DELIVERABLE 1

Jesús García Ayala

Contents

1. Introduction

In today's competitive market, understanding customer behavior is crucial for businesses to optimize their marketing strategies and enhance customer retention. For this reason, we have selected this dataset, as we believe it is highly valuable.

This dataset was chosen because it allows us to analyze key factors influencing customer decision-making, such as household income, family structure, spending patterns, and response to promotional campaigns. By exploring this data, we can identify trends, segment customers, and develop data-driven strategies to improve marketing effectiveness.

Additionally, the dataset contains a variety of numerical and categorical variables, making it an excellent choice for applying preprocessing techniques. This diversity enables us to utilize different analytical methods, including statistical analysis.

By working with this dataset, we aim to gain a deeper understanding of how marketing efforts impact different customer groups, how businesses can enhance their strategies to maximize engagement and revenue, and how to effectively analyze a dataset, including both its preprocessing and postprocessing stages.

Database context:

The database is part of a case study designed to simulate the real challenges faced by data analysts at iFood, the leading food delivery platform in Brazil. The company operates in over a thousand cities and serves millions of customers annually. Maintaining a high level of customer engagement is essential to consolidating its market leadership.

The iFood data team must work on open-ended analytical projects to identify business opportunities, optimize marketing strategies, and support data-driven decision-making. We will put ourselves in the shoes of iFood’s data team.

The dataset provides simulated information on customer profiles based on real data and their interactions with iFood’s marketing campaigns, including socio-demographic and firmographic data from 2,240 customers, randomly selected for a pilot marketing campaign.

The main challenge is to develop a predictive model that more accurately identifies customers with a higher likelihood of purchasing, allowing iFood to focus its efforts on the most responsive segments.

By applying data-driven strategies, iFood aims to refine its marketing approach, reduce inefficiencies, and strengthen its relationship with customers, solidifying its position as the leader in the food delivery industry.

Objective of the project

The goal of this project is to analyze customer behavior in a sales company by exploring their demographic characteristics, purchasing habits, and response to marketing campaigns. The aim is to extract useful insights that can help improve commercial strategies and better personalize offers for customers.

Methods used

- Exploratory Data Analysis (EDA) to understand data distribution and potential anomalies.
- Handling missing values and outliers to ensure data quality.
- Comparison of variables before and after preprocessing to observe the impact on analysis.
- Use of **R** for visualizations and basic predictive modeling on the dataset.

2. Data Font (where is data from?)

https://www.kaggle.com/datasets/jackdaoud/marketing-data?resource=download&select=dictionary.png

The dataset used comes from a customer database of a company that offers various types of products. It contains information about:

- **Customer profile** (income, age, marital status, education level).
- **Purchasing habits** (products bought, purchase channels).
- **Response to marketing campaigns** (acceptance of promotions).
- **Interaction with the service** (website visits, recorded complaints).

The original data has been preprocessed to remove inconsistencies and improve interpretability, as detailed in the following sections.

3. Metadata

This table provides an explanation of all the variables in our dataset, including the variable name, data type, variable type classification, description, and possible values.

Variable name	Variable Type	Description	Possibles values
Id	ID	The ID of the customer	Random number between 3-5 digits
Income	Numerical Continuous	Customer’s yearly household income.	> 0
Kidhome	Numerical Discrete	Number of small children in the customer’s household.	0-5
Teenhome	Numerical Discrete	Number of teenagers in the customer’s household.	0-5
Recency	Numerical Discrete	Number of days since the last purchase.	0-365(aprox)
MntWines	Numerical Continuous	Amount spent on wine in the last 2 years.	≥ 0
MntFruits	Numerical Continuous	Amount spent on fruits in the last 2 years.	≥ 0
MntMeatProducts	Numerical Continuous	Amount spent on meat products in the last 2 years.	≥ 0

MntFishProducts	Numerical Continuous	Amount spent on fish products in the last 2 years.	≥ 0
MntSweetProducts	Numerical Continuous	Amount spent on sweet products in the last 2 years.	≥ 0
MntGoldProds	Numerical Continuous	Amount spent on gold products in the last 2 years.	≥ 0
NumDealsPurchases	Numerical Discrete	Number of purchases made with a discount.	≥ 0
NumWebPurchases	Numerical Discrete	Number of purchases made through the website.	≥ 0
NumCatalogPurchases	Numerical Discrete	Number of purchases made using a catalog.	≥ 0
NumStorePurchases	Numerical Discrete	Number of purchases made directly in a store.	≥ 0
NumWebVisitsMonth	Numerical Discrete	Number of visits to the company's website by the customer in the last month.	≥ 0
AcceptedCmp1	Categorical Binary	Whether the customer accepted the offer in the 1st campaign.	0, 1
AcceptedCmp2	Categorical Binary	Whether the customer accepted the offer in the 2nd campaign.	0, 1

AcceptedCmp3	Categorical Binary	Whether the customer accepted the offer in the 3rd campaign.	0, 1
AcceptedCmp4	Categorical Binary	Whether the customer accepted the offer in the 4th campaign.	0, 1
AcceptedCmp5	Categorical Binary	Whether the customer accepted the offer in the 5th campaign.	0, 1
Complain	Categorical Binary	Whether the customer has filed a complaint in the last 2 years.	0, 1
Z_CostContact	Numerical Discrete	Fixed cost of contacting the customer.	3
Z_Revenue	Numerical Discrete	Fixed revenue value.	11
Response	Categorical Binary	Whether the customer accepted the offer in the last campaign.	0, 1
Year_birth	Numerical Discrete	Customer's year of birth.	1900-2020
Dt_Customer	Numerical Discrete	Date when the customer enrolled .	Date between 1-1-1900/31-12-2020
Marital	Categorical Nominal	Whether the customer is divorced.	Divorced, Married,Single,Together,Widow
Education	Categorical Nominal	The level of studies the consumer has.	2nd cycle,Basic,Graduation,Master PhD

4. Data description

Basic descriptive = basic description of the dataset through descriptive statistics and visualizations.

4.1 Dataset dimensions

The original dataset contains 2240 rows and 29 columns, representing different customer attributes.

4.2 Types of variables

The dataset includes:

- **Numerical (quantitative) variables:** such as income, age, spending on different products, number of purchases through each channel, etc.
- **Categorical (qualitative) variables:** such as marital status, education level and customer classification based on marketing campaign responses.

4.3 Descriptive statistics

Basic statistics for numerical variables include:

- **Mean, median, minimum, maximum and standard deviation.**
- **Histograms to analyze distribution.**
- **Boxplots to identify outliers.**

Categorical variables are summarized using:

- **Frequency tables** to examine category distributions.
- **Bar and pie charts** for a visual representation of common values.

4.4 Preliminary insights

- Initial patterns observed in the data are highlighted.
- Variables with extreme or highly skewed values are discussed.
- Initial hypotheses are formulated for further analysis.

Configuración para mejorar la ejecución

```
knitr::opts_chunk$set(echo = TRUE, warning = FALSE, message = FALSE)
```

Cargar dataset

```
ifood <- read.csv("ifood_no_preprocessed.csv", sep=",", header=TRUE)
```

Mostrar las primeras filas

```
head(ifood)
```

```
##     ID Year_Birth  Education Marital_Status Income Kidhome Teenhome Dt_Customer
## 1 5524       1957 Graduation         Single  58138       0        0  2012-09-04
## 2 2174       1954 Graduation         Single  46344       1        1  2014-03-08
## 3 4141       1965 Graduation       Together  71613       0        0  2013-08-21
## 4 6182       1984 Graduation       Together  26646       1        0  2014-02-10
## 5 5324       1981        PhD        Married  58293       1        0  2014-01-19
## 6 7446       1967     Master       Together  62513       0        1  2013-09-09
##   Recency MntWines MntFruits MntMeatProducts MntFishProducts MntSweetProducts
## 1      58      635        88             546             172               88
## 2      38       11         1               6               2                1
## 3      26      426        49             127             111               21
## 4      26       11         4              20              10                3
## 5      94      173        43             118              46               27
## 6      16      520        42              98               0               42
##   MntGoldProds NumDealsPurchases NumWebPurchases NumCatalogPurchases
## 1           88                 3               8                  10
## 2            6                 2               1                   1
## 3           42                 1               8                   2
## 4            5                 2               2                   0
## 5           15                 5               5                   3
## 6           14                 2               6                   4
##   NumStorePurchases NumWebVisitsMonth AcceptedCmp3 AcceptedCmp4 AcceptedCmp5
## 1                 4                 7            0            0            0
## 2                 2                 5            0            0            0
## 3                10                 4            0            0            0
## 4                 4                 6            0            0            0
## 5                 6                 5            0            0            0
## 6                10                 6            0            0            0
##   AcceptedCmp1 AcceptedCmp2 Complain Z_CostContact Z_Revenue Response
## 1            0            0        0             3        11        1
## 2            0            0        0             3        11        0
## 3            0            0        0             3        11        0
## 4            0            0        0             3        11        0
## 5            0            0        0             3        11        0
## 6            0            0        0             3        11        0
```

Ver estructura del dataset

```
str(ifood)
```

```
## 'data.frame':	2240 obs. of  29 variables:
##  $ ID                 : int  5524 2174 4141 6182 5324 7446 965 6177 4855 5899 ...
##  $ Year_Birth         : int  1957 1954 1965 1984 1981 1967 1971 1985 1974 1950 ...
##  $ Education          : chr  "Graduation" "Graduation" "Graduation" "Graduation" ...
##  $ Marital_Status     : chr  "Single" "Single" "Together" "Together" ...
##  $ Income             : int  58138 46344 71613 26646 58293 62513 55635 33454 30351 5648 ...
##  $ Kidhome            : int  0 1 0 1 1 0 0 1 1 1 ...
##  $ Teenhome           : int  0 1 0 0 0 1 1 0 0 1 ...
##  $ Dt_Customer        : chr  "2012-09-04" "2014-03-08" "2013-08-21" "2014-02-10" ...
##  $ Recency            : int  58 38 26 26 94 16 34 32 19 68 ...
##  $ MntWines           : int  635 11 426 11 173 520 235 76 14 28 ...
##  $ MntFruits          : int  88 1 49 4 43 42 65 10 0 0 ...
##  $ MntMeatProducts    : int  546 6 127 20 118 98 164 56 24 6 ...
##  $ MntFishProducts    : int  172 2 111 10 46 0 50 3 3 1 ...
##  $ MntSweetProducts   : int  88 1 21 3 27 42 49 1 3 1 ...
##  $ MntGoldProds       : int  88 6 42 5 15 14 27 23 2 13 ...
##  $ NumDealsPurchases  : int  3 2 1 2 5 2 4 2 1 1 ...
##  $ NumWebPurchases    : int  8 1 8 2 5 6 7 4 3 1 ...
##  $ NumCatalogPurchases: int  10 1 2 0 3 4 3 0 0 0 ...
##  $ NumStorePurchases  : int  4 2 10 4 6 10 7 4 2 0 ...
##  $ NumWebVisitsMonth  : int  7 5 4 6 5 6 6 8 9 20 ...
##  $ AcceptedCmp3       : int  0 0 0 0 0 0 0 0 0 1 ...
##  $ AcceptedCmp4       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ AcceptedCmp5       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ AcceptedCmp1       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ AcceptedCmp2       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Complain           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Z_CostContact      : int  3 3 3 3 3 3 3 3 3 3 ...
##  $ Z_Revenue          : int  11 11 11 11 11 11 11 11 11 11 ...
##  $ Response           : int  1 0 0 0 0 0 0 0 1 0 ...
```

```
##  NumWebPurchases  NumCatalogPurchases NumStorePurchases NumWebVisitsMonth
##  Min.   : 0.000   Min.   : 0.000      Min.   : 0.00     Min.   : 0.000
##  1st Qu.: 2.000   1st Qu.: 0.000      1st Qu.: 3.00     1st Qu.: 3.000
##  Median : 4.000   Median : 2.000      Median : 5.00     Median : 6.000
##  Mean   : 4.085   Mean   : 2.662      Mean   : 5.79     Mean   : 5.317
##  3rd Qu.: 6.000   3rd Qu.: 4.000      3rd Qu.: 8.00     3rd Qu.: 7.000
##  Max.   :27.000   Max.   :28.000      Max.   :13.00     Max.   :20.000
##
##   AcceptedCmp3      AcceptedCmp4      AcceptedCmp5      AcceptedCmp1
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.00000   Median :0.00000   Median :0.00000   Median :0.00000
##  Mean   :0.07277   Mean   :0.07455   Mean   :0.07277   Mean   :0.06429
##  3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.00000   Max.   :1.00000
##
##   AcceptedCmp2        Complain         Z_CostContact   Z_Revenue
##  Min.   :0.00000   Min.   :0.000000   Min.   :3      Min.   :11
##  1st Qu.:0.00000   1st Qu.:0.000000   1st Qu.:3      1st Qu.:11
##  Median :0.00000   Median :0.000000   Median :3      Median :11
##  Mean   :0.01339   Mean   :0.009375   Mean   :3      Mean   :11
##  3rd Qu.:0.00000   3rd Qu.:0.000000   3rd Qu.:3      3rd Qu.:11
##  Max.   :1.00000   Max.   :1.000000   Max.   :3      Max.   :11
##
##     Response
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.1491
##  3rd Qu.:0.0000
##  Max.   :1.0000
##
```

Resumen Estadístico General

```
# Resumen de todas las variables
summary(ifood)
```

