Project developed by Thiago Bulgarelli

Contact: bugath36@gmail.com

Electric Car Energy Consumption Forecast with Machine Learning

Description and Introduction to the Business Problem

"A transportation and logistics company wants to transition its fleet to electric cars in order to reduce costs. Before making the decision, the company would like to forecast the energy consumption of electric cars based on various usage factors and vehicle characteristics.

Using a dataset with publicly available real data, we need to build a machine learning model capable of predicting the energy consumption of electric cars based on various factors, such as the type and number of electric motors, vehicle weight, cargo capacity, and other attributes.

For the construction of this project, we will use the R programming language. The dataset can be found at the following link: https://data.mendeley.com/datasets/tb9yrptydn/2.

This dataset lists all fully electric cars currently available on the market, along with their attributes (properties). The collection does not include data on hybrid cars and electric cars with "range extenders." Hydrogen cars are also not included in the dataset due to the insufficient number of mass-produced models and their different specificities compared to electric vehicles, including different charging methods.

The dataset includes cars that, as of December 2, 2020, could be purchased as new in Poland from authorized dealers, as well as those available for public pr-sale, provided a publicly available price list was available. The list does not include discontinued cars that cannot be purchased as new from an authorized dealer (even if they are not available in stock).

The electric car dataset includes all fully electric cars in the primary market, obtained from official materials (technical specifications and catalogs) provided by car manufacturers licensed to sell cars in Poland. These materials were downloaded from their official websites. If the data provided by the manufacturer was incomplete, the information was supplemented with data from the Auto Catalog SAMAR.

Our task is to build an ML model capable of predicting the energy consumption of electric vehicles."

Project developed by Thiago Bulgarelli

Contact: bugath36@gmail.com

Data Dictionary

Variable Name	Description
Car Full Name	Full name of the vehicle
Make	Vehicle manufacturer
Model	Vehicle model
Minimal price	Minimum selling price of the vehicle in [US]
Engine power	Power declared by the vehicle's engine manufacturer, measured in [kW]
Maximum torque	Maximum torque of the engine measured in [Nm]
Type of brakes	Type of brake system
Drive type	Type of drive
Battery capacity	Battery capacity in [kWh]
Range	Battery range in [km]
Wheelbase	Wheelbase distance in [cm]
Length	Vehicle length in [cm]
Width	Vehicle width in [cm]
Heigth	Vehicle height in [cm]
Minimal Empty weight	Empty weight measured in [kg]
Permissable gross weight	Maximum allowed weight measured in [kg]
Maximum load capacity	Maximum load capacity measured in [kg]
Number of seats	Number of seats in the vehicle
Number of doors	Number of doors in the vehicle
Tire size	Tire radius size measured in [inches]
Maximum speed	Maximum speed measured in [km/h]
Boot capacity	Trunk capacity measured in [l]
Acceleration 0-100	Acceleration time from 0 to 100 [s]
Maximum DC charging power	Maximum battery charging capacity measured in [kW]
Mean Energy Consumption [kWh/100 km]	Average battery consumption measured in kWh per 100 km traveled

Packages Used

Using the data from the dataset, our objective is to predict the electricity consumption of the base models and identify important insights and potential improvement opportunities.

Project developed by Thiago Bulgarelli

Contact: bugath36@gmail.com

```
# Pacotes utilizados
  library(readxl)
  library(thinkr)
  library(usefun)
  library(Amelia)
```

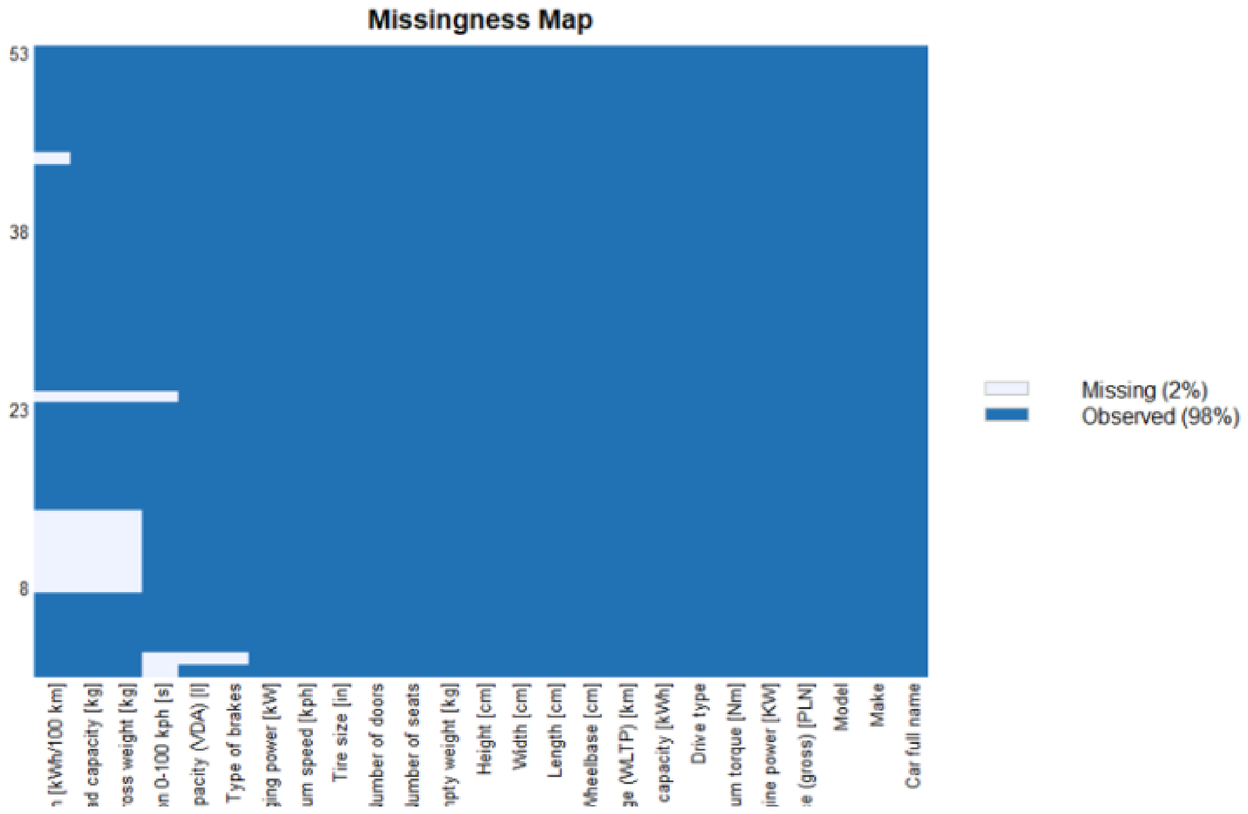
Loading the Data and Adjusting the Dataset

```
# Carregando e Ajustando o Dataset
  getwd()
  ?read_xlsx

  dadosv00 <- read_xlsx('dados/fev_dataset.xlsx', na = 'NaN')

  View(dadosv00)
  dim(dadosv00)
  str(dadosv00)
```

```
>   str(dadosv00)
tibble [53 × 25] (S3: tbl_df/tbl/data.frame)
 $ Car full name                         : chr [1:53] "Audi e-tron 55 quattro" "Audi e-tron 50 quattro" "Audi e-tron S quattro" "Audi e-tron Sportback 50 quattro" ...
 $ Make                                  : chr [1:53] "Audi" "Audi" "Audi" "Audi" ...
 $ Model                                 : chr [1:53] "e-tron 55 quattro" "e-tron 50 quattro" "e-tron S quattro" "e-tron Sportback 50 quattro" ...
 $ Minimal price (gross) [PLN]           : num [1:53] 345700 308400 414900 319700 357000 ...
 $ Engine power [KW]                     : num [1:53] 360 313 503 313 360 503 170 184 286 136 ...
 $ Maximum torque [Nm]                   : num [1:53] 664 540 973 540 664 973 250 270 400 260 ...
 $ Type of brakes                        : chr [1:53] "disc (front + rear)" "disc (front + rear)" "disc (front + rear)" "disc (front + rear)" ...
 $ Drive type                            : chr [1:53] "4WD" "4WD" "4WD" "4WD" ...
 $ Battery capacity [kWh]                : num [1:53] 95 71 95 71 95 95 42.2 42.2 80 50 ...
 $ Range (WLTP) [km]                     : num [1:53] 438 340 364 346 447 369 359 345 460 350 ...
 $ Wheelbase [cm]                        : num [1:53] 293 293 293 293 293 ...
 $ Length [cm]                           : num [1:53] 490 490 490 490 490 ...
 $ Width [cm]                            : num [1:53] 194 194 198 194 194 ...
 $ Height [cm]                           : num [1:53] 163 163 163 162 162 ...
 $ Minimal empty weight [kg]             : num [1:53] 2565 2445 2695 2445 2595 ...
 $ Permissable gross weight [kg]         : num [1:53] 3130 3040 3130 3040 3130 ...
 $ Maximum load capacity [kg]            : num [1:53] 640 670 565 640 670 565 440 440 540 459 ...
 $ Number of seats                       : num [1:53] 5 5 5 5 5 5 4 4 5 5 ...
 $ Number of doors                       : num [1:53] 5 5 5 5 5 5 5 5 5 5 ...
 $ Tire size [in]                        : num [1:53] 19 19 20 19 19 20 19 20 19 16 ...
 $ Maximum speed [kph]                   : num [1:53] 200 190 210 190 200 210 160 160 180 150 ...
 $ Boot capacity (VDA) [l]               : num [1:53] 660 660 660 615 615 615 260 260 510 380 ...
 $ Acceleration 0-100 kph [s]            : num [1:53] 5.7 6.8 4.5 6.8 5.7 4.5 8.1 6.9 6.8 9.5 ...
 $ Maximum DC charging power [kW]        : num [1:53] 150 150 150 150 150 150 50 50 150 100 ...
 $ mean - Energy consumption [kWh/100 km]: num [1:53] 24.4 23.8 27.6 23.3 23.9 ...
```

