

Projeto com Feedback 1 - Machine Learning em Logística Prevendo o Consumo de Energia de Carros Elétricos

Marcos Vinícius Gonzaga - Aluno da Formação Cientista de Dados na Data Science Academy

Projeto com Feedback 1:

Machine Learning em Logística Prevendo o Consumo de Energia de Carros Elétricos

Problema de Negócio

Uma empresa da área de transporte e logística deseja migrar sua frota para

carros elétricos com o objetivo de reduzir os custos. Antes de tomar a decisão,

a empresa gostaria de prever o consumo de energia de carros elétricos com base

em diversos fatores de utilização e características dos veículos.

Definindo Diretório de Trabalho

```
setwd("C:/Users/marcos/Documents/Cientista_de_Dados/BigDataRAzure/Projetos_com_Feedback/Arquivos/Projetos")  
getwd()
```

```
## [1] "C:/Users/marcos/Documents/Cientista_de_Dados/BigDataRAzure/Projetos_com_Feedback/Arquivos/Projetos"
```

Etapa 1: Carregando Pacotes

```
library(tidyverse)
```

```
## -- Attaching packages ----- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr 0.3.4
## v tibble 3.1.7       v dplyr 1.0.10
## v tidyr 1.2.0        v stringr 1.4.0
## v readr 2.1.2        v forcats 0.5.2
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()
```

```
library(ggplot2)
library(readxl)
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
library(caTools)
```

Etapa 2: Carregando Dados

```
df <- read_excel("FEV-data-Excel.xlsx")
class(df)
```

```
## [1] "tbl_df"      "tbl"        "data.frame"
```

Etapa 3: ##### Análise Exploratória dos Dados - Limpeza dos Dados

Informações sobre o Dataframe

```
dim(df)
```

```
## [1] 53 25
```

```
str(df)
```

```
## tibble [53 x 25] (S3: tbl_df/tbl/data.frame)
##  $ Car full name      : chr [1:53] "Audi e-tron 55 quattro" "Audi e-tron 50 quattro" ...
##  $ Make                : chr [1:53] "Audi" "Audi" "Audi" "Audi" ...
##  $ Model               : chr [1:53] "e-tron 55 quattro" "e-tron 50 quattro" "e-tron 50 quattro" ...
##  $ Minimal price (gross) [PLN] : num [1:53] 345700 308400 414900 319700 357000 ...
##  $ Engine power [KM]    : 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)" ...
##  $ 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 ...
## $ Permissible 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 ...
```

```
head(df)
```

```
## # A tibble: 6 x 25
##   'Car full name' Make Model 'Minimal price~' 'Engine power ~' 'Maximum torqu~'
##   <chr>          <chr> <chr>          <dbl>          <dbl>          <dbl>
## 1 Audi e-tron 55~ Audi e-tr~          345700          360          664
## 2 Audi e-tron 50~ Audi e-tr~          308400          313          540
## 3 Audi e-tron S ~ Audi e-tr~          414900          503          973
## 4 Audi e-tron Sp~ Audi e-tr~          319700          313          540
## 5 Audi e-tron Sp~ Audi e-tr~          357000          360          664
## 6 Audi e-tron Sp~ Audi e-tr~          426200          503          973
## # ... with 19 more variables: 'Type of brakes' <chr>, 'Drive type' <chr>,
## #   'Battery capacity [kWh]' <dbl>, 'Range (WLTP) [km]' <dbl>,
## #   'Wheelbase [cm]' <dbl>, 'Length [cm]' <dbl>, 'Width [cm]' <dbl>,
## #   'Height [cm]' <dbl>, 'Minimal empty weight [kg]' <dbl>,
## #   'Permissible gross weight [kg]' <dbl>, 'Maximum load capacity [kg]' <dbl>,
## #   'Number of seats' <dbl>, 'Number of doors' <dbl>, 'Tire size [in]' <dbl>,
## #   'Maximum speed [kph]' <dbl>, 'Boot capacity (VDA) [l]' <dbl>, ...
```

Visualizando os Dados

```
View(df)
```

Etapa 3.1: Verificando se existe valores ausentes (NA)

```
verifica_na <- function(x){
  colSums(is.na(x))
}
```

Etapa 3.1.1: Chamando a Função “verifica_na”

```
verifica_na(df)
```

```
##           Car full name           Make
##           0           0
##           Model       Minimal price (gross) [PLN]
##           0           0
##           Engine power [KM]       Maximum torque [Nm]
##           0           0
##           Type of brakes           Drive type
##           1           0
##           Battery capacity [kWh]       Range (WLTP) [km]
##           0           0
##           Wheelbase [cm]           Length [cm]
##           0           0
##           Width [cm]           Height [cm]
##           0           0
##           Minimal empty weight [kg]       Permissable gross weight [kg]
##           0           8
##           Maximum load capacity [kg]       Number of seats
##           8           0
##           Number of doors           Tire size [in]
##           0           0
##           Maximum speed [kph]       Boot capacity (VDA) [l]
##           0           1
##           Acceleration 0-100 kph [s]       Maximum DC charging power [kW]
##           3           0
## mean - Energy consumption [kWh/100 km]
##           9
```

Etapa 3.2: Verificando quantas observações possuem casos completos

```
casos_completos <- sum(complete.cases(df))
casos_completos
```

```
## [1] 42
```

Etapa 3.2.1: Verificando quantas observações possuem casos incompletos

```
casos_incompletos <- sum(!complete.cases(df))
casos_incompletos
```

```
## [1] 11
```

Etapa 3.3.2: Percentual de casos incompletos no Dataset

```
percentual = (casos_incompletos/53) * 100  
percentual
```

```
## [1] 20.75472
```

Etapa 3.3.3: Removendo o objeto anterior para liberar memória RAM

```
rm(percentual)
```

Conclusão preliminar da Análise de Dados acima:

O Dataset contém uma quantidade considerável de valores NA, 20.75% do total de

Observações, concentrando a maioria na variável target. Para a criação do

modelo preditivo, as observações com valores NA serão omitidas.

Etapa 4: Removendo a primeira coluna (“Car full name”) do Dataset.

