio alo peculiar. Todos los metodos obtuvieron el mismo error, puesto que los valores no cambian. Se considera que esto sucede debido a que el rango de los datos es muy pequeño, haciendo que los metodos coincidan en su completación.
- Para la columna Parch el mejor metodo fue la regresión lineal simple, obteniendo la menor suma de errores cuadrados. Sin embargo, cabe mencionar que al igual que con la variable SibSp, el rango de los datos era muy bajo, de modo que los metodos resultaron muy igualados, y la diferencia fue mínima.
- Para la columna Fare, el metodo con la menor suma de errores cuadrados fue la imputación de la media.

Metodos de eliminación outliers:

- Para la columna Age el método de outliers que tuvo una media y desviación estandar más cercana a la de la columna real fue el de 2 desviaciones estandar de distancia.
- Para la columna SibSp el método de outliers que tuvo una media y desviación estandar más cercana a la de la columna real fue el de 2 desviaciones estandar de distancia.
- Para la columna Parch el ambos métodos obtuvieron la misma variación respecto a la media y desviacion estandar real. Nuevamente se considera que el resultado se debe a el corto rango de los datos.
- Para la columna Fare el método de outliers que tuvo una media y desviación estandar más cercana a la de la columna real tambien fue el de 2 desviaciones estandar de distancia.

Variables categóricas:

- Para las variables categóricas se completaron los datos utilizando imputación de la moda, ya que los otros métodos son aplicables exclusivamente para variables numéricas.
- Para la variable Sex, la cantidad total de errores fue de 24. Cabe mencionar que para esta variable, la tasa de error es sumamente alta. Esto se debe a que a pesar de que la moda fue “male”, la diferencia era poca, de modo que este método no es muy recomendable para escenarios de este tipo.
- Para la variable Embarked la cantidad total de errores fue de 6.

Parte 2

1)

Para la parte 2, se busca ver el efecto que tienen distintos métodos de estandarización sobre variables con datos faltantes. Para esta parte se incluyon únicamente la variable Age y la variable Fare. Esto debido a que SibSp y Parch son variables discretas y de rango muy corto, de modo que las transformaciones no son muy aplicables para variables de este estilo.

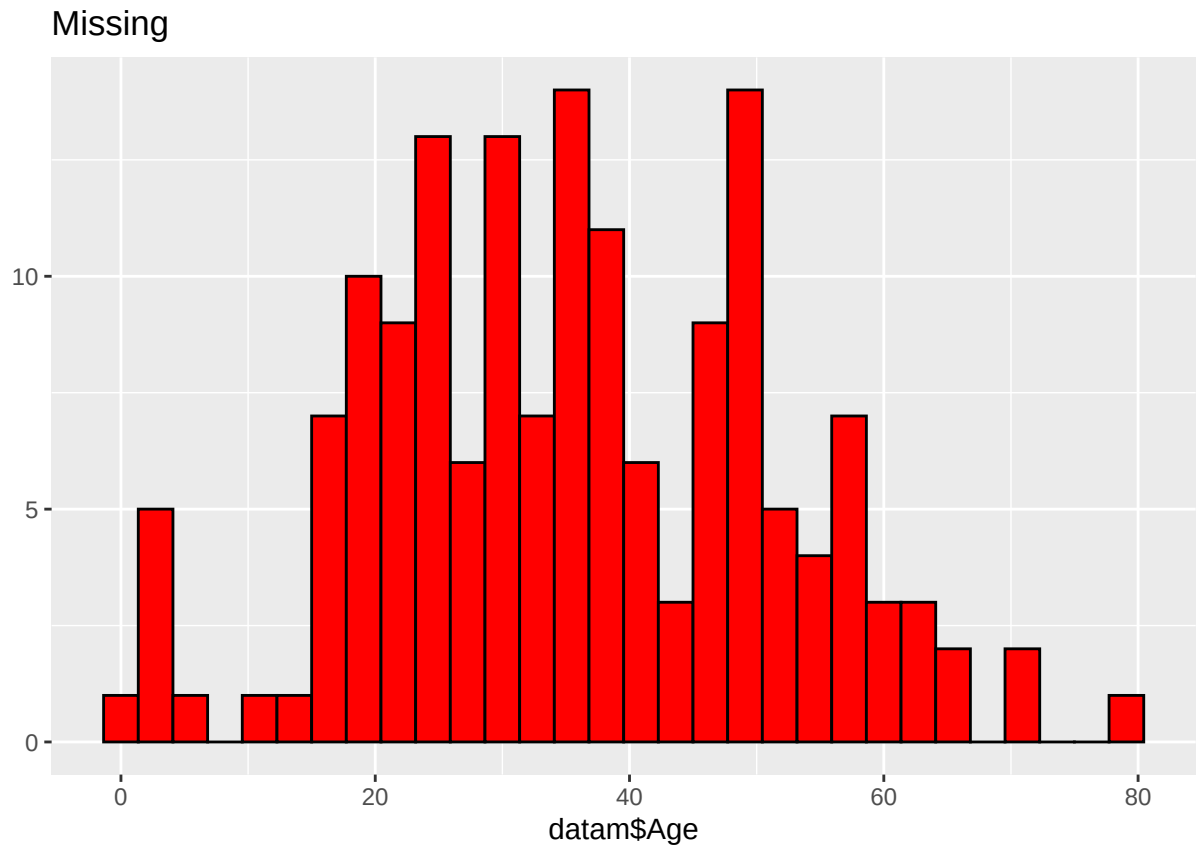
Como primer paso se ralizó la estandarización de las variables, tanto con datos faltantes como completos. Para cada variable se realizron 3 métodos, Normalización, MinMaxScaling y MaxAbsScaler.

```
#Normalizacion
norm_age1 <- datam %>% mutate(Age_norm = (Age-mean(Age,na.rm = TRUE))/sd(Age,na.rm = TRUE)) %>% select(A
norm_age2 <- data %>% mutate(Age_norm = (Age-mean(Age,na.rm = TRUE))/sd(Age,na.rm = TRUE)) %>% select(Ag

norm_Fare1 <- datam %>% mutate(Fare_norm = (Fare-mean(Fare,na.rm = TRUE))/sd(Fare,na.rm = TRUE)) %>% sel
norm_Fare2 <- data %>% mutate(Fare_norm = (Fare-mean(Fare,na.rm = TRUE))/sd(Fare,na.rm = TRUE)) %>% sele

#MinMaxScaling
minmax_age1 <- datam %>% mutate(Age_minmax = (Age-min(Age,na.rm = TRUE)) / (max(Age,na.rm = TRUE)-min(Ag
minmax_age2 <- data %>% mutate(Age_minmax = (Age-min(Age,na.rm = TRUE)) / (max(Age,na.rm = TRUE)-min(Age

minmax_Fare1 <- datam %>% mutate(Fare_minmax = (Fare-min(Fare,na.rm = TRUE)) / (max(Fare,na.rm = TRUE)-m
minmax_Fare2 <- data %>% mutate(Fare_minmax = (Fare-min(Fare,na.rm = TRUE)) / (max(Fare,na.rm = TRUE)-mi

#MaxAbsScaler
maxabs_age1 <- datam %>% mutate(Age_maxabs = (Age - mean(c(max(Age,na.rm = TRUE),min(Age,na.rm = TRUE)))
maxabs_age2 <- data %>% mutate(Age_maxabs = (Age - mean(c(max(Age,na.rm = TRUE),min(Age,na.rm = TRUE))))

maxabs_Fare1 <- datam %>% mutate(Fare_maxabs = (Fare - mean(c(max(Fare,na.rm = TRUE),min(Fare,na.rm = TR
maxabs_Fare2 <- data %>% mutate(Fare_maxabs = (Fare - mean(c(max(Fare,na.rm = TRUE),min(Fare,na.rm = TRU
```