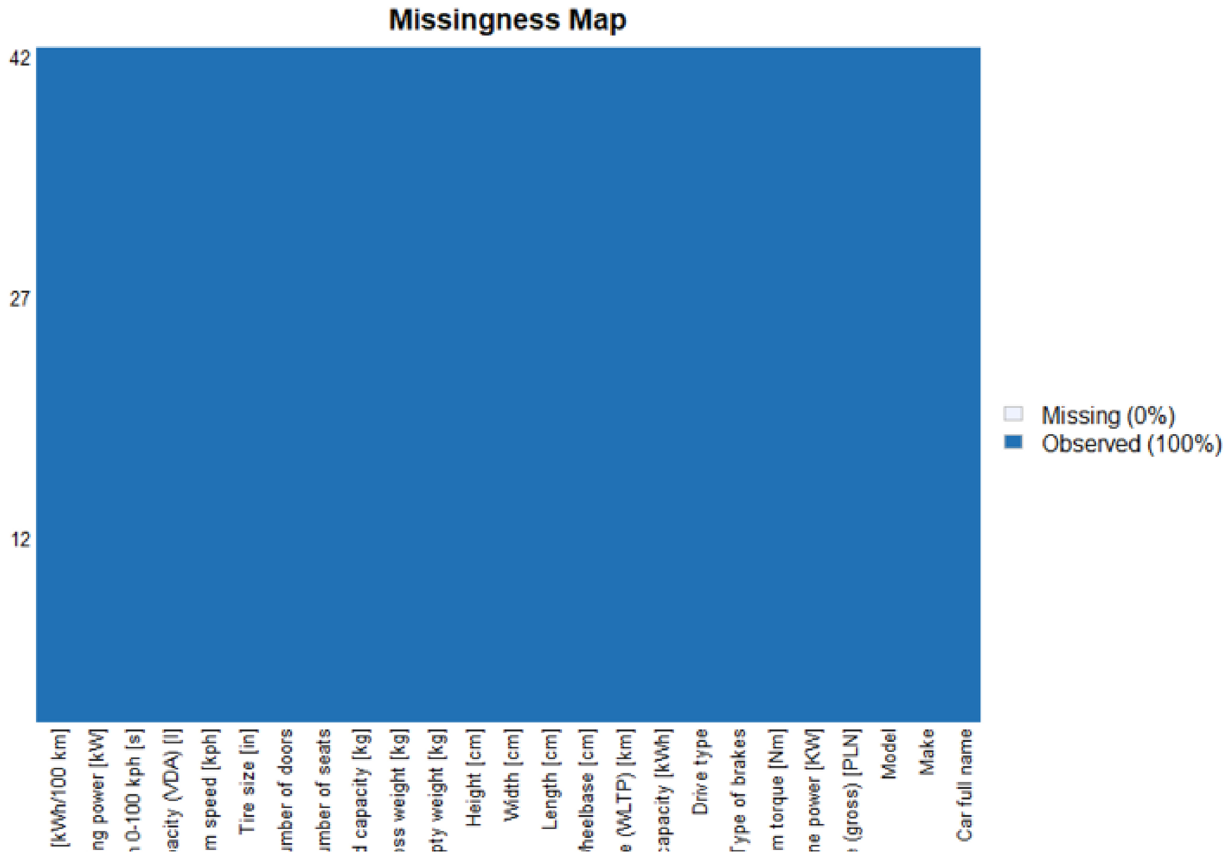
```
# Convertendo o objeto para DataFrame
  dadosv00 <- as.data.frame(dadosv00)

# Verificando Dados NaN
  summary(is.na(dadosv00))
  colSums(is.na(dadosv00))
  missmap(dadosv00)
```

```
>   summary(is.na(dadosv00))
 Car full name       Make            Model          Minimal price (gross) [PLN] Engine power [KW]
 Mode :logical   Mode :logical   Mode :logical   Mode :logical               Mode :logical
 FALSE:53        FALSE:53        FALSE:53        FALSE:53                    FALSE:53

 Maximum torque [Nm] Type of brakes  Drive type      Battery capacity [kWh] Range (WLTP) [km]
 Mode :logical       Mode :logical   Mode :logical   Mode :logical          Mode :logical
 FALSE:53            FALSE:52        FALSE:53        FALSE:53               FALSE:53
                     TRUE :1
 Wheelbase [cm]  Length [cm]     Width [cm]      Height [cm]     Minimal empty weight [kg]
 Mode :logical   Mode :logical   Mode :logical   Mode :logical   Mode :logical
 FALSE:53        FALSE:53        FALSE:53        FALSE:53        FALSE:53

 Permissable gross weight [kg] Maximum load capacity [kg] Number of seats Number of doors
 Mode :logical                 Mode :logical              Mode :logical   Mode :logical
 FALSE:45                      FALSE:45                   FALSE:53        FALSE:53
 TRUE :8                       TRUE :8
 Tire size [in]  Maximum speed [kph] Boot capacity (VDA) [l] Acceleration 0-100 kph [s]
 Mode :logical   Mode :logical       Mode :logical           Mode :logical
 FALSE:53        FALSE:53            FALSE:52                FALSE:50
                                     TRUE :1                 TRUE :3
 Maximum DC charging power [kW] mean - Energy consumption [kWh/100 km]
 Mode :logical                  Mode :logical
 FALSE:53                       FALSE:44
                                TRUE :9
```



We have some empty rows in the dataset, specifically in the Average Consumption column, which is our target variable. We will simply remove these incomplete data to avoid biasing the prediction error during our predictive modeling. We could have applied some imputation technique, but for now, we will proceed without the incomplete data and see what results we can achieve.

```
# Eliminando os dados Na
  dadosv01 <- na.omit(dadosv00)

  summary(is.na(dadosv01))
  colSums(is.na(dadosv01))
  missmap(dadosv01)
```



Project developed by Thiago Bulgarelli

Contact: bugath36@gmail.com

```
# Renomeando as Colunas
  nomesCol <- c('Carro', 'Fabricante', 'Modelo', 'PrecoMin', 'Potencia',
                'TorqMax', 'Freios','Cambio', 'CapBat', 'Autonomia',
                'DistEixos', 'Comprimento', 'Largura', 'Altura', 'PesoVazio',
                'PesoCheio', 'CapMax', 'NumAssentos', 'NumeroPortas',
                'TamPneu','VelMax', 'BootCap', 'Acc', 'CapMaxBat',
                'ConsMedio')
  colnames(dadosv01) <- nomesCol
  View(dadosv01)
```

Carro	Fabricante	Modelo	PrecoMin	Potencia	TorqMax	Freios	Can
Audi e-tron 55 quattro	Audi	e-tron 55 quattro	345700	360	664	disc (front + rear)	4WI
Audi e-tron 50 quattro	Audi	e-tron 50 quattro	308400	313	540	disc (front + rear)	4WI
Audi e-tron S quattro	Audi	e-tron S quattro	414900	503	973	disc (front + rear)	4WI
Audi e-tron Sportback 50 quattro	Audi	e-tron Sportback 50 quattro	319700	313	540	disc (front + rear)	4WI
Audi e-tron Sportback 55 quattro	Audi	e-tron Sportback 55 quattro	357000	360	664	disc (front + rear)	4WI
Audi e-tron Sportback S quattro	Audi	e-tron Sportback S quattro	426200	503	973	disc (front + rear)	4WI
BMW i3	BMW	i3	169700	170	250	disc (front + rear)	2WI
BMW i3s	BMW	i3s	184200	184	270	disc (front + rear)	2WI

Performing Label Encoding of Categorical Variables

As we intend to create a regression model, we need to identify which categorical variables we will transform into numerical ones.

We have the following categorical variables: Manufacturer, Brakes, and Transmission. We will discard the Model and Car variables as they are not relevant to the analysis of the target variable, as instructed by the business area.

```
# Realizando Label Encoding das Variáveis Categóricas

  dadosv01$Fabricante <- as.numeric(as.factor(dadosv01$Fabricante))

  dadosv01$Freios <- as.numeric(as.factor(dadosv01$Freios))

  dadosv01$Cambio <- as.numeric(as.factor(dadosv01$Cambio))

  str(dadosv01)
```

Manufacturer Variable Dictionary

- Audi = 1
- BMW = 2
- Citroen = 3
- DS = 4
- Honda = 5
- Hyundai = 6
- Jaguar = 7
- Kia = 8
- Mazda = 9
- Mercedes-Benz = 10
- Mini = 11
- Nissan = 12
- Opel = 13
- Peugeot = 14
- Porshe = 15
- Renault = 16
- Skoda = 17
- Smart = 18
- Volkswagen = 19

Brake Type Variable Dictionary

- Four-wheel Disc Brakes = 1
- Front Disc Brakes and Rear Drum Brakes = 2

Drive Type Variable Dictionary

- 4WD = 3
- 2WD(rear) = 2
- 2WD(front) = 1

Saving Modified Dataset to Disk

```
file = 'dados/dados_ajustados.csv'
  save_df_to_file(dadosv01, file = file)
```

At this point, we are ready to begin our exploratory analysis and prepare our dataset for building the Machine Learning models.

Project developed by Thiago Bulgarelli

Contact: bugath36@gmail.com

Exploratory Analysis

```
# Pacotes

  library(kim)
  library(plotly)
  library(ggplot2)
  library(dplyr)
  library(corrplot)
  library(GGally)
  library(gridExtra)
```

```
# Verificando e ajustando os tipos de variáveis

  dados <- as.data.frame(read_csv(name = 'dados_ajustados', head = TRUE,
                                  dirname = 'dados' ))

  dados <- dados[, -1]
  dados$Fabricante <- as.factor(dados$Fabricante)
  dados$Freios <- as.factor(dados$Freios)
  dados$Cambio <- as.factor(dados$Cambio)

  str(dados)
  View(dados)
```

```
'data.frame':	42 obs. of  25 variables:
 $ Carro       : chr  "Audi e-tron 55 quattro" "Audi e-tron 50 quattro" "
ttro" "Audi e-tron Sportback 50 quattro" ...
 $ Fabricante  : Factor w/ 19 levels "1","2","3","4",..: 1 1 1 1 1 1 2 2
 $ Modelo      : chr  "e-tron 55 quattro" "e-tron 50 quattro" "e-tron S q
portback 50 quattro" ...
 $ PrecoMin    : int  345700 308400 414900 319700 357000 426200 169700 18
0 ...
 $ Potencia    : int  360 313 503 313 360 503 170 184 286 136 ...
 $ TorqMax     : int  664 540 973 540 664 973 250 270 400 260 ...
 $ Freios      : Factor w/ 2 levels "1","2": 1 1 1 1 1 1 1 1 1 1 ...
 $ Cambio      : Factor w/ 3 levels "1","2","3": 3 3 3 3 3 3 2 2 2 1 ...
 $ CapBat      : num  95 71 95 71 95 95 42.2 42.2 80 50 ...
 $ Autonomia   : int  438 340 364 346 447 369 359 345 460 320 ...
 $ DistEixos   : num  293 293 293 293 293 ...
 $ Comprimento : num  490 490 490 490 490 ...
 $ Largura     : num  194 194 198 194 194 ...
 $ Altura      : num  163 163 163 162 162 ...
 $ PesoVazio   : int  2565 2445 2695 2445 2595 2695 1440 1460 2260 1523 .
 $ PesoCheio   : int  3130 3040 3130 3040 3130 3130 1730 1730 2725 1975 .
 $ CapMax      : int  640 670 565 640 670 565 440 440 540 450 ...
 $ NumAssentos : int  5 5 5 5 5 5 4 4 5 5 ...
 $ NumeroPortas: int  5 5 5 5 5 5 5 5 5 5 ...
 $ TamPneu     : int  19 19 20 19 19 20 19 20 19 17 ...
 $ VelMax      : int  200 190 210 190 200 210 160 160 180 150 ...
 $ BootCap     : int  660 660 660 615 615 615 260 260 510 350 ...
 $ Acc         : num  5.7 6.8 4.5 6.8 5.7 4.5 8.1 6.9 6.8 8.7 ...
 $ CapMaxBat   : int  150 150 150 150 150 150 50 50 150 100 ...
 $ ConsMedio   : num  24.4 23.8 27.6 23.3 23.9 ...
```