Já existe variáveis com a mesma informação dentro do próprio conjunto de dados

```
df[1] <- NULL
```

```
View(df)  
head(df)
```

```
## # A tibble: 6 x 24  
##   Make Model           'Minimal price~' 'Engine power ~' 'Maximum torqu~'  
##   <chr> <chr>                <dbl>          <dbl>          <dbl>  
## 1 Audi  e-tron 55 quattro      345700         360           664  
## 2 Audi  e-tron 50 quattro      308400         313           540  
## 3 Audi  e-tron S quattro      414900         503           973
```

```
## 4 Audi e-tron Sportback 50 ~ 319700 313 540
## 5 Audi e-tron Sportback 55 ~ 357000 360 664
## 6 Audi e-tron Sportback S q~ 426200 503 973
## # ... with 19 more variables: 'Type of brakes' <chr>, 'Drive type' <chr>,
## # 'Battery capacity [kWh]' <dbl>, 'Range (WLTP) [km]' <dbl>,
## # 'Wheelbase [cm]' <dbl>, 'Length [cm]' <dbl>, 'Width [cm]' <dbl>,
## # 'Height [cm]' <dbl>, 'Minimal empty weight [kg]' <dbl>,
## # 'Permissible gross weight [kg]' <dbl>, 'Maximum load capacity [kg]' <dbl>,
## # 'Number of seats' <dbl>, 'Number of doors' <dbl>, 'Tire size [in]' <dbl>,
## # 'Maximum speed [kph]' <dbl>, 'Boot capacity (VDA) [l]' <dbl>, ...
```

```
dim(df)
```

```
## [1] 53 24
```

Etapa 5: Alterando os nomes das colunas, e gravando em um vetor

```
new_names <- colnames(df)
new_names
```

```
## [1] "Make"
## [2] "Model"
## [3] "Minimal price (gross) [PLN]"
## [4] "Engine power [KM]"
## [5] "Maximum torque [Nm]"
## [6] "Type of brakes"
## [7] "Drive type"
## [8] "Battery capacity [kWh]"
## [9] "Range (WLTP) [km]"
## [10] "Wheelbase [cm]"
## [11] "Length [cm]"
## [12] "Width [cm]"
## [13] "Height [cm]"
## [14] "Minimal empty weight [kg]"
## [15] "Permissible gross weight [kg]"
## [16] "Maximum load capacity [kg]"
## [17] "Number of seats"
## [18] "Number of doors"
## [19] "Tire size [in]"
## [20] "Maximum speed [kph]"
## [21] "Boot capacity (VDA) [l]"
## [22] "Acceleration 0-100 kph [s]"
## [23] "Maximum DC charging power [kW]"
## [24] "mean - Energy consumption [kWh/100 km]"
```

Etapa 5.1: Nomeando as colunas do vetor

```

new_names[1] <- "Fabricante"
new_names[2] <- "Modelo"
new_names[3] <- "Preco_Minimo(PLN)"
new_names[4] <- "Potencia_Motor(KM)"
new_names[5] <- "Torque(Nm)"
new_names[6] <- "Tipo_Freio"
new_names[7] <- "Transmissao"
new_names[8] <- "Capacidade_Bateria(kwh)"
new_names[9] <- "Autonomia(WLTP)(km)"
new_names[10] <- "Distancia_Eixos(cm)"
new_names[11] <- "Comprimento(cm)"
new_names[12] <- "Largura(cm)"
new_names[13] <- "Altura(cm)"
new_names[14] <- "Peso_Vazio(kg)"
new_names[15] <- "Peso_Admissivel(kg)"
new_names[16] <- "Carga_Max(kg)"
new_names[17] <- "N_Assentos"
new_names[18] <- "N_Portas"
new_names[19] <- "Raio_Pneu(in)"
new_names[20] <- "Vel_Maxima(kph)"
new_names[21] <- "Mala(VDA)(L)"
new_names[22] <- "Aceleracao_0_100(kph)(s)"
new_names[23] <- "Potencia_Max_Carregamento(kW)"
new_names[24] <- "Consumo_Energia(KWh/100km)"

```

Visualizando o resultado

```
new_names
```

```

## [1] "Fabricante"           "Modelo"
## [3] "Preco_Minimo(PLN)"    "Potencia_Motor(KM)"
## [5] "Torque(Nm)"           "Tipo_Freio"
## [7] "Transmissao"          "Capacidade_Bateria(kwh)"
## [9] "Autonomia(WLTP)(km)" "Distancia_Eixos(cm)"
## [11] "Comprimento(cm)"      "Largura(cm)"
## [13] "Altura(cm)"           "Peso_Vazio(kg)"
## [15] "Peso_Admissivel(kg)"  "Carga_Max(kg)"
## [17] "N_Assentos"           "N_Portas"
## [19] "Raio_Pneu(in)"        "Vel_Maxima(kph)"
## [21] "Mala(VDA)(L)"         "Aceleracao_0_100(kph)(s)"
## [23] "Potencia_Max_Carregamento(kW)" "Consumo_Energia(KWh/100km)"

```

Etapa 5.2: Atribuindo os nomes do vetor ao Dataset

```
colnames(df) <- new_names
```

Etapa 5.3: Removendo os objetos anteriores para liberar memória RAM

```
rm(new_names)
```

Visualizando o Dataset com as colunas renomeadas

```
View(df)
head(df)

## # A tibble: 6 x 24
##   Fabricante Modelo   'Preco_Minimo(~' 'Potencia_Moto~' 'Torque(Nm)' Tipo_Freio
##   <chr>         <chr>         <dbl>         <dbl>         <dbl> <chr>
## 1 Audi         e-tron 5~         345700         360         664 disc (fro~
## 2 Audi         e-tron 5~         308400         313         540 disc (fro~
## 3 Audi         e-tron S~         414900         503         973 disc (fro~
## 4 Audi         e-tron S~         319700         313         540 disc (fro~
## 5 Audi         e-tron S~         357000         360         664 disc (fro~
## 6 Audi         e-tron S~         426200         503         973 disc (fro~
## # ... with 18 more variables: Transmissao <chr>,
## #   'Capacidade_Bateria(kwh)' <dbl>, 'Autonomia(WLTP)(km)' <dbl>,
## #   'Distancia_Eixos(cm)' <dbl>, 'Comprimento(cm)' <dbl>, 'Largura(cm)' <dbl>,
## #   'Altura(cm)' <dbl>, 'Peso_Vazio(kg)' <dbl>, 'Peso_Admissivel(kg)' <dbl>,
## #   'Carga_Max(kg)' <dbl>, N_Assentos <dbl>, N_Portas <dbl>,
## #   'Raio_Pneu(in)' <dbl>, 'Vel_Maxima(kph)' <dbl>, 'Mala(VDA)(L)' <dbl>,
## #   'Aceleracao_0_100(kph)(s)' <dbl>, ...
```

Etapa 6: Criando uma nova variável, relação “peso/potencia” para determinar

o desempenho do carro

Etapa 6.1: Copiando os dados da variável “Potencia_Motor(KM)” para uma

variável temporária

```
potencia <- df$`Potencia_Motor(KM)`
potencia

## [1] 360 313 503 313 360 503 170 184 286 136 136 136 154 136 136 204 400 136 204
## [20] 136 204 145 408 184 150 217 136 136 136 136 435 490 625 625 108 135 83 82
## [39] 82 285 372 480 525 772 525 772 83 204 204 204 136 204 109
```