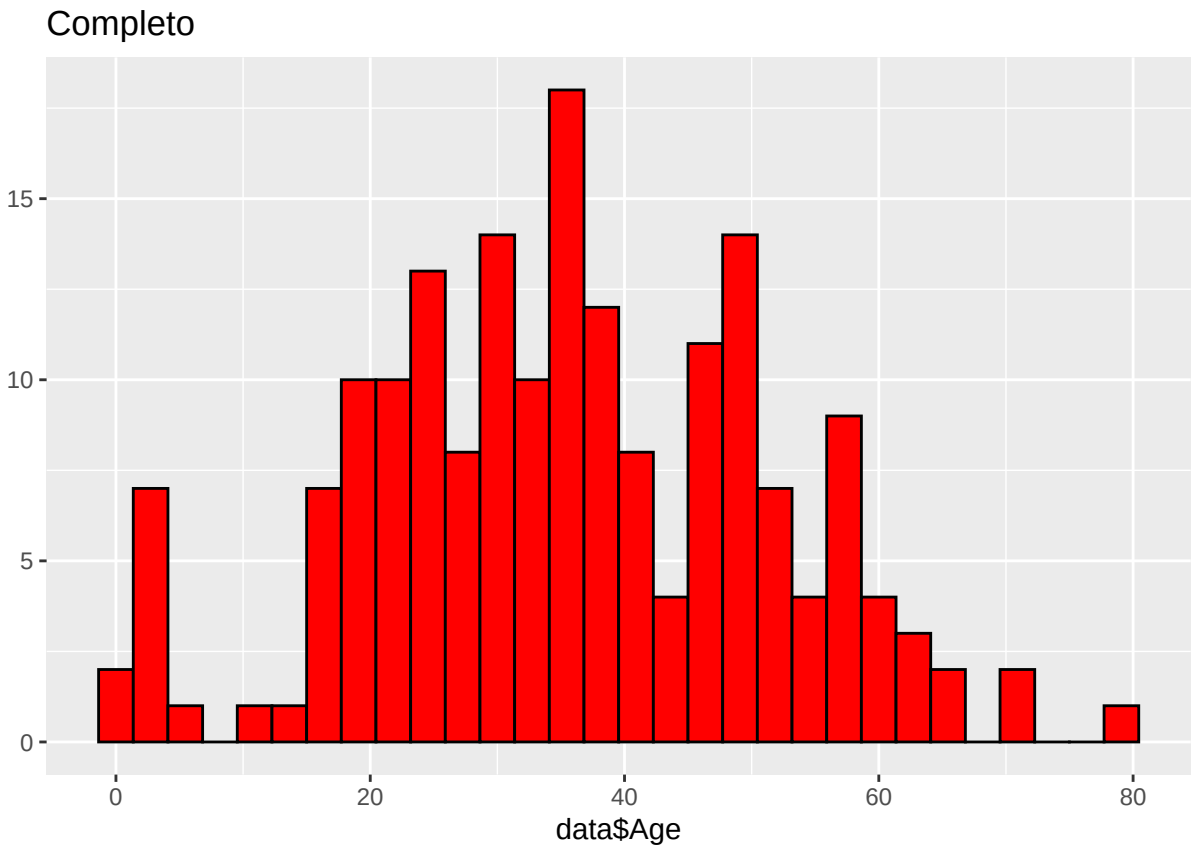
Posteriormente se graficáron histogramas de frecuencia para los datos faltantes y los completos antes y despues de la normalizaón, con el obketivo de comparar los resultados.

Edad sin estandarizar

```
qplot(datam$Age,geom="histogram",fill=I("red"),color=I("black"),main = "Missing")
```

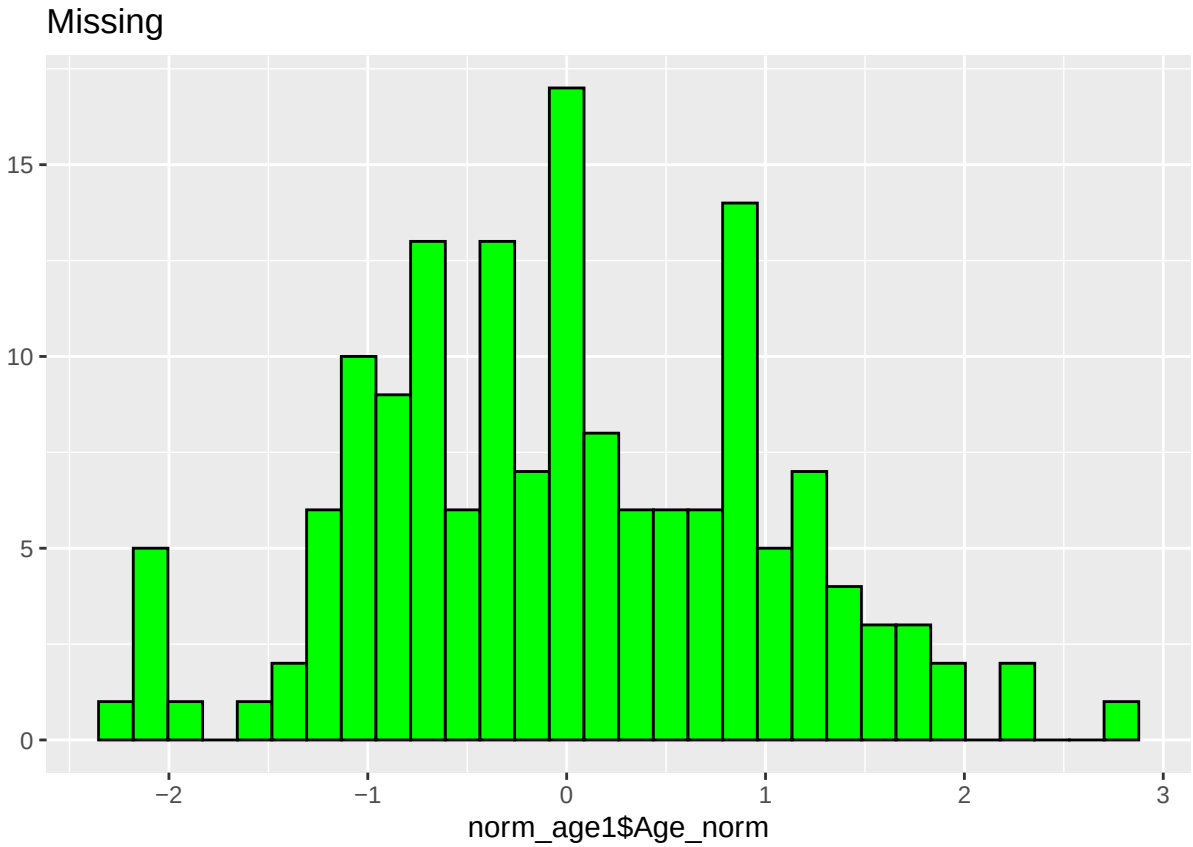



```
qplot(data$Age,geom="histogram",fill=I("red"),color=I("black"),main = "Completo")
```

