HandsOn 02

This is an R Markdown Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Ctrl+Shift+Enter*.

```
library(dlookr)
library(kableExtra)
env <- new.env()
with(env, library(dlookr))
```

```
## install.packages("knitr")
## install.packages("tidytext")
library(ggplot2)
library(vegan)
```

```
Loading required package: permute
Loading required package: lattice
This is vegan 2.6-4
```

```
library(conflicted)
library(dplyr)
library(tidyverse)
```

```
── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
✔ forcats   1.0.0     ✔ stringr   1.5.0
✔ lubridate 1.9.2     ✔ tibble    3.2.1
✔ purrr     1.0.1     ✔ tidyr     1.3.0
✔ readr     2.1.4
```

```
## with(env, knitr::knit("Hands_On 02.Rmd", "Hands_On 02.pdf"))
```

Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Ctrl+Alt+I*.

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Ctrl+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.

CALIDAD DE DATOS Y PRE-PROCESAMIENTO

1.- Evaluación de la Calidad de los Datos

Cargar los siguientes paquetes dplyr, na.tools, tidyimpute (versión de github decisionpatterns/tidyimpute") Cargar el data set carInsurance que trata de las puntuaciones de riesgo de seguro de los carros basado en varias caracteristicas de cada carro

a) Revisar si hay algún valor no agregado

```
load("C:/Users/Edgar/Documents/GitHub/DataMining_and_MachineLearning_EdgarV/data/HandsOn_Data/carInsurance.Rdat
a")
## ls()
df <- carIns
##df
any(is.na(df))
```

```
[1] TRUE
```

```
## si hay valores NA
```

b) Contar el numero de casos que tienen almenos un valor no agregado

```
## ls()
df <- carIns
##df
df1 <- df %>%
    dplyr::filter(any(is.na(df)))%>%
    count()
## el numero de NA´s es:
df1
```

n
<int>
205

1 row

c) crear un nuevo data set a partir de la remoción de todos los casos que tienen valores no agregados

```
load("C:/Users/Edgar/Documents/GitHub/DataMining_and_MachineLearning_EdgarV/data/HandsOn_Data/carInsurance.Rdat
a")
## ls()
df <- carIns
new_dataset <- df %>%
    drop_na()
## de esta forma se eliminan las filas que contienen NA
head(new_dataset)
```

sy…	normLoss	m…	fuelType	aspiration	nDoors	bodyStyle	driveWheels	engineLocation	wheelBase
<int>	<int>	<fctr>	<fctr>	<fctr>	<fctr>	<fctr>	<fctr>	<fctr>	<dbl>
2	164	audi	gas	std	four	sedan	fwd	front	99.8
2	164	audi	gas	std	four	sedan	4wd	front	99.4
1	158	audi	gas	std	four	sedan	fwd	front	105.8
1	158	audi	gas	turbo	four	sedan	fwd	front	105.8
2	192	bmw	gas	std	two	sedan	rwd	front	101.2
0	192	bmw	gas	std	four	sedan	rwd	front	101.2

6 rows | 1-10 of 26 columns

d) crear un nuevo dataset a partir del reemplazo de todos los valores no agregados por 0

```
load("C:/Users/Edgar/Documents/GitHub/DataMining_and_MachineLearning_EdgarV/data/HandsOn_Data/carInsurance.Rdat
a")
## ls()
df <- carIns
new_dataset <- df %>%
    mutate_all(~ifelse(is.na(.), 0, .))
## observamos que se reemplaza todos los NA por 0 y en el caso que haya un NA en categorias, le cambia a int y la
categoria  se transforma en numero
head(new_dataset)
```

sy…	normLoss	m…	fuelType	aspiration	nDoors	bodyStyle	driveWheels	engineLocation	wheelBase
<int>	<dbl>	<int>	<int>	<int>	<dbl>	<int>	<int>	<int>	<dbl>
3	0	1	2	1	2	1	3	1	88.6
3	0	1	2	1	2	1	3	1	88.6
1	0	1	2	1	2	3	3	1	94.5
2	164	2	2	1	1	4	2	1	99.8
2	164	2	2	1	1	4	1	1	99.4
2	0	2	2	1	2	4	2	1	99.8