Etapa 6.1.1: Converterdo a unidade de medida da variável “Potencia_Motor(KM)”

para cavalo de força (hp)

```
potencia <- potencia * 1.341
potencia
```

```
## [1] 482.760 419.733 674.523 419.733 482.760 674.523 227.970 246.744
## [9] 383.526 182.376 182.376 182.376 206.514 182.376 182.376 273.564
## [17] 536.400 182.376 273.564 182.376 273.564 194.445 547.128 246.744
## [25] 201.150 290.997 182.376 182.376 182.376 182.376 583.335 657.090
## [33] 838.125 838.125 144.828 181.035 111.303 109.962 109.962 382.185
## [41] 498.852 643.680 704.025 1035.252 704.025 1035.252 111.303 273.564
## [49] 273.564 273.564 182.376 273.564 146.169
```

Etapa 6.2: Copiando os dados da variável “Peso_Vazio(kg)” para uma

variável temporária

```
peso <- df$`Peso_Vazio(kg)`
peso
```

```
## [1] 2565 2445 2695 2445 2595 2695 1440 1460 2260 1541 1523 1514 1543 1527 1535
## [16] 1685 2208 1592 1737 1535 1535 1645 2495 1300 1545 1705 1530 1598 1455 1548
## [31] 2215 2295 2380 2370 1502 1502 1178 1035 1140 1626 1862 1862 2391 2417 2464
## [46] 2524 1235 1805 1934 2124 1969 2710 1592
```

Etapa 6.3: Criando a variável “desempenho” que será incorporada no Dataset

```
desempenho <- peso/potencia
desempenho
```

```
## [1] 5.313199 5.825132 3.995416 5.825132 5.375342 3.995416 6.316621
## [8] 5.917064 5.892690 8.449577 8.350880 8.301531 7.471648 8.372812
## [15] 8.416678 6.159436 4.116331 8.729219 6.349520 8.416678 5.611118
## [22] 8.459976 4.560176 5.268618 7.680835 5.859167 8.389262 8.762118
## [29] 7.978023 8.487959 3.797132 3.492672 2.839672 2.827740 10.370923
## [36] 8.296738 10.583722 9.412342 10.367218 4.254484 3.732570 2.892742
## [43] 3.396186 2.334697 3.499876 2.438054 11.095837 6.598090 7.069644
## [50] 7.764179 10.796377 9.906274 10.891502
```

Etapa 6.4: Adicionando a nova coluna ao Dataframe

```
df$`Desempenho(kg/hp)` <- c(desempenho)
View(df)
head(df)
```

```
## # A tibble: 6 x 25
##   Fabricante Modelo   'Preco_Minimo(~' 'Potencia_Moto~' 'Torque(Nm)' Tipo_Freio
##   <chr>         <chr>         <dbl>         <dbl>         <dbl> <chr>
## 1 Audi         e-tron 5~         345700         360         664 disc (fro~
## 2 Audi         e-tron 5~         308400         313         540 disc (fro~
## 3 Audi         e-tron S~         414900         503         973 disc (fro~
## 4 Audi         e-tron S~         319700         313         540 disc (fro~
## 5 Audi         e-tron S~         357000         360         664 disc (fro~
## 6 Audi         e-tron S~         426200         503         973 disc (fro~
## # ... with 19 more variables: Transmissao <chr>,
## #   'Capacidade_Bateria(kwh)' <dbl>, 'Autonomia(WLTP)(km)' <dbl>,
## #   'Distancia_Eixos(cm)' <dbl>, 'Comprimento(cm)' <dbl>, 'Largura(cm)' <dbl>,
## #   'Altura(cm)' <dbl>, 'Peso_Vazio(kg)' <dbl>, 'Peso_Admissivel(kg)' <dbl>,
## #   'Carga_Max(kg)' <dbl>, N_Assentos <dbl>, N_Portas <dbl>,
## #   'Raio_Pneu(in)' <dbl>, 'Vel_Maxima(kph)' <dbl>, 'Mala(VDA)(L)' <dbl>,
## #   'Aceleracao_0_100(kph)(s)' <dbl>, ...
```

```
dim(df)
```

```
## [1] 53 25
```

Etapa 7: Removendo os objetos anteriores para liberar memória RAM

```
rm(peso)
rm(potencia)
rm(desempenho)
```

Etapa 8: Omitindo valores ausêntes do Dataset, e criando uma nova variável “df2”

```
df2 <- na.omit(df)
```

Vizualizando o Novo Dataset com os valores ausêntes removidos

```
View(df2)
head(df2)
```

```
## # A tibble: 6 x 25
##   Fabricante Modelo   'Preco_Minimo(~' 'Potencia_Moto~' 'Torque(Nm)' Tipo_Freio
##   <chr>         <chr>         <dbl>         <dbl>         <dbl> <chr>
## 1 Audi         e-tron 5~         345700         360           664 disc (fro~
## 2 Audi         e-tron 5~         308400         313           540 disc (fro~
## 3 Audi         e-tron S~         414900         503           973 disc (fro~
## 4 Audi         e-tron S~         319700         313           540 disc (fro~
## 5 Audi         e-tron S~         357000         360           664 disc (fro~
## 6 Audi         e-tron S~         426200         503           973 disc (fro~
## # ... with 19 more variables: Transmissao <chr>,
## #   'Capacidade_Bateria(kwh)' <dbl>, 'Autonomia(WLTP)(km)' <dbl>,
## #   'Distancia_Eixos(cm)' <dbl>, 'Comprimento(cm)' <dbl>, 'Largura(cm)' <dbl>,
## #   'Altura(cm)' <dbl>, 'Peso_Vazio(kg)' <dbl>, 'Peso_Admissivel(kg)' <dbl>,
## #   'Carga_Max(kg)' <dbl>, N_Assentos <dbl>, N_Portas <dbl>,
## #   'Raio_Pneu(in)' <dbl>, 'Vel_Maxima(kph)' <dbl>, 'Mala(VDA)(L)' <dbl>,
## #   'Aceleracao_0_100(kph)(s)' <dbl>, ...
```

```
dim(df2)
```

```
## [1] 42 25
```