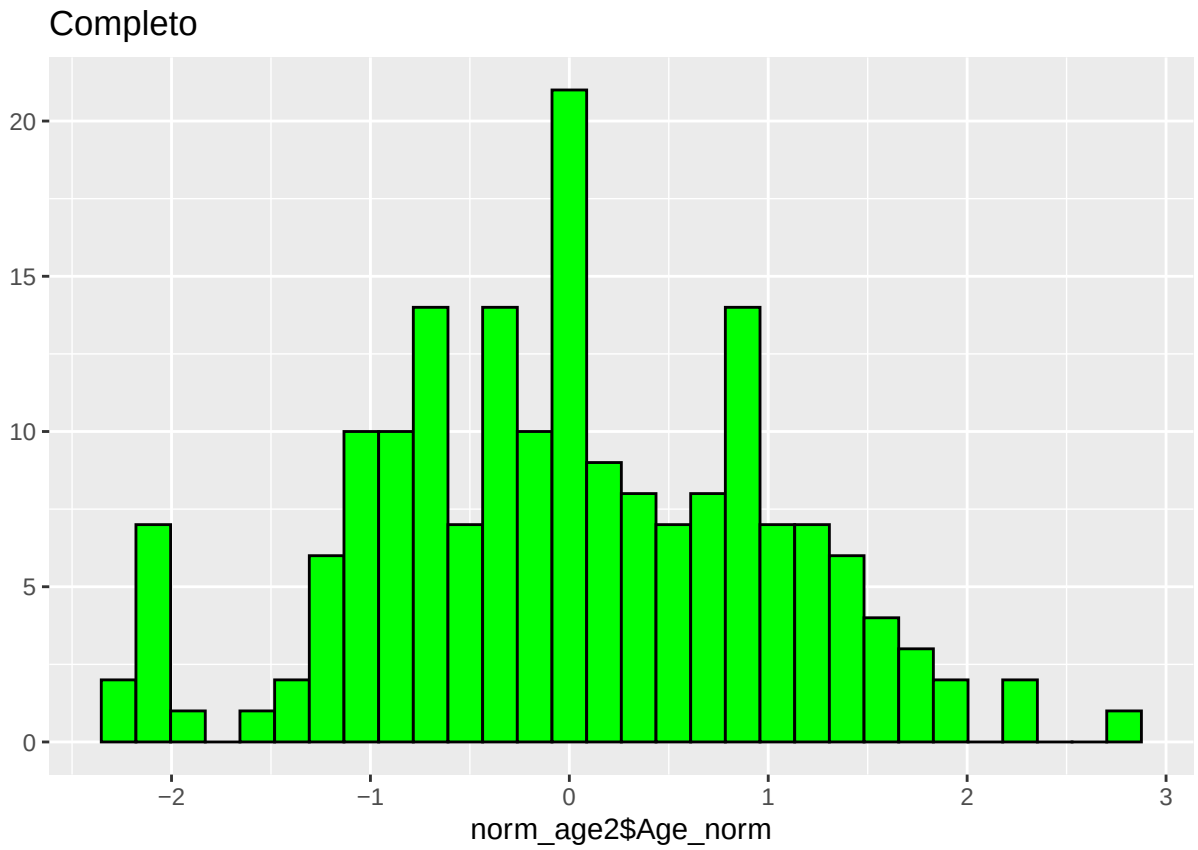


Edad Normalizada

```
qplot(norm_age1$Age_norm, geom="histogram",fill=I("green"),color=I("black"),main = "Missing")
```

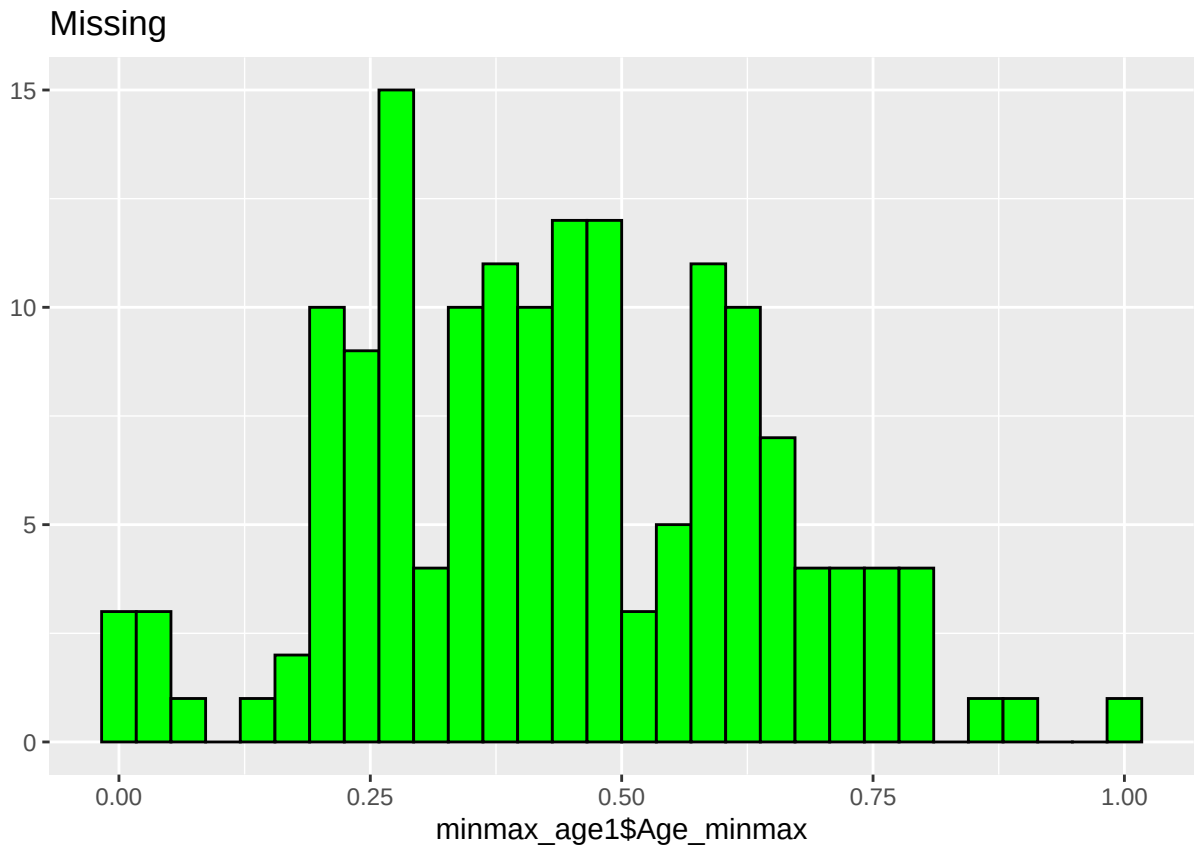



```
qplot(norm_age2$Age_norm, geom="histogram",fill=I("green"),color=I("black"),main = "Completo")
```