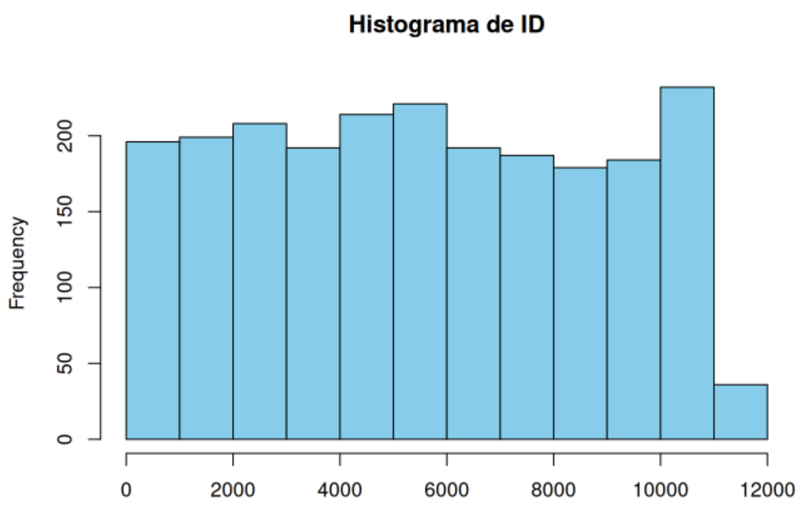
```
##        ID          Year_Birth    Education          Marital_Status
##  Min.   :    0   Min.   :1893   Length:2240        Length:2240
##  1st Qu.: 2828   1st Qu.:1959   Class :character   Class :character
##  Median : 5458   Median :1970   Mode  :character   Mode  :character
##  Mean   : 5592   Mean   :1969
##  3rd Qu.: 8428   3rd Qu.:1977
##  Max.   :11191   Max.   :1996
##
##      Income          Kidhome          Teenhome       Dt_Customer
##  Min.   :  1730   Min.   :0.0000   Min.   :0.0000   Length:2240
##  1st Qu.: 35303   1st Qu.:0.0000   1st Qu.:0.0000   Class :character
##  Median : 51382   Median :0.0000   Median :0.0000   Mode  :character
##  Mean   : 52247   Mean   :0.4442   Mean   :0.5062
##  3rd Qu.: 68522   3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :666666   Max.   :2.0000   Max.   :2.0000
##  NA's   :24
##     Recency          MntWines          MntFruits       MntMeatProducts
##  Min.   : 0.00   Min.   :   0.00   Min.   :  0.0   Min.   :   0.0
##  1st Qu.:24.00   1st Qu.:  23.75   1st Qu.:  1.0   1st Qu.:  16.0
##  Median :49.00   Median : 173.50   Median :  8.0   Median :  67.0
##  Mean   :49.11   Mean   : 303.94   Mean   : 26.3   Mean   : 166.9
##  3rd Qu.:74.00   3rd Qu.: 504.25   3rd Qu.: 33.0   3rd Qu.: 232.0
##  Max.   :99.00   Max.   :1493.00   Max.   :199.0   Max.   :1725.0
##
##  MntFishProducts  MntSweetProducts  MntGoldProds     NumDealsPurchases
##  Min.   :  0.00   Min.   :  0.00   Min.   :  0.00   Min.   : 0.000
##  1st Qu.:  3.00   1st Qu.:  1.00   1st Qu.:  9.00   1st Qu.: 1.000
##  Median : 12.00   Median :  8.00   Median : 24.00   Median : 2.000
##  Mean   : 37.53   Mean   : 27.06   Mean   : 44.02   Mean   : 2.325
##  3rd Qu.: 50.00   3rd Qu.: 33.00   3rd Qu.: 56.00   3rd Qu.: 3.000
##  Max.   :259.00   Max.   :263.00   Max.   :362.00   Max.   :15.000
##
```

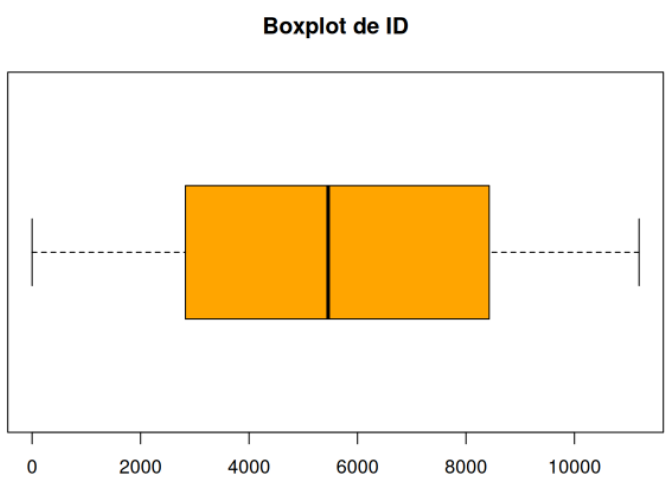
Análisis de Variables Numéricas

```r
# Seleccionar variables numéricas
numericas <- sapply(ifood, is.numeric)
numericas <- names(ifood)[numericas]
```

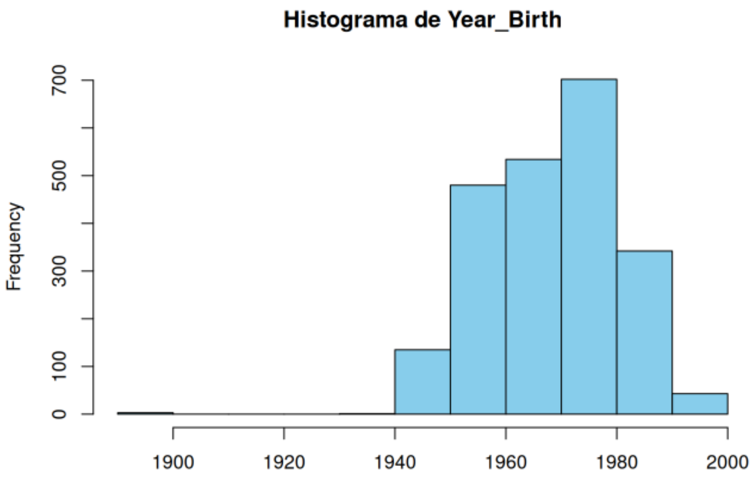
```r
# Histograma y boxplot para cada variable numérica
for (var in numericas) {
  cat("Variable -> ", var, "\n\n")

  # Histograma
  hist(ifood[[var]], main=paste("Histograma de", var), col="skyblue", border="black")

  # Boxplot
  boxplot(ifood[[var]], main=paste("Boxplot de", var), col="orange", horizontal=TRUE)

  # Tabla de frecuencias y resumen estadístico

  # Muestra solo los primeros 20 valores

  print(head(ifood[[var]], 20))
  print(summary(ifood[[var]], 20))
  cat("\n\n")
}
```

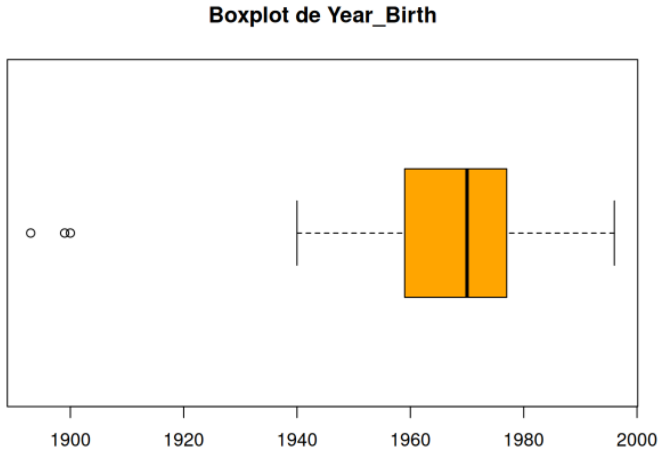
```
## Variable ->  ID
```



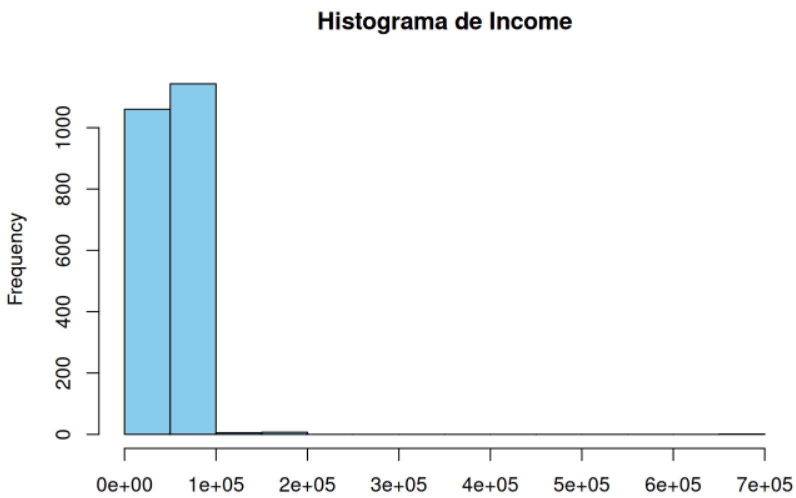


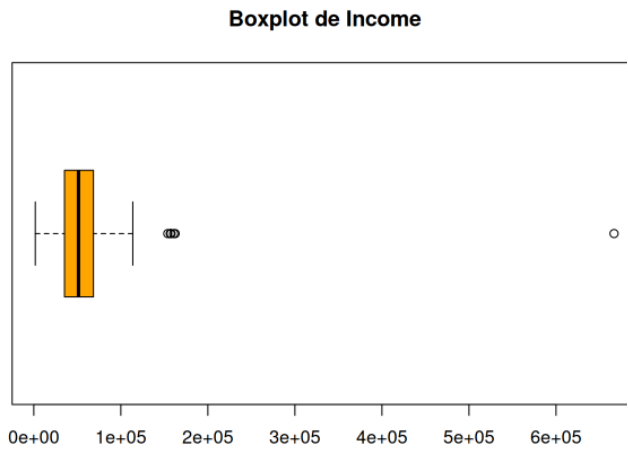

```
##  [1] 5524 2174 4141 6182 5324 7446  965 6177 4855 5899 1994  387 2125 8180 2569
## [16] 2114 9736 4939 6565 2278
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0    2828    5458    5592    8428   11191
##
##
## Variable ->  Year_Birth
```



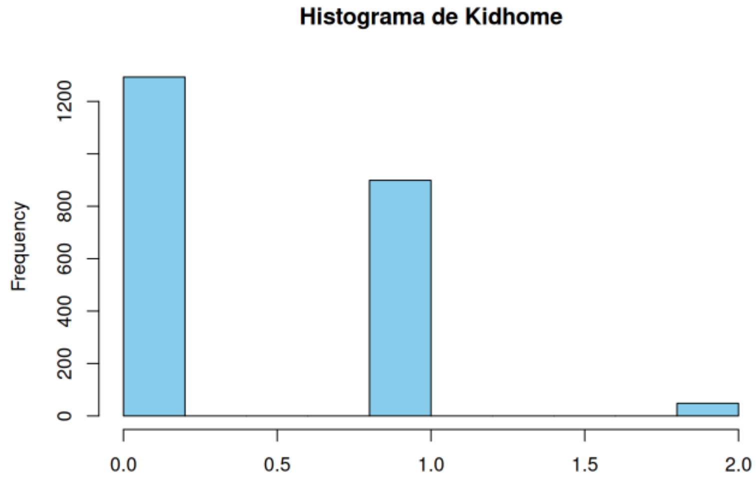


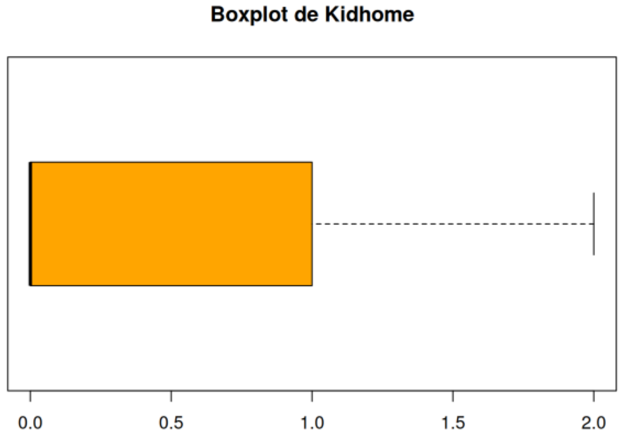

```
##  [1] 1957 1954 1965 1984 1981 1967 1971 1985 1974 1950 1983 1976 1959 1952 1987
## [16] 1946 1980 1946 1949 1985
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    1893    1959    1970    1969    1977    1996 
## 
## 
## Variable ->  Income
```



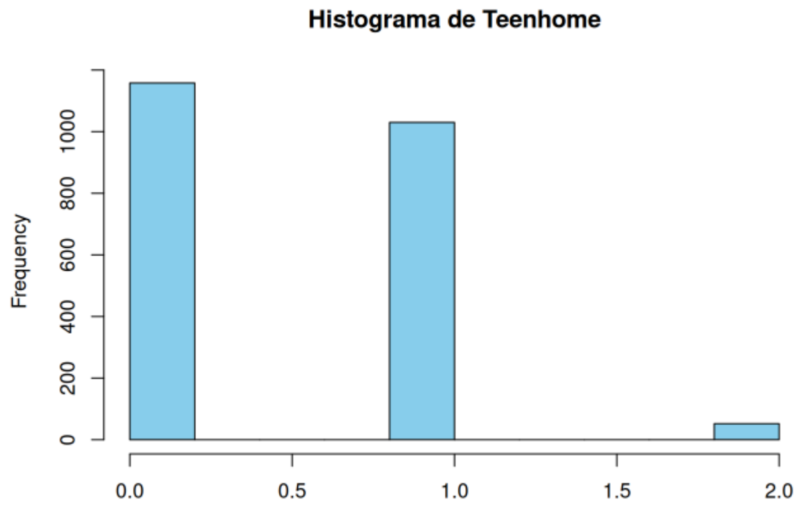


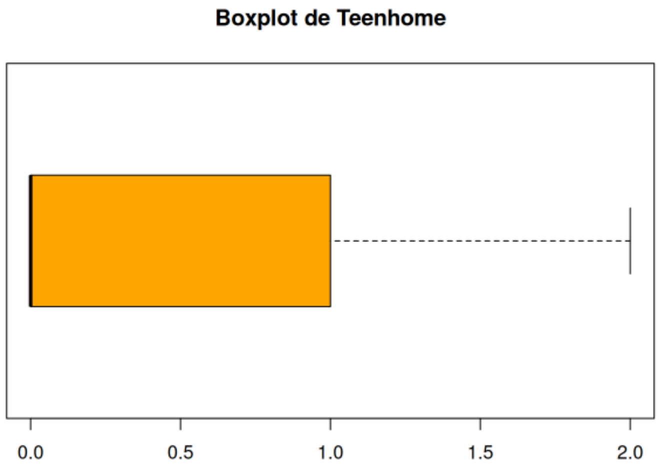

```
##  [1] 58138 46344 71613 26646 58293 62513 55635 33454 30351  5648    NA  7500
## [13] 63033 59354 17323 82800 41850 37760 76995 33812
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##    1730   35303   51382   52247   68522  666666      24 
## 
## 
## Variable ->  Kidhome
```



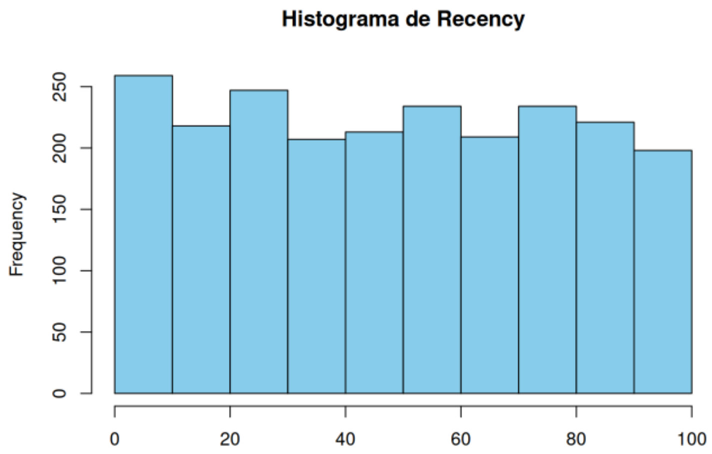


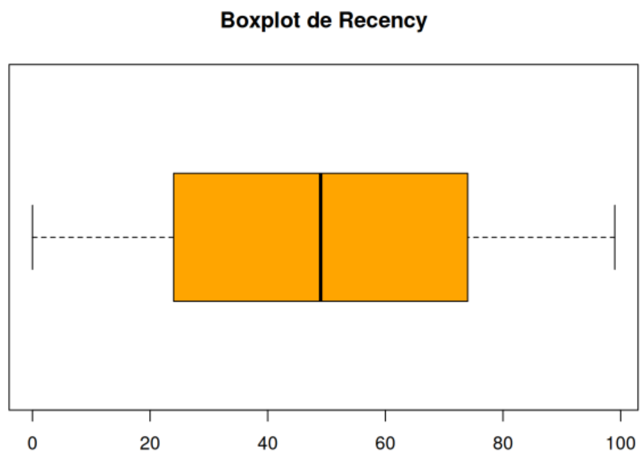

```
##  [1] 0 1 0 1 1 0 0 1 1 1 1 0 0 1 0 0 1 0 0 1
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  0.0000  0.4442  1.0000  2.0000
##
##
## Variable ->  Teenhome
```