6 rows | 1-10 of 26 columns

e) Crear un nuevo dataset a partir del ingreso del promedio en todas las columnas las cuales tienen sus datos de tipo double

```
load("C:/Users/Edgar/Documents/GitHub/DataMining_and_MachineLearning_EdgarV/data/HandsOn_Data/carInsurance.Rdat
a")
## ls()
df <- carIns
new_dataset <- df %>%
    mutate_all(~ifelse(is.na(.), 0, .))

media_doubles <- new_dataset %>%
                  select_if(is.double) %>%
                  summarize(across(everything(), mean))

media_doubles
```

normLoss	nDoors	wheelBase	length	width	height	bore	stroke	compressionRatio	horsePower
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
97.6	1.42439	98.75659	174.0493	65.9078	53.72488	3.26478	3.191902	10.14254	103.239

1 row | 1-10 of 12 columns

```
NA
```

f) Crear un nuevo dataset a partir del ingreso de la moda en todas las columnas con valores de tipo Integer

```
load("C:/Users/Edgar/Documents/GitHub/DataMining_and_MachineLearning_EdgarV/data/HandsOn_Data/carInsurance.Rdat
a")
## ls()
df <- carIns
new_dataset <- df %>%
    mutate_all(~ifelse(is.na(.), 0, .))
############################FUNCION MODA##############################
Moda <- function(x) {
  ux <- unique(x)
  ux[which.max(tabulate(match(x, ux)))]
}
######################################################################
moda_integer <- new_dataset %>%
                  select_if(is.integer) %>%
                  summarize(across(everything(), Moda))

moda_integer
```

sy…	m…	fuelType	aspiration	bodyStyle	driveWheels	engineLocation	curbWeight	engineType
<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>
0	20	2	1	4	2	1	2385	4

1 row | 1-9 of 14 columns

```
NA
```

f) Crear un nuevo data set a partir del ingreso de valores mas frecuentes para la columna “inDoors”.

```
load("C:/Users/Edgar/Documents/GitHub/DataMining_and_MachineLearning_EdgarV/data/HandsOn_Data/carInsurance.Rdat
a")
## ls()
df <- carIns
new_dataset <- df %>%
    mutate_all(~ifelse(is.na(.), 0, .))

tabla_nDoors <- new_dataset %>%
                select(nDoors)
tabla_nDoors
```

nDoors
<dbl>
2
2
2
1
1
2
1
1
1
2

1-10 of 205 rows Previous 1 2 3 4 5 6 ... 21 Next

```
tabla_mas_frecuente <- table(tabla_nDoors)

valor_mas_frecuente <- names(tabla_mas_frecuente)[tabla_mas_frecuente == max(tabla_mas_frecuente)]

## muestra el valor mas frecuente entre 0, 1, 2
valor_mas_frecuente
```

```
[1] "1"
```

g) Combinar los tres últimas imputaciones para obtener un dataset final, ¿hay algunos casos duplicados?

```
load("C:/Users/Edgar/Documents/GitHub/DataMining_and_MachineLearning_EdgarV/data/HandsOn_Data/carInsurance.Rdat
a")
## ls()
df <- carIns
new_dataset <- df %>%
    mutate_all(~ifelse(is.na(.), 0, .))

media_doubles <- new_dataset %>%
                  select_if(is.double) %>%
                  summarize(across(everything(), mean))
##########DATASET 1###############
media_doubles
```

normLoss	nDoors	wheelBase	length	width	height	bore	stroke	compressionRatio	horsePower
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
97.6	1.42439	98.75659	174.0493	65.9078	53.72488	3.26478	3.191902	10.14254	103.239

1 row | 1-10 of 12 columns

```
##################################

############################FUNCION MODA##############################
Moda <- function(x) {
  ux <- unique(x)
  ux[which.max(tabulate(match(x, ux)))]
}
######################################################################
moda_integer <- new_dataset %>%
                  select_if(is.integer) %>%
                  summarize(across(everything(), Moda))
##########DATASET 2###############
moda_integer
```

sy…	m…	fuelType	aspiration	bodyStyle	driveWheels	engineLocation	curbWeight	engineType
<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>
0	20	2	1	4	2	1	2385	4