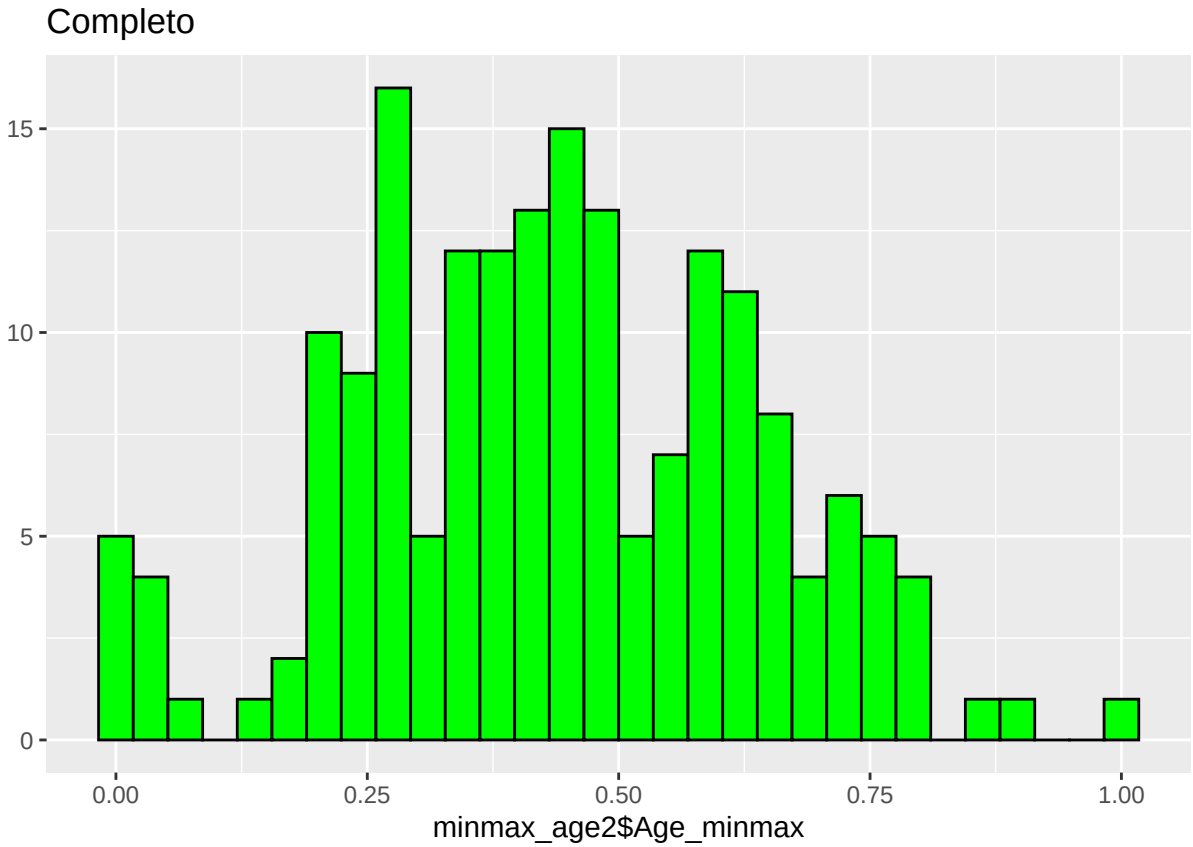


Edad MinMaxScaling

```
qplot(minmax_age1$Age_minmax, geom="histogram",fill=I("green"),color=I("black"),main = "Missing")
```

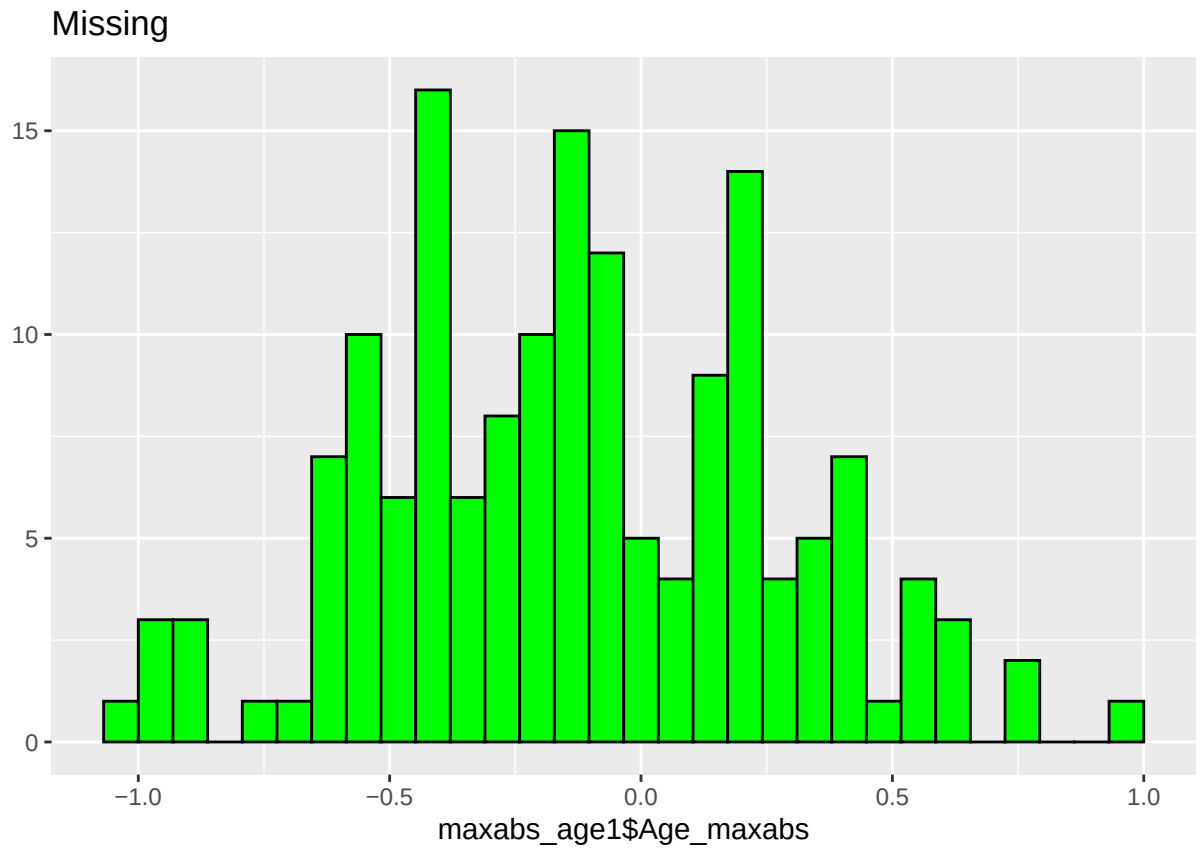



```
qplot(minmax_age2$Age_minmax, geom="histogram",fill=I("green"),color=I("black"),main = "Completo")
```