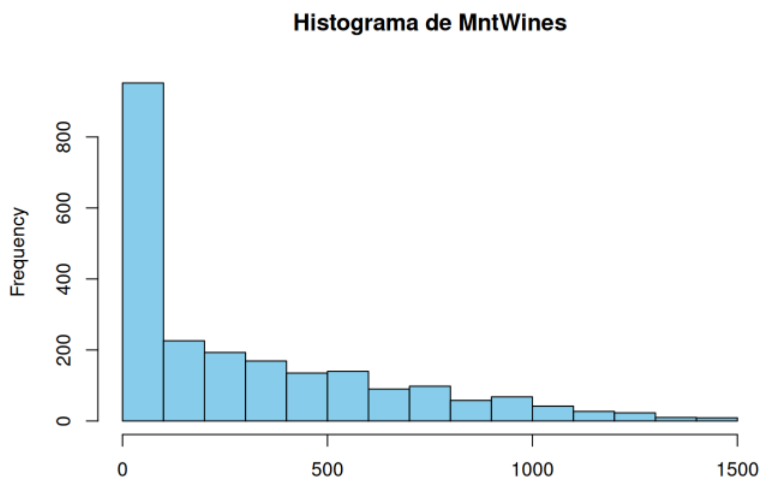


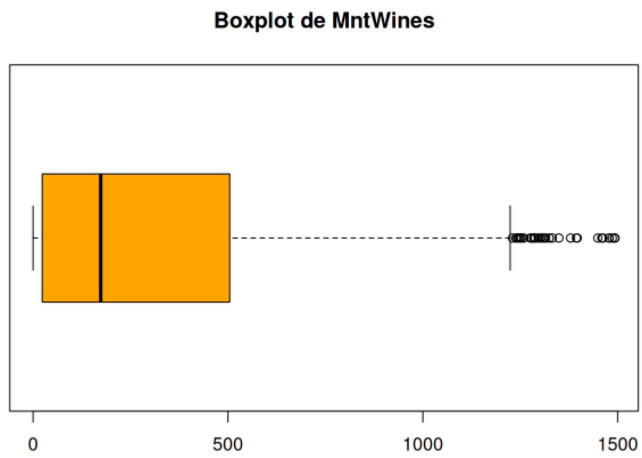

```
##  [1] 0 1 0 0 0 1 1 0 0 1 0 0 0 1 0 0 1 0 1 0
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  0.0000  0.5062  1.0000  2.0000
##
##
## Variable ->  Recency
```



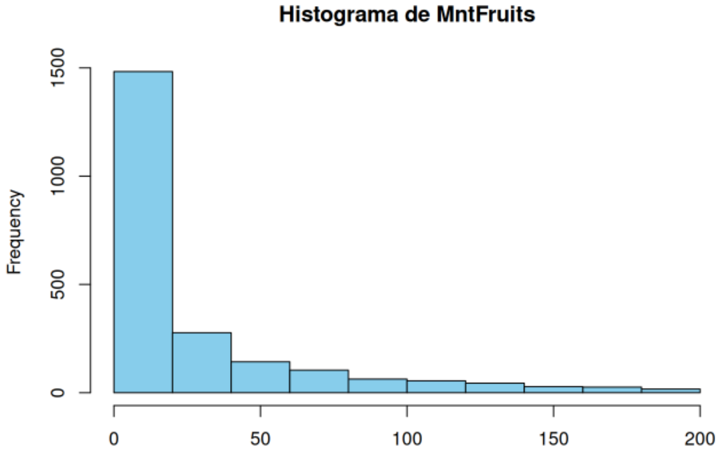


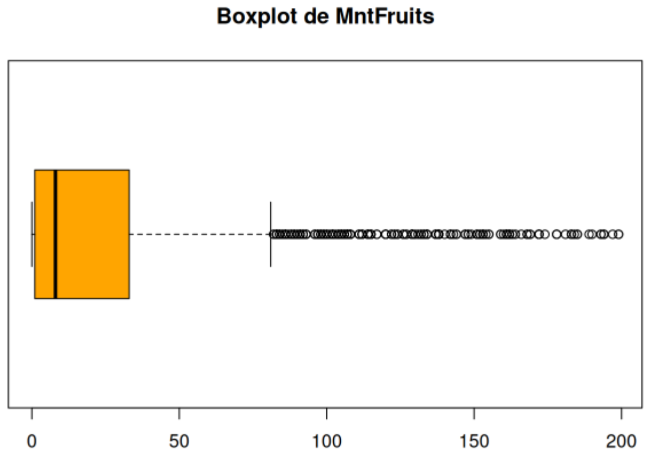

```
## [1] 58 38 26 26 94 16 34 32 19 68 11 59 82 53 38 23 51 20 91 86
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   24.00   49.00   49.11   74.00   99.00
##
##
## Variable ->  MntWines
```



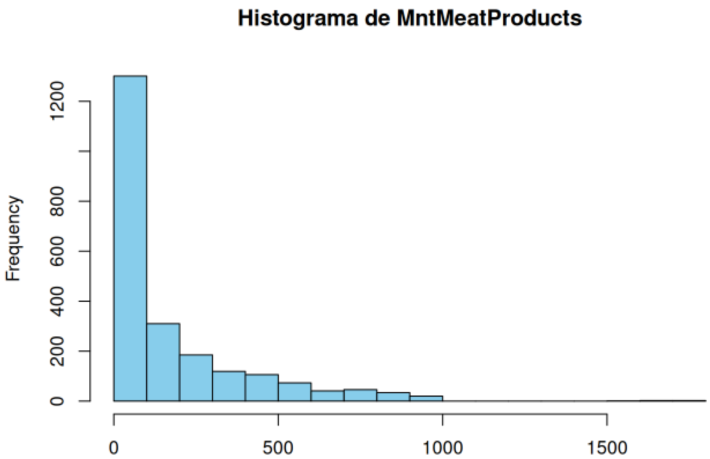


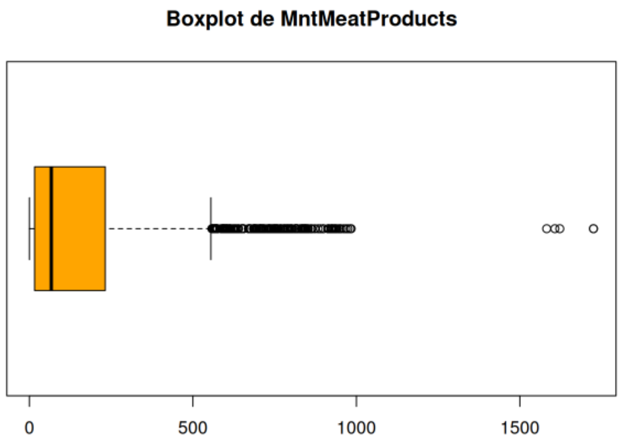

```
## [1]  635   11  426   11  173  520  235   76   14   28    5    6  194  233    3
## [16] 1006   53   84 1012    4
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   23.75  173.50  303.94  504.25 1493.00
##
##
## Variable ->  MntFruits
```



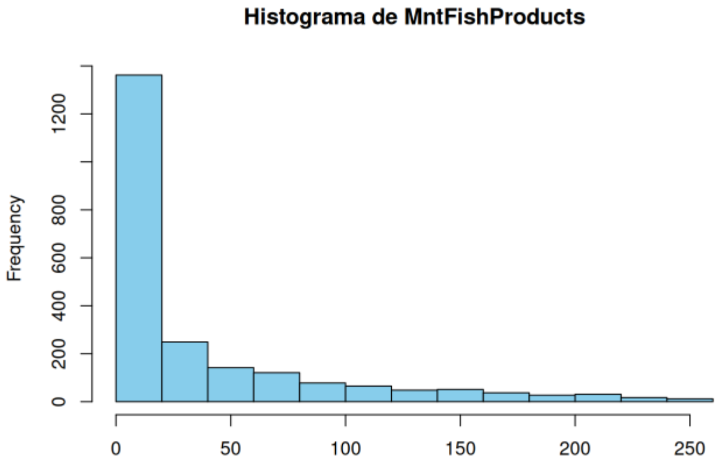


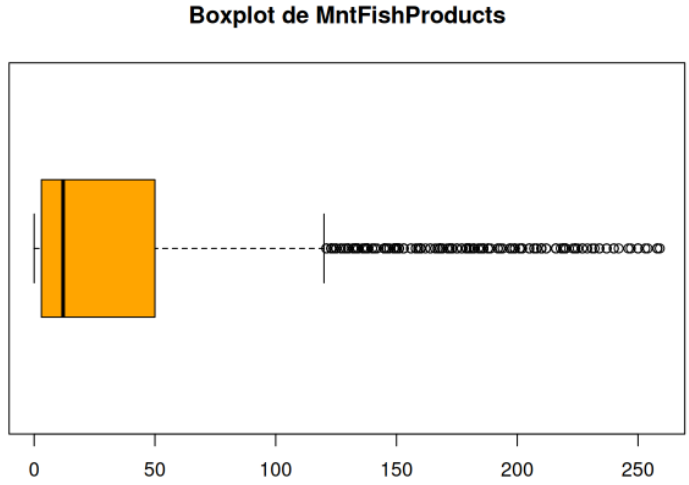

```
##  [1] 88  1 49  4 43 42 65 10  0  0  5 16 61  2 14 22  5  5 80 17
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0     1.0     8.0    26.3    33.0   199.0
##
##
## Variable ->  MntMeatProducts
```



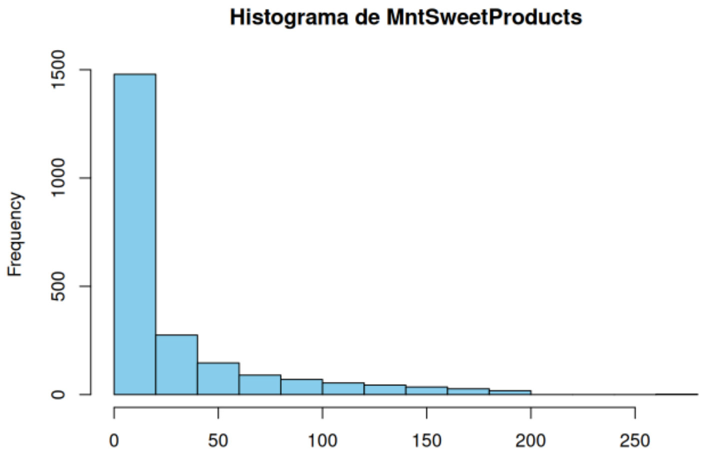


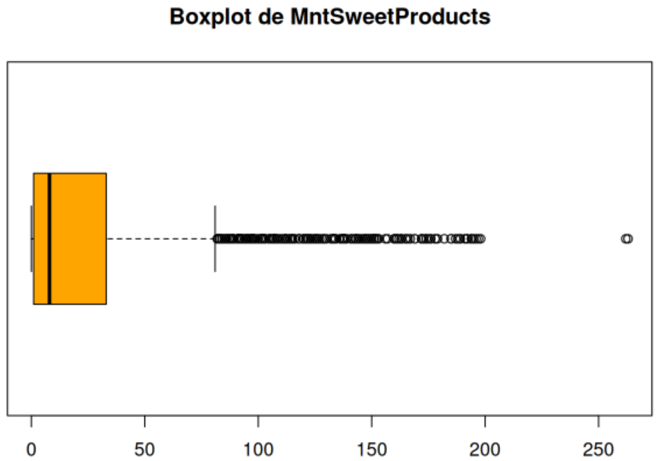

```
##  [1] 546   6 127  20 118  98 164  56  24   6   6  11 480  53  17 115  19  38 498
## [20]  19
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0    16.0    67.0   166.9   232.0  1725.0
##
##
## Variable ->  MntFishProducts
```



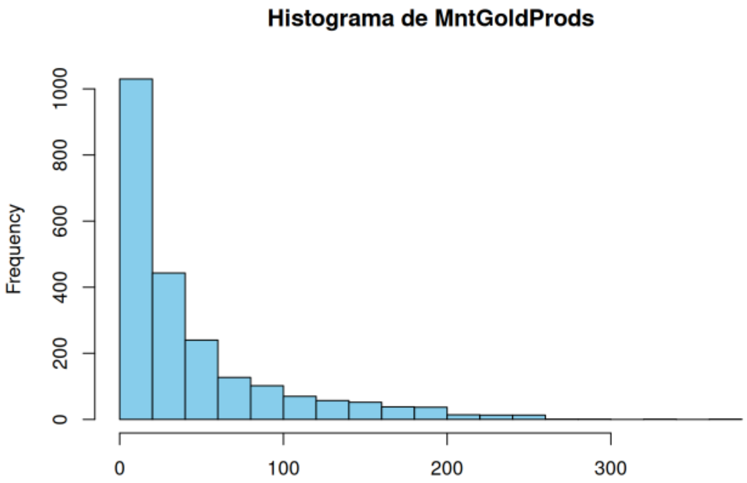


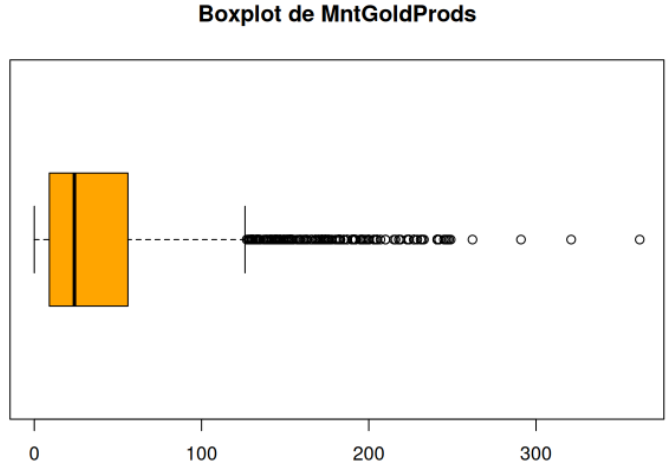

```
##  [1] 172   2 111  10  46   0  50   3   3   1   0  11 225   3   6  59   2 150   0
## [20]  30
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    0.00    3.00   12.00   37.53   50.00  259.00 
## 
## 
## Variable ->  MntSweetProducts
```