Etapa 9: Visualizando as colunas removidas

```
removed_col <- setdiff(df,df2)
removed_col
```

```
## # A tibble: 11 x 25
##   Fabricante Modelo   'Preco_Minimo(~' 'Potencia_Moto~' 'Torque(Nm)' Tipo_Freio
##   <chr>         <chr>         <dbl>         <dbl>         <dbl> <chr>
## 1 Citroën      ë-C4         125000         136           260 disc (fro~
## 2 Peugeot      e-2008       149400         136           260 disc (fro~
## 3 Tesla        Model~       195490         285           450 disc (fro~
## 4 Tesla        Model~       235490         372           510 disc (fro~
## 5 Tesla        Model~       260490         480           639 disc (fro~
## 6 Tesla        Model~       368990         525           755 disc (fro~
## 7 Tesla        Model~       443990         772           1140 disc (fro~
## 8 Tesla        Model~       407990         525           755 disc (fro~
## 9 Tesla        Model~       482990         772           1140 disc (fro~
## 10 Mercedes-Be~ EQV (~       339480         204           362 <NA>
## 11 Nissan       e-NV2~       164328         109           254 disc (fro~
## # ... with 19 more variables: Transmissao <chr>,
## #   'Capacidade_Bateria(kwh)' <dbl>, 'Autonomia(WLTP)(km)' <dbl>,
## #   'Distancia_Eixos(cm)' <dbl>, 'Comprimento(cm)' <dbl>, 'Largura(cm)' <dbl>,
## #   'Altura(cm)' <dbl>, 'Peso_Vazio(kg)' <dbl>, 'Peso_Admissivel(kg)' <dbl>,
## #   'Carga_Max(kg)' <dbl>, N_Assentos <dbl>, N_Portas <dbl>,
## #   'Raio_Pneu(in)' <dbl>, 'Vel_Maxima(kph)' <dbl>, 'Mala(VDA)(L)' <dbl>,
## #   'Aceleracao_0_100(kph)(s)' <dbl>, ...
```

Etapa 10: Extraindo as variáveis numéricas do Dataset “df2”

```
variaveis_numericas <- sapply(df2, is.numeric)
dados_numericos <- df2[variaveis_numericas]
```

```
View(dados_numericos)
head(dados_numericos)
```

```
## # A tibble: 6 x 21
##   'Preco_Minimo(PLN)' 'Potencia_Motor(KM)' 'Torque(Nm)' 'Capacidade_Bateria(kw~'
##           <dbl>           <dbl>           <dbl>           <dbl>
## 1           345700           360           664           95
## 2           308400           313           540           71
## 3           414900           503           973           95
## 4           319700           313           540           71
## 5           357000           360           664           95
## 6           426200           503           973           95
## # ... with 17 more variables: 'Autonomia(WLTP)(km)' <dbl>,
## #   'Distancia_Eixos(cm)' <dbl>, 'Comprimento(cm)' <dbl>, 'Largura(cm)' <dbl>,
## #   'Altura(cm)' <dbl>, 'Peso_Vazio(kg)' <dbl>, 'Peso_Admissivel(kg)' <dbl>,
## #   'Carga_Max(kg)' <dbl>, N_Assentos <dbl>, N_Portas <dbl>,
## #   'Raio_Pneu(in)' <dbl>, 'Vel_Maxima(kph)' <dbl>, 'Mala(VDA)(L)' <dbl>,
## #   'Aceleracao_0_100(kph)(s)' <dbl>, 'Potencia_Max_Carregamento(kW)' <dbl>,
## #   'Consumo_Energia(KWh/100km)' <dbl>, 'Desempenho(kg/hp)' <dbl>
```

```
dim(dados_numericos)
```

```
## [1] 42 21
```

Etapa 10.1: Removendo o objeto anterior para liberar memória RAM

```
rm(variaveis_numericas)
```

Plots e Estatísticas

Etapa 11: Criando uma Matrix de Correlação para compreender o relacionamento

entre as variáveis numéricas

Etapa 11.1: Salvando a Matrix de Correlção em uma variável

```
interacao <- cor(dados_numericos)
interacao
```

##	Preco_Minimo(PLN)	Potencia_Motor(KM)	Torque(Nm)
##	Preco_Minimo(PLN)	1.00000000	0.96090955 0.902694431
##	Potencia_Motor(KM)	0.96090955	1.00000000 0.952062140
##	Torque(Nm)	0.90269443	0.95206214 1.000000000
##	Capacidade_Bateria(kwh)	0.79408068	0.86852061 0.831894406
##	Autonomia(WLTP) (km)	0.45771017	0.53700763 0.450305128
##	Distancia_Eixos(cm)	0.62435370	0.64498383 0.635060942
##	Comprimento(cm)	0.73475225	0.77315650 0.760928943
##	Largura(cm)	0.47621771	0.47187218 0.457055008
##	Altura(cm)	-0.23587726	-0.19381690 -0.092389876
##	Peso_Vazio(kg)	0.78670749	0.85533411 0.858799330
##	Peso_Admissivel(kg)	0.76776036	0.81507279 0.803782763
##	Carga_Max(kg)	0.47506858	0.43270799 0.390599129
##	N_Assentos	-0.09068844	-0.08745365 -0.001363035
##	N_Portas	-0.30669290	-0.26089020 -0.133803691
##	Raio_Pneu(in)	0.75024149	0.80097775 0.749802241
##	Vel_Maxima(kph)	0.94381157	0.95168277 0.877643876
##	Mala(VDA) (L)	0.57594755	0.64427362 0.676182162
##	Aceleracao_0_100(kph) (s)	-0.81099044	-0.89669357 -0.832416924
##	Potencia_Max_Carregamento(kW)	0.88771930	0.85407706 0.770809047
##	Consumo_Energia(KWh/100km)	0.79944742	0.82464843 0.827639009
##	Desempenho(kg/hp)	-0.79623326	-0.88970536 -0.818446587
##	Capacidade_Bateria(kwh)	Autonomia(WLTP) (km)	
##	Preco_Minimo(PLN)	0.79408068	0.4577102
##	Potencia_Motor(KM)	0.86852061	0.5370076
##	Torque(Nm)	0.83189441	0.4503051
##	Capacidade_Bateria(kwh)	1.00000000	0.8092553
##	Autonomia(WLTP) (km)	0.80925526	1.0000000
##	Distancia_Eixos(cm)	0.74460203	0.5075445
##	Comprimento(cm)	0.84183683	0.6044038
##	Largura(cm)	0.52470674	0.3366901
##	Altura(cm)	0.04163423	-0.0112427
##	Peso_Vazio(kg)	0.92096505	0.5913838
##	Peso_Admissivel(kg)	0.88501111	0.5497372
##	Carga_Max(kg)	0.50753901	0.2736420
##	N_Assentos	0.16724829	0.1305064
##	N_Portas	0.03800588	0.1525001
##	Raio_Pneu(in)	0.78821091	0.6188445
##	Vel_Maxima(kph)	0.82387115	0.5611077
##	Mala(VDA) (L)	0.79423877	0.5088502
##	Aceleracao_0_100(kph) (s)	-0.81940251	-0.6276312
##	Potencia_Max_Carregamento(kW)	0.78890175	0.5512162
##	Consumo_Energia(KWh/100km)	0.75786574	0.2744891
##	Desempenho(kg/hp)	-0.78529848	-0.6052826
##	Distancia_Eixos(cm)	Comprimento(cm)	Largura(cm)
##	Preco_Minimo(PLN)	0.6243537	0.7347522 0.47621771
##	Potencia_Motor(KM)	0.6449838	0.7731565 0.47187218
##	Torque(Nm)	0.6350609	0.7609289 0.45705501
##	Capacidade_Bateria(kwh)	0.7446020	0.8418368 0.52470674
##	Autonomia(WLTP) (km)	0.5075445	0.6044038 0.33669012