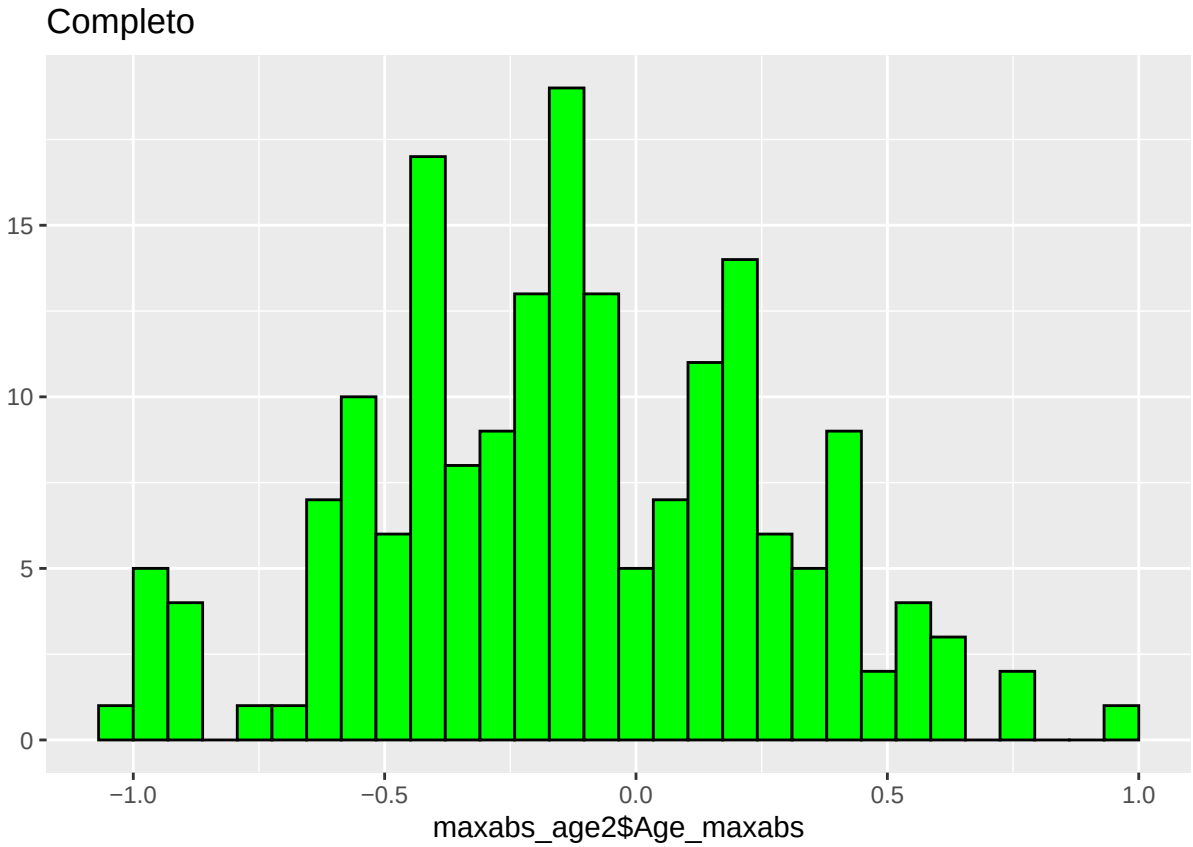


Edad MaxAbsScaler

```
qplot(maxabs_age1$Age_maxabs, geom="histogram",fill=I("green"),color=I("black"),main = "Missing")
```



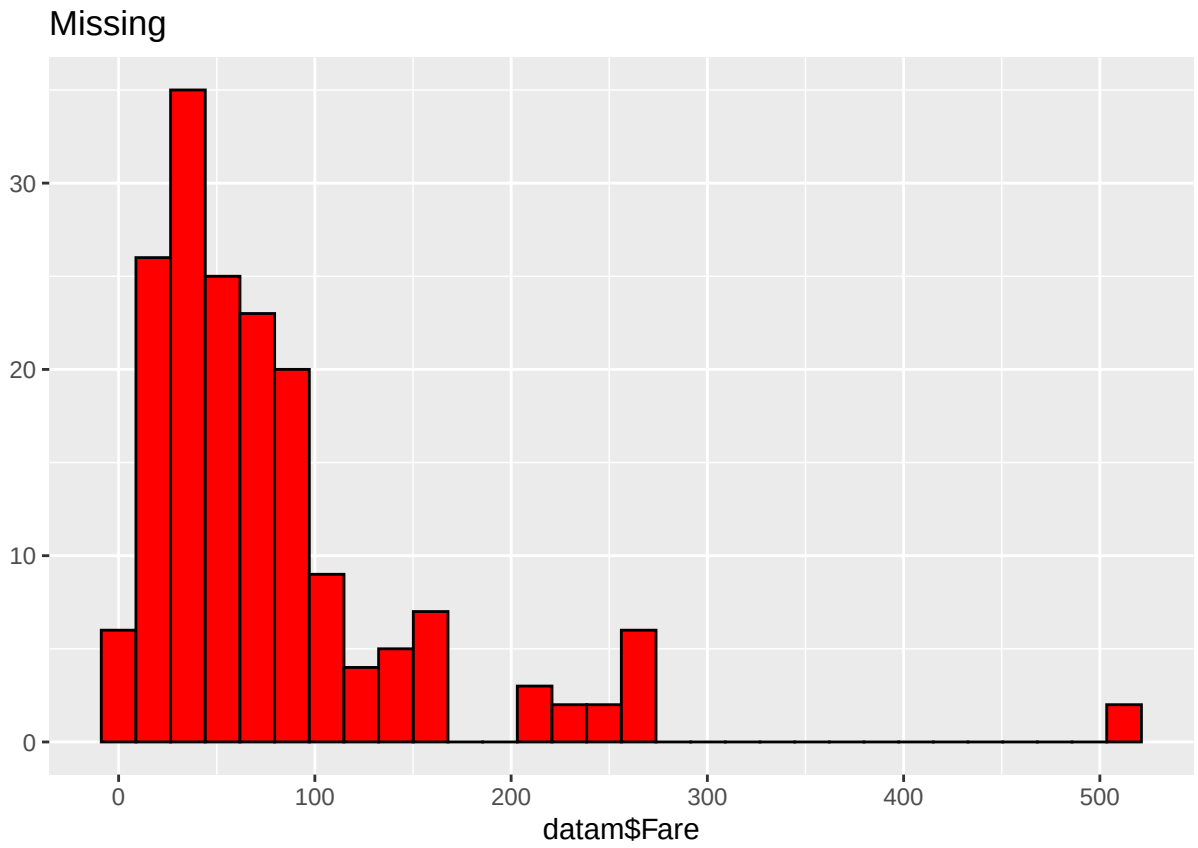

```
qplot(maxabs_age2$Age_maxabs, geom="histogram",fill=I("green"),color=I("black"),main = "Completo")
```



Tarifa sin estandarizar

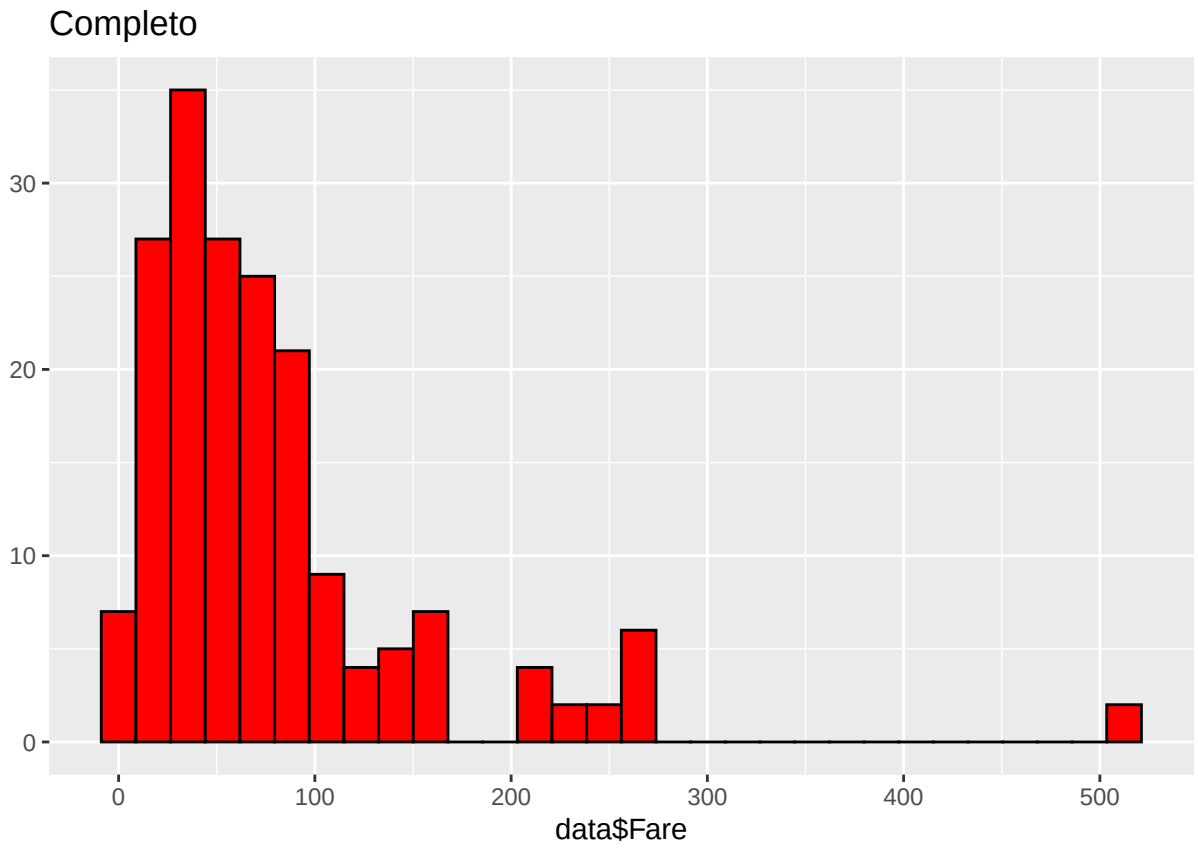
```
qplot(datam$Fare,geom="histogram",fill=I("red"),color=I("black"),main = "Missing")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
qplot(data$Fare,geom="histogram",fill=I("red"),color=I("black"),main = "Completo")
```