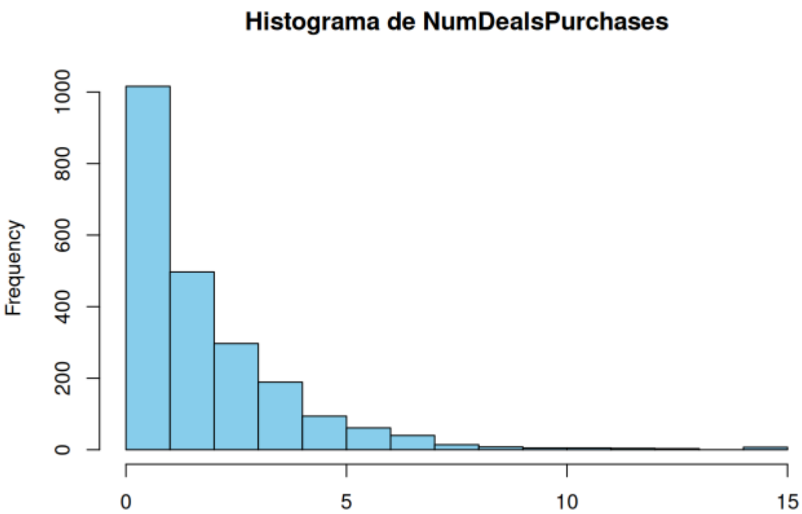


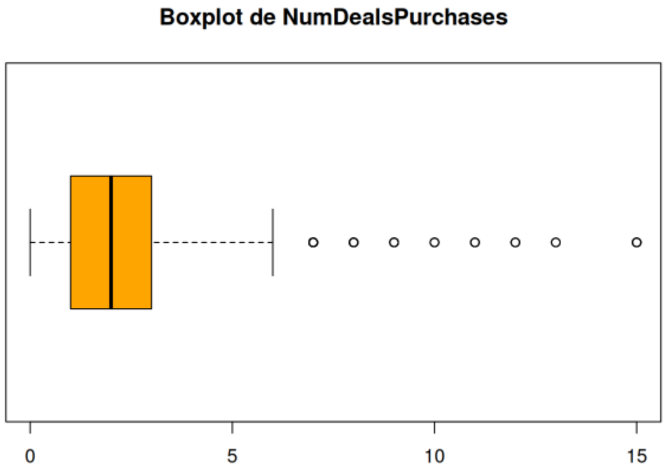

```
##  [1]  88   1  21   3  27  42  49   1   3   1   2   1 112   5   1  68  13  12  16
## [20]  24
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    0.00    1.00    8.00   27.06   33.00  263.00 
## 
## 
## Variable ->  MntGoldProds
```



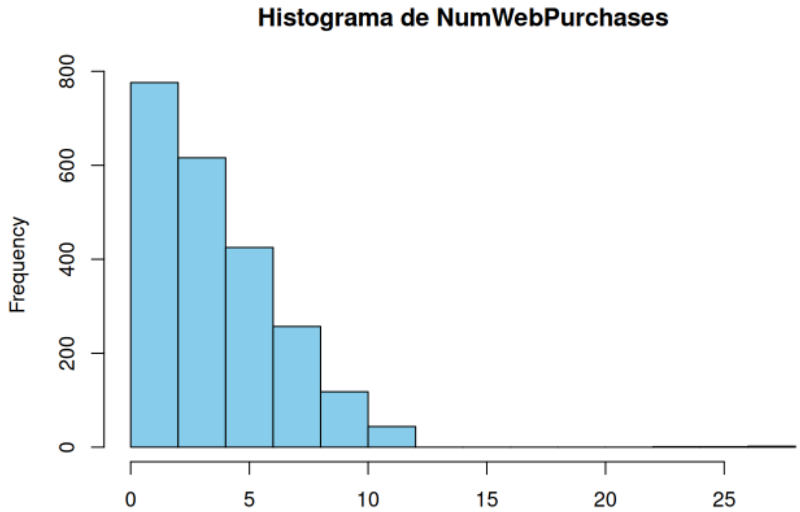


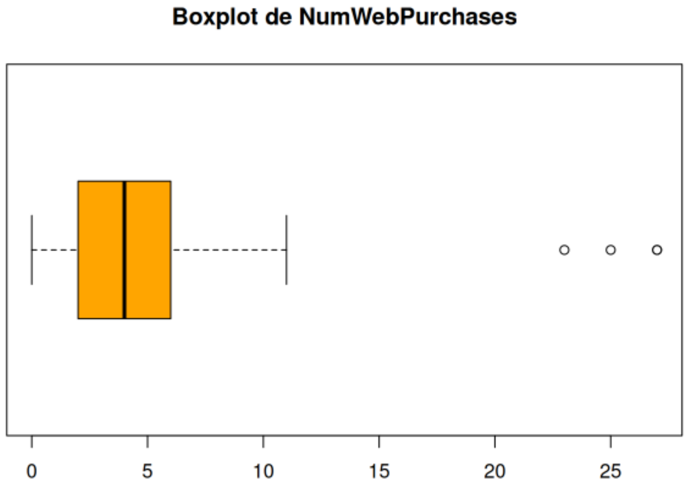

```
##  [1]  88   6  42   5  15  14  27  23   2  13   1  16  30  14   5  45   4  28 176
## [20]  39
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    9.00   24.00   44.02   56.00  362.00
##
##
## Variable ->  NumDealsPurchases
```



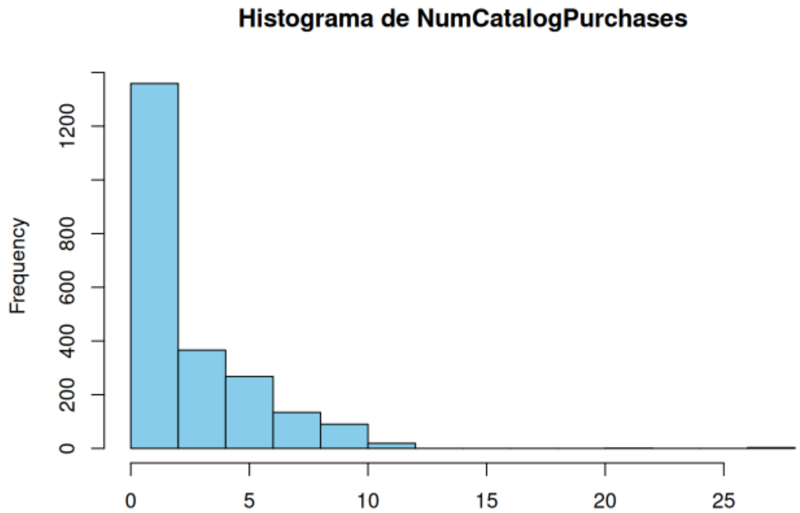


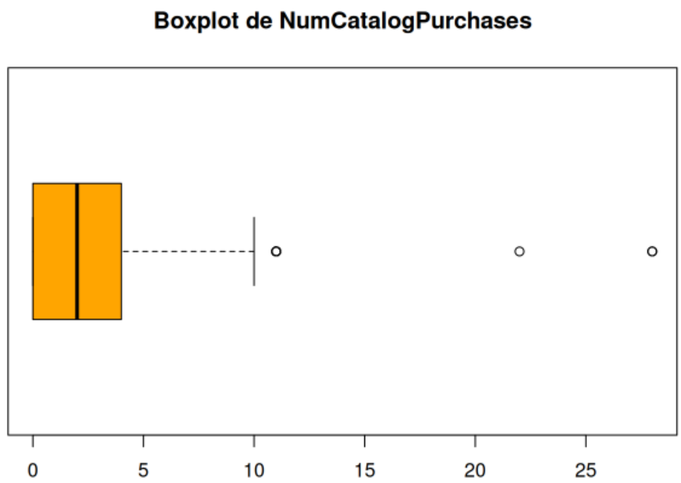

```
##  [1] 3 2 1 2 5 2 4 2 1 1 1 1 1 3 1 1 3 2 2 2
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   1.000   2.000   2.325   3.000  15.000
##
##
## Variable ->  NumWebPurchases
```



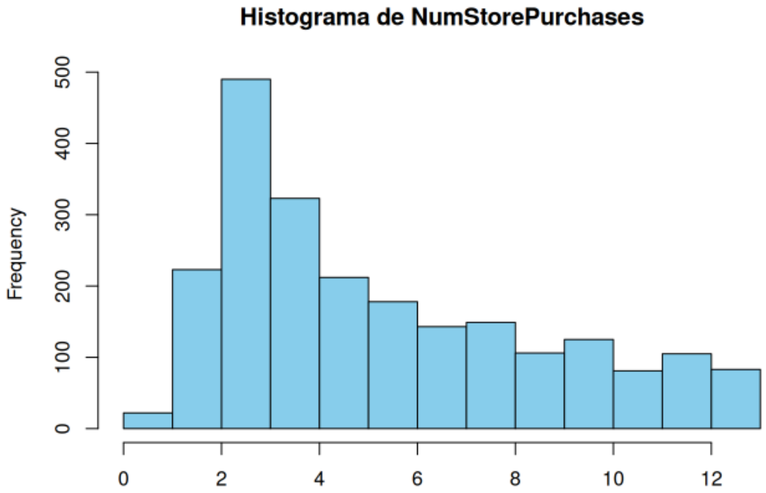


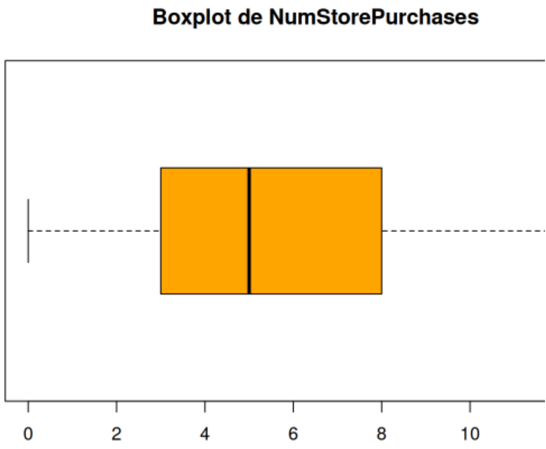

```
##  [1]  8  1  8  2  5  6  7  4  3  1  1  2  3  6  1  7  3  4 11  2
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   2.000   4.000   4.085   6.000  27.000
##
##
## Variable ->  NumCatalogPurchases
```



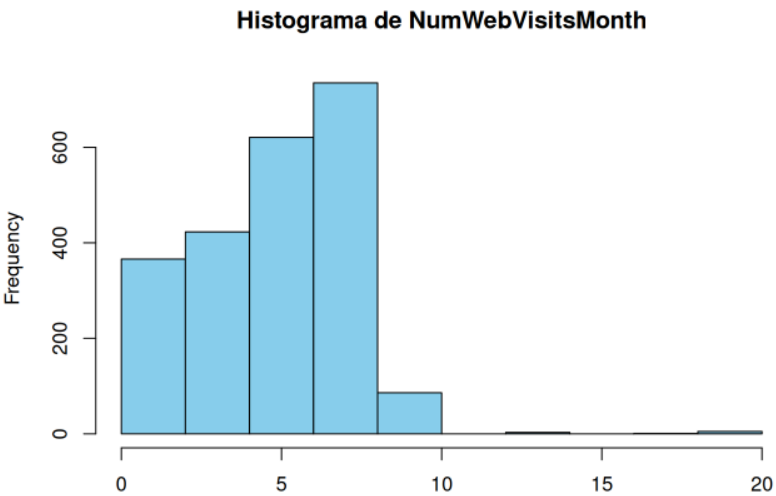


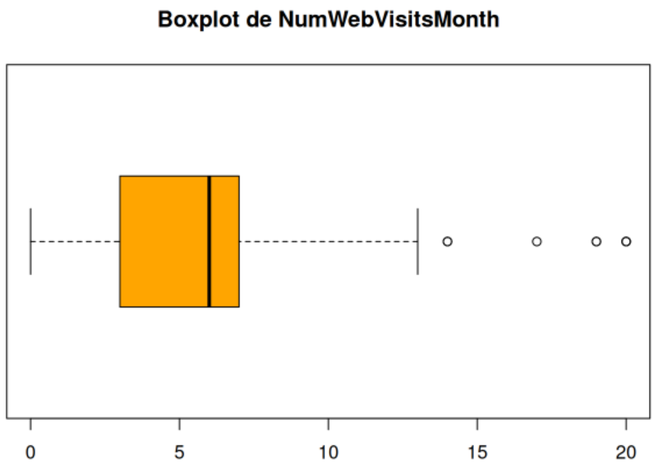

```
##  [1] 10  1  2  0  3  4  3  0  0  0  0  0  4  1  0  6  0  1  4  1
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.000   2.000   2.662   4.000  28.000
##
##
## Variable ->  NumStorePurchases
```



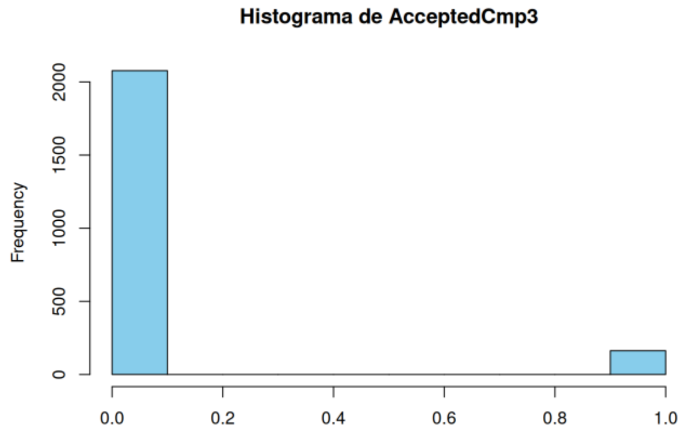


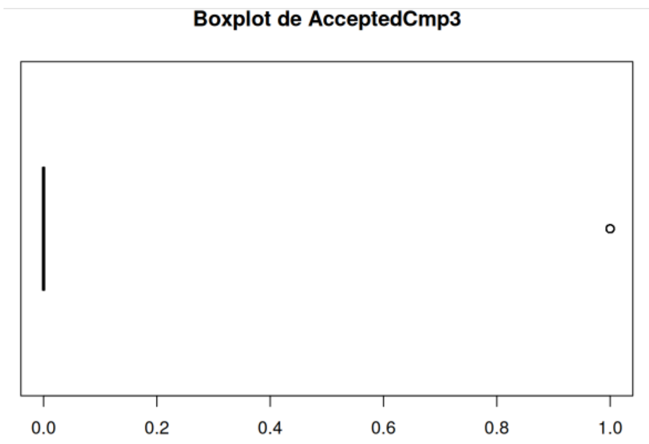

```
## [1]  4  2 10  4  6 10  7  4  2  0  2  3  8  5  3 12  3  6  9  3
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    3.00    5.00    5.79    8.00   13.00
##
##
## Variable ->  NumWebVisitsMonth
```