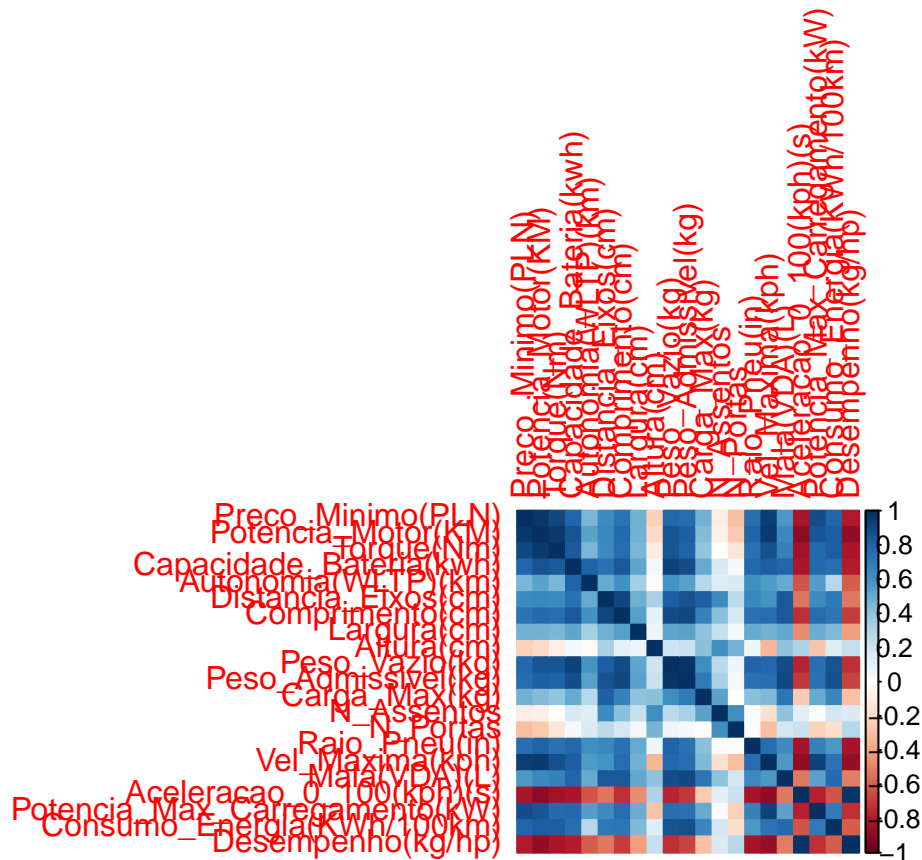
## Distancia_Eixos(cm)	1.0000000	0.9137165	0.48806451
## Comprimento(cm)	0.9137165	1.0000000	0.54354711
## Largura(cm)	0.4880645	0.5435471	1.00000000
## Altura(cm)	0.3284534	0.1011494	0.07603541
## Peso_Vazio(kg)	0.8280735	0.8964352	0.52459686
## Peso_Admissivel(kg)	0.8679398	0.9058960	0.53667898
## Carga_Max(kg)	0.8105733	0.6770901	0.40205020
## N_Assentos	0.6256672	0.4172107	0.23262103
## N_Portas	0.2646807	0.1712633	0.04435635
## Raio_Pneu(in)	0.6433391	0.7467170	0.50014251
## Vel_Maxima(kph)	0.5959374	0.7636150	0.47471759
## Mala(VDA) (L)	0.8442445	0.8418493	0.51408200
## Aceleracao_0_100(kph) (s)	-0.5372283	-0.7348261	-0.46355915
## Potencia_Max_Carregamento(kW)	0.6274435	0.7633754	0.49159807
## Consumo_Energia(KWh/100km)	0.7164781	0.7210755	0.44686912
## Desempenho(kg/hp)	-0.5236950	-0.6969048	-0.41573146
##	Altura(cm)	Peso_Vazio(kg)	Peso_Admissivel(kg)
## Preco_Minimo(PLN)	-0.23587726	0.78670749	0.76776036
## Potencia_Motor(KM)	-0.19381690	0.85533411	0.81507279
## Torque(Nm)	-0.09238988	0.85879933	0.80378276
## Capacidade_Bateria(kwh)	0.04163423	0.92096505	0.88501111
## Autonomia(WLTP) (km)	-0.01124270	0.59138380	0.54973719
## Distancia_Eixos(cm)	0.32845344	0.82807346	0.86793980
## Comprimento(cm)	0.10114942	0.89643525	0.90589604
## Largura(cm)	0.07603541	0.52459686	0.53667898
## Altura(cm)	1.00000000	0.20215504	0.21625601
## Peso_Vazio(kg)	0.20215504	1.00000000	0.97972500
## Peso_Admissivel(kg)	0.21625601	0.97972500	1.00000000
## Carga_Max(kg)	0.41431910	0.61316189	0.70931840
## N_Assentos	0.61912209	0.26314414	0.32961612
## N_Portas	0.46312924	0.06579237	0.04193092
## Raio_Pneu(in)	0.06801116	0.80848609	0.74973766
## Vel_Maxima(kph)	-0.33049793	0.78021624	0.75049975
## Mala(VDA) (L)	0.40282847	0.88201835	0.90614041
## Aceleracao_0_100(kph) (s)	0.30330802	-0.76406884	-0.70482336
## Potencia_Max_Carregamento(kW)	-0.28568743	0.75193494	0.74551349
## Consumo_Energia(KWh/100km)	0.13285838	0.86190952	0.87477101
## Desempenho(kg/hp)	0.23634874	-0.73366462	-0.67231954
##	Carga_Max(kg)	N_Assentos	N_Portas
## Preco_Minimo(PLN)	0.475068582	-0.090688436	-0.306692899
## Potencia_Motor(KM)	0.432707985	-0.087453650	-0.260890205
## Torque(Nm)	0.390599129	-0.001363035	-0.133803691
## Capacidade_Bateria(kwh)	0.507539014	0.167248289	0.038005880
## Autonomia(WLTP) (km)	0.273642010	0.130506408	0.152500139
## Distancia_Eixos(cm)	0.810573337	0.625667199	0.264680729
## Comprimento(cm)	0.677090091	0.417210653	0.171263315
## Largura(cm)	0.402050200	0.232621034	0.044356345
## Altura(cm)	0.414319101	0.619122086	0.463129238
## Peso_Vazio(kg)	0.613161895	0.263144144	0.065792365
## Peso_Admissivel(kg)	0.709318405	0.329616122	0.041930925
## Carga_Max(kg)	1.000000000	0.536034513	-0.002314982
## N_Assentos	0.536034513	1.000000000	0.616216585
## N_Portas	-0.002314982	0.616216585	1.000000000
## Raio_Pneu(in)	0.419874021	0.024790506	-0.016257241