1 row | 1-9 of 14 columns

```
##################################

tabla_nDoors <- new_dataset %>%
                select(nDoors)
tabla_nDoors
```

nDoors
<dbl>
2
2
2
1
1
2
1
1
1
2

1-10 of 205 rows Previous 1 2 3 4 5 6 ... 21 Next

```
tabla_mas_frecuente <- table(tabla_nDoors)

valor_mas_frecuente <- names(tabla_mas_frecuente)[tabla_mas_frecuente == max(tabla_mas_frecuente)]

## muestra el valor mas frecuente entre 0, 1, 2
##########DATASET 3###############
valor_mas_frecuente
```

```
[1] "1"
```

```
###################################

df_final <- merge(media_doubles,moda_integer)

df_final
```

normLoss	nDoors	wheelBase	length	width	height	bore	stroke	compressionRatio	horsePower
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
97.6	1.42439	98.75659	174.0493	65.9078	53.72488	3.26478	3.191902	10.14254	103.239

1 row | 1-10 of 26 columns

```
dup <- any(duplicated(df_final))
## no se encontró valores duplicados
dup
```

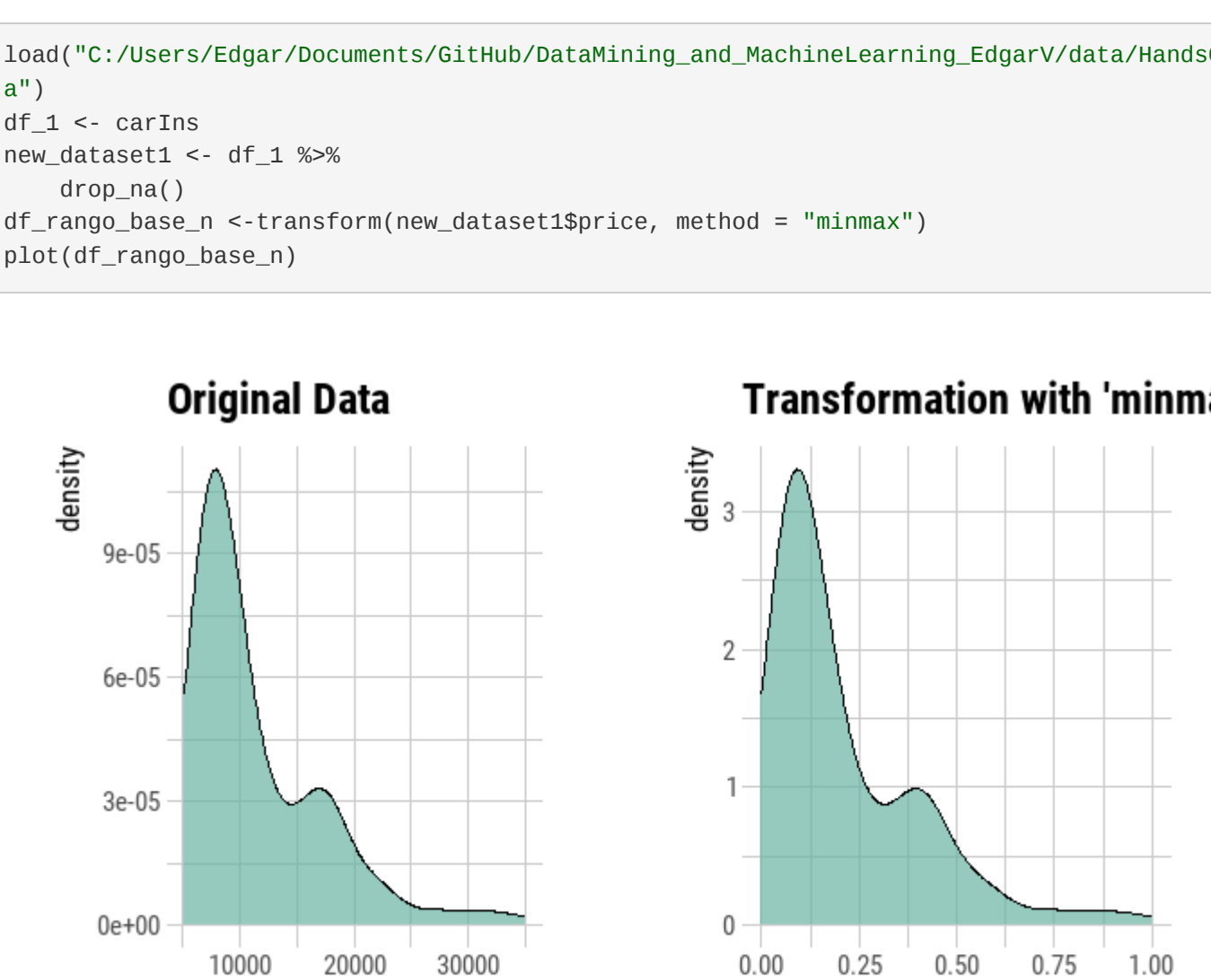
```
[1] FALSE
```

2.- Pre procesamiento de datos

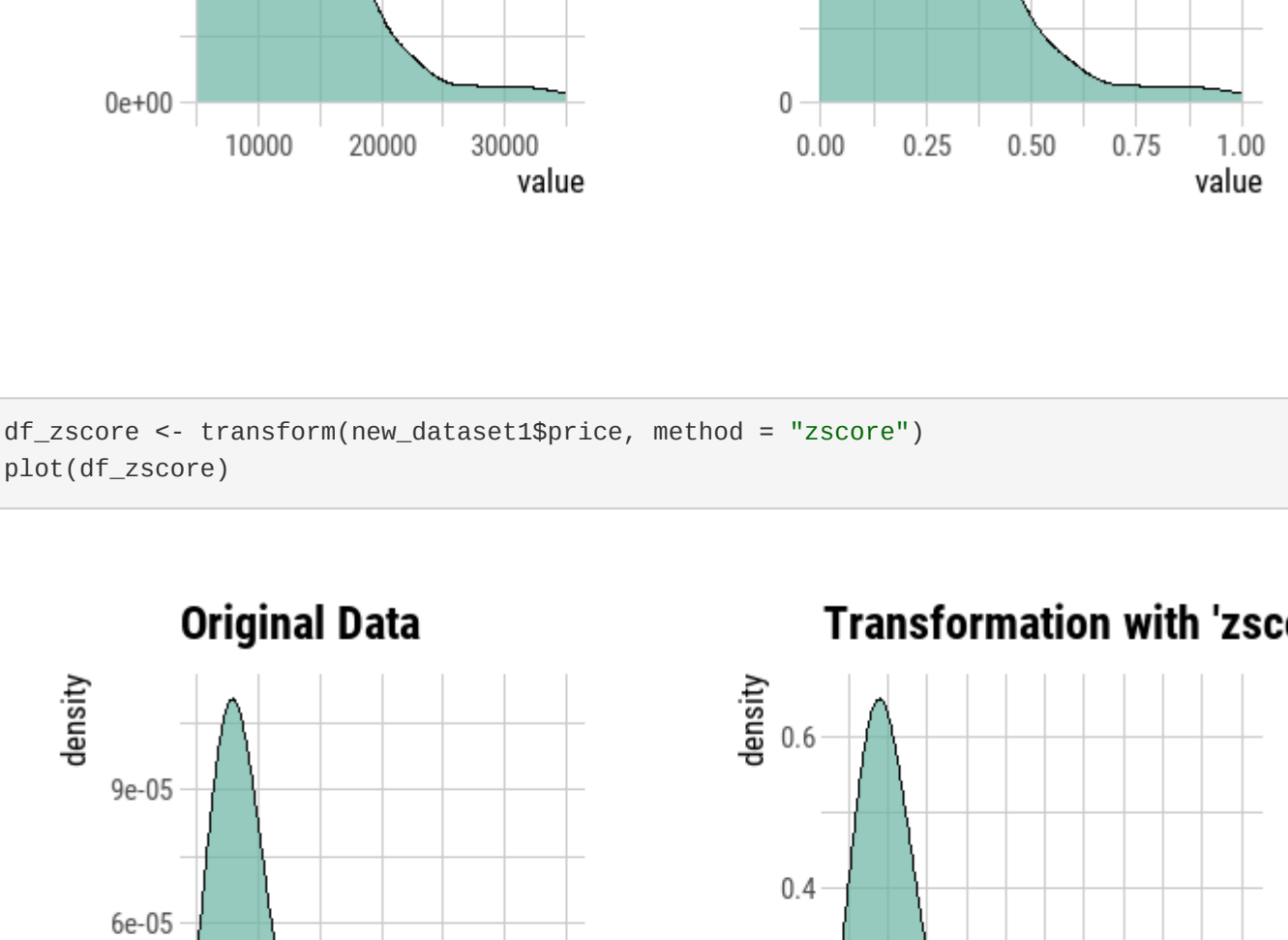
2. Cargar el paquete dlookr, utilizar el mismo dataset carInsurance y aplicar las siguientes transformacionespara el atrbuto precio. ser critico con los resultados obtenidos.