We started our Exploratory Analysis by examining a general summary of the data. However, since we have many variables, it is more practical to analyze them separately: Numerical Variables, Categorical Variables, and the Target Variable.

```
# Análise Exploratória dos Dados

  summary(dados)

    # Podemos reparar que a Média e a Mediana possuem valores próximos em todas
    # as variáveis numéricas, o que nos indica uma possível distribuição normal.

  # Vamos separar as Variáveis em Numérias, Categoricas e Resposta para
  # analisarmos de forma separada.

    VarNum <- dados %>% select(c(4:6), c(9:24))
    VarCat <- dados %>% select(c(1:3), c(7:8))
    VarResp <- dados$ConsMedio
```

Analysis of the Target Variable

```
# Variável Resposta

    summary(VarResp) # Min. 13.10 / 1st Qu.15.60 / Median 16.88 / Mean 18.61
                     #  3rd Qu. 22.94 / Max. 27.55

    diff(range(VarResp)) # 14.45

    sd(VarResp) # 4.134293

    var(VarResp) # 17.09238


    hist(VarResp, main = 'Distribuição do Consumo Médio em Carros Elétricos',
       xlab = 'Consumo de Energia [kW/h]')
```

```
 Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
13.10   15.60   16.88   18.61   22.94   27.55
```

Right from the start, we notice that the median is lower than the mean of the data, indicating that our distribution is right-skewed (positively skewed). This means that the tail of the distribution extends towards higher values.

Project developed by Thiago Bulgarelli

Contact: bugath36@gmail.com

```
>     skewness(VarResp)
[1] 0.7857014
```

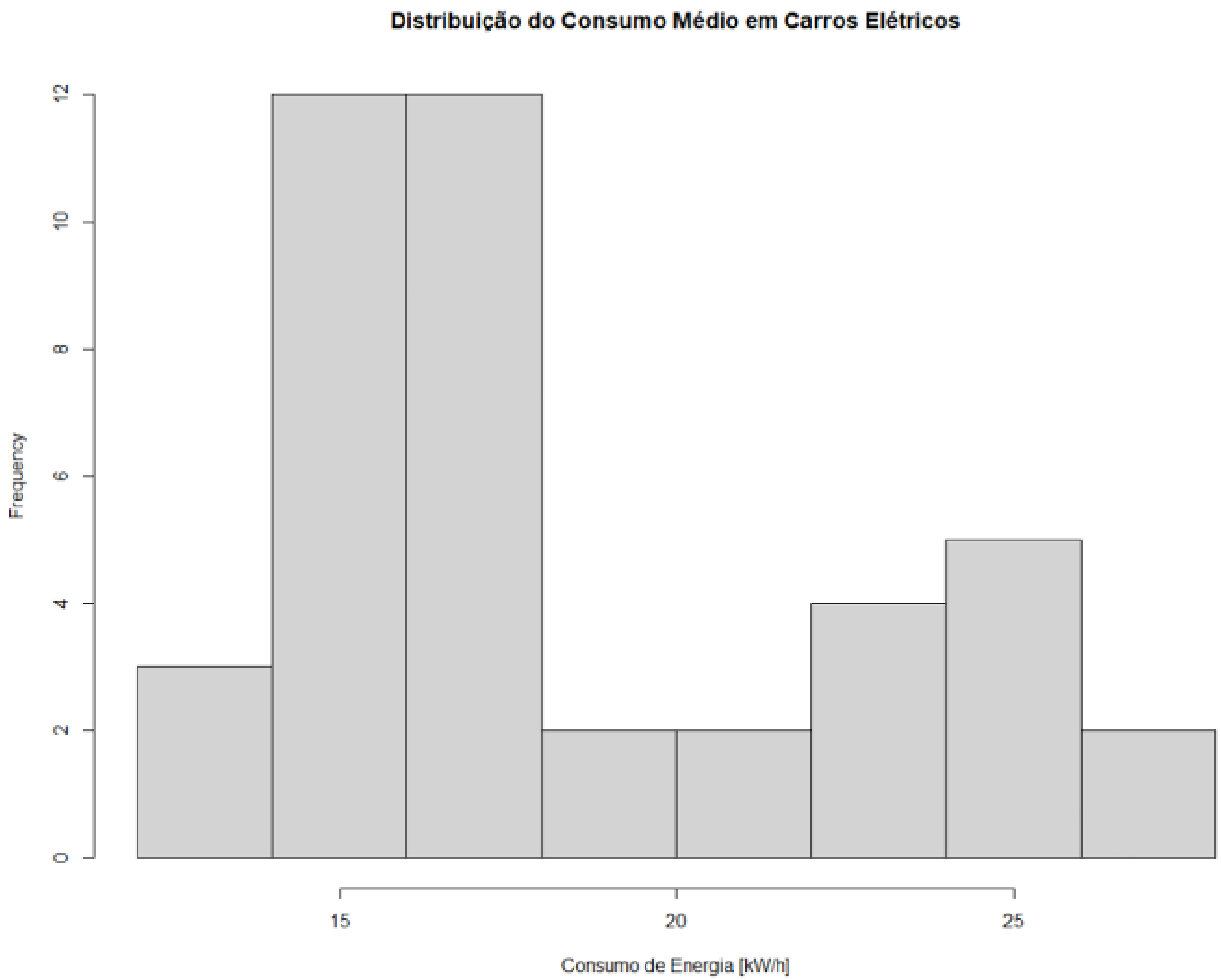
The maximum range distance is 14.45, as we can verify:

This indicates that the largest difference between the models does not exceed 15 kWh in consumption.

Our standard deviation is 4.13 kWh, and the variance is 17.09.

```
>     sd(VarResp) # 4.134293
[1] 4.134293
>
>     var(VarResp) # 17.09238
[1] 17.09238
```

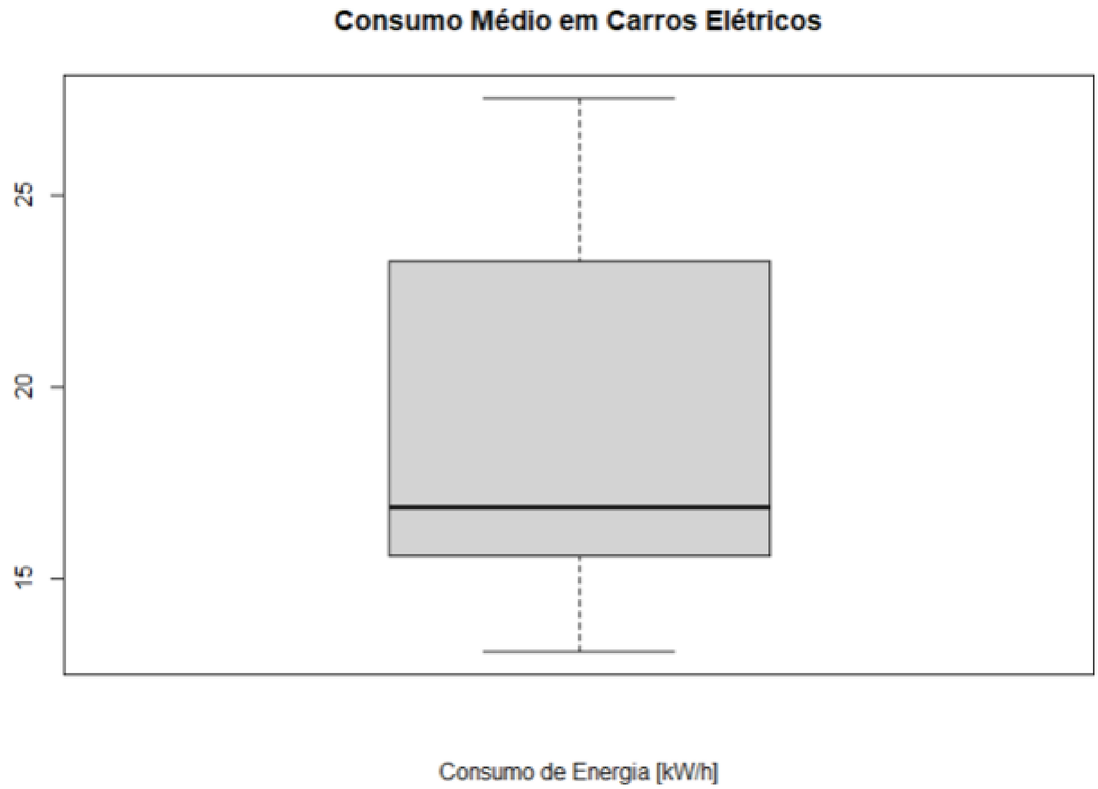
We can observe the issue of Skewness in the histogram below:



Project developed by Thiago Bulgarelli

Contact: bugath36@gmail.com

We have some outliers, indicated by values significantly higher than the median of the data. We can further observe this by constructing a boxplot.



We observe that we do not have any outliers, thus maintaining the data with a reasonable pattern for analysis.

We cannot claim that the data follows a normal distribution, but we can perform a Shapiro test to check for normality in the data distribution.

- H0 → We consider the data to be normally distributed.
- H1 → We cannot consider the data to be normally distributed.
  - If the p-value is greater than 0.05, we cannot reject H0.
  - If the p-value is less than 0.05, we reject H0.

```
shapiro.test(VarResp)
```

```
        Shapiro-Wilk normality test

data:  VarResp
W = 0.86663, p-value = 0.0001665
```

We can observe that the p-value is less than 0.05, indicating that we cannot consider the data of the target variable to follow a normal distribution. Therefore, we reject H0.

