Laboratorio 8

Daniela Domínguez

11/15/2020

library(readr)  
library(psych)  
titanic\_MD <- read\_csv("titanic\_MD.csv")  
titanic\_MD$Sex <- gsub(pattern = "[?]",replacement = NA,x = titanic\_MD$Sex)  
head(titanic\_MD)

## # A tibble: 6 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 2 1 1 Cumi… <NA> 38 1 0 PC 17… 71.3 C85   
## 2 4 1 1 Futr… fema… 35 1 0 113803 53.1 C123   
## 3 7 0 1 McCa… male 54 0 0 17463 51.9 E46   
## 4 11 1 3 Sand… fema… NA 1 NA PP 95… 16.7 G6   
## 5 12 1 1 Bonn… fema… 58 NA 0 113783 26.6 C103   
## 6 22 1 2 Bees… male 34 0 0 248698 13 D56   
## # … with 1 more variable: Embarked <chr>

# 1. Reporte detallado de todas las columnas

La base de datos “titanic\_MD) es un dataset con missing data, es decir con datos faltantes. Está compuesta por 183 observaciones y 12 variables. Estas variables son: “PassengerId” que es un código único identificando a cada pasajero; “Survived” es una variable dummy que indica si el pasajero sobrevivió el accidente o no; “Pclass” indica si el pasajero viajaba en primera, segunda o tercera clase; “Name” indica el nombre del pasajero; “Sex” indica si la persona es de sexo femenino o masculino; “Age” indica la edad del pasajero al momento del accidente; “SibSp”; “Parch” si el pasajero estaba deshidratado; “Tickte” es el número de ticket del pasajero; “Fare” es la tarifa que el pasajero pagó por el viaje; “Cabin” es el número de cabina del pasajero; y “Embarked” es de qué puerto embarcó el cliente.

Algunas de las columnas tienen datos faltantes. Las columnas que están completas son: PassengerId, Survived, Pclass, Name, Ticket, y Cabin. Eso nos deja con 6 columnas con datos faltantes.

Sex: (chr) tiene únicamente 132 observaciones, lo que nos deja con 51 faltantes. Esto representa un porcentaje de 27.86% de datos faltantes. Age: (num) tiene 158 observaciones y 25 faltantes, lo que representa un 13.66% de data faltante. SibSp: (num) tiene 180 observaciones y 3 faltantes, es decir 1.63% de data faltante. Parch: (num) tiene 171 observaciones y 12 faltantes, lo que representa un 6.56% de data faltante. Fare: (num) tiene 175 observaciones y 8 observaciones, lo que representa 4.37% de data faltante. Embarked: (chr) tiene 171 observaciones y 12 faltantes, nuevamente esto es 6.56% de faltantes.

La variable que más faltantes tiene es sexo, con más de ¼ de faltantes, mientras que la variable SibSp es la que menos faltantes tiene.

summary(titanic\_MD)

## 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

describe(titanic\_MD)

## vars n mean sd median trimmed mad min max range  
## PassengerId 1 183 455.37 247.05 457.00 457.90 302.45 2.00 890.00 888.00  
## Survived 2 183 0.67 0.47 1.00 0.71 0.00 0.00 1.00 1.00  
## Pclass 3 183 1.19 0.52 1.00 1.05 0.00 1.00 3.00 2.00  
## Name\* 4 183 92.00 52.97 92.00 92.00 68.20 1.00 183.00 182.00  
## Sex\* 5 132 1.52 0.50 2.00 1.52 0.00 1.00 2.00 1.00  
## Age 6 158 35.69 15.64 35.50 35.48 17.05 0.92 80.00 79.08  
## SibSp 7 180 0.46 0.65 0.00 0.37 0.00 0.00 3.00 3.00  
## Parch 8 171 0.46 0.75 0.00 0.31 0.00 0.00 4.00 4.00  
## Ticket\* 9 183 64.26 36.27 64.00 64.50 45.96 1.00 127.00 126.00  
## Fare 10 175 78.96 77.03 56.93 64.94 45.04 0.00 512.33 512.33  
## Cabin\* 11 183 69.63 38.63 69.00 69.67 48.93 1.00 133.00 132.00  
## Embarked\* 12 171 2.30 0.95 3.00 2.37 0.00 1.00 3.00 2.00  
## skew kurtosis se  
## PassengerId -0.05 -1.12 18.26  
## Survived -0.73 -1.48 0.03  
## Pclass 2.64 5.82 0.04  
## Name\* 0.00 -1.22 3.92  
## Sex\* -0.06 -2.01 0.04  
## Age 0.12 -0.35 1.24  
## SibSp 1.44 2.37 0.05  
## Parch 1.56 2.12 0.06  
## Ticket\* -0.03 -1.16 2.68  
## Fare 2.69 10.31 5.82  
## Cabin\* 0.01 -1.20 2.86  
## Embarked\* -0.62 -1.61 0.07