## Vel_Maxima(kph)	0.447107139	-0.160005945	-0.302964824
## Mala(VDA) (L)	0.728206393	0.407679455	0.188547338
## Aceleracao_0_100(kph) (s)	-0.255273354	0.122769541	0.212901102
## Potencia_Max_Carregamento(kW)	0.509866271	0.021871644	-0.236286284
## Consumo_Energia(KWh/100km)	0.650026546	0.183648113	-0.172312792
## Desempenho(kg/hp)	-0.291637429	0.151263253	0.283626169
##	Raio_Pneu(in)	Vel_Maxima(kph)	Mala(VDA) (L)
## Preco_Minimo(PLN)	0.75024149	0.9438116	0.5759475
## Potencia_Motor(KM)	0.80097775	0.9516828	0.6442736
## Torque(Nm)	0.74980224	0.8776439	0.6761822
## Capacidade_Bateria(kwh)	0.78821091	0.8238711	0.7942388
## Autonomia(WLTP) (km)	0.61884454	0.5611077	0.5088502
## Distancia_Eixos(cm)	0.64333906	0.5959374	0.8442445
## Comprimento(cm)	0.74671705	0.7636150	0.8418493
## Largura(cm)	0.50014251	0.4747176	0.5140820
## Altura(cm)	0.06801116	-0.3304979	0.4028285
## Peso_Vazio(kg)	0.80848609	0.7802162	0.8820183
## Peso_Admissivel(kg)	0.74973766	0.7504997	0.9061404
## Carga_Max(kg)	0.41987402	0.4471071	0.7282064
## N_Assentos	0.02479051	-0.1600059	0.4076795
## N_Portas	-0.01625724	-0.3029648	0.1885473
## Raio_Pneu(in)	1.00000000	0.7693357	0.6394526
## Vel_Maxima(kph)	0.76933570	1.0000000	0.5758061
## Mala(VDA) (L)	0.63945261	0.5758061	1.0000000
## Aceleracao_0_100(kph) (s)	-0.82744844	-0.8870691	-0.5249693
## Potencia_Max_Carregamento(kW)	0.67054887	0.9209751	0.5345822
## Consumo_Energia(KWh/100km)	0.57797942	0.7304696	0.7721157
## Desempenho(kg/hp)	-0.82928991	-0.8766168	-0.5115341
##	Aceleracao_0_100(kph) (s)		
## Preco_Minimo(PLN)		-0.8109904	
## Potencia_Motor(KM)		-0.8966936	
## Torque(Nm)		-0.8324169	
## Capacidade_Bateria(kwh)		-0.8194025	
## Autonomia(WLTP) (km)		-0.6276312	
## Distancia_Eixos(cm)		-0.5372283	
## Comprimento(cm)		-0.7348261	
## Largura(cm)		-0.4635592	
## Altura(cm)		0.3033080	
## Peso_Vazio(kg)		-0.7640688	
## Peso_Admissivel(kg)		-0.7048234	
## Carga_Max(kg)		-0.2552734	
## N_Assentos		0.1227695	
## N_Portas		0.2129011	
## Raio_Pneu(in)		-0.8274484	
## Vel_Maxima(kph)		-0.8870691	
## Mala(VDA) (L)		-0.5249693	
## Aceleracao_0_100(kph) (s)		1.0000000	
## Potencia_Max_Carregamento(kW)		-0.7727057	
## Consumo_Energia(KWh/100km)		-0.6268448	
## Desempenho(kg/hp)		0.9714489	
##	Potencia_Max_Carregamento(kW)		
## Preco_Minimo(PLN)		0.88771930	
## Potencia_Motor(KM)		0.85407706	
## Torque(Nm)		0.77080905	

## Capacidade_Bateria(kwh)	0.78890175	
## Autonomia(WLTP) (km)	0.55121621	
## Distancia_Eixos(cm)	0.62744351	
## Comprimento(cm)	0.76337541	
## Largura(cm)	0.49159807	
## Altura(cm)	-0.28568743	
## Peso_Vazio(kg)	0.75193494	
## Peso_Admissivel(kg)	0.74551349	
## Carga_Max(kg)	0.50986627	
## N_Assentos	0.02187164	
## N_Portas	-0.23628628	
## Raio_Pneu(in)	0.67054887	
## Vel_Maxima(kph)	0.92097511	
## Mala(VDA) (L)	0.53458220	
## Aceleracao_0_100(kph) (s)	-0.77270571	
## Potencia_Max_Carregamento(kW)	1.00000000	
## Consumo_Energia(KWh/100km)	0.70932838	
## Desempenho(kg/hp)	-0.73170880	
##	Consumo_Energia(KWh/100km)	Desempenho(kg/hp)
## Preco_Minimo(PLN)	0.7994474	-0.7962333
## Potencia_Motor(KM)	0.8246484	-0.8897054
## Torque(Nm)	0.8276390	-0.8184466
## Capacidade_Bateria(kwh)	0.7578657	-0.7852985
## Autonomia(WLTP) (km)	0.2744891	-0.6052826
## Distancia_Eixos(cm)	0.7164781	-0.5236950
## Comprimento(cm)	0.7210755	-0.6969048
## Largura(cm)	0.4468691	-0.4157315
## Altura(cm)	0.1328584	0.2363487
## Peso_Vazio(kg)	0.8619095	-0.7336646
## Peso_Admissivel(kg)	0.8747710	-0.6723195
## Carga_Max(kg)	0.6500265	-0.2916374
## N_Assentos	0.1836481	0.1512633
## N_Portas	-0.1723128	0.2836262
## Raio_Pneu(in)	0.5779794	-0.8292899
## Vel_Maxima(kph)	0.7304696	-0.8766168
## Mala(VDA) (L)	0.7721157	-0.5115341
## Aceleracao_0_100(kph) (s)	-0.6268448	0.9714489
## Potencia_Max_Carregamento(kW)	0.7093284	-0.7317088
## Consumo_Energia(KWh/100km)	1.0000000	-0.6085481
## Desempenho(kg/hp)	-0.6085481	1.0000000

Etapa 11.2: Criando o Mapa de Correlação com a Matrix acima

```
corrplot(interacao, method = "color")
```

Conclusão da Análise do Mapa de Correlação:

A maioria das variáveis possuem uma forte correlação tanto positiva quanto

negativa em relação a variável target (Consumo_Energia(kwh/100km)), no entanto

há 4 variáveis que apresentaram os menores valores (próximo a zero) de interação.