Analysis of Numerical Variables

Initially, let's examine a summary of each variable:

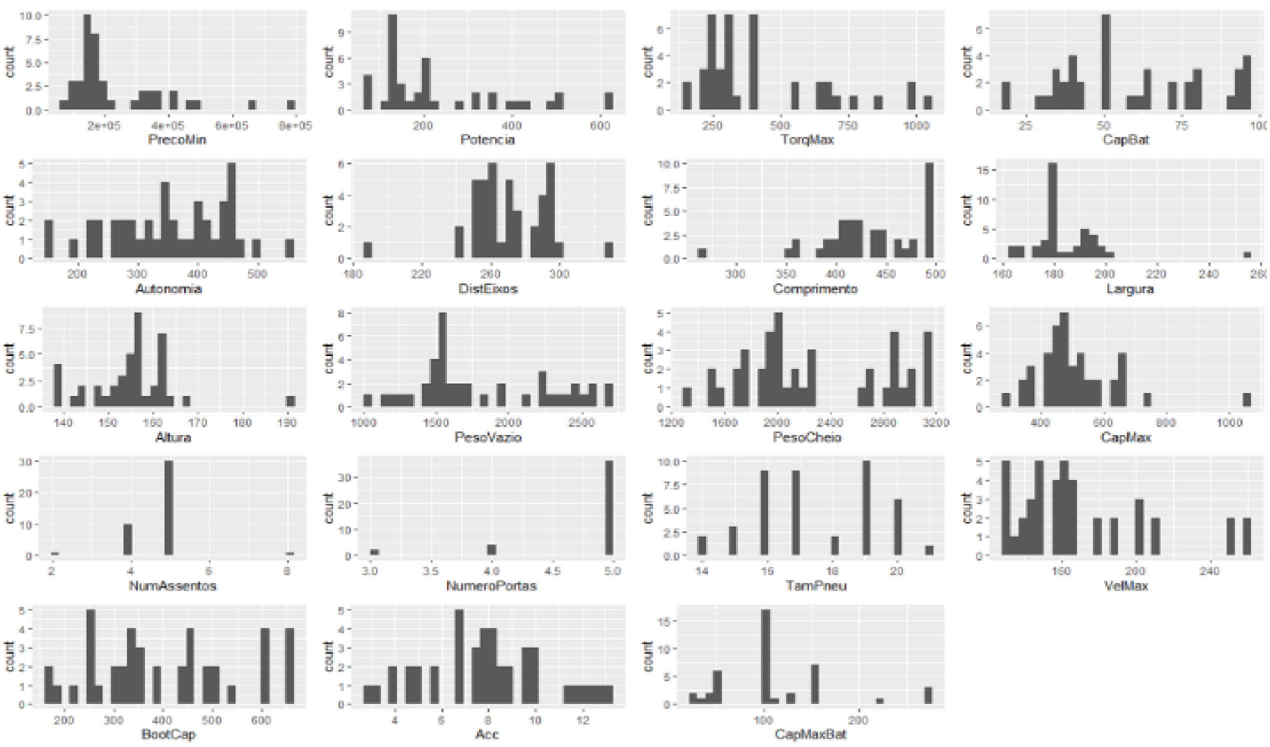
```
>  summary(VarNum)
    PrecoMin         Potencia         TorqMax           CapBat          Autonomia
 Min.   : 82050   Min.    : 82.0   Min.    : 160.0   Min.    :17.60   Min.    :148.0
 1st Qu.:140650   1st Qu.:136.0   1st Qu.: 260.0   1st Qu.:39.20   1st Qu.:279.2
 Median :166945   Median :184.0   Median : 317.5   Median :52.00   Median :352.5
 Mean   :235066   Mean   :237.7   Mean   : 425.2   Mean   :58.84   Mean   :351.7
 3rd Qu.:316875   3rd Qu.:313.0   3rd Qu.: 540.0   3rd Qu.:78.65   3rd Qu.:434.8
 Max.   :794000   Max.   :625.0   Max.   :1050.0   Max.   :95.00   Max.   :549.0
   DistEixos      Comprimento        Largura          Altura         PesoVazio      PesoCheio
 Min.   :187.3   Min.   :269.5   Min.   :164.5   Min.   :137.8   Min.   :1035   Min.   :1310
 1st Qu.:256.3   1st Qu.:406.6   1st Qu.:178.7   1st Qu.:151.2   1st Qu.:1516   1st Qu.:1882
 Median :270.0   Median :431.8   Median :180.2   Median :156.0   Median :1622   Median :2100
 Mean   :269.8   Mean   :433.5   Mean   :184.8   Mean   :155.0   Mean   :1821   Mean   :2268
 3rd Qu.:290.0   3rd Qu.:475.5   3rd Qu.:193.5   3rd Qu.:160.5   3rd Qu.:2249   3rd Qu.:2855
 Max.   :327.5   Max.   :496.3   Max.   :255.8   Max.   :190.0   Max.   :2695   Max.   :3130
     CapMax         NumAssentos      NumeroPortas      TamPneu          VelMax
 Min.   : 290.0   Min.   :2.000   Min.   :3.00   Min.   :14.00   Min.   :130.0
 1st Qu.: 440.0   1st Qu.:4.250   1st Qu.:5.00   1st Qu.:16.00   1st Qu.:146.2
 Median : 485.5   Median :5.000   Median :5.00   Median :17.00   Median :160.0
 Mean   : 510.5   Mean   :4.762   Mean   :4.81   Mean   :17.55   Mean   :169.5
 3rd Qu.: 565.0   3rd Qu.:5.000   3rd Qu.:5.00   3rd Qu.:19.00   3rd Qu.:187.5
 Max.   :1056.0   Max.   :8.000   Max.   :5.00   Max.   :21.00   Max.   :260.0
    BootCap           Acc             CapMaxBat
 Min.   :171.0   Min.   : 2.800   Min.   : 22.0
 1st Qu.:310.2   1st Qu.: 6.800   1st Qu.: 62.5
 Median :371.0   Median : 7.900   Median :100.0
 Mean   :404.3   Mean   : 7.893   Mean   :109.7
 3rd Qu.:497.0   3rd Qu.: 9.650   3rd Qu.:143.8
 Max.   :660.0   Max.   :13.100   Max.   :270.0
```

Let's visually analyze a multi-histogram:

Project developed by Thiago Bulgarelli

Contact: bugath36@gmail.com

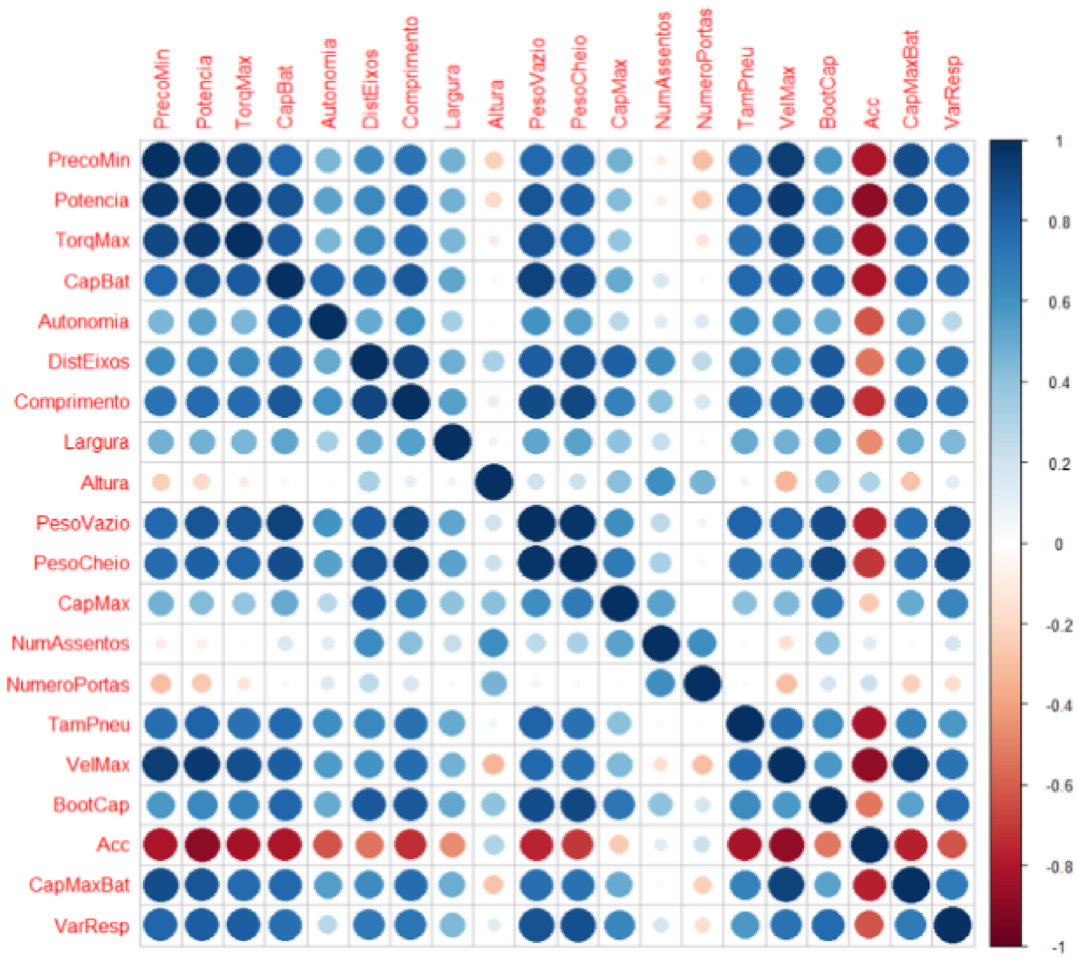
```
# Multi-Hitogramas de cada Variável

     VarNum01 <- ggplot(VarNum) + geom_histogram(aes(x = PrecoMin))
     VarNum02 <- ggplot(VarNum) + geom_histogram(aes(x = Potencia))
     VarNum03 <- ggplot(VarNum) + geom_histogram(aes(x = TorqMax))
     VarNum04 <- ggplot(VarNum) + geom_histogram(aes(x = CapBat))
     VarNum05 <- ggplot(VarNum) + geom_histogram(aes(x = Autonomia))
     VarNum06 <- ggplot(VarNum) + geom_histogram(aes(x = DistEixos))
     VarNum07 <- ggplot(VarNum) + geom_histogram(aes(x = Comprimento))
     VarNum08 <- ggplot(VarNum) + geom_histogram(aes(x = Largura))
     VarNum09 <- ggplot(VarNum) + geom_histogram(aes(x = Altura))
     VarNum10 <- ggplot(VarNum) + geom_histogram(aes(x = PesoVazio))
     VarNum11 <- ggplot(VarNum) + geom_histogram(aes(x = PesoCheio))
     VarNum12 <- ggplot(VarNum) + geom_histogram(aes(x = CapMax))
     VarNum13 <- ggplot(VarNum) + geom_histogram(aes(x = NumAssentos))
     VarNum14 <- ggplot(VarNum) + geom_histogram(aes(x = NumeroPortas))
     VarNum15 <- ggplot(VarNum) + geom_histogram(aes(x = TamPneu))
     VarNum16 <- ggplot(VarNum) + geom_histogram(aes(x = VelMax))
     VarNum17 <- ggplot(VarNum) + geom_histogram(aes(x = BootCap))
     VarNum18 <- ggplot(VarNum) + geom_histogram(aes(x = Acc))
     VarNum19 <- ggplot(VarNum) + geom_histogram(aes(x = CapMaxBat))

     grid.arrange(VarNum01, VarNum02, VarNum03, VarNum04, VarNum05,
                  VarNum06, VarNum07, VarNum08, VarNum09, VarNum10,
                  VarNum11, VarNum12, VarNum13, VarNum14, VarNum15,
                  VarNum16, VarNum17, VarNum18,VarNum19)
```



We notice that in our dataset, we have cars from different categories, despite having the same propulsion fuel. Therefore, it is a mixed dataset that does not condition the category of the vehicles. It might be interesting to obtain more data on vehicles and categorize them (label them) so that we can analyze them separately. We can support this statement when we look at extreme data points in Maximum Torque, Engine Power, Minimal Price, Battery Capacity, and Maximum Speed.