# 2. Modelos a utilizar (hipótesis)

* Sex: como esta variable no es numérica, sino que aparece como “male” y “female”, en lugar de convertirla a una variable *dummy* se podría hacer una imputación general por la moda, ya que este acepta no solo número, sino que también con la frecuencia con que aparecen palabras.
* Age: Creo que lo major sería reemplazar los valores faltantes por el promedio de edad.
* SibSp: Esta variable posiblemetne sea más acertada utilizando un modelo de regresión lineal simple.
* Parch: Al igual que con la variable anterior, esta variable posiblemente sea más acertada trabajarla con un modelo de regresión lineal simple.
* Fare: De igual forma que SibSp y Parch, para esta variable posibelmente sea mejor usar un modelo de regresión lineal simple
* Embarked: esta variable al igual que Sex no es numérica. También usaría la moda para reemplazar los datos faltantes.

# 3. Filas completas

Como se había mencionado antes, las columnas que no tienen datos faltantes, es decir que están completos son las siguientes 6 variables: PassengerId, Survived, Pclass, Name, Ticket, y Cabin. Todas están compuestos por 183 observaciones.

PassengerId: (num) aunque solo tenemos 183 observaciones, el id de todos los pasajeros no son números contínuos (por ejmplo de 1 a 183), sino que están en un rango entre 2 y 890. Survived: (num) es una variable Dummy. 1 representa si sobrevivió y 0 si no sobrevivió. De nuestra base de datos 123 pasajeros sobrevivieron y 60 fallecieron en el incidente. Pclass: (num) esta variable indica si el pasajero viajaba en primera, segunda o tercera clase. De nuestra base de datos 158 pasajeros viajaban en 1era clase, 15 pasajeros viajaban en 2nda clase, y 10 pasajeros viajaban en 3era clase. Name: (chr) indica el nombre del pasajero. Está en formato “Nombre, Mr/Mrs/Miss/Dr Apellido(s)”, aunque también hay algunos que tienen también su título como por ejemplo “Countess” o “Captain”, entre otros. Ticket: (chr) La razón por la que este dato no es numérico, es porque en algunas ocasiones el número de ticket tiene indicado un código en letra, por ejemplo “WE/P 5735”. Cabin: (chr) Este dato está compuesto por una letra (representando en qué nivel del barco está ubicado) más un número indicando el número de cabina. Hay pasajeros que tenían pedidos hasta cuatro cabinas para una misma persona.

# 4. Prueba con distintos modelos

Para saber qué modelo es el que más acertado sería para cada variable, se hicieron pruebas con los siguientes métodos para *missing values*: Pairwise; Imputación general usando la media, la mediana, la moda y reemplacando con cero; Imputación sectorizada; Modelo de regresión lineal simple, Eliminación de outliers: Standard deviation approach; y por último Eliminación de outliers: Percentile approach. A continuación se presenta el proceso de cada una de estas.

## Pairwise

a <- titanic\_MD  
mean(a$Age)

## [1] NA

mean(a$Age,na.rm = TRUE)

## [1] 35.69253

cor(a$Age,a$Fare, use = "pairwise.complete.obs")

## [1] -0.1309791

a[!is.na(a$Sex),]

## # A tibble: 132 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 11 1 3 Sand… fema… NA 1 NA PP 95… 16.7 G6   
## 4 12 1 1 Bonn… fema… 58 NA 0 113783 26.6 C103   
## 5 22 1 2 Bees… male 34 0 0 248698 13 D56   
## 6 53 1 1 Harp… fema… 49 1 NA PC 17… 76.7 D33   
## 7 55 0 1 Ostb… male 65 0 1 113509 62.0 B30   
## 8 63 0 1 Harr… male 45 1 0 36973 83.5 C83   
## 9 67 1 2 Nye,… fema… 29 0 0 C.A. … 10.5 F33   
## 10 76 0 3 Moen… male 25 0 0 348123 NA F G73  
## # … with 122 more rows, and 1 more variable: Embarked <chr>

cor(a$Age,a$Fare, use = "pairwise.complete.obs")