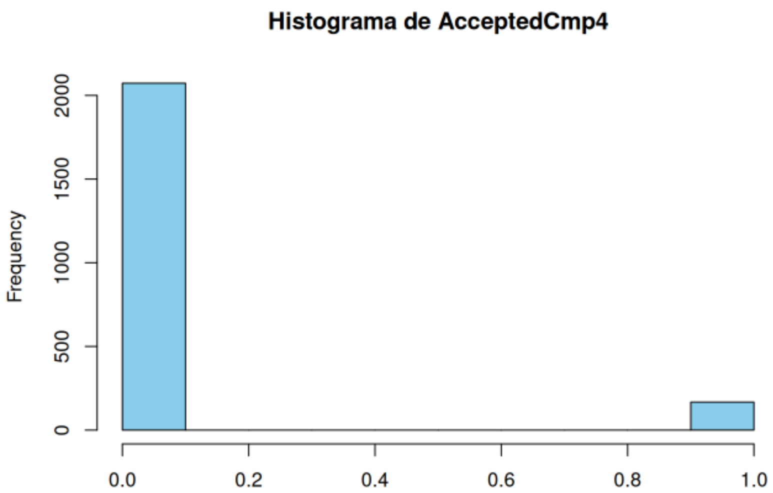


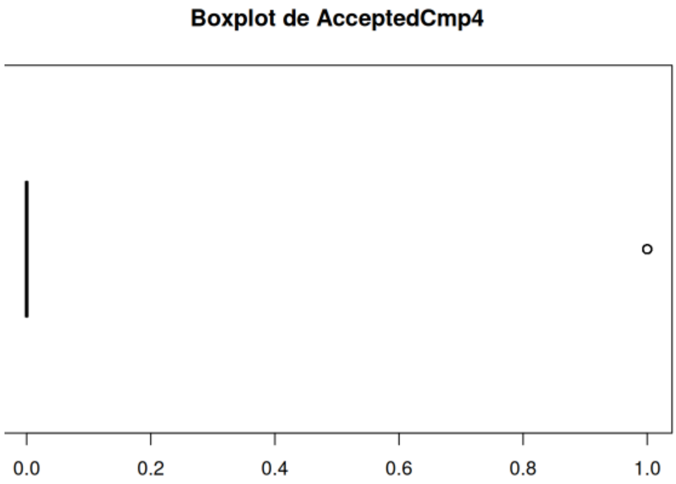

```
## [1]  7  5  4  6  5  6  6  8  9 20  7  8  2  6  8  3  8  7  5  6
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   3.000   6.000   5.317   7.000  20.000
##
##
## Variable ->  AcceptedCmp3
```



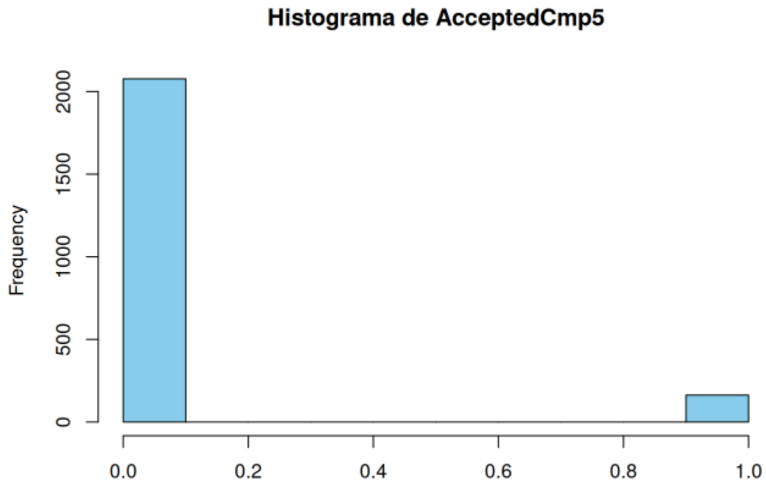


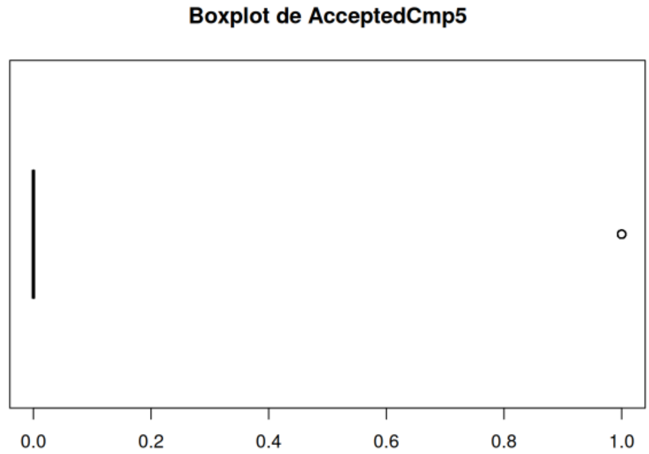

```
##  [1] 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00000 0.00000 0.00000 0.07277 0.00000 1.00000
##
##
## Variable ->  AcceptedCmp4
```



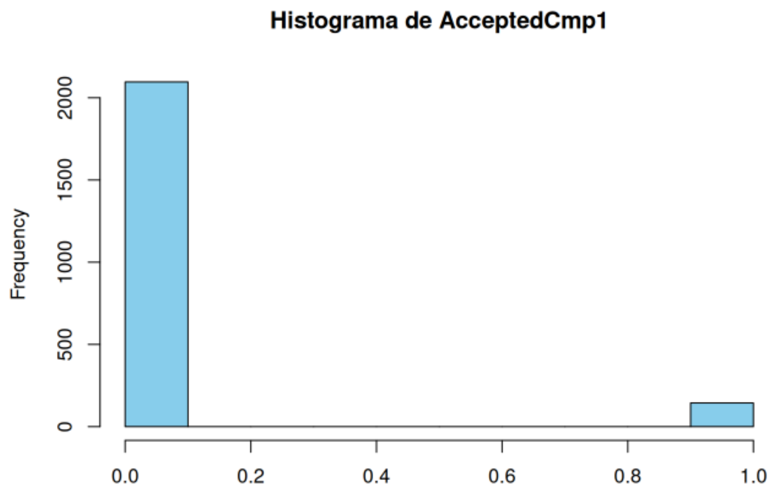


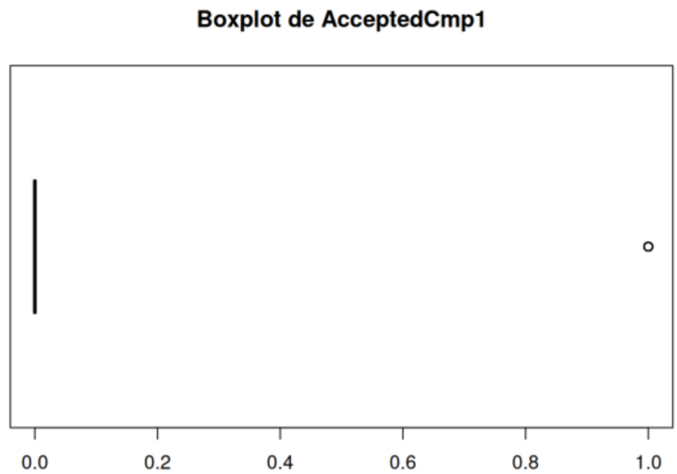

```
##  [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00000 0.00000 0.00000 0.07455 0.00000 1.00000
##
##
## Variable ->  AcceptedCmp5
```



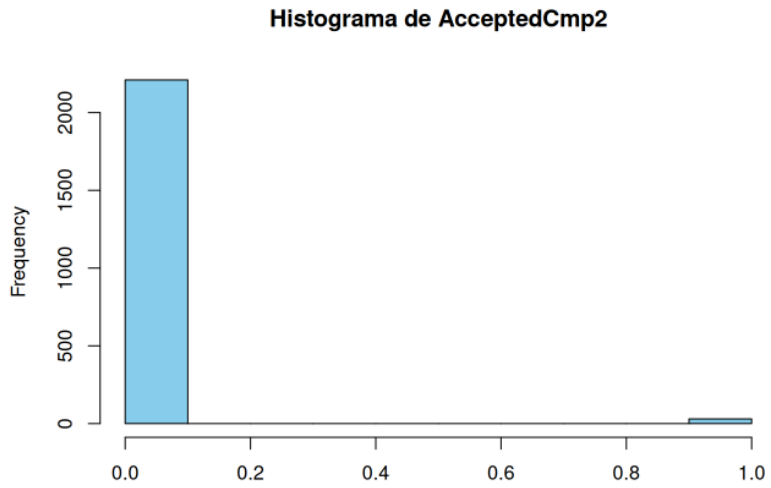


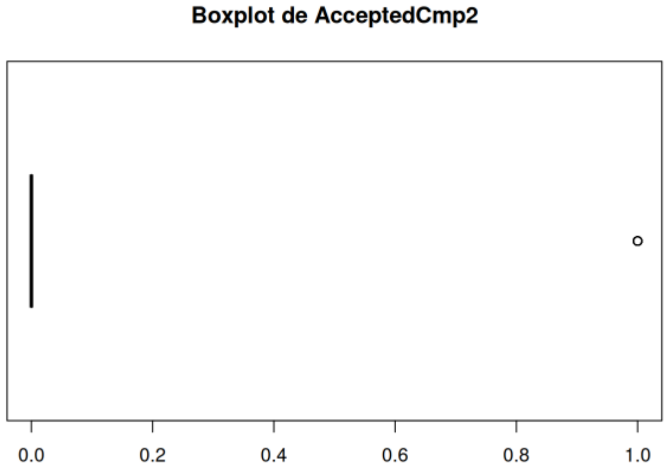

```
##  [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00000 0.00000 0.00000 0.07277 0.00000 1.00000
##
##
## Variable ->  AcceptedCmp1
```



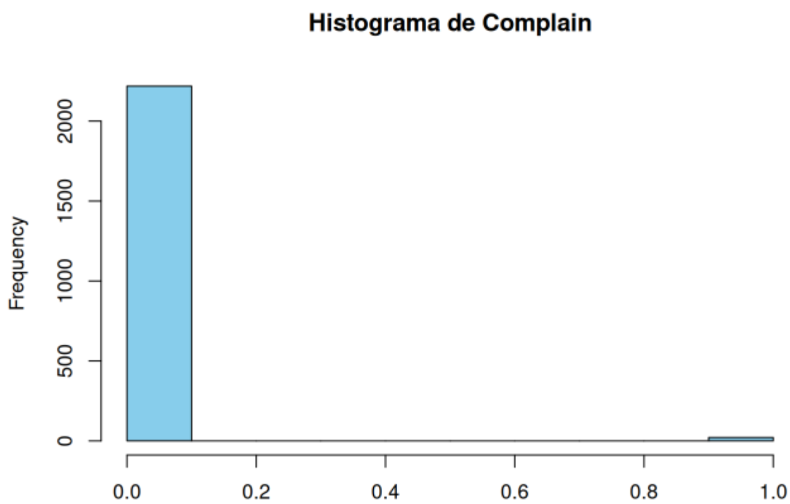


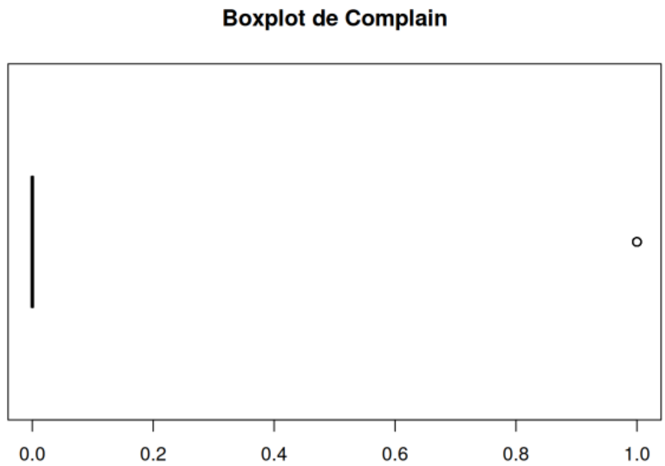

```
##  [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00000 0.00000 0.00000 0.06429 0.00000 1.00000
##
##
## Variable ->  AcceptedCmp2
```



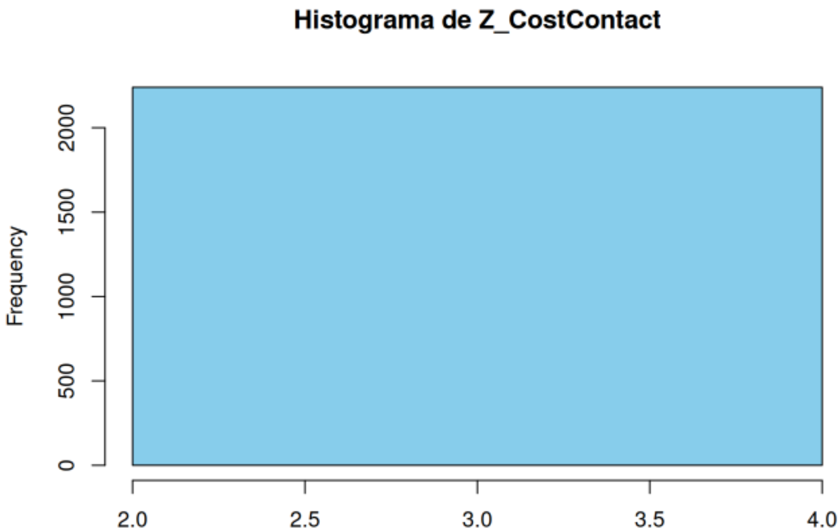


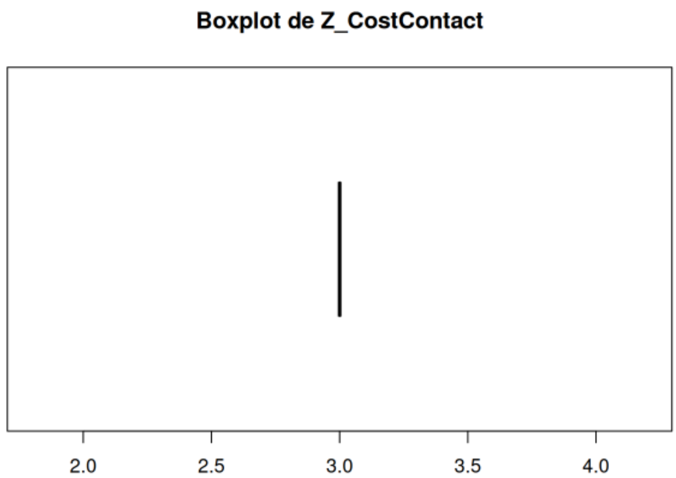

```
##  [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00000 0.00000 0.00000 0.01339 0.00000 1.00000
##
##
## Variable ->  Complain
```