Another important point for our analysis is to identify how the variables relate to each other and to the target variable. We will perform a correlation analysis to identify issues such as multicollinearity.

```
# Análise de Multicolinearidade

      corrplot(cor(cbind(VarNum, VarResp)))
```



We can identify an excessive correlation between several variables, which can potentially cause confusion during our predictive modeling. Correlations such as Weight/Power and Wheelbase/MaxLoad are relatively obvious from a business standpoint. However, when observing our target variable, we can see high correlations with variables such as Price, Power, Max Torque, Battery Capacity, Empty Weight, Gross Weight, Boot Capacity, and Acceleration. Nonetheless, we will use a linear regression algorithm to help us better select our variables.

To analyze the correlation between variables more effectively, let's create a multivariate plot.

```
# Scatterplots de Cada Variável Numérica Dependente e a Variável Resposta

      DadosIntResp <-dados %>% select(c(4:6), c(9:25))
      Multiplot <- function(DadosIntResp, mapping, method = "loess", ...){
        p <- ggplot(data = data, mapping = mapping) +
          geom_point()+
          geom_smooth(method = method, ...)
        p
      }
      ggpairs(DadosIntResp, lower = list(continous = Multiplot))
```

Analysis of Categorical Variables

Let's ignore the manufacturer for now, as it doesn't seem to be relevant since we have 19 manufacturers for 43 observations, which does not provide us with a reasonable sample size to analyze from that perspective.

We will begin by analyzing the proportions of data for each factor in the Brake and Transmission variables.

```
prop.table(table(VarCat$Freios))*100
prop.table(table(VarCat$Cambio))*100
```

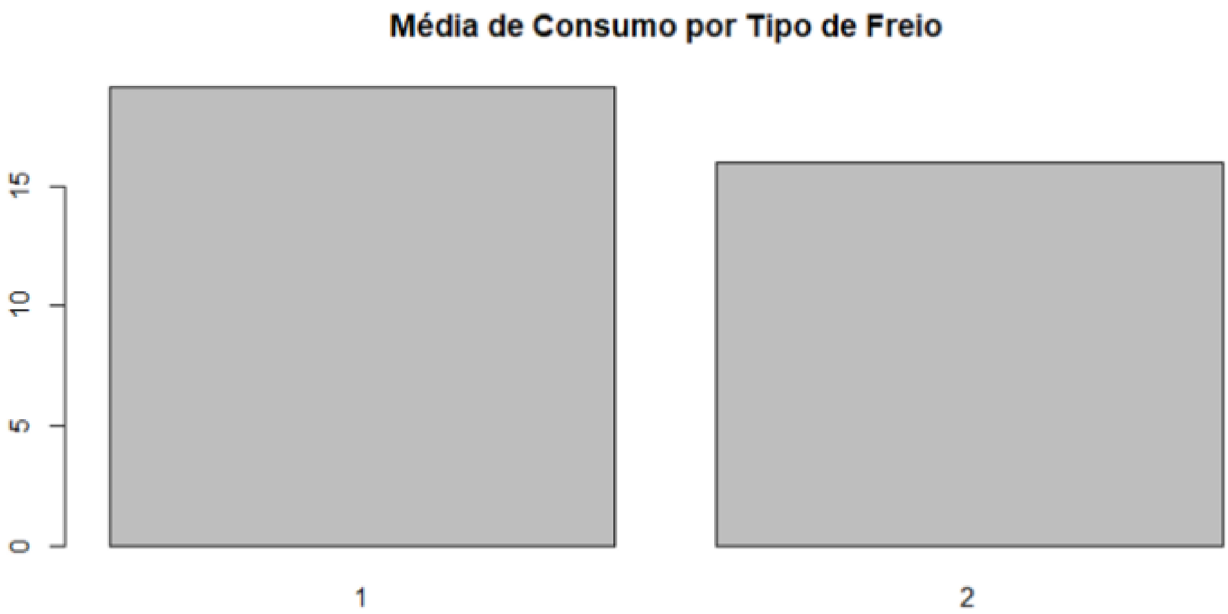
```
> prop.table(table(VarCat$Freios))*100

       1        2
83.33333 16.66667
>     prop.table(table(VarCat$Cambio))*100

       1        2        3
47.61905 23.80952 28.57143
```

We can see that we have a lot more data for cars with four-wheel disc brakes than for cars with only front disc brakes.

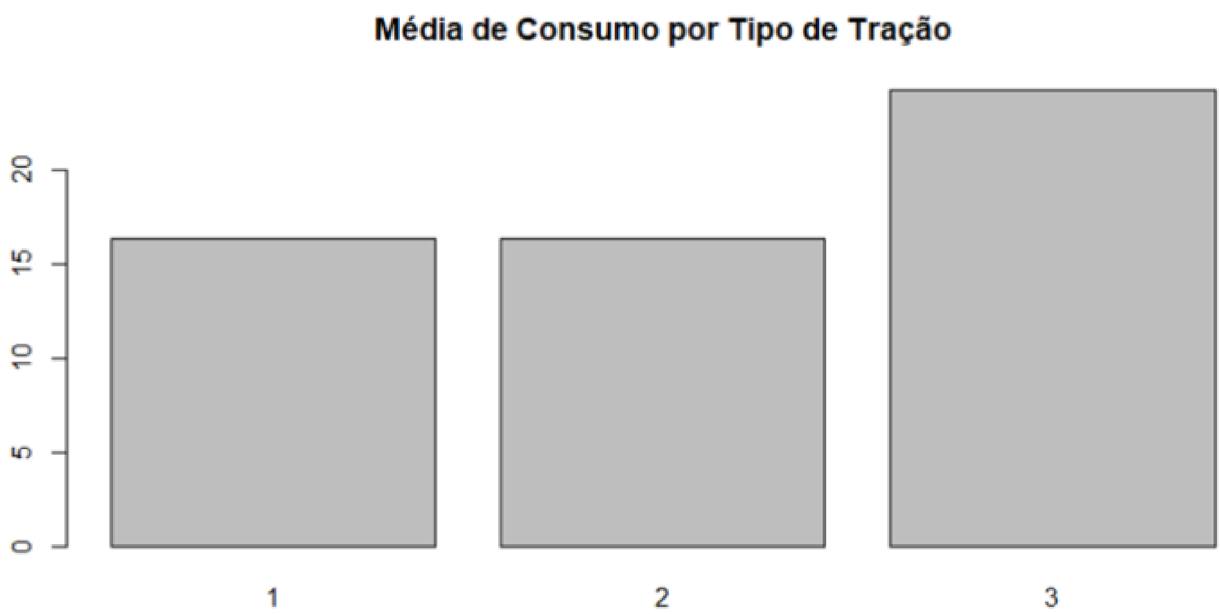
We can also observe a certain balance in the transmission variable, which is better for our predictive model.

Let's analyze the energy consumption for each variable type by calculating the average for each category.

```
media_por_tipo_de_freio = dados %>% group_by(Freios) %>%
      summarise(avg_consumo = mean(ConsMedio))

    media_por_tipo_de_tracao = dados %>% group_by(Cambio) %>%
      summarise(avg_consumo = mean(ConsMedio))

    barplot(media_por_tipo_de_freio$avg_consumo,
            names.arg = media_por_tipo_de_freio$Freios,
            main = 'Média de Consumo por Tipo de Freio')

    barplot(media_por_tipo_de_tracao$avg_consumo,
            names.arg = media_por_tipo_de_tracao$Cambio,
            main = 'Média de Consumo por Tipo de Tração')
```



We can clearly see that cars with front and rear disc brakes have higher energy consumption per hour. Here, we are not observing causality, but simply noting the fact that cars with this characteristic, on average, consume more electrical energy.



In this variable, we can observe that cars with 4-wheel drive have higher energy consumption. This is directly related to the nature of the 4WD transmission, as it requires energy to be delivered to all four wheels to provide traction, resulting in a higher energy usage compared to just two wheels.

However, an interesting finding is that the average consumption is very similar between cars with front-wheel drive and rear-wheel drive.

We have concluded our Data Exploration phase and are now ready to proceed with the Preparation stage for building the predictive model and addressing the defined business problem.