(a) Apply range-based normalization and z-score normalization.

```
load("C:/Users/Edgar/Documents/GitHub/DataMining_and_MachineLearning_EdgarV/data/HandsOn_Data/carInsurance.Rdat
a")
df_1 <- carIns
new_dataset1 <- df_1 %>%
    drop_na()
df_rango_base_n <-transform(new_dataset1$price, method = "minmax")
plot(df_rango_base_n)
```



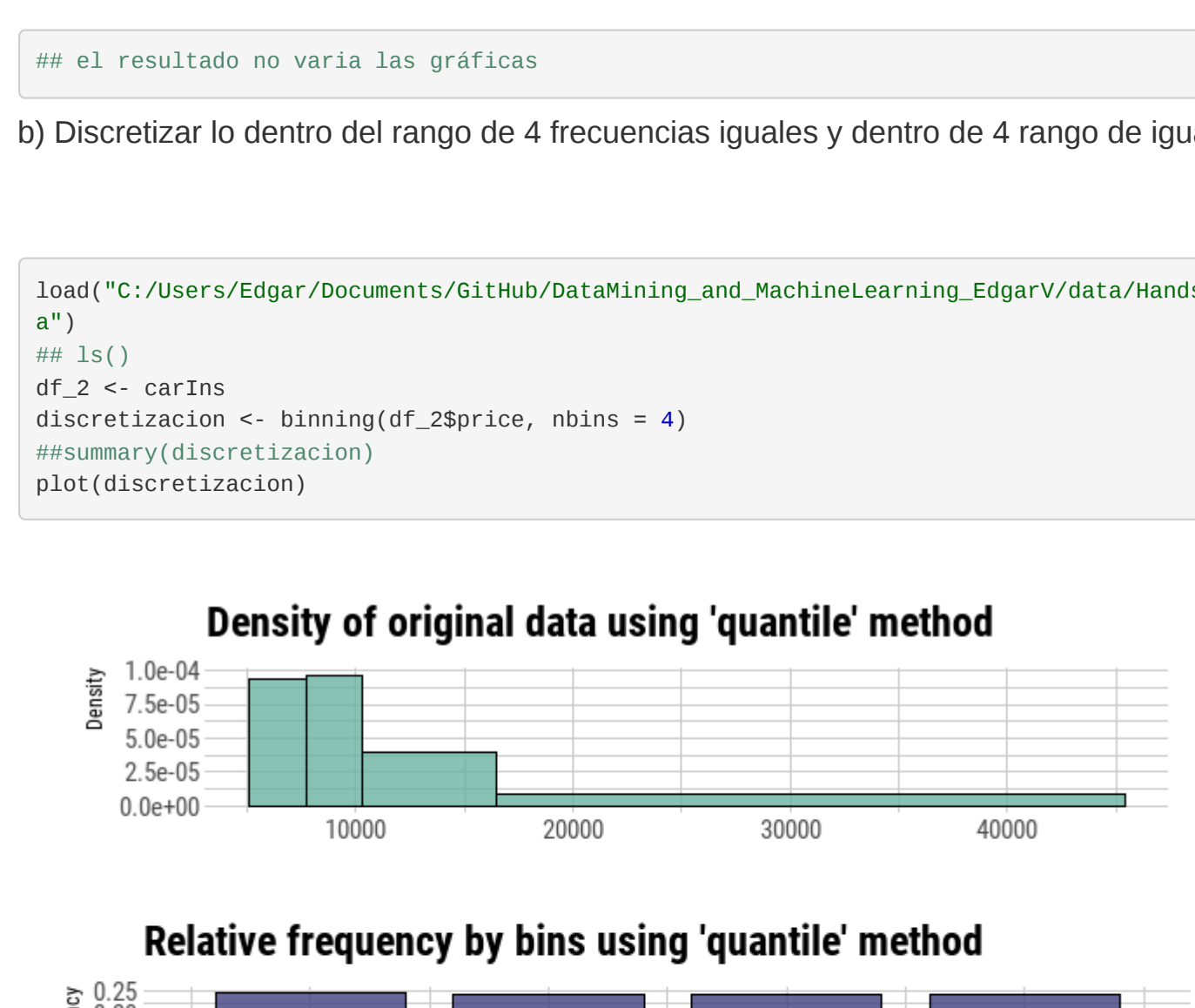

```
df_zscore <- transform(new_dataset1$price, method = "zscore")
plot(df_zscore)
```




```
## el resultado no varia las gráficas
```