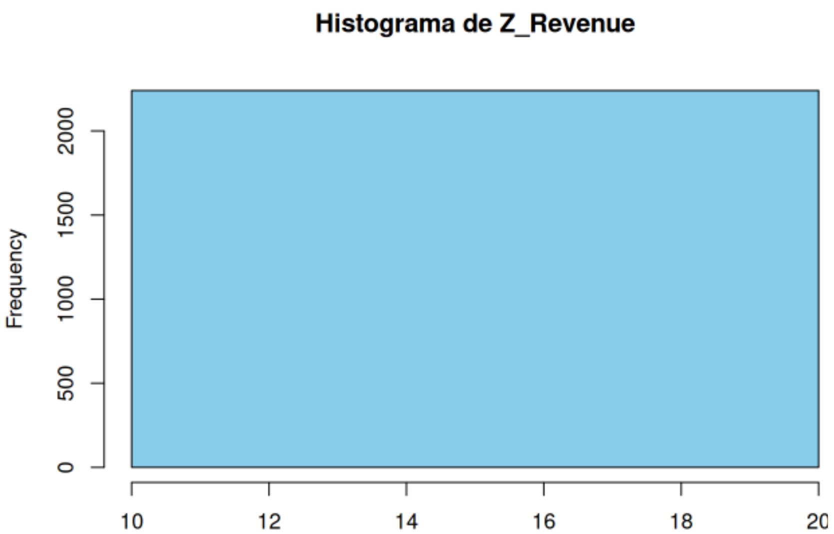


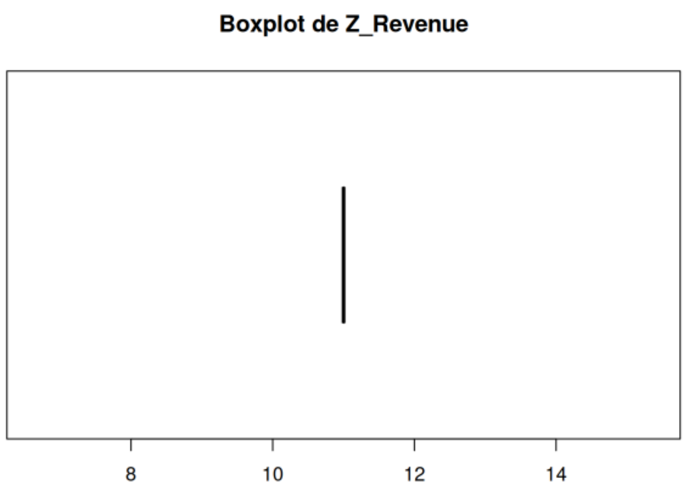

```
##  [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.000000 0.000000 0.000000 0.009375 0.000000 1.000000
##
##
## Variable ->  Z_CostContact
```



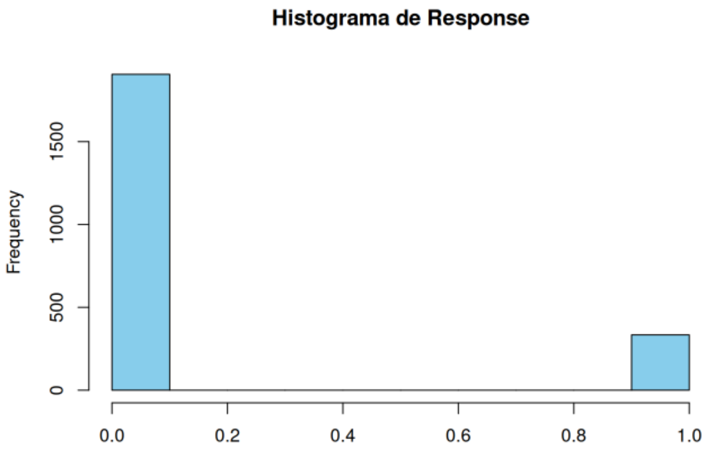


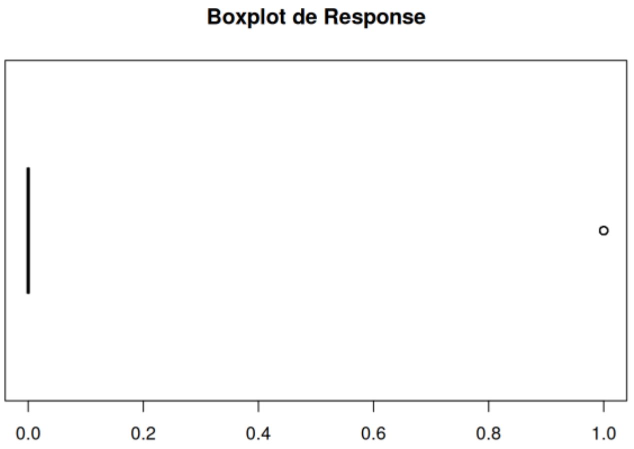

```
##  [1] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       3       3       3       3       3       3
##
##
## Variable ->  Z_Revenue
```






```
##  [1] 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      11      11      11      11      11      11
##
##
## Variable ->  Response
```



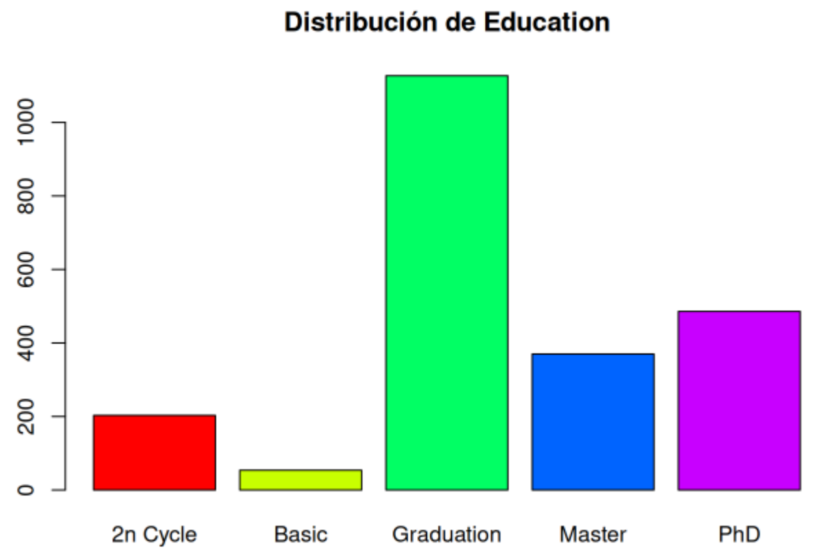


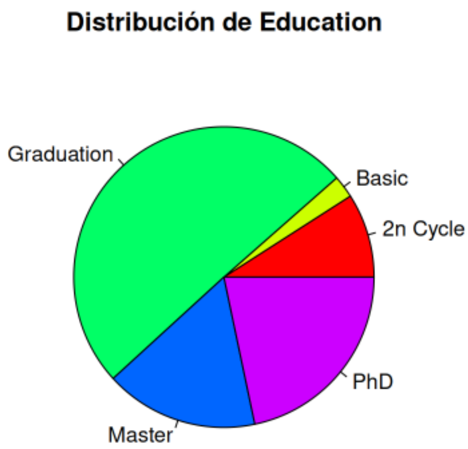

```
##  [1] 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  0.0000  0.1491  0.0000  1.0000
```

Análisis de Variables Categóricas

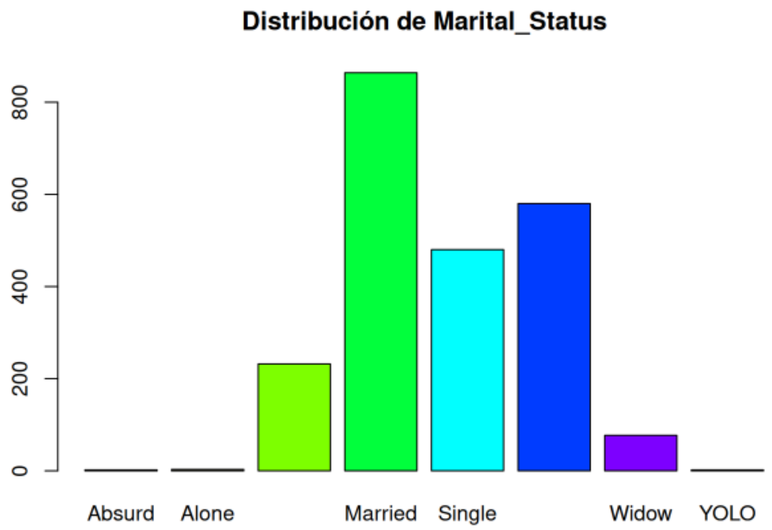
```
# Seleccionar variables categóricas
categoricas <- sapply(ifood, function(x) is.factor(x) | is.character(x))
categoricas <- names(ifood)[categoricas]

# Análisis para cada variable categórica
for (var in categoricas) {
  cat("###", var, "\n\n")

  # Tabla de frecuencias
  print(table(ifood[[var]]))

  # Gráfico de barras
  barplot(table(ifood[[var]]), main=paste("Distribución de", var), col=rainbow(length(unique(ifood[[var]]))))

  # Gráfico de pastel
  pie(table(ifood[[var]]), main=paste("Distribución de", var), col=rainbow(length(unique(ifood[[var]]))))
}
```

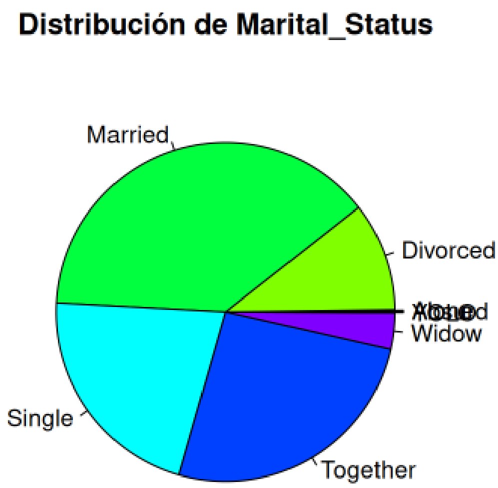
```
## ### Education
##
##
##   2n Cycle      Basic Graduation     Master        PhD
##        203         54       1127        370        486
```






```
## ### Marital_Status
##
##
##    Absurd     Alone  Divorced   Married    Single  Together     Widow      YOLO
##         2         3       232       864       480       580        77         2
```





5. Preprocessing

Explain which preprocessing steps and methods you have applied to your dataset.

Load required libraries

```
# Suppress startup messages of library dplyr
suppressPackageStartupMessages(library(dplyr))
# Loading required libraries
library(dplyr, quietly = TRUE)
library(class, quietly = TRUE)
```

0. Load raw dataset

```
ifood <- read.csv("ml_project1_data.csv", sep=",", header=TRUE, stringsAsFactors = FALSE)
```

1. Remove irrelevant columns

```
ifood <- ifood[, !names(ifood) %in% c("ID", "Z_CostContact", "Z_Revenue")]
```

2. Transform date-related variables

```
ifood$Age <- 2020 - ifood$Year_Birth
ifood <- ifood[, !names(ifood) %in% c("Year_Birth")]
reference_date <- as.Date("2020-12-31")
ifood$CustDays <- as.numeric(reference_date - as.Date(ifood$Dt_Customer, format="%Y-%m-%d"))
ifood <- ifood[, !names(ifood) %in% c("Dt_Customer")]
```

3. Rename columns for easier access

```
colnames(ifood) <- gsub("NumDealsPurchases", "DealsPurc", colnames(ifood))
colnames(ifood) <- gsub("NumWebPurchases", "WebPurc", colnames(ifood))
colnames(ifood) <- gsub("NumStorePurchases", "StorePurc", colnames(ifood))
colnames(ifood) <- gsub("NumWebVisitsMonth", "WebVisits", colnames(ifood))
colnames(ifood) <- gsub("AcceptedCmpOverall", "CmpOverall", colnames(ifood))
colnames(ifood) <- gsub("MntWines", "WineExp", colnames(ifood))
colnames(ifood) <- gsub("MntFruits", "FruitExp", colnames(ifood))
colnames(ifood) <- gsub("MntMeatProducts", "MeatExp", colnames(ifood))
colnames(ifood) <- gsub("MntFishProducts", "FishExp", colnames(ifood))
colnames(ifood) <- gsub("MntSweetProducts", "SweetExp", colnames(ifood))
colnames(ifood) <- gsub("MntGoldProds", "GoldExp", colnames(ifood))
colnames(ifood) <- gsub("Marital_Status", "MaritalSts", colnames(ifood))
colnames(ifood) <- gsub("NumCatalogPurchases", "CatalogPurc", colnames(ifood))
colnames(ifood) <- gsub("AcceptedCmp1", "AccCmp1", colnames(ifood))
colnames(ifood) <- gsub("AcceptedCmp2", "AccCmp2", colnames(ifood))
colnames(ifood) <- gsub("AcceptedCmp3", "AccCmp3", colnames(ifood))
colnames(ifood) <- gsub("AcceptedCmp4", "AccCmp4", colnames(ifood))
colnames(ifood) <- gsub("AcceptedCmp5", "AccCmp5", colnames(ifood))
```

4. Handle outliers

```
ifood$Age <- ifelse(ifood$Age > 80, 80, ifood$Age)
```

5. Handle missing values

```
ifood <- ifood[!ifood$MaritalSts %in% c("YOLO", "Absurd"),]
ifood$MaritalSts[ifood$MaritalSts == "Alone"] <- "Single"
```

6. Impute missing Income using KNN

```
ifood$Income <- ifelse(ifood$Income < 12500, NA, ifood$Income)

num_vars <- sapply(ifood, is.numeric)
complete_vars <- colnames(ifood)[num_vars]
missing_threshold <- 0.2 * nrow(ifood)
complete_vars <- complete_vars[colSums(is.na(ifood[, complete_vars])) < missing_threshold]
aux <- ifood[, complete_vars]

var <- "Income"
aux1 <- aux[!is.na(ifood[[var]]), , drop = FALSE]
aux2 <- aux[is.na(ifood[[var]]), , drop = FALSE]

cols_na <- colnames(aux2)[colSums(is.na(aux2)) > 0]
if (length(cols_na) > 0) {
  aux1 <- aux1[, !(colnames(aux1) %in% cols_na), drop = FALSE]
  aux2 <- aux2[, !(colnames(aux2) %in% cols_na), drop = FALSE]
}

knn_impute <- knn(aux1, aux2, ifood[[var]][!is.na(ifood[[var]])], k = 1)
ifood[[var]][is.na(ifood[[var]])] <- as.numeric(as.character(knn_impute))
```

7. Correct calculation of `TotAccCmp`

```
ifood$TotAccCmp <- ifood$AccCmp1 + ifood$AccCmp2 + ifood$AccCmp3 + ifood$AccCmp4 + ifood$AccCmp5
```

8. Remove duplicate records

```
ifood <- ifood %>% arrange(desc(Response)) %>% distinct_at(vars(-Response), .keep_all = TRUE)
```

9. Create `TotalExp` before using it

```
ifood$TotalExp <- rowSums(ifood[, c("WineExp", "FruitExp", "MeatExp", "FishExp", "SweetExp", "GoldExp")], na.rm = TRUE)
```

10. Save cleaned dataset

```
write.csv(ifood, "ifood_cleaned.csv", row.names = FALSE)
```

Variable Creation

Second-Generation

Total Purchases

```
ifood$TotalPurchases <- ifood$DealsPurc + ifood$WebPurc + ifood$CatalogPurc + ifood$StorePurc
```

Purchase Frequency

```
ifood$PurchaseFrequency <- ifelse(ifood$CustDays > 0, ifood$TotalPurchases / (ifood$CustDays / 30), 0)
```

Preferred Product Category

```
product_categories <- c("WineExp", "FruitExp", "MeatExp", "FishExp", "SweetExp", "GoldExp")
max_index <- apply(ifood[ , product_categories], 1, which.max)
ifood$PreferredProductCategory <- product_categories[max_index]
ifood$PreferredProductCategory <- as.factor(ifood$PreferredProductCategory)
```

Preferred Purchase Channel

```
channels <- c("DealsPurc", "WebPurc", "CatalogPurc", "StorePurc")
max_ch_index <- apply(ifood[ , channels], 1, which.max)
ifood$PreferredChannel <- channels[max_ch_index]
ifood$PreferredChannel <- as.factor(ifood$PreferredChannel)
```

Average Spend Per Purchase

```
ifood$AvgSpendPerPurchase <- ifelse(ifood$TotalPurchases > 0, ifood$TotalExp / ifood$TotalPurchases, 0)
```