Data Preprocessing

In this stage, we will apply Feature Selection and determine which variables we will consider for the construction of our Base Model. The rule here is to find a good model efficiency, keeping it as simple as possible.

```
# Carregando Pacotes
  #
    library(dplyr)
    library(kim)
    library(caret)

  # Carregando o Dataset

    dados <- as.data.frame(read_csv(name = 'dados_ajustados', head = TRUE,
                                    dirname = 'dados' ))
    dim(dados)
    dados <- dados[, 5:26]
    dim(dados)
    str(dados)
    dados$Freios <- as.factor(dados$Freios)
    dados$Cambio <- as.factor(dados$Cambio)
    dados$NumAssentos <- as.factor(dados$NumAssentos)
    dados$NumeroPortas <- as.factor(dados$NumeroPortas)
    str(dados)
    View(dados)FeaturesLM <- lm(ConsMedio ~ .,
                   data = dados)
    summary(FeaturesLM)
```

Feature Selection

We will use a simple linear model with all variables to determine which variables are most significant in explaining our target variable.

```
FeaturesLM <- lm(ConsMedio ~ .,
                 data = dados)
summary(FeaturesLM)
```

```
Call:
lm(formula = ConsMedio ~ ., data = dados)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.2542 -0.3364  0.0000  0.3086  1.1067 

Coefficients:
                Estimate Std. Error t value Pr(>|t|)   
(Intercept)    6.217e+01  1.628e+01   3.820  0.00151 **
PrecoMin       4.431e-06  6.499e-06   0.682  0.50506   
Potencia       6.931e-03  1.138e-02   0.609  0.55091   
TorqMax       -1.455e-03  4.577e-03  -0.318  0.75469   
Freios2       -2.390e-02  1.318e+00  -0.018  0.98576   
Cambio2        3.808e+00  1.278e+00   2.981  0.00883 **
Cambio3        5.277e+00  2.220e+00   2.378  0.03024 * 
CapBat         8.619e-02  6.756e-02   1.276  0.22028   
Autonomia     -1.652e-02  7.108e-03  -2.324  0.03364 * 
DistEixos     -2.622e-01  1.198e-01  -2.188  0.04384 * 
Comprimento    2.159e-02  4.208e-02   0.513  0.61499   
Largura       -1.965e-02  1.428e-02  -1.377  0.18762   
Altura         4.024e-02  7.665e-02   0.525  0.60674   
PesoVazio      2.635e-03  7.014e-03   0.376  0.71209   
PesoCheio     -2.122e-03  3.314e-03  -0.640  0.53114   
CapMax        -3.181e-03  7.710e-03  -0.413  0.68541   
NumAssentos4   1.823e+01  7.864e+00   2.318  0.03402 * 
NumAssentos5   2.157e+01  9.505e+00   2.269  0.03748 * 
NumAssentos8   4.114e+01  1.861e+01   2.211  0.04196 * 
NumeroPortas4  1.676e+00  2.917e+00   0.575  0.57353   
NumeroPortas5 -4.450e+00  2.216e+00  -2.008  0.06180 . 
TamPneu       -7.817e-01  2.807e-01  -2.785  0.01325 * 
VelMax         1.474e-02  7.138e-02   0.207  0.83896   
BootCap        2.489e-02  8.726e-03   2.852  0.01153 * 
Acc           -2.294e-01  3.102e-01  -0.739  0.47040   
CapMaxBat     -8.928e-03  1.731e-02  -0.516  0.61304   
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.8818 on 16 degrees of freedom
Multiple R-squared:  0.9822,	Adjusted R-squared:  0.9545 
F-statistic: 35.41 on 25 and 16 DF,  p-value: 8.055e-10
```

We can see that Range, Transmission, Wheelbase, Number of Seats, Tire Size, and Boot Capacity are statistically significant for our model. The metrics show that with an Adjusted R^2 of 95.45%, we can conclude that our model can explain 95% of the variability in the mean consumption based on the listed variables.

However, this information may indicate that our model is overfitting, meaning it is overly trained and not very generalizable. Nevertheless, the purpose of this model was solely to identify the most important variables, which it served perfectly.

An important observation we made is that the data is on different scales, which could affect the outcome of the model's identification of the most important variables. Therefore, we redid the model with normalized data. We used a MinMax function to scale all the variables to the same range.

```
# Aplicando Padronização ao Dataset

      # Criando um função de Padronizar

        padronizar <- function(x) {
          return ((x - min(x)) / (max(x) - min(x)))
        }

      # Primeiramente vamos dividir nosso modelo em dados de treino e teste.

        separador <- createDataPartition(y = dados$ConsMedio, p = 0.75,
                                         list = FALSE)
        dados_treino <- dados[separador,]
        dados_teste <- dados[-separador,]

        str(dados_teste)
        str(dados_treino)
```

```
      # Aplicando a Função aos Dados Treino Numéricos


        dados_testeNumP <- dados_teste[,!unlist(lapply(dados_teste,
                                                       is.factor))]
        str(dados_testeNumP)

        dados_testeFac <- dados_teste[, unlist(lapply(dados_teste
                                                      , is.factor))]
        str(dados_testeFac)

        dados_treinoNumP <- dados_treino[,!unlist(lapply(dados_treino,
                                                         is.factor))]
        str(dados_treinoNumP)

        dados_treinoFac <- dados_treino[, unlist(lapply(dados_treino
                                                        , is.factor))]
        str(dados_treinoFac)
```

Project developed by Thiago Bulgarelli

Contact: bugath36@gmail.com

```
        dados_testeNumP <- as.data.frame(lapply(dados_testeNumP[,-18]
                                        , padronizar))
        dados_testeNumP <- cbind(dados_testeNumP, dados_teste$ConsMedio)
        colnames(dados_testeNumP)[18] <- 'ConsMedio'
        str(dados_testeNumP)

        dados_treinoNumP <- as.data.frame(lapply(dados_treinoNumP[, -18],
                                        padronizar))
        dados_treinoNumP <- cbind(dados_treinoNumP, dados_treino$ConsMedio)
        colnames(dados_treinoNumP)[18] <- 'ConsMedio'
        str(dados_treinoNumP)

      # Unificando os Data Frames

        dados_treino_final <- cbind(dados_treinoNumP, dados_treinoFac)
        dados_teste_final <- cbind(dados_testeNumP, dados_testeFac)

        str(dados_treino_final)
        str(dados_teste_final)
```

```
# Reavaliando as Variáveis de Maior Significância

     dados_final <- rbind(dados_treino_final, dados_teste_final)
      FeaturesLMS <- lm(ConsMedio ~ ., data = dados_final)
      summary(FeaturesLMS)
```

```
Call:
lm(formula = ConsMedio ~ ., data = dados_final)

Residuals:
     Min       1Q   Median       3Q      Max
-0.95879 -0.33334 -0.01356  0.34427  1.05642

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    16.6138     2.2395   7.419 1.46e-06 ***
PrecoMin        1.2369     4.2953   0.288 0.777061
Potencia        8.0887     5.1687   1.565 0.137158
TorqMax        -0.3851     2.7555  -0.140 0.890606
CapBat         10.4525     3.6385   2.873 0.011048 *
Autonomia      -7.5258     2.1371  -3.522 0.002832 **
DistEixos     -12.6267     4.5659  -2.765 0.013789 *
Comprimento     4.8813     4.8253   1.012 0.326780
Largura        -0.5382     1.0920  -0.493 0.628791
Altura          2.6975     2.7209   0.991 0.336245
PesoVazio       3.7471     8.2349   0.455 0.655199
PesoCheio      -4.0409     5.0726  -0.797 0.437338
CapMax          3.9142     1.7945   2.181 0.044432 *
```

```
TamPneu        -7.2467     1.7481  -4.145 0.000761 ***
VelMax        -10.6552     5.2359  -2.035 0.058761 .
BootCap         5.1904     2.7796   1.867 0.080280 .
Acc            -2.4776     2.4640  -1.006 0.329618
CapMaxBat       2.7448     3.0502   0.900 0.381527
Freios2        -1.2319     1.0308  -1.195 0.249484
Cambio2         3.1542     0.8691   3.629 0.002255 **
Cambio3         3.8343     1.6382   2.341 0.032540 *
NumAssentos4    5.0579     2.2070   2.292 0.035825 *
NumAssentos5    5.1798     2.6637   1.945 0.069624 .
NumAssentos8    9.4808     4.2587   2.226 0.040716 *
NumeroPortas4   1.2353     2.2536   0.548 0.591154
NumeroPortas5  -2.3233     1.3482  -1.723 0.104113
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.8516 on 16 degrees of freedom
Multiple R-squared:  0.9834,    Adjusted R-squared:  0.9576
F-statistic: 38.02 on 25 and 16 DF,  p-value: 4.666e-10
```

So we have a similar adjusted R^2 value, but with different variables. Therefore, we will consider the following variables: Battery Capacity, Range, Wheelbase, Maximum Load Capacity, Transmission, and Number of Seats.

```
dados_prep_teste <- dados_teste_final %>% select(CapBat, Autonomia, DistEixos,
                                                  CapMax, Cambio, NumAssentos,
                                                  ConsMedio)
    str(dados_prep_teste)

    dados_prep_treino <- dados_treino_final %>% select(CapBat, Autonomia, DistEixos,
                                                   CapMax, Cambio, NumAssentos,
                                                   ConsMedio)
    str(dados_prep_treino)


    write.csv2(dados_prep_treino,file = 'dados/dados_prep_treino.csv')

    write.csv2(dados_prep_teste,file = 'dados/dados_prep_teste.csv')
```

Predictive Modeling

```
# Parte 04 - Modelagem Preditiva

  # Neste etapa, vamos criar modelos de machine learning e analisar as métricas
  # de cada modelo e definir qual realiza previsões com mais eficiência,
  # utilizando o dataset preparado e levando em consideração nosso modelo
  # preditivo base.
```

```
# Carregando Pacotes

  library(dplyr)
  library(caret)
  library(ggplot2)
  library(forecast)

# Carregando o Dataset

  dados_treino <- read.csv2('dados/dados_prep_treino.csv', header = TRUE)[-1]
  str(dados_treino)
  dados_teste <- read.csv2('dados/dados_prep_teste.csv', header = TRUE)[-1]
  str(dados_teste)

  summary(is.na(dados_treino))
  summary(is.na(dados_teste))
```

Building the Base Model and Analyzing the Result

Base Model

Let's construct the base model using the lm() function from the base utils package in the R language. This function allows us to fit a linear regression model to our data.

```
# Construindo nosso Modelo Base

      # Vamos utilizar a função lm() para construir nosso modelo base, sendo o
      # algoritmo mais simples que conhecemos.

      ModeloBase <- lm(ConsMedio ~ ., dados_treino)
      summary(ModeloBase)

        # Nosso modelo Base possui em treinamento, um R² de 92,67% ajustado, o
        # que siginifica que conseguimos explicar o consumo com 92,67% de
        # variabilidade dessas variáveis.

      # Acurácia do Modelo Base de Treino
        prev_treino_ModeloB <- predict(ModeloBase, dados_treino[-8])
        accuracy(prev_treino_ModeloB, dados_treino$ConsMedio)

          # Acurácia de RMSE 1.407
```

```
Call:
lm(formula = ConsMedio ~ ., data = dados_treino)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.3493 -0.7016  0.0883  0.5107  2.4363 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  14.0363     1.9469   7.209 1.17e-07 ***
CapBat       17.6938     2.1127   8.375 7.41e-09 ***
Autonomia   -14.1841     1.6552  -8.570 4.76e-09 ***
DistEixos    -3.8796     4.1306  -0.939  0.35626    
CapMax        6.8222     2.0469   3.333  0.00259 ** 
Cambio        0.7917     0.4980   1.590  0.12399    
NumAssentos   0.2588     0.5429   0.477  0.63760    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.157 on 26 degrees of freedom
Multiple R-squared:  0.9404,	Adjusted R-squared:  0.9267 
F-statistic: 68.41 on 6 and 26 DF,  p-value: 1.114e-14
```