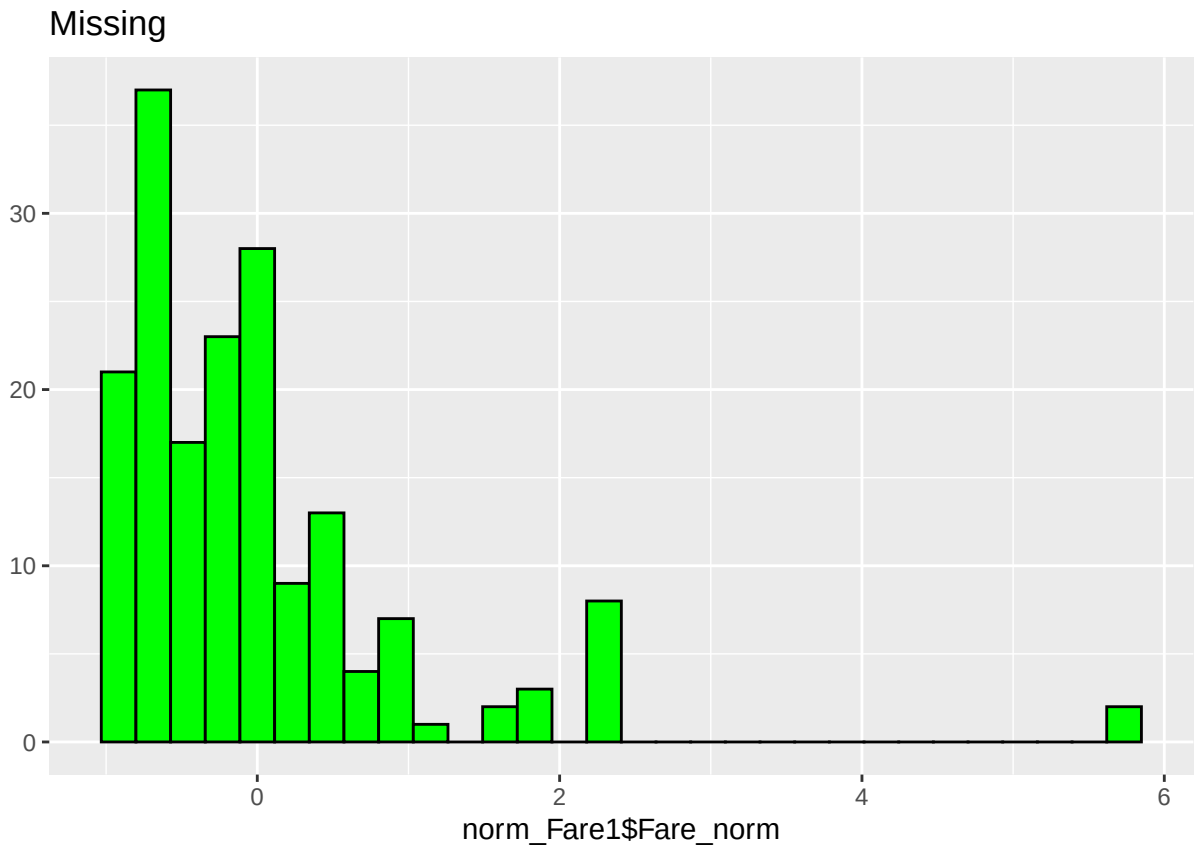
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Tarifa Normalizada

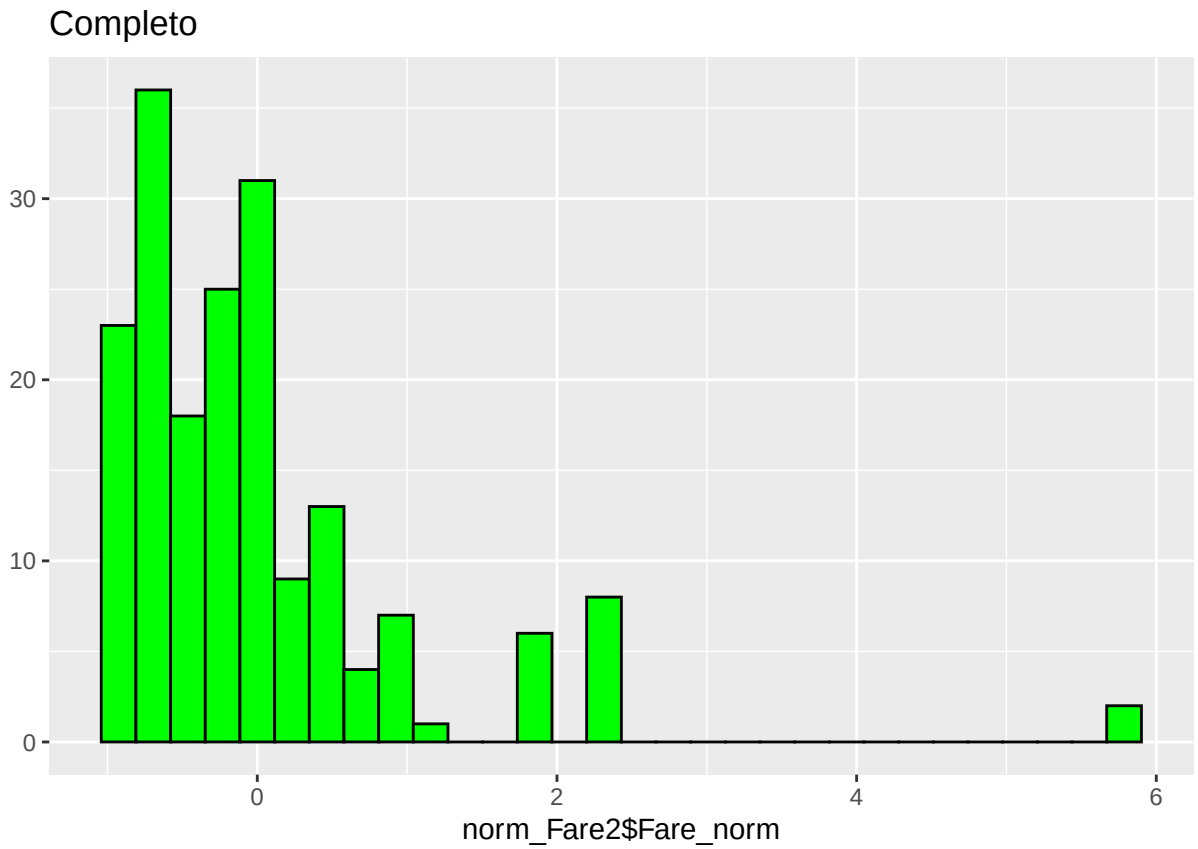
```
qplot(norm_Fare1$Fare_norm, geom="histogram",fill=I("green"),color=I("black"),main = "Missing")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
qplot(norm_Fare2$Fare_norm, geom="histogram",fill=I("green"),color=I("black"),main = "Completo")
```