## [1] -0.1309791

cor(a[,c(6:8,10)],use = "pairwise.complete.obs")

## 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

## Imputación general

### Media

b1 <- titanic\_MD  
b1$Age[is.na(b1$Age)] <- mean(b1$Age,na.rm = TRUE)  
b1$SibSp[is.na(b1$SibSp)] <- mean(b1$SibSp,na.rm = TRUE)  
b1$Parch[is.na(b1$Parch)] <- mean(b1$Parch,na.rm = TRUE)  
b1$Fare[is.na(b1$Fare)] <- mean(b1$Fare,na.rm = TRUE)  
mb1 <- summary(b1[,c(6:8,10)])  
summary(b1[,c(6:8,10)])

## Age SibSp Parch Fare   
## Min. : 0.92 Min. :0.0000 Min. :0.000 Min. : 0.00   
## 1st Qu.:25.00 1st Qu.:0.0000 1st Qu.:0.000 1st Qu.: 30.00   
## Median :35.69 Median :0.0000 Median :0.000 Median : 61.17   
## Mean :35.69 Mean :0.4611 Mean :0.462 Mean : 78.96   
## 3rd Qu.:46.50 3rd Qu.:1.0000 3rd Qu.:1.000 3rd Qu.: 90.00   
## Max. :80.00 Max. :3.0000 Max. :4.000 Max. :512.33

### Moda

b3 <- titanic\_MD  
b31 <- b3$Age  
b3$Age[is.na(b3$Age)] <- as.numeric(names(table(b31))[table(b31)==max(table(b31))])  
b32 <- b3$SibSp  
b3$SibSp[is.na(b3$SibSp)] <- as.numeric(names(table(b32))[table(b32)==max(table(b32))])  
b33 <- b3$Parch  
b3$Parch[is.na(b3$Parch)] <- as.numeric(names(table(b33))[table(b33)==max(table(b33))])  
b34 <- b3$Fare  
b3$Fare[is.na(b3$Fare)] <- as.numeric(names(table(b34))[table(b34)==max(table(b34))])  
b35 <- b3$Sex  
b3$Sex[is.na(b3$Sex)] <- names(table(b35))[table(b35)==max(table(b35))]  
b36 <- b3$Embarked  
b3$Embarked[is.na(b3$Embarked)] <- names(table(b36))[table(b36)==max(table(b36))]  
  
mb3 <- summary(b3[,c(6:8,10)])  
summary(b3[,c(5:8,10,12)])

## Sex Age SibSp Parch   
## Length:183 Min. : 0.92 Min. :0.0000 Min. :0.0000   
## Class :character 1st Qu.:24.00 1st Qu.:0.0000 1st Qu.:0.0000   
## Mode :character Median :31.00 Median :0.0000 Median :0.0000   
## Mean :34.10 Mean :0.4536 Mean :0.4317   
## 3rd Qu.:46.50 3rd Qu.:1.0000 3rd Qu.:1.0000   
## Max. :80.00 Max. :3.0000 Max. :4.0000   
## Fare Embarked   
## Min. : 0.00 Length:183   
## 1st Qu.: 26.55 Class :character   
## Median : 55.44 Mode :character   
## Mean : 76.67   
## 3rd Qu.: 90.00   
## Max. :512.33

### Mediana

b2 <- titanic\_MD  
b2$Age[is.na(b2$Age)] <- median(b2$Age,na.rm = TRUE)  
b2$SibSp[is.na(b2$SibSp)] <- median(b2$SibSp,na.rm = TRUE)  
b2$Parch[is.na(b2$Parch)] <- median(b2$Parch,na.rm = TRUE)  
b2$Fare[is.na(b2$Fare)] <- median(b2$Fare,na.rm = TRUE)  
mb2 <- summary(b2[,c(6:8,10)])  
summary(b2[,c(6:8,10)])