HasChildren

```
ifood$HasChildren <- ifelse(ifood$Kidhome + ifood$Teenhome > 0, 1, 0)
```

IncomeSegment

```
income_quantiles <- quantile(ifood$Income, probs = c(0.33, 0.66), na.rm = TRUE)
ifood$IncomeSegment <- cut(ifood$Income, breaks = c(-Inf, income_quantiles[1], income_quantiles[2], Inf),
                           labels = c("Low", "Medium", "High"))
```

CustomerTenure

```
ifood$CustomerTenure <- ifood$CustDays / 365
```

CampaignAcceptanceRate

```
ifood$CampaignAcceptanceRate <- ifelse(ifood$TotAccCmp > 0, ifood$TotAccCmp / 5, 0)
```

Third-Generation

Third-Generation Feature 1: Customer Segmentation via Clustering

Prepare data for clustering: use Recency, TotalPurchases (frequency), and TotalExp (monetary)

```
cluster_data <- ifood %>% select(Recency, TotalPurchases, TotalExp)
```

Scale the data for clustering

```
cluster_data_scaled <- scale(cluster_data)
```

Perform k-means clustering with 3 clusters (as an example)

```
set.seed(123)  # for reproducibility
k3 <- kmeans(cluster_data_scaled, centers = 3, nstart = 25)  # nstart for better convergence
```

Add the cluster assignment as a new feature

```
ifood$CustomerSegment <- as.factor(k3$cluster)
```

(Customers are now labeled 1, 2, or 3 based on their cluster segment)

Third-Generation Feature 2: Propensity Score via Logistic Regression

Fit a logistic regression model to predict campaign response (Response) using relevant features

```
propensity_model <- glm(Response ~ Income + Recency + TotalExp + TotalPurchases + TotAccCmp + Age + MaritalSts,
                        data = ifood, family = binomial)
```

Get predicted probabilities (propensity to respond)

```
ifood$PropensityScore <- predict(propensity_model, ifood, type = "response")
```

(PropensityScore is the model's predicted probability of Response=1 for each customer)

Quick summary of PropensityScore range

```
summary(ifood$PropensityScore)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00000 0.03488 0.07725 0.15313 0.18046 0.99312
```

Third-Generation Feature 3: Engagement Index

Normalize components between 0 and 1

Note: For Recency, a lower value means more recent (more engaged), so we invert it.

```
recency_norm   <- (max(ifood$Recency) - ifood$Recency) / max(ifood$Recency)          # invert recency
frequency_norm <- ifood$TotalPurchases / max(ifood$TotalPurchases)                  # purchases normalized
monetary_norm  <- ifood$TotalExp / max(ifood$TotalExp)                              # spending normalized
campaign_norm  <- (ifood$TotAccCmp + ifood$Response) / 6                            # campaign acceptance (out of 6 campaig
ns total including last response)
webvisit_norm  <- ifood$WebVisits / max(ifood$WebVisits)                            # web visits normalized
```

Calculate engagement index as average of all five components, scaled to 0-100

```
ifood$EngagementIndex <- (recency_norm + frequency_norm + monetary_norm + campaign_norm + webvisit_norm) / 5 * 100
```

Preview EngagementIndex distribution

```
summary(ifood$EngagementIndex)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.892  20.859  27.401  28.453  35.209  62.573
```

Save enriched dataset

```
write.csv(ifood, "ifood_enriched.csv", row.names = FALSE)
```

5.1 Explanation

The dataset preprocessing has been designed to ensure data quality, eliminate inconsistencies, and improve usability for analysis and modeling. Below, we explain the issues detected in the raw dataset and the reasoning behind each preprocessing step.

5.1.1 Issues Detected in the Raw Dataset

Before processing the data, we identified several issues that could affect the quality of the analysis:

1️⃣ **Irrelevant Variables**

- Some columns (**ID, Z_CostContact, Z_Revenue**) did not provide useful information for analysis.
- These columns did not contain relevant values for customer segmentation or behavior prediction.

2️⃣ **Inadequate Formatting of Temporal Variables**

- **Year_Birth**: Represented the customer's birth year. It was not directly interpretable, so it was transformed into **Age** to facilitate its use in models.

- **Dt_Customer**: Date of the first purchase in date format, which made certain analyses difficult. It was transformed into **CustDays** (days since the first purchase until 12/31/2020), converting it into a numerical metric for customer seniority.

3 **Column Name Errors**

- Some variable names were too long or confusing, making them difficult to use in code.
- They were renamed to more manageable versions. Example:
  - **NumWebPurchases → WebPurc**

4 **Outlier Values**

- **Age**: Extremely high values (>120 years) were found, indicating data entry errors or fictitious customers.
  - **Age** was capped at **80 years**, as the life expectancy in Brazil is approximately **76 years**, but many people exceed 80 (for example, women's life expectancy is 79 years).
  - This decision affected only **3 records**, preventing biases from erroneous data while keeping real elderly customers.
- **Income**: Abnormally low incomes (<12500, the annual minimum wage in Brazil in 2020) were detected, likely representing erroneous or outlier values in the distribution.

5 **Missing Data and Invalid Values**

- **MaritalSts**: Contained invalid values (**"YOLO", "Absurd"**), which did not represent legitimate marital status categories.
- **Income**:
  - Some values were **missing (NA)**.
  - Extremely low values (<12500) were considered erroneous and treated as missing data.

6 **Calculation of TotAccCmp**

- **TotAccCmp** was not correctly calculated. Instead of reflecting the total number of accepted campaigns, its value was incorrect.

7 **Duplicate Records**

- **Exact duplicate records** were found, which could bias the analysis if certain customers appear more times than they should.

5.1.2 Step-by-Step Preprocessing Explained

To address these issues, a structured pipeline for data cleaning and transformation was implemented.

1️⃣Removal of Irrelevant Columns

```
JavaScript
ifood<-ifood[,     !names(ifood)     %in%     c("ID",
"Z_CostContact", "Z_Revenue")]
```

Reason: These variables do not provide useful information for the analysis.

- **ID** is a simple identifier.
- **Z_CostContact** and **Z_Revenue** appear to be constant or irrelevant.

2️⃣Transformation of Temporal Variables

```
JavaScript
ifood$Age <- 2020 - ifood$Year_Birth
ifood$CustDays<-as.numeric(reference_date-as.Date(ifoo
d$Dt_Customer, format="%Y-%m-%d"))
```

Reason:

- **Year_Birth** is converted into **Age** because models better interpret age rather than birth year.
- **Dt_Customer** is transformed into **CustDays** (days since the first purchase) so that customer tenure becomes numerical and easy to use.

3️⃣Renaming Columns for Clarity

```
JavaScript
colnames(ifood)<-gsub("NumWebPurchases","WebPurc",coln
ames(ifood))
```

Reason: Some column names were too long (**NumWebPurchases** → **WebPurc**), making dataset writing and analysis more difficult.

4️⃣Handling Outliers

```
JavaScript
ifood$Age<-ifelse(ifood$Age > 80, 80, ifood$Age)
ifood$Income<-ifelse(ifood$Income<12500,          NA,
ifood$Income)
```

Reason:

- Ages over **80** are considered errors or unrepresentative in Brazil, but still allow for the inclusion of real elderly customers.
- Incomes below **12,500** are considered incorrect and are imputed as missing values.

5 Removal of Invalid Data in MaritalSts

```
JavaScript
ifood<-ifood[!ifood$MaritalSts %in% c("YOLO",
"Absurd"), ]
```

Reason: YOLO and **Absurd** are not legitimate marital status categories, so they are removed from the dataset (only **4 records** were affected).

6 Imputation of Missing Values in Income

```
JavaScript
knn_impute<-knn(aux1,aux2,ifood[[var]][!is.na(ifood[[v
ar]])], k = 1)
ifood[[var]][is.na(ifood[[var]])]<-as.numeric(as.chara
cter(knn_impute))
```

Reason: Instead of simply removing missing values in **Income**, they are imputed using **KNN**, leveraging information from other variables.

7 Correct Calculation of TotAccCmp

```
JavaScript
ifood$TotAccCmp<-ifood$AccCmp1+ifood$AccCmp2+ifood$Acc
Cmp3 + ifood$AccCmp4 + ifood$AccCmp5
```

Reason: The variable was not correctly calculated. It now correctly reflects the **total number of accepted campaigns**

8 Removal of Duplicate Records

```
JavaScript
ifood<-ifood%>%arrange(desc(Response))%>%distinct_at(v
ars(-Response), .keep_all = TRUE)
```

Reason: Duplicate records are removed, keeping only **one unique customer per row**. If duplicates exist, priority is given to those who responded **positively** to campaigns.

9 Creation of TotalExp Before Its Use

```JavaScript
ifood$TotalExp <- rowSums(ifood[, c("WineExp", "FruitExp", "MeatExp", "FishExp", "SweetExp", "GoldExp")], na.rm = TRUE)
```

Reason: This variable **did not previously exist**, but it was required for several subsequent calculations.

5.2 Conclusion

This preprocessing process has been essential for correcting errors, improving the coherence of the dataset, and generating new variables useful for marketing analysis. The actions performed are detailed below:

- **Elimination of irrelevant information**: Variables and records that did not contribute value to the analysis were discarded, ensuring a cleaner and more focused dataset.
- **Transformation of dates and column names**: Date formats were standardized, and column names were renamed to facilitate interpretation and ensure better understanding of the data.
- **Correction of errors in age and atypical income**: Incorrect ages were adjusted, and income values that were out of range were normalized, improving the quality of the data for subsequent analysis.
- **Elimination of invalid values (YOLO, Absurd)**: Records with anomalous values, such as "YOLO" or "Absurd", which were clearly erroneous data, were removed.
- **Imputation of missing values in "Income" using KNN**: An imputation process was applied to replace missing values in the "Income" field, using the KNN (K-Nearest Neighbors) algorithm, which ensures accurate estimation based on nearby data.
- **Correct calculation of "TotAccCmp"**: The formula for "TotAccCmp" was corrected to ensure that the values were consistent and accurate, improving the integrity of the dataset.
- **Elimination of duplicates**: Duplicate records were removed to avoid bias in the analysis and ensure that each observation in the dataset is unique.
- **Creation of the "TotalExp" variable**: A new variable, "TotalExp", was created to facilitate the analysis of customers' total expenditures, enabling more effective segmentation and study of consumption behavior.

With these adjustments, the data is now ready to be used in prediction models and advanced segmentation.

5.3 Creation of New Variables:

Once the iFood dataset has been cleaned and refined, it is crucial to generate new variables to enhance the quality of the analysis and better align with the needs of data mining models (an idea based on the paper by Karina Gisbert: *A Survey on Pre-processing Techniques: Relevant Issues in the Context of Environmental Data Mining*).

The new variables are derived from the original ones using various techniques we have implemented and represent our proposed comprehensive preprocessing for the dataset.

5.3.1 Second-Generation Variables:

These refer to all variables based on expert knowledge, derived from domain-specific understanding. In our case, the "experts" are ourselves, as we have researched and studied the field to create these variables, allowing us to represent the concepts used by industry professionals in their reasoning.
These variables are derived by combining or transforming original data to better capture certain customer behaviors or characteristics.

- **TotalPurchases**: The total number of purchases a customer made across all channels.

  Calculated as the sum of *DealsPurc + WebPurc + CatalogPurc + StorePurc*.

  Why? This single metric captures overall purchase volume, indicating how active the customer is in transactions.

- **PurchaseFrequency**: Purchase frequency per month.

  Calculated as: *PurchaseFrequency = TotalPurchases / (CustDays / 30)*, where CustDays represents the total number of days since the customer's first purchase.

  Why? This metric quantifies how often a customer makes a purchase, normalized per month so high values indicate frequent buyers, while low values suggest occasional or inactive customers. (e.g., useful for identifying loyal, high-frequency customers vs. sporadic buyers, enabling better-targeted retention and promotional strategies).

- **AverageSpendPerPurchase**: Average expenditure per transaction.

  Computed as *TotalExp / TotalPurchases* (the total monetary spend divided by the total number of purchases).

  Why? This reveals whether the customer tends to make large purchases in each visit or smaller, lower-value transactions. It's useful for identifying big spenders vs. bargain shoppers.

- **PreferredProductCategory**: Favorite product category based on spend.

For each customer, identify which of the product categories (WineExp, FruitExp, MeatExp, FishExp, SweetExp, GoldExp) is highest.

Why? This variable describes the customer's primary interest, allowing more personalized marketing – e.g., wine lovers vs. meat lovers have different profiles.

- **PreferredChannel**: Preferred shopping channel.

  Determine the channel through which the customer made the most purchases: compare WebPurc, CatalogPurc, StorePurc. The new variable is a category like "Web", "Store", "Catalog", or "Deals" for the channel with the highest purchase count for that customer. (We are considering Deals as another channel for purchasing.)

  Why? This shows where the customer is most comfortable shopping, informing channel-specific strategies — e.g., online-oriented vs. in-store shoppers.

- **HasChildren**: Customers with children.

  Calculated as a binary flag: *1 if TotalChildren > 0, else 0*.

  Why? This explicitly distinguishes customers with families from those without, which could be insightful for segmentation.

- **IncomeSegment**: Customer income segment.

  Categorizes customers into Low, Medium, or High income based on terciles of the Income distribution.

  Calculated by computing income quantiles (33rd and 66th percentiles) and assigning each customer to one of the three segments:

  - *Low*: Income in the bottom third.
  - *Medium*: Income in the middle third.
  - *High*: Income in the top third.

  Why? This segmentation allows for differentiated marketing strategies based on spending power. — e.g., High-income customers might be more interested in gold products, or Low-income customers in promotions.

- **CustomerTenure**: Customer tenure in years. Measures how long the customer has been with the company.

  Calculated as: *CustomerTenure = CustDays / 365*, converting days since first purchase into years.

  Why? Identifies new vs. loyal customers for tailored retention strategies. e.g., Long-tenure customers may deserve loyalty rewards, while newer customers might need onboarding incentives.

- **CampaignAcceptanceRate**: Campaign acceptance rate. The percentage of marketing campaigns the customer has accepted.

  Computed as: *CampaignAcceptanceRate = TotAccCmp / 5*, where TotAccCmp is the total number of accepted campaigns (sum of AccCmp1 to AccCmp5).

  Why? High values indicate customers highly responsive to promotions, ideal for frequent marketing engagement. e.g., Low values signal less reactive customers, requiring stronger incentives or different approaches.

5.3.2 Third-Generation Variables:

These variables have been developed through independent research on various techniques, and it is essential to understand that this is a proposed approach. We consider it an additional enhancement to the given task (even though it is technically necessary).

We refer to variables generated using more sophisticated data analysis techniques (e.g., Clustering: k-means) to synthesize multiple original variables into more compact and useful indicators.

- **CustomerSegment**: A segment label for each customer, determined via clustering on key behaviors (e.g., spending, frequency, recency).

  We have performed k-means clustering to group similar customers and assign a segment number.

  Why? Identifies high-value active customers vs. low-value customers. (e.g., Targeted marketing)

- **PropensityScore**: A predictive score indicating the customer's propensity to respond to marketing campaigns.