Etapa 12: Removendo as variáveis com nível de significância mais baixo.

```
df2$`Autonomia(WLTP)(km)` <- NULL
df2$`Altura(cm)` <- NULL
df2$N_Assentos <- NULL
df2$N_Portas <- NULL
```

Visualizando o Dataset

```
View(df2)
head(df2)
```

```
## # A tibble: 6 x 21
##   Fabricante Modelo   'Preco_Minimo(~' 'Potencia_Moto~' 'Torque(Nm)' Tipo_Freio
##   <chr>      <chr>      <dbl>          <dbl>          <dbl> <chr>
## 1 Audi      e-tron 5~      345700          360          664 disc (fro~
## 2 Audi      e-tron 5~      308400          313          540 disc (fro~
## 3 Audi      e-tron S~      414900          503          973 disc (fro~
## 4 Audi      e-tron S~      319700          313          540 disc (fro~
## 5 Audi      e-tron S~      357000          360          664 disc (fro~
## 6 Audi      e-tron S~      426200          503          973 disc (fro~
## # ... with 15 more variables: Transmissao <chr>,
## #   'Capacidade_Bateria(kwh)' <dbl>, 'Distancia_Eixos(cm)' <dbl>,
## #   'Comprimento(cm)' <dbl>, 'Largura(cm)' <dbl>, 'Peso_Vazio(kg)' <dbl>,
## #   'Peso_Admissivel(kg)' <dbl>, 'Carga_Max(kg)' <dbl>, 'Raio_Pneu(in)' <dbl>,
## #   'Vel_Maxima(kph)' <dbl>, 'Mala(VDA)(L)' <dbl>,
## #   'Aceleracao_0_100(kph)(s)' <dbl>, 'Potencia_Max_Carregamento(kW)' <dbl>,
## #   'Consumo_Energia(KWh/100km)' <dbl>, 'Desempenho(kg/hp)' <dbl>
```

```
dim(df2)
```

```
## [1] 42 21
```

Etapa 13: Análise por Fabricante

Etapa 13.1: Obtendo a Média de Consumo de Energia por Fabricante

```
group_1 <- df2 %>%
  group_by(Fabricante) %>%
  summarise(Media_Consumo = mean(`Consumo_Energia(KWh/100km)`)) %>%
  arrange(desc(Media_Consumo))
```

```
View(group_1)
head(group_1)
```

```
## # A tibble: 6 x 2
##   Fabricante   Media_Consumo
##   <chr>         <dbl>
## 1 Citroën       25.2
## 2 Audi          25.0
## 3 Porsche       24.4
## 4 Mercedes-Benz 21.8
## 5 Jaguar        21.2
## 6 Nissan        17.8
```

Conclusão da análise acima:

Os carros da fabricante Citroën possuem a média de consumo de combustível em

25.2(KWh/100km), enquanto o menor consumo é da fabricante Mazda, 15.5(KWh/100km).

Etapa 13.2: Relação Consumo x Desempenho

```
group_2 <- df2 %>%
  group_by(Fabricante) %>%
  select(Fabricante, Modelo, `Preco_Minimo(PLN)`, `Consumo_Energia(KWh/100km)`, `Desempenho(kg/hp)`) %>%
  arrange(desc(`Consumo_Energia(KWh/100km)`))
```

```
head(group_2)
```

```
## # A tibble: 6 x 5
## # Groups:   Fabricante [3]
##   Fabricante Modelo      `Preco_Minimo(~` `Consumo_Energ~` `Desempenho(kg~`
##   <chr>      <chr>          <dbl>          <dbl>          <dbl>
## 1 Audi       e-tron S quattro      414900          27.6           4.00
## 2 Audi       e-tron Sportbac~      426200          27.2           4.00
## 3 Citroën    ë-Spacetourer (~      215400          25.2          10.8
## 4 Porsche    Taycan Turbo S        794000          25.1           2.83
## 5 Porsche    Taycan Turbo          653000          24.8           2.84
## 6 Audi       e-tron 55 quatt~      345700          24.4           5.31
```

```
tail(group_2)
```

```
## # A tibble: 6 x 5
## # Groups:   Fabricante [4]
##   Fabricante Modelo      `Preco_Minimo(~` `Consumo_Energ~` `Desempenho(kg~`
##   <chr>      <chr>          <dbl>          <dbl>          <dbl>
## 1 Hyundai    Kona electric 3~      154400          15            8.42
## 2 Mazda      MX-30             142900          14.5           8.46
## 3 BMW        i3s               184200          14.3           5.92
## 4 Volkswagen e-up!           97990          14            11.1
## 5 Hyundai    Ioniq electric      184500          13.8           8.37
## 6 BMW        i3                169700          13.1           6.32
```

Conclusão da análise acima:

Os carros da Audi lidera as duas primeiras posições do ranking com o maior

consumo de energia e com um desempenho igual a 4.00(kg/hp) para os seus carros.

Enquanto a BMW i3 é a última do ranking, mas com um equilíbrio maior entre

o consumo de energia e o desempenho do carro. Olhando apenas para esses dados,

e comparando as duas marcas, podemos afirmar que o melhor custo benefício entre

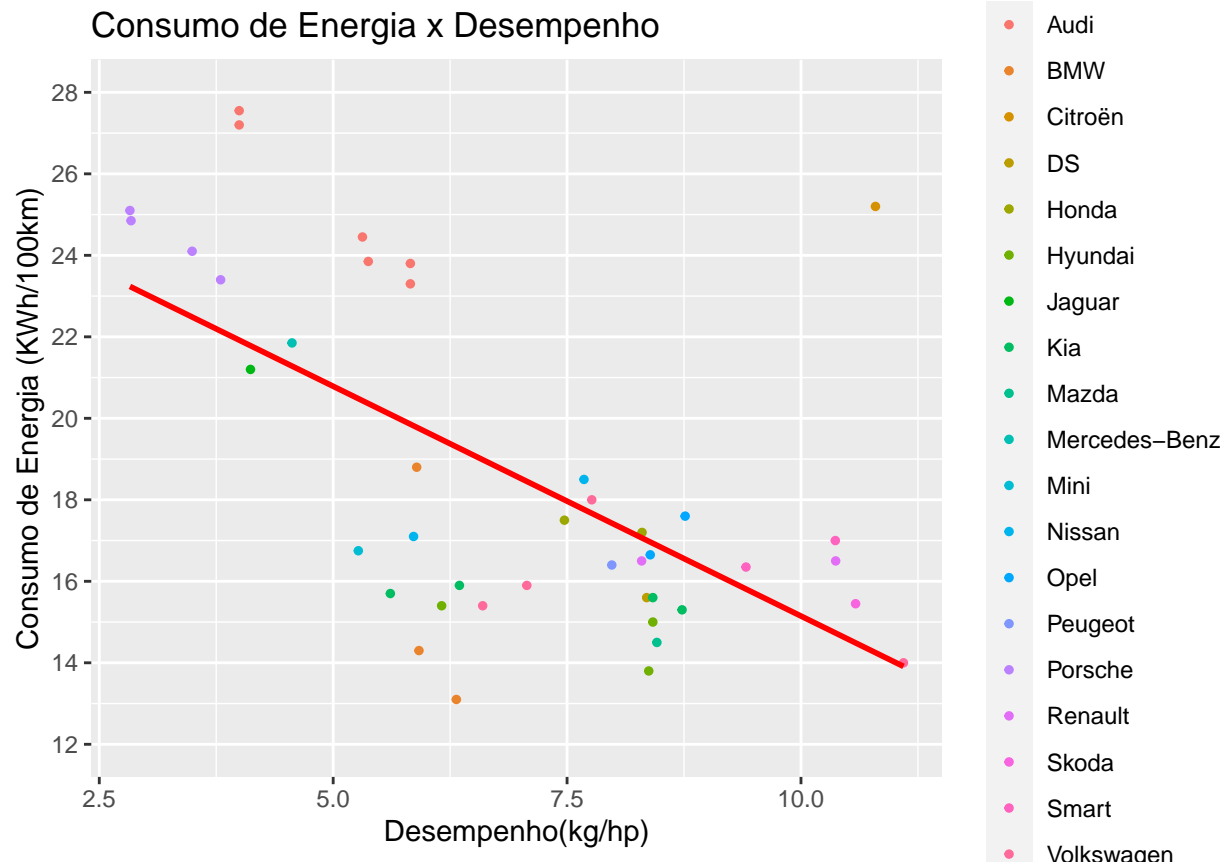
os carros Audi e-tron 55 quattro, Audi e-tron 55 Sportback S quattro e a BMW i3