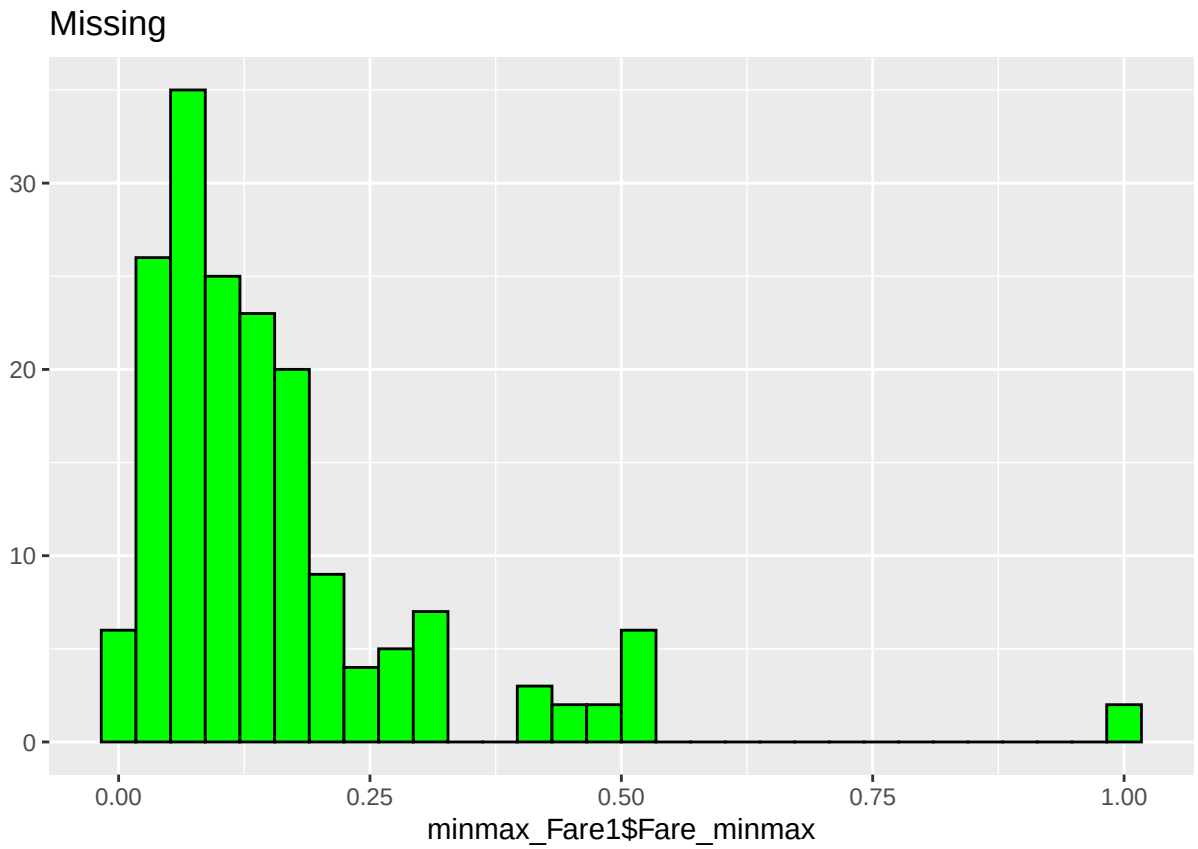
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Tarifa MinMaxScaling

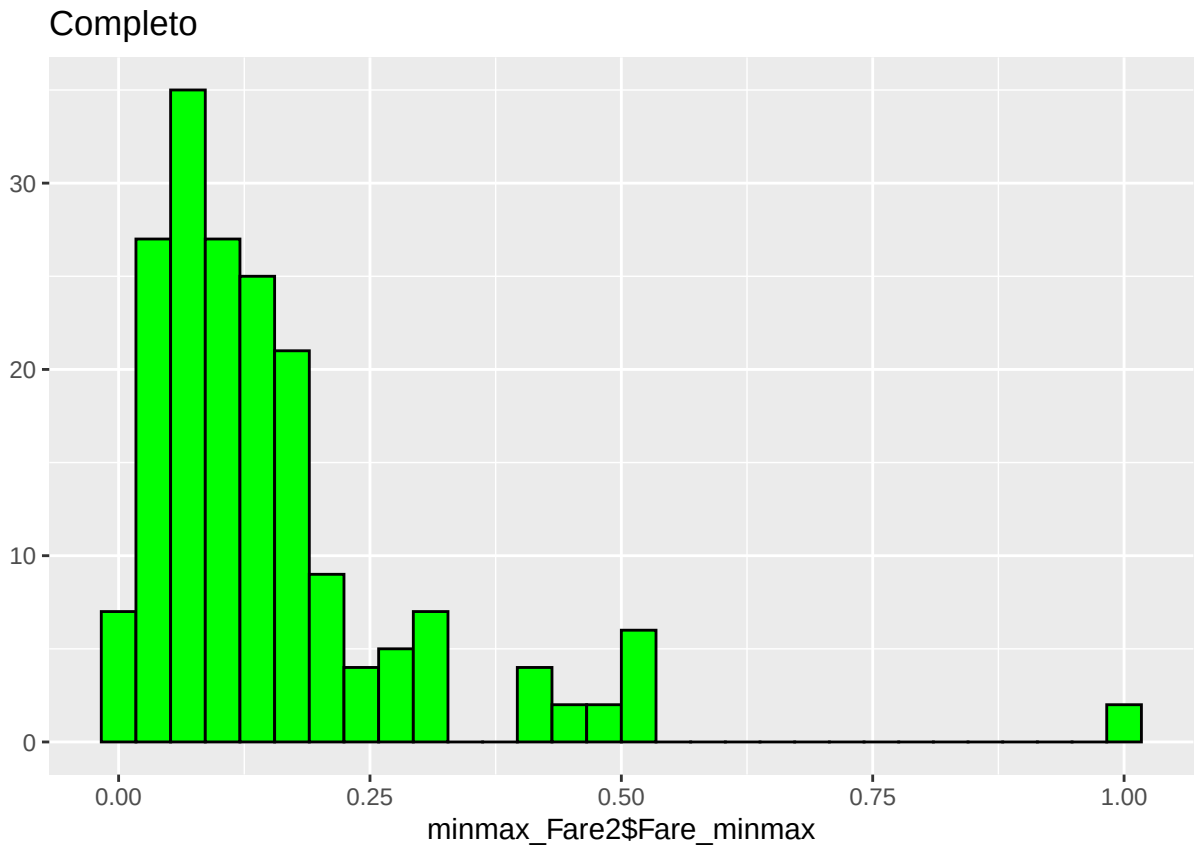
```
qplot(minmax_Fare1$Fare_minmax, geom="histogram",fill=I("green"),color=I("black"),main = "Missing")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
qplot(minmax_Fare2$Fare_minmax, geom="histogram",fill=I("green"),color=I("black"),main = "Completo")
```