  We trained a simple logistic regression model using existing data (e.g., previous campaign acceptances and customer attributes) to estimate the probability of a positive response, and use this probability as the propensity score.

  Why? Helps prioritize high-propensity customers for marketing campaigns.

  EngagementIndex: A composite index that quantifies overall customer engagement.

  This index will combine multiple factors (recency of purchases, frequency of purchases, total spending, campaign responsiveness, and website visits) into a single score (e.g., scaled 0–100).

  Why? Captures overall customer involvement across purchases, campaigns, and web interactions.

We compute each of these step by step:

1. Customer Segmentation via Clustering

We use k-means clustering on key features to segment customers. Here we used RFM-style features: Recency, Frequency, and Monetary value (Recency, TotalPurchases, and TotalExp respectively). Before clustering, we scale these features to ensure equal weight. We choose 3 clusters for simplicity (this can be adjusted in the near future). After clustering, we assign each customer a segment label (1, 2, 3):

- Third-Generation Feature 1: Customer Segmentation via Clustering -

To prepare the data for clustering, we selected the features: Recency, TotalPurchases (frequency), and TotalExp (monetary). The next step is to scale the data for clustering purposes:

```python
# Prepare data for clustering: use Recency, TotalPurchases (frequency), and TotalExp (monetary)
cluster_data <- ifood %>% select(Recency, TotalPurchases, TotalExp)
# Scale the data for clustering
cluster_data_scaled <- scale(cluster_data)
# Perform k-means clustering with 3 clusters (as an example)
set.seed(123)  # for reproducibility
k3 <- kmeans(cluster_data_scaled, centers = 3, nstart = 25)  # nstart for better convergence
# Add the cluster assignment as a new feature
ifood$CustomerSegment <- as.factor(k3$cluster)
# (Customers are now labeled 1, 2, or 3 based on their cluster segment)
```

We used 3 clusters to define broad segments (e.g., Segment 1, Segment 2, Segment 3). Depending on the clustering outcome, these might correspond to profiles such as "High-Value Customers", "Occasional Buyers", etc., but for now, they are simply numerical labels. (We converted `CustomerSegment` to a factor for clarity.)

2. Propensity Score (Predicted Response Probability)

To estimate each customer's propensity to respond to marketing campaigns, we fit a logistic regression model using the cleaned dataset. The model predicts the probability of a customer accepting the next campaign (response variable) based on their characteristics (e.g., income, recency, past campaign acceptances, etc.). We then extract the predicted probability as the `PropensityScore` for each customer:

```
Python
# --- Third-Generation Feature 2: Propensity Score via Logistic Regression ---
# Fit a logistic regression model to predict campaign response (Response) using relevant features
propensity_model <- glm(Response ~ Income + Recency + TotalExp + TotalPurchases + TotAccCmp + Age + MaritalSts, data = ifood, family = binomial)
# Get predicted probabilities (propensity to respond)
ifood$PropensityScore <- predict(propensity_model, ifood, type = "response")
# (PropensityScore is the model's predicted probability of Response=1 for each customer)
# Quick summary of PropensityScore range
summary(ifood$PropensityScore)
```

In this case, we included features such as Income, Recency, TotalExpenditure, TotalPurchases, total past accepted campaigns, Age, and Marital Status as predictors. The resulting `PropensityScore` is a number between 0 and 1 (closer to 1 means the model predicts a higher likelihood the customer will respond positively to a campaign). In the near future, we might refine feature selection or use more advanced models.

3. Engagement Index

The `EngagementIndex` is designed to summarize how actively engaged a customer is with the brand across various dimensions. We combine multiple behaviors into one score. Specifically, we include:

- **Recency** (how recently the customer purchased, with more recent = more engaged),
- **Frequency** (total purchases),
- **Monetary** (total spending),
- **Campaign responsiveness** (number of campaigns accepted),
- **Web visit frequency** (number of web visits).

We will normalize each component to a 0–1 scale, then take an average (and scale to 0–100 for convenience):

```
Python
# --- Third-Generation Feature 3: Engagement Index ---
# Normalize components between 0 and 1
```

```r
# Note: For Recency, a lower value means more recent (more engaged), so we invert it.
recency_norm <- (max(ifood$Recency) - ifood$Recency) / max(ifood$Recency) # invert recency
frequency_norm <- ifood$TotalPurchases / max(ifood$TotalPurchases) # purchases normalized
monetary_norm <- ifood$TotalExp / max(ifood$TotalExp) # spending normalized
campaign_norm <- (ifood$TotAccCmp + ifood$Response) / 6 # campaign acceptance (out of 6 campaigns total including last response)
webvisit_norm <- ifood$WebVisits / max(ifood$WebVisits) # web visits normalized
# Calculate engagement index as average of all five components, scaled to 0-100
ifood$EngagementIndex <- (recency_norm + frequency_norm + monetary_norm + campaign_norm + webvisit_norm) / 5 * 100
# Preview EngagementIndex distribution
summary(ifood$EngagementIndex)
```

In the code above, we treated a customer as more engaged if they have recent purchases (low `Recency` -> high `recency_norm`), frequent purchases (high `frequency_norm`), high spending (high `monetary_norm`), multiple accepted campaigns (`campaign_norm` accounts for 5 previous campaigns plus the latest response), and frequent web visits (high `webvisit_norm`). The final `EngagementIndex` is an average of these factors on a 0–100 scale. This provides a single metric to compare overall engagement levels across customers.

Conclusion of Third-Generation Features:

This set of third-generation variables is entirely our own proposal, developed through independent research. While not originally required, we have designed and implemented these variables to enhance the dataset by applying advanced data analysis techniques, ensuring a deeper and more structured understanding of customer behavior.

6. Basic description of post processing in this variables with changes

After preprocessing and enriching the dataset, several transformations have been applied to certain variables.

In general terms, the impact of preprocessing on the dataset has resulted in a reduction from 2.240 instances to 2.031 and an expansion from 29 variables to 35.

Original dataset:

```
# Cargar dataset
ifood <- read.csv("ml_project1_data.csv", sep=",", header=TRUE)
```

```
# Ver estructura del dataset
str(ifood)

## 'data.frame':    2240 obs. of  29 variables:
```

Enriched dataset:

```
# Cargar dataset
ifood <- read.csv("ifood_enriched.csv", sep=",", header=TRUE)
```

```
# Ver estructura del dataset
str(ifood)

## 'data.frame':    2031 obs. of  35 variables:
```

6.1 Main changes in variables

Based on the comparison of the analyses, the main changes we did in the following variables are:

- Numerical variables:
  - Income, Kidhome, Teenhome, and Recency: outlier correction and handling of missing values.
  - Product expenditures (WineExp, FruitExp, MeatExp, FishExp, SweetExp, GoldExp): summarized into a new variable (TotalExp).
  - Number of purchases per channel (WebPurc, StorePurc, CatalogPurc, DealsPurc): summarized into a new variable (TotalPurchases).
  - Accepted campaigns (AcceptedCmp1, AcceptedCmp2, AcceptedCmp3, AcceptedCmp4, and AcceptedCmp5): summarized into a new variable (TotalAccCmp).

- PurchaseFrequency, CustomerSegment, PropensityScore, and EngagementIndex: new derived variables added to the enriched dataset.
- Other variables were simply renamed to make them shorter (for example, WebVisitsMonth to WebVisits).
- Year_Birth was transformed into Age.

- Categorical variables:
  - Education and Marital_Status: outlier correction and handling of missing values.
  - PreferredProductCategory and PreferredChannel: new derived variables added to the enriched dataset.

6.2 Visualization of changes

In the analysis documents (created from "ifood_analysis.Rmd"), we have included histograms and boxplots for numerical variables, where we can observe the following:

- Histograms: more homogeneous distributions after preprocessing.
- Boxplots: reduction of outliers, indicating effective data cleaning.

Additionally, bar charts and pie charts are included for categorical variables, showing, after preprocessing, a more uniform distribution of previously redundant or residual instances.

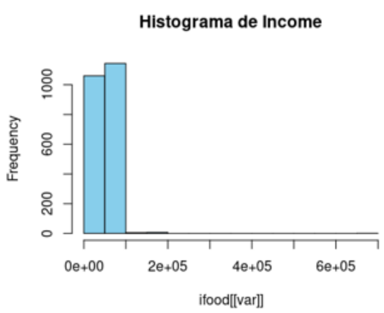
6.3 Statistics' summary after processing

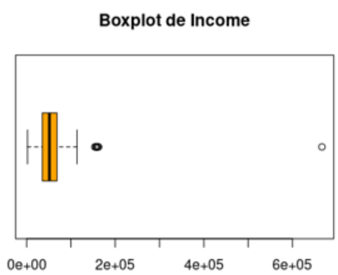
After preprocessing, the descriptive statistics reveal:

- Reduction of extreme values: variables such as "Income" exhibit less extreme variability:

Original dataset:

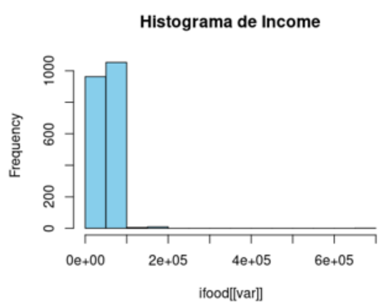
```
## ### Income
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    1730   35303   51382   52247   68522  666666      24
```

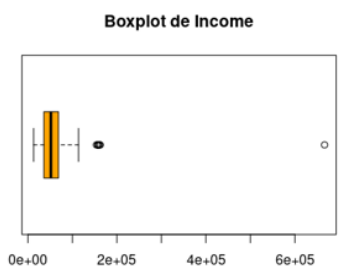




Enriched dataset:

```
## ### Income
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   12571   35828   51563   52844   68656  666666
```





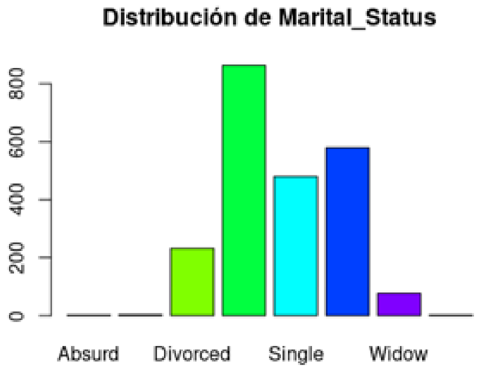
- Uniformization of categorical variable values, for example, in “Marital_Status” (merging of categories like “Single”, “Alone”, etc.):

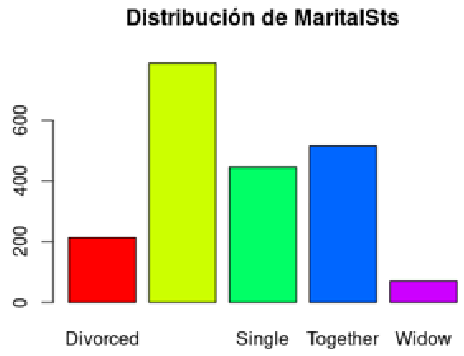
Original dataset:

```
## ### Marital_Status
##
##
##    Absurd     Alone Divorced  Married   Single  Together     Widow
YOLO
##         2         3      232      864      480      580        77
```

Enriched dataset:

```
## ### MaritalSts
##
##
## Divorced  Married   Single Together    Widow
##      213      787      445      516       70
```





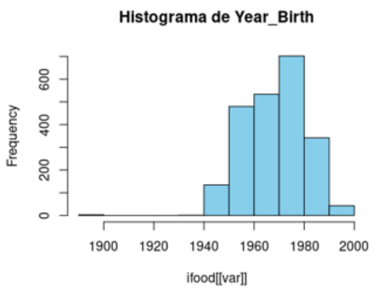
- More balanced distributions: fewer outliers can be observed in the enriched dataset, such as in the “Year_Birth” variable, which was transformed into “Age”:

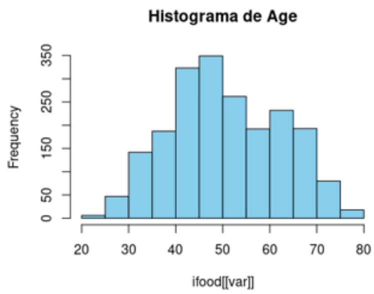
Original dataset:

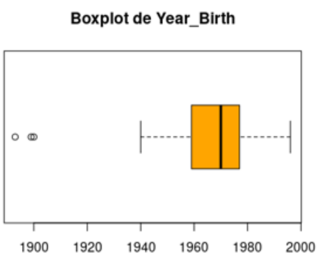
```
## ### Year_Birth
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1893    1959    1970    1969    1977    1996
```

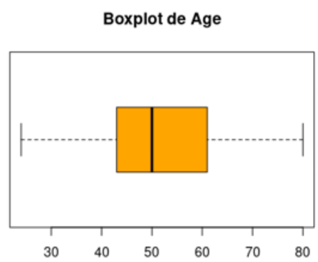
Enriched dataset:

```
## ### Age
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    24.0    43.0    50.0    51.2    61.0    80.0
```



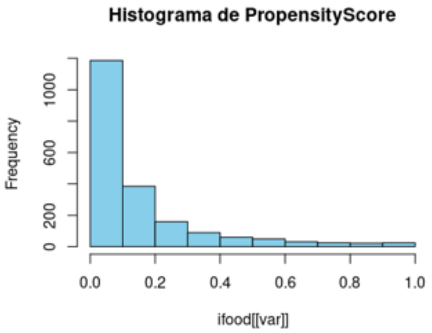


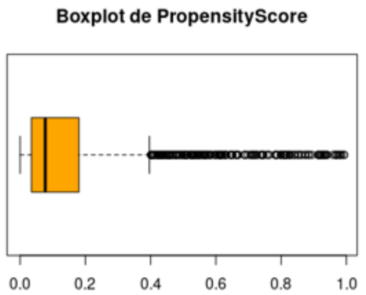




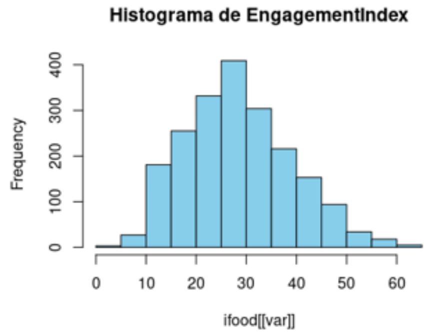
- New significant variables: “PropensityScore” and “EngagementIndex” provide additional analytical insights:
  Enriched dataset:

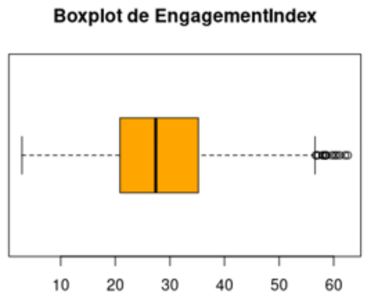
```
## ### PropensityScore
##     Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00000 0.03488 0.07725 0.15313 0.18046 0.99312
```






```
## ### EngagementIndex
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.892  20.859  27.401  28.453  35.209  62.573
```





6.4 Conclusions

The adjustments we made have allowed for:

- A more precise and refined dataset for analysis.
- The introduction of new variables such as “EngagementIndex” and “PropensityScore” to generate more meaningful insights.
- Improved reliability of descriptive statistics, thanks to the handling of outliers and missing values.