fica com a BMW.

Etapa 13.3: Gráfico 1 - Consumo de Energia x Desempenho

```
ggplot(group_2, aes(x = `Desempenho(kg/hp)`, y = `Consumo_Energia(KWh/100km)` ,  
                    colour = (Fabricante))) + geom_point(size = 1) +  
  scale_y_continuous(limits = c(12,28), breaks = seq(12,29,2)) +  
  geom_smooth(method = lm , color = "red", se = FALSE) +  
  ggtitle("Consumo de Energia x Desempenho") +  
  theme(legend.position = "right") +  
  xlab ("Desempenho(kg/hp)") +  
  ylab ("Consumo de Energia (KWh/100km)")
```

```
## 'geom_smooth()' using formula 'y ~ x'
```



Conclusão do Gráfico acima:

O gráfico confirma a informação do Mapa de Correlação, existe uma forte

correlação negativa entre as duas variáveis, a medida que o desempenho do

carro aumenta o consumo de energia do veículo diminui.

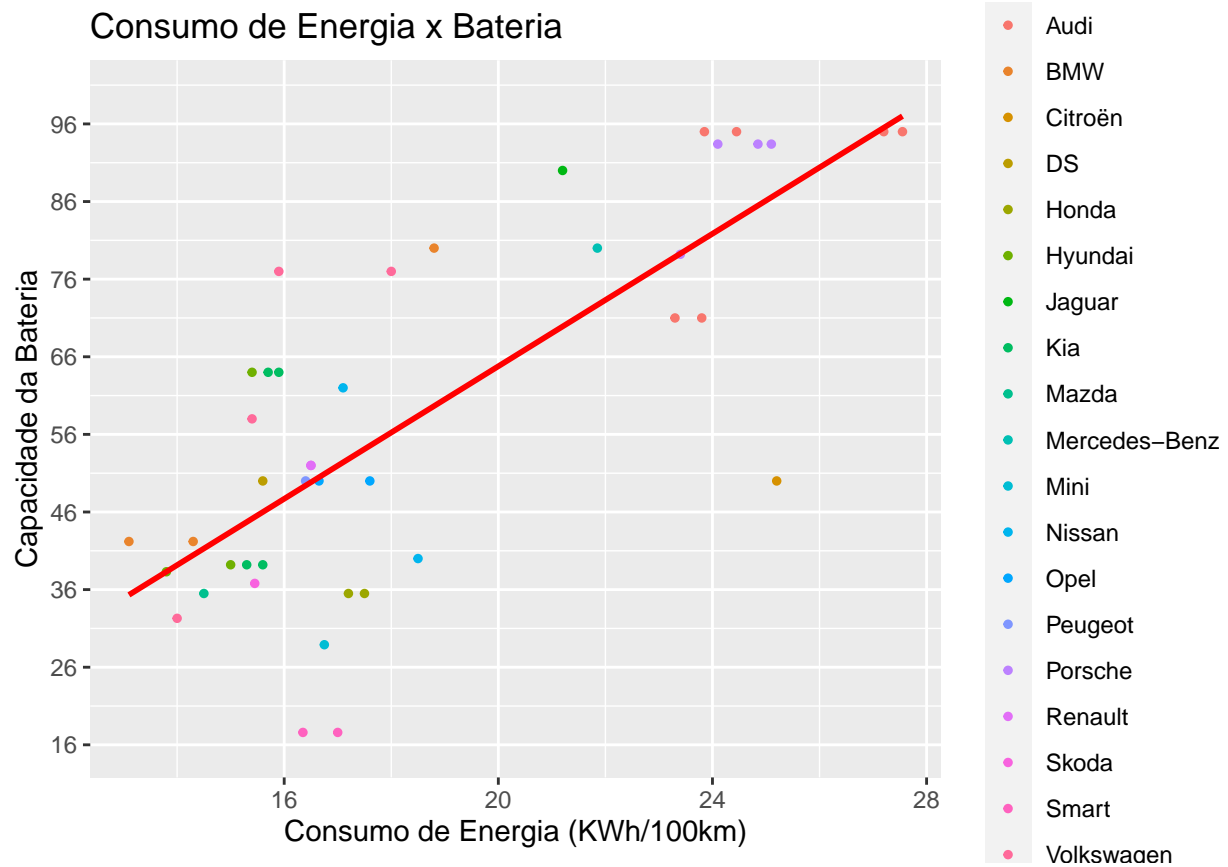
Etapa 13.4: Gráfico 2 - Consumo de Energia x Bateria

```
summary(df2$`Capacidade_Bateria(kwh)`)
```

##	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
##	17.60	39.20	52.00	58.84	78.65	95.00

```
ggplot(df2, aes(x = `Consumo_Energia(KWh/100km)`, y = `Capacidade_Bateria(kwh)`,
  colour = (Fabricante))) + geom_point(size = 1) +
  scale_y_continuous(limits = c(16,100), breaks = seq(16,96,10)) +
  geom_smooth(method = lm , color = "red", se = FALSE) +
  ggtitle("Consumo de Energia x Bateria") +
  theme(legend.position = "right") +
  xlab ("Consumo de Energia (KWh/100km)") +
  ylab ("Capacidade da Bateria")
```

'geom_smooth()' using formula 'y ~ x'



Conclusão Gráfico 2:

O gráfico mostra que há uma forte correlação positiva, o que confirma que

para um consumo maior de energia o carro tem que ter uma bateria com maior

capacidade.