Our base model, in training, has an adjusted R^2 of 92.67%, indicating that we can explain 92% of the variability in average consumption using the independent variables. This is a good value for the metric, as well as the accuracy of the base training model with an RMSE of 1.027.

```
>        accuracy(prev_treino_ModeloB, dados_treino$ConsMedio)
                   ME     RMSE       MAE        MPE     MAPE
Test set 2.691476e-15 1.027163 0.7549713 -0.3647261 4.476753
```

Let's make predictions on the test data and analyze the residuals.

```
# Testando e Avaliando o Modelo

      Previsao01 <- predict(ModeloBase, dados_teste[-8])
      Previsao01

      # Analisando a Acurárcia do Teste para o Modelo Base

        accuracy(Previsao01, dados_teste$ConsMedio)

          # Obtivemos uma acurácia relativamente mais alta que em treino
          # no valor de RMSE 2.907. !! Ponto de Atenção
```

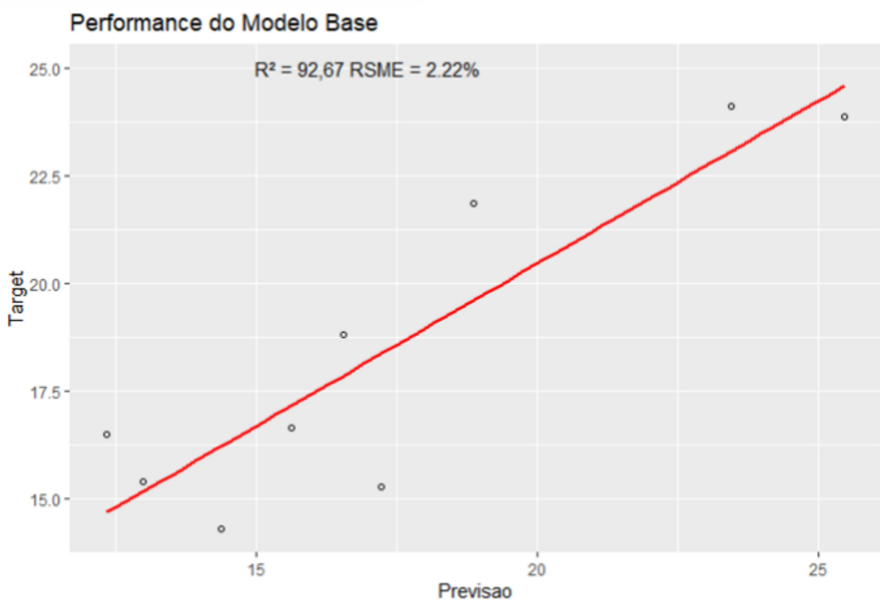
Project developed by Thiago Bulgarelli

Contact: bugath36@gmail.com

```
# Analisando o resíduo do modelo

  ConsumoTeste <- dados_teste$ConsMedio

  Res_ModeloBase <- ConsumoTeste - Previsao01

  FitModeloBase <- data.frame(Target = ConsumoTeste,
                              Previsao = Previsao01,
                              Residuo = Res_ModeloBase)
  head(FitModeloBase)
  summary(FitModeloBase)

# Scatter Plot Comparativo

  camada1 <- geom_point(shape = 1)

  camada2 <- geom_smooth(method = lm, color = 'red', se = FALSE)

  ggplot(FitModeloBase, aes(x = Previsao, y = Target)) + camada1 + camada2 +
    ggtitle('Performance do Modelo Base') + annotate(geom = 'text', x = 17,
                                                     y = 25,
                                                     label = 'R² = 92,67 RSME = 2,22%')
```

We can see that the residuals have a mean close to zero and a small range between the minimum and maximum values. Let's visualize the regression graphically.

As a result, we have excellent accuracy and few errors compared to the measured results in the test set.



```
> head(FitModeloBase)
  Target Previsao     Residuo
1  23.85 25.46181 -1.61180984
2  14.30 14.36639 -0.06639241
3  18.80 16.56364  2.23636045
4  15.40 12.98300  2.41699755
5  15.30 17.23469 -1.93468759
6  21.85 18.87666  2.97333844
```

```
> summary(FitModeloBase)
     Target          Previsao         Residuo
 Min.   :14.30   Min.   :12.35   Min.   :-1.93469
 1st Qu.:15.40   1st Qu.:14.37   1st Qu.:-0.06639
 Median :16.65   Median :16.56   Median : 1.01638
 Mean   :18.53   Mean   :17.43   Mean   : 1.09301
 3rd Qu.:21.85   3rd Qu.:18.88   3rd Qu.: 2.41700
 Max.   :24.10   Max.   :25.46   Max.   : 4.15171
```

Model V02

For model V02, we will change the algorithm but still use the regression method. We will use the Caret package with the 'lm' method.

```
# Construindo Modelo Versão 02

# Para este modelo, utilizaremo o pacote Caret com o método de Regressão
# Linear, sem Trainig Control e Tuning.

  ModeloV02 <- train(ConsMedio ~ ., data = dados_treino, method = 'lm')
  summary(ModeloV02)

# Podemos reparar que obtivemos um R² de 92,67%, mesmo valor do modelo base.
# Vamos realizar a Previsão de teste e avaliar
# graficamente.

  Previsao02 <- predict(ModeloV02, dados_teste[-8])
  Previsao02

  accuracy(Previsao02, dados_teste$ConsMedio)

    # Mesma acurácia do modelo base RMSE 2.907

  # Analisando o resíduo do modelo

  Res_ModeloV02 <- ConsumoTeste - Previsao02

  FitModeloV02 <- data.frame(Target = ConsumoTeste,
                             Previsao = Previsao02,
                             Residuo = Res_ModeloV02)
  head(FitModeloV02)
  summary(FitModeloV02)
```

```r
# Scatter Plot Comparativo

camada1 <- geom_point(shape = 1)

camada2 <- geom_smooth(method = lm, color = 'red', se = FALSE)

ggplot(FitModeloV02, aes(x = Previsao, y = Target)) + camada1 + camada2 +
  ggtitle('Performance do Modelo V02') + annotate(geom = 'text', x = 17,
                                                  y = 25,
                                                  label = 'R² = 92,67 RSME = 2.22%')
```

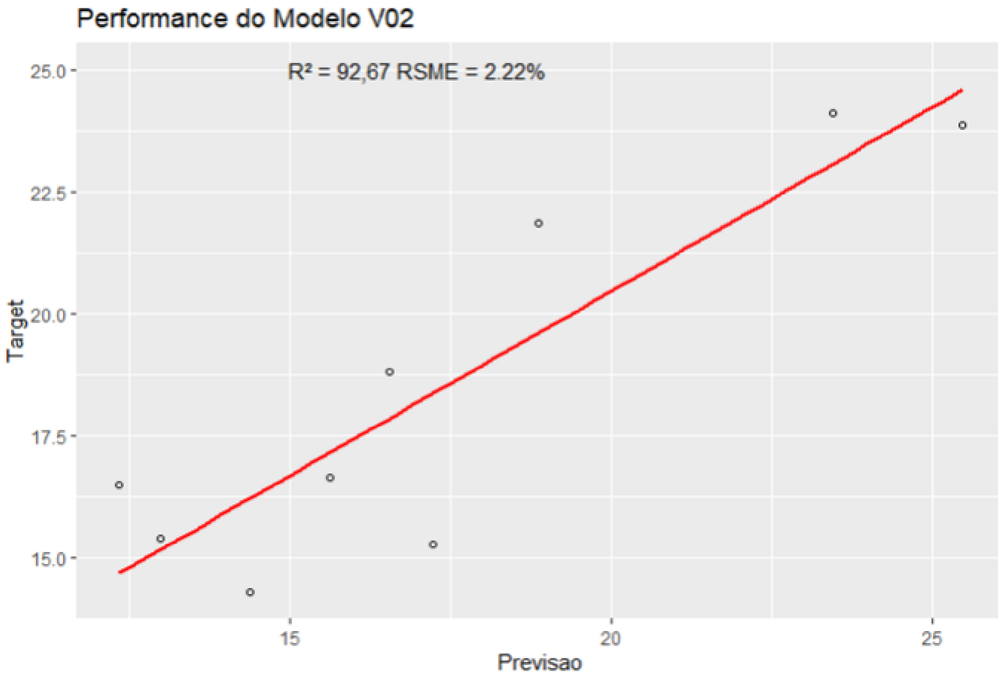
```
Call:
lm(formula = .outcome ~ ., data = dat)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.3493 -0.7016  0.0883  0.5107  2.4363 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  14.0363     1.9469   7.209 1.17e-07 ***
CapBat       17.6938     2.1127   8.375 7.41e-09 ***
Autonomia   -14.1841     1.6552  -8.570 4.76e-09 ***
DistEixos    -3.8796     4.1306  -0.939  0.35626    
CapMax        6.8222     2.0469   3.333  0.00259 ** 
Cambio        0.7917     0.4980   1.590  0.12399    
NumAssentos   0.2588     0.5429   0.477  0.63760    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.157 on 26 degrees of freedom
Multiple R-squared:  0.9404,	Adjusted R-squared:  0.9267 
F-statistic: 68.41 on 6 and 26 DF,  p-value: 1.114e-14
```

Notice that by simply changing the package, we didn't observe any difference in the results.

```
>     accuracy(Previsao02, dados_teste$ConsMedio)
               ME     RMSE      MAE      MPE     MAPE
Test set 1.093011 2.229308 1.895876 6.146204 10.56118
```

```
>     head(FitModeloV02)
  Target Previsao     Residuo
1  23.85 25.46181 -1.61180984
2  14.30 14.36639 -0.06639241
3  18.80 16.56364  2.23636045
4  15.40 12.98300  2.41699755
5  15.30 17.23469 -1.93468759
6  21.85 18.87666  2.97333844
```

```
     Target         Previsao         Residuo       
 Min.   :14.30   Min.   :12.35   Min.   :-1.93469  
 1st Qu.:15.40   1st Qu.:14.37   1st Qu.:-0.06639  
 Median :16.65   Median :16.56   Median : 1.01638  
 Mean   :18.53   Mean   :17.43   Mean   : 1.09301  
 3rd Qu.:21.85   3rd Qu.:18.88   3rd Qu.: 2.41700  
 Max.   :24.10   Max.   :25.46   Max.   : 4.15171  
```



Model V03

For this model, we will change the linear regression method to Boosted Linear Regression.