## Age SibSp Parch Fare   
## Min. : 0.92 Min. :0.0000 Min. :0.0000 Min. : 0.00   
## 1st Qu.:25.00 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.: 30.00   
## Median :35.50 Median :0.0000 Median :0.0000 Median : 56.93   
## Mean :35.67 Mean :0.4536 Mean :0.4317 Mean : 78.00   
## 3rd Qu.:46.50 3rd Qu.:1.0000 3rd Qu.:1.0000 3rd Qu.: 90.00   
## Max. :80.00 Max. :3.0000 Max. :4.0000 Max. :512.33

### Cero

b4 <- titanic\_MD  
b4$Age[is.na(b4$Age)] <- 0  
b4$SibSp[is.na(b4$SibSp)] <- 0  
b4$Parch[is.na(b4$Parch)] <- 0  
b4$Fare[is.na(b4$Fare)] <- 0  
mb4 <- summary(b4[,c(6:8,10)])  
summary(b4[,c(6:8,10)])

## Age SibSp Parch Fare   
## Min. : 0.00 Min. :0.0000 Min. :0.0000 Min. : 0.00   
## 1st Qu.:19.00 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.: 26.55   
## Median :31.00 Median :0.0000 Median :0.0000 Median : 55.44   
## Mean :30.82 Mean :0.4536 Mean :0.4317 Mean : 75.51   
## 3rd Qu.:46.50 3rd Qu.:1.0000 3rd Qu.:1.0000 3rd Qu.: 90.00   
## Max. :80.00 Max. :3.0000 Max. :4.0000 Max. :512.33

## Imputación sectorizada

summary(cars)

## speed dist   
## Min. : 4.0 Min. : 2.00   
## 1st Qu.:12.0 1st Qu.: 26.00   
## Median :15.0 Median : 36.00   
## Mean :15.4 Mean : 42.98   
## 3rd Qu.:19.0 3rd Qu.: 56.00   
## Max. :25.0 Max. :120.00

## Modelo de regresión lineal simple

Para este modelo no se hizo una función que trabajara todas las columnas deseadas al mismo tiempo. Se trabajó cada variable una por una, lo que hace que el código sea muy largo, pero todos se trabajaron de la misma forma. A continuación se presenta únicamente lo que se trabajó con la variable Age.

d <- titanic\_MD  
  
Ind <- function(t){  
 x <- dim(length(t))  
 x[which(!is.na(t))] = 1  
 x[which(is.na(t))] = 0  
 return(x)  
}  
d$Iage <- Ind(d$Age)  
d$ISibSp <- Ind(d$SibSp)  
d$IParch <- Ind(d$Parch)  
d$IFare <- Ind(d$Fare)  
d

## # A tibble: 183 x 16  
## PassengerId Survived Pclass Name Sex Age SibSp Parch Ticket Fare Cabin  
## <dbl> <dbl> <dbl> <chr> <chr> <dbl> <dbl> <dbl> <chr> <dbl> <chr>  
## 1 2 1 1 Cumi… <NA> 38 1 0 PC 17… 71.3 C85   
## 2 4 1 1 Futr… fema… 35 1 0 113803 53.1 C123   
## 3 7 0 1 McCa… male 54 0 0 17463 51.9 E46   
## 4 11 1 3 Sand… fema… NA 1 NA PP 95… 16.7 G6   
## 5 12 1 1 Bonn… fema… 58 NA 0 113783 26.6 C103   
## 6 22 1 2 Bees… male 34 0 0 248698 13 D56   
## 7 24 1 1 Slop… <NA> NA 0 0 113788 35.5 A6   
## 8 28 0 1 Fort… <NA> 19 3 2 19950 263 C23 …  
## 9 53 1 1 Harp… fema… 49 1 NA PC 17… 76.7 D33   
## 10 55 0 1 Ostb… male 65 0 1 113509 62.0 B30   
## # … with 173 more rows, and 5 more variables: Embarked <chr>, Iage <dbl>,  
## # ISibSp <dbl>, IParch <dbl>, IFare <dbl>

cor(a[,c(6:8,10)],use = "complete.obs")

## Age SibSp Parch Fare  
## Age 1.00000000 -0.05113744 -0.2920879 -0.1036419  
## SibSp -0.05113744 1.00000000 0.1760052 0.2442869  
## Parch -0.29208793 0.17600517 1.0000000 0.3708889  
## Fare -0.10364194 0.24428693 0.3708889 1.0000000

# Por la correlación, usaremos:  
# Parch para predecir Age  
# Fare para predecir SibSp  
# Fare para predecir Parch  
# y Parch para predecir Fare

# Age:  
lm(Age ~ Parch, data = d)

##   
## Call:  
## lm(formula = Age ~ Parch, data = d)  
##   
## Coefficients:  
## (Intercept) Parch   
## 38.08 -5.55

summary(lm(Age ~ Parch, data = d))

##   
## Call:  
## lm(formula = Age ~ Parch, data = d)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -30.527 -10.190 -2.078 9.922 48.123   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 38.078 1.438 26.476 < 2e-16 \*\*\*  
## Parch -5.550 1.578 -3.518 0.00058 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 14.88 on 146 degrees of freedom  
## (35 observations deleted due to missingness)  
## Multiple R-squared: 0.07815, Adjusted R-squared: 0.07183   
## F-statistic: 12.38 on 1 and 146 DF, p-value: 0.0005798

for (i in 1:nrow(d)) {  
 if (d$Iage[i] == 0 && d$IParch[i] != 0 ) {  
 d$Age[i] <- 37.98950 - 0.02655\*d$Parch[i]  
 } else if (d$Iage[i] == 0 && d$IParch[i] == 0 ) {  
 d$Age[i] <- 37.98950 - 0.02655\*0  
 }  
}  
  
summary(d[6])

## Age   
## Min. : 0.92   
## 1st Qu.:25.00   
## Median :37.94   
## Mean :36.01   
## 3rd Qu.:46.50   
## Max. :80.00

## Eliminación de outliers: Standard deviation approach

Igual que con el método de *missing data* anterior, para este modelo no se hizo una función que trabajara todas las columnas deseadas al mismo tiempo. Se trabajó cada variable una por una, lo que hace que el código sea muy largo, pero todos se trabajaron de la misma forma. A continuación, se presenta únicamente lo que se trabajó con la variable Age. Los siguientes métodos hará lo mismo.

e <- titanic\_MD  
x <- 3  
# Age  
lower1 <- mean(e$Age, na.rm = TRUE) - sd(e$Age, na.rm = TRUE \* x)  
upper1 <- mean(e$Age, na.rm = TRUE) + sd(e$Age, na.rm = TRUE \* x)  
lower1

## [1] 20.05167

upper1

## [1] 51.33339

e1 <- e[(e$Age>=lower1) & (e$Age<=upper1) & (!is.na(e$Age)),]  
summary(e1$Age)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 21.00 27.25 35.50 35.33 42.75 51.00

## Eliminación de outliers: Percentile approach

f <- titanic\_MD  
# Age  
p1\_lower <- quantile(f$Age,na.rm = TRUE,probs = 0.1)  
p1\_upper <- quantile(f$Age,na.rm = TRUE,probs = 0.9)  
p1\_lower

## 10%   
## 17.7

p1\_upper

## 90%   
## 56

f1 <- f[(f$Age>=p1\_lower) & (f$Age<=p1\_upper) & (!is.na(f$Age)),]  
summary(f1$Age)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 18.00 26.00 36.00 35.63 45.50 56.0

## 5. Comparación de métodos

Adjunto a este archive se encuentra un Excel llamado “Resultados\_abreviados.xlsx”, en la que se ordenó y juntó los resultados de estas pruebas para que sea más fácil de entender a simple vista. Sin embargo, adjuntamos aquí los resultados más relevantes.

En la primera columna dice qué valor es el que se está mostrando, por ejemplo el número mínimo o la media. La segunda columna muestra los datos del dataset completo llamado “titanic.csv” para tener con qué hacer la comparación. La tercera columna muestra los datos con los diferentes métodos con los que se trabajó el *missing data*. Y por último se muestra el porcentaje de diferencia que hubo entre los datos del modelo versus los originales completos.

Las siguientes tablas muestran qué modelo tuvo menos diferencias contra la base de datos completa. Para Fare lo que mejor funcionó fue usar una regresión lineal simple. Para SibsSp lo que mejor funcionó fue usar una imputación general utilizando la moda. Para Age fue similar, solo que en lugar de usar la moda, lo mejor fue usar la media. Y para Parch hubo un empate entre la una imputación general utilizando la media o utilizando la mediana.

Para Age y para Embarked, lo mejor fue utilizar la imputación general utilizando la moda.

Table

Description automatically generated

Table

Description automatically generated

Table

Description automatically generated

Table

Description automatically generated

## 6. Conclusiones

Cuando se está trabajando con datos, la mayoría de las veces estos están en desorden o más comúnmente tienen datos faltantes. Por eso frecuentemente uno se tarda muchísimo más tiempo limpiando y organizando la data que programar y crear algoritmos, etc.

Es importante entender que cuando se tienen datos faltantes puede ser por varias razones. Una de ellas puede ser que la persona a quien se entrevistó no quiso contestar una pregunta, pero esto en sí mismo no proporciona información que el individuo no se sentía cómodo con un tema. También puede ser que simplemente al recolectar los datos no se hizo correctamente.

Antes de conocer de este tema, las únicas formas que yo conocía para lidear con datos faltantes o NAs, era obviando completamente las filas, o mi segunda opción era sustituír los faltantes con el promedio de la variable. Esto sin embargo, no servía cuando se tenían datos en caracteres (por ejemplo nombres de países) en lugar de ser numéricos (5, 7, 18,e tc.).

Por esto es por lo que me sorprendió bastante cuando conocí que había muchísimos más métodos que únicamente estos dos. Existen incluso más formas de trabajar *missing* data de los que se utilizaron para este trabajo.

Los menos útiles en el caso de los datos del titanic fueron los dos métodos de eliminación de outliers, tanto el Standard deviation approach como el de Percentile apprach. Cortaba demasiados datos que no se alejaban tanto de el resto de los datos. Otro método que no usaría frecuentemente es el de imputación general con cero, ya que crea un gran sesgo al resto de datos.

# PARTE 2

library(psych)  
library(dplyr)  
# install.packages("gensvm")  
library(gensvm)  
# install.packages("sparklyr")  
library(sparklyr)

## Standarization

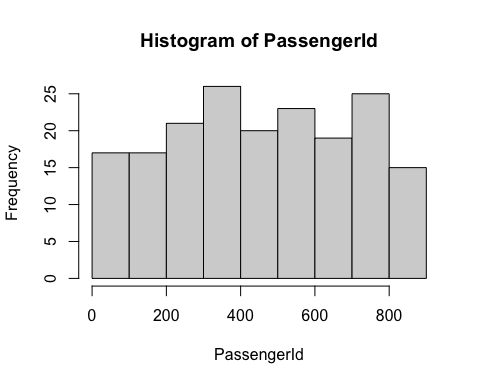
A continuación se presenta el proceso de trabajo de la variable PassengerId para estandarizar los valores a valores Z. La fórmula que se usó para esto es la siguiente.

A picture containing object, clock

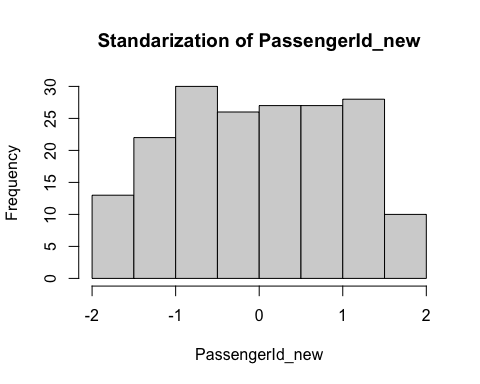
Description automatically generated

Además se muestra un histograma con los valores originales de esta variable, al igual que un seguno histograma con los valores estandarizados como valores Z.

a21 <- titanic\_MD  
  
# PassengerID  
a211 <- a21 %>%   
 mutate(PassengerId\_new = ((PassengerId - mean(PassengerId,na.rm = TRUE))/sd(PassengerId,na.rm = TRUE))) %>%   
 select(PassengerId,PassengerId\_new)   
hist(a21$PassengerId, main = paste("Histogram of PassengerId"), xlab = "PassengerId")



hist(a211$PassengerId\_new,main = paste("Standarization of PassengerId\_new"), xlab = "PassengerId\_new")

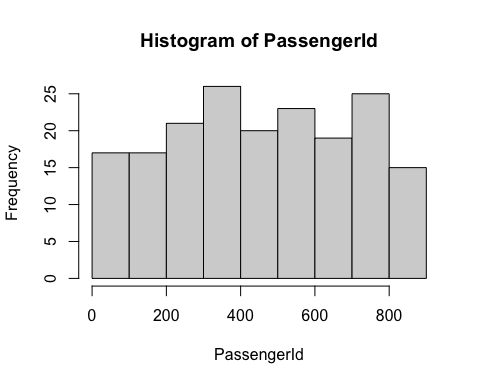


summary(a211)

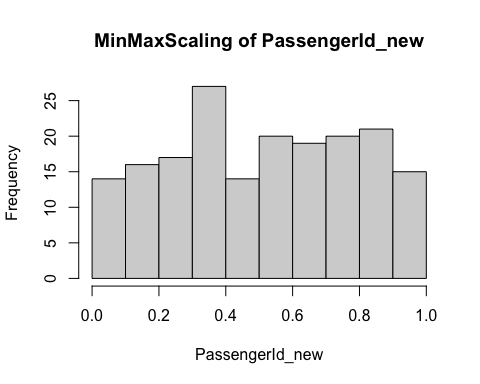
## PassengerId PassengerId\_new   
## Min. : 2.0 Min. :-1.835101   
## 1st Qu.:263.5 1st Qu.:-0.776621   
## Median :457.0 Median : 0.006613   
## Mean :455.4 Mean : 0.000000   
## 3rd Qu.:676.0 3rd Qu.: 0.893065   
## Max. :890.0 Max. : 1.759278

## MinMaxScaling

b22 <- titanic\_MD  
  
# PassengerID  
b221 <- b22 %>%   
 mutate(PassengerId\_new = ((PassengerId - min(PassengerId,na.rm = TRUE))/  
 (max(PassengerId,na.rm = TRUE) - min(PassengerId,na.rm = TRUE)))) %>%   
 select(PassengerId,PassengerId\_new)   
hist(b22$PassengerId, main = paste("Histogram of PassengerId"), xlab = "PassengerId")



hist(b221$PassengerId\_new,main = paste("MinMaxScaling of PassengerId\_new"), xlab = "PassengerId\_new")



summary(b221)

## PassengerId PassengerId\_new   
## Min. : 2.0 Min. :0.0000   
## 1st Qu.:263.5 1st Qu.:0.2945   
## Median :457.0 Median :0.5124   
## Mean :455.4 Mean :0.5105   
## 3rd Qu.:676.0 3rd Qu.:0.7590   
## Max. :890.0 Max. :1.0000

## MaxAbsScaler

Solo se puede hacer con survived porque es la única variable que se encuentra entre -1 y 1.

c22 <- titanic\_MD  
  
# Survived  
c221 <- gensvm.maxabs.scale(c22$Survived,x.test = NULL)  
summary(c221)

## V1   
## Min. :0.0000   
## 1st Qu.:0.0000   
## Median :1.0000   
## Mean :0.6721   
## 3rd Qu.:1.0000   
## Max. :1.0000

# Comparación

Para poder hacer la comparación de las columnas numéricas con los últimos tres métodos presentados con los valores del dataset completo “titanic.csv”, es necesario convertir también las columnas de estos datos originales para que puedan ser comparables.

titanic <- read\_csv("titanic.csv")  
  
# PassengerId  
d211 <- titanic %>%   
 mutate(PassengerId\_new = ((PassengerId - mean(PassengerId,na.rm = TRUE))/sd(PassengerId,na.rm = TRUE))) %>%   
 select(PassengerId,PassengerId\_new)   
summary(d211)

## PassengerId PassengerId\_new   
## Min. : 2.0 Min. :-1.835101   
## 1st Qu.:263.5 1st Qu.:-0.776621   
## Median :457.0 Median : 0.006613   
## Mean :455.4 Mean : 0.000000   
## 3rd Qu.:676.0 3rd Qu.: 0.893065   
## Max. :890.0 Max. : 1.759278

summary(cars)

## speed dist   
## Min. : 4.0 Min. : 2.00   
## 1st Qu.:12.0 1st Qu.: 26.00   
## Median :15.0 Median : 36.00   
## Mean :15.4 Mean : 42.98   
## 3rd Qu.:19.0 3rd Qu.: 56.00   
## Max. :25.0 Max. :120.00

Adjunto a este archive se encuentra un Excel llamado “Resultados\_abreviados.xlsx”, en la que se ordenó y juntó los resultados de estas pruebas para que sea más fácil de entender a simple vista. Sin embargo, adjuntamos aquí los resultados más relevantes.