b) Discretizar lo dentro del rango de 4 frecuencias iguales y dentro de 4 rango de igual amplitud

```
load("C:/Users/Edgar/Documents/GitHub/DataMining_and_MachineLearning_EdgarV/data/HandsOn_Data/carInsurance.Rdat
a")
## ls()
df_2 <- carIns
discretizacion <- binning(df_2$price, nbins = 4)
##summary(discretizacion)
plot(discretizacion)
```



3 con la semilla 111019 obtener las siguientes muestras sobre la dataset carInsurance

una muestra aleatoria del 60% de los casos con reemplazo

```
load("C:/Users/Edgar/Documents/GitHub/DataMining_and_MachineLearning_EdgarV/data/HandsOn_Data/carInsurance.Rdat
a")
## ls()
df_3 <- carIns

sampled_df <- df_3 %>% sample_frac(0.6)
sampled_df
```

sy…	normLoss	make	fuelType	aspiration	nDoors	bodyStyle	driveWheels	engineLocation
<int>	<int>	<fctr>	<fctr>	<fctr>	<fctr>	<fctr>	<fctr>	<fctr>
2	137	honda	gas	std	two	hatchback	fwd	front
2	168	nissan	gas	std	two	hardtop	fwd	front
1	101	honda	gas	std	two	hatchback	fwd	front
-1	65	toyota	gas	std	four	hatchback	fwd	front
-2	103	volvo	gas	std	four	sedan	rwd	front
0	85	honda	gas	std	four	sedan	fwd	front
2	83	subaru	gas	std	two	hatchback	4wd	front
0	89	subaru	gas	std	four	wagon	fwd	front
0	128	nissan	gas	std	four	sedan	fwd	front
1	103	nissan	gas	std	four	wagon	fwd	front

1-10 of 123 rows | 1-9 of 26 columns Previous 1 2 3 4 5 6 ... 13 Next

```
## de las 205 filas me ha tomado una muestra solo del 60% que serian 123 filas
```

un muestreo estratificado del 60% de los casos de carros.de acuerdo al atributo tipo de combustible (fuelType)

```
load("C:/Users/Edgar/Documents/GitHub/DataMining_and_MachineLearning_EdgarV/data/HandsOn_Data/carInsurance.Rdat
a")
## ls()
df_4 <- carIns

stratified_sample <- df_4 %>%
  group_by(fuelType) %>%
  sample_frac(0.6)
##ahora observamos que nos trae el 60% de la muestra pero agrupado por tipo de combustible
stratified_sample
```

sy…	normLoss	make	fuelType	aspiration	nDoors	bodyStyle	driveWheels	engineLocation
<int>	<int>	<fctr>	<fctr>	<fctr>	<fctr>	<fctr>	<fctr>	<fctr>
-1	93	mercedes-benz	diesel	turbo	four	wagon	rwd	front
-1	95	volvo	diesel	turbo	four	sedan	rwd	front
0	161	peugot	diesel	turbo	four	sedan	rwd	front
0	93	mercedes-benz	diesel	turbo	two	hardtop	rwd	front
0	161	peugot	diesel	turbo	four	sedan	rwd	front
0	91	toyota	diesel	std	four	sedan	fwd	front
-1	93	mercedes-benz	diesel	turbo	four	sedan	rwd	front
0	161	peugot	diesel	turbo	four	sedan	rwd	front
0	NA	peugot	diesel	turbo	four	wagon	rwd	front
2	94	volkswagen	diesel	std	four	sedan	fwd	front

1-10 of 123 rows | 1-9 of 26 columns Previous 1 2 3 4 5 6 ... 13 Next

Utilizar la tabla de funciones para inspeccionar la distribución en cada uno de las dos muestra de arriba.

4.

```
library(corrplot)
```

```
Warning: package ‘corrplot’ was built under R version 4.3.1corrplot 0.92 loaded
```

```
numeric_data<- carIns[,sapply(carIns,is.numeric)]
cor_matrix<- cor(numeric_data)
corrplot(cor_matrix,method = "circle")
```