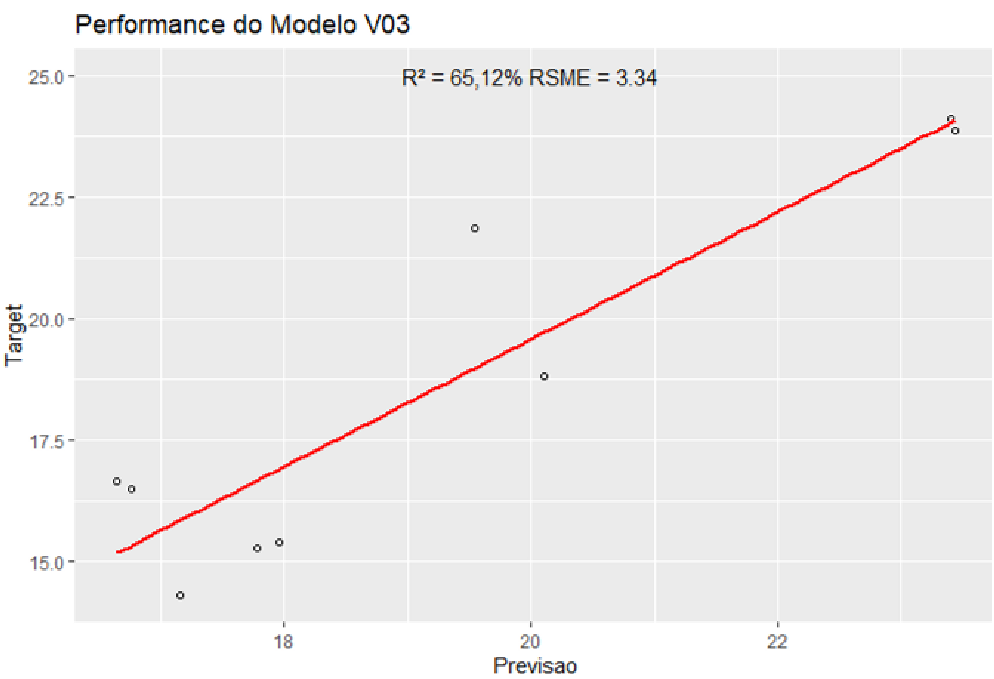
```
 # Construindo Modelo V03

# Para este modelo vamos alterar o método de Regressão Linear para Boosted
# Linear Regression e analisar os resultados.
# Utilizaremos o mesmo pacote Caret.


  ModeloV03 <- train(ConsMedio ~ ., data = dados_treino, method = 'BstLm')
  ModeloV03

  Previsao03 <- predict(ModeloV03, dados_teste[-8])
```

```
# Analisando o resíduo do modelo

Res_ModeloV03 <- ConsumoTeste - Previsao03

FitModeloV03 <- data.frame(Target = ConsumoTeste,
                           Previsao = Previsao03,
                           Residuo = Res_ModeloV03)
head(FitModeloV03)
summary(FitModeloV03)

# Scatter Plot Comparativo

ggplot(FitModeloV03, aes(x = Previsao, y = Target)) +
  geom_point(shape = 1) +
  geom_smooth(method = lm, color = 'red', se = FALSE) +
  ggtitle('Performance do Modelo V03')+
  annotate(geom = 'text', x = 17, y = 25, label = 'R² = 65,12% RSME = 3.34')
```

```
>       ModeloV03
Boosted Linear Model

33 samples
 6 predictor

No pre-processing
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 33, 33, 33, 33, 33, 33, ...
Resampling results across tuning parameters:

  mstop  RMSE      Rsquared   MAE
   50    3.833294  0.5773610  3.431336
  100    3.566225  0.6157972  3.111424
  150    3.341078  0.6512431  2.854399

Tuning parameter 'nu' was held constant at a value of 0.1
RMSE was used to select the optimal model using the
 smallest value.
The final values used for the model were mstop = 150 and nu
 = 0.1.
```

```
  Target Previsao    Residuo
1  23.85 23.43588  0.4141222
2  14.30 17.16161 -2.8616109
3  18.80 20.10335 -1.3033470
4  15.40 17.96319 -2.5631871
5  15.30 17.78603 -2.4860301
6  21.85 19.54537  2.3046302
```

```
     Target          Previsao         Residuo
 Min.   :14.30   Min.   :16.65   Min.   :-2.8616
 1st Qu.:15.40   1st Qu.:17.16   1st Qu.:-2.4860
 Median :16.65   Median :17.96   Median :-0.2657
 Mean   :18.53   Mean   :19.20   Mean   :-0.6733
 3rd Qu.:21.85   3rd Qu.:20.10   3rd Qu.: 0.4141
 Max.   :24.10   Max.   :23.44   Max.   : 2.3046
```



Project developed by Thiago Bulgarelli

Contact: bugath36@gmail.com

In this model, we achieved an R^2 of approximately 65.12% with 25 repeated cross-validations. In this method, we performed multiple iterations with smaller samples of the dataset and identified the one with the lowest RMSE and consequently the highest R^2. However, we obtained a worse result than in our Base Model.

Model V04

For this model, we will use another method from the Caret package called "glmnet".

```
# Contruindo Modelo V04

    # Para este modelo vamos alterar o método de Regressão Linear para glmnet
    # e analisar os resultados.
    # Utilizaremos o mesmo pacote Caret.

      ModeloV04 <- train(ConsMedio ~ ., data = dados_treino, method = 'glmnet')
      ModeloV04

      # Tivemos uma piora drástica do R² em relação ao Modelo V03, porém ainda
      # assim nosso modelo base é melhor em relação a métrica de varibilidade
      # das variáveis em relação a variável resposta.
```

```
      Previsao04 <- predict(ModeloV04, dados_teste[-8])
      Previsao04

      # Analisando o resíduo do modelo

      Res_ModeloV04 <- ConsumoTeste - Previsao04

      FitModeloV04 <- data.frame(Target = ConsumoTeste,
                                 Previsao = Previsao04,
                                 Residuo = Res_ModeloV04)
      head(FitModeloV03)
      summary(FitModeloV04)

      # Scatter Plot Comparativo

      ggplot(FitModeloV04, aes(x = Previsao, y = Target)) +
        geom_point(shape = 1) +
        geom_smooth(method = lm, color = 'red', se = FALSE) +
        ggtitle('Performance do Modelo V04')+
        annotate(geom = 'text', x = 17, y = 25, label = 'R² = 77,77%')
```

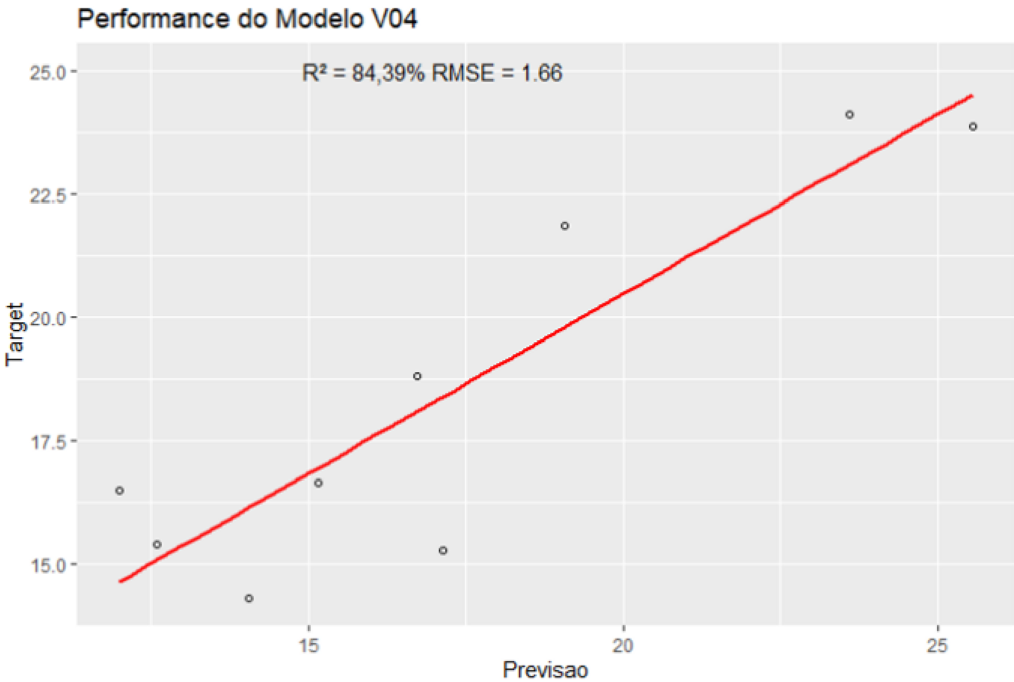
Project developed by Thiago Bulgarelli

Contact: bugath36@gmail.com

```
>       ModeloV04
glmnet

33 samples
 6 predictor

No pre-processing
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 33, 33, 33, 33, 33, 33, ...
Resampling results across tuning parameters:

  alpha  lambda       RMSE      Rsquared   MAE
  0.10   0.006295296  1.664000  0.8439565  1.236413
  0.10   0.062952965  1.704398  0.8422168  1.290303
  0.10   0.629529645  2.169265  0.7640937  1.710190
  0.55   0.006295296  1.660722  0.8450119  1.232211
  0.55   0.062952965  1.687921  0.8464261  1.276365
  0.55   0.629529645  2.483541  0.6968285  1.942585
  1.00   0.006295296  1.656760  0.8460774  1.228075
  1.00   0.062952965  1.686541  0.8458511  1.275791
  1.00   0.629529645  2.813177  0.6168897  2.166624

RMSE was used to select the optimal model using the
 smallest value.
The final values used for the model were alpha = 1 and lambda
 = 0.006295296.
```

```
  Target Previsao    Residuo
1  23.85 23.43588  0.4141222
2  14.30 17.16161 -2.8616109
3  18.80 20.10335 -1.3033470
4  15.40 17.96319 -2.5631871
5  15.30 17.78603 -2.4860301
6  21.85 19.54537  2.3046302
```

```
    Target           Previsao          Residuo
 Min.   :14.30   Min.   :12.00   Min.   :-1.8369
 1st Qu.:15.40   1st Qu.:14.05   1st Qu.: 0.2538
 Median :16.65   Median :16.73   Median : 1.5001
 Mean   :18.53   Mean   :17.32   Mean   : 1.2085
 3rd Qu.:21.85   3rd Qu.:19.07   3rd Qu.: 2.7768
 Max.   :24.10   Max.   :25.55   Max.   : 4.5012
```



We have achieved an R^2 of 84.39% in this model, which represents a significant improvement compared to Model V03, although it is still lower than our baseline model. However, the model has a better accuracy of 1.66.

Conclusion

In conclusion, we have completed our work and identified Model V04 as the most suitable for making predictions for the business area. It showed improved performance in terms of metrics and generated residuals compared to previous models.

It is important to note that any new data to be applied to the model's variables must be treated in the same way as during the preprocessing process. This includes removing outliers, handling missing values, and normalizing the data to ensure consistent scaling.

We conclude the project with this report, which includes a description of the script implemented in the R programming language.