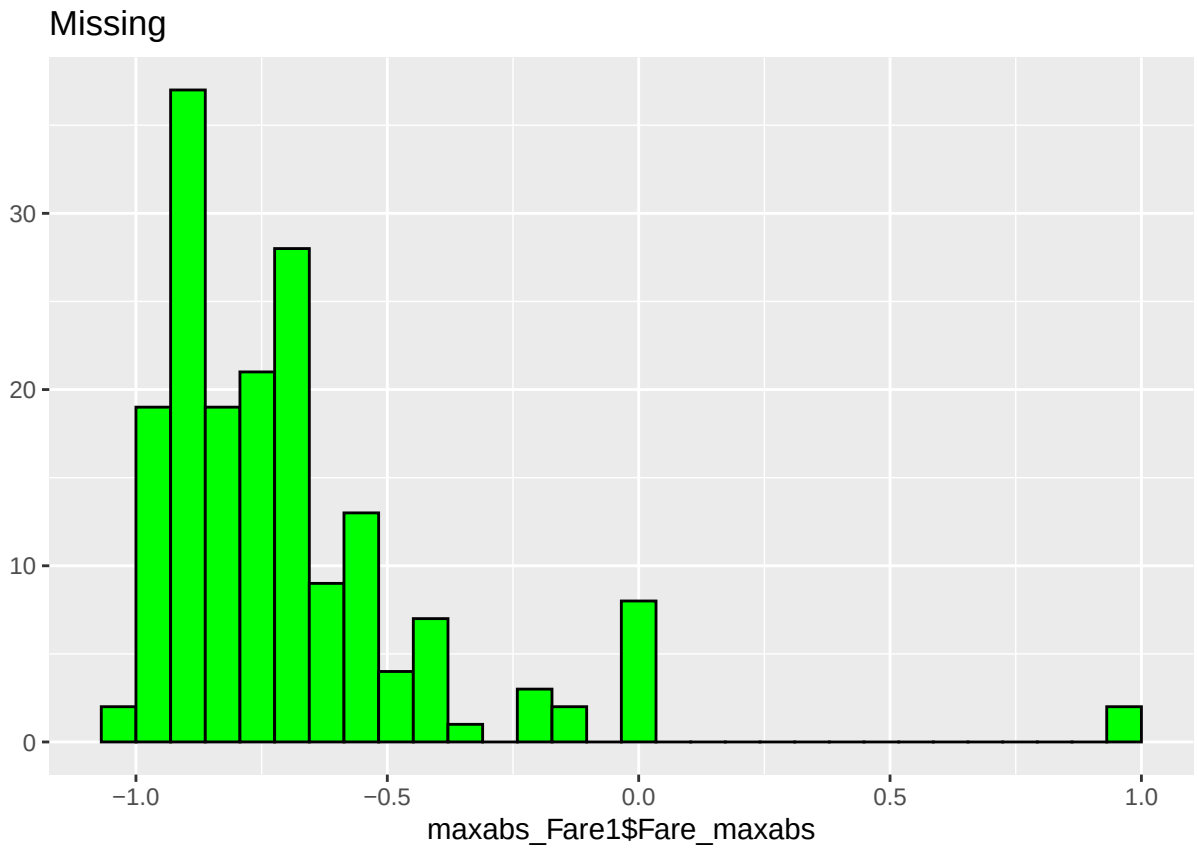
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Tarifa MaxAbsScaler

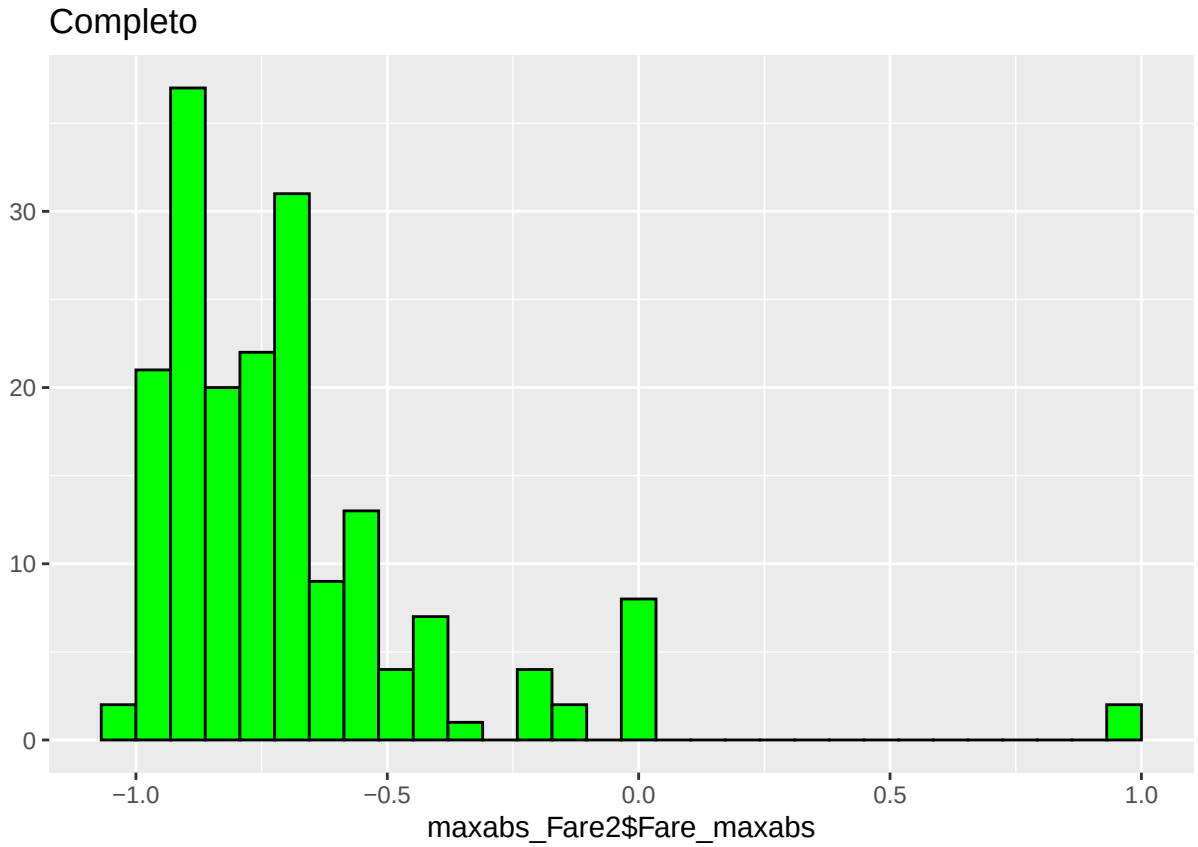
```
qplot(maxabs_Fare1$Fare_maxabs, geom="histogram",fill=I("green"),color=I("black"),main = "Missing")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
qplot(maxabs_Fare2$Fare_maxabs, geom="histogram",fill=I("green"),color=I("black"),main = "Completo")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Conclusiones

Para la variable de edad se logra ver con total claridad que el histograma de frecuencia con datos faltantes difiere considerablemente del histograma con datos completos. Sin embargo, tras realizar la estandarización de los datos (el efecto se presenta en los tres métodos usados) se observa que los histogramas de datos faltantes son mucho más similares. En el caso de la normalización y MinMaxScaler la cercanía es casoperfecta.

En la variabe de tarifa, vemos que los histogramas originales (antes de realizar la estandarización) son ya muy similares. Se considera que la cercanía se debe a que en esta columna la cantidad de datos faltantes era relativamente baja (8 de 183). Las transformaciones como era de esperarse tambien mantienen similitud entre si, pero difieren de la original, especialmente la delmétodo MinMaxScaling, la cual muestra una distribución más suavizada, ya que no cuenta con el cráter en su punto más alto que se presenta tanto en los histogramas originales como en los estandarizados con los otros métodos.

Finalmente se concluye que las estandarizaciones pueden ser una herramienta muy útil al trabajar con datos faltantes, ya que estas mejoran el rendimiento de los modelos, debido a que suavizan el efecto de sesgo que generan los datos faltantes en un set de datos. Adicionalmente permiten llegar a una distribución más apropiada, y que en la mayoría de ocasiones será mejor recibida por los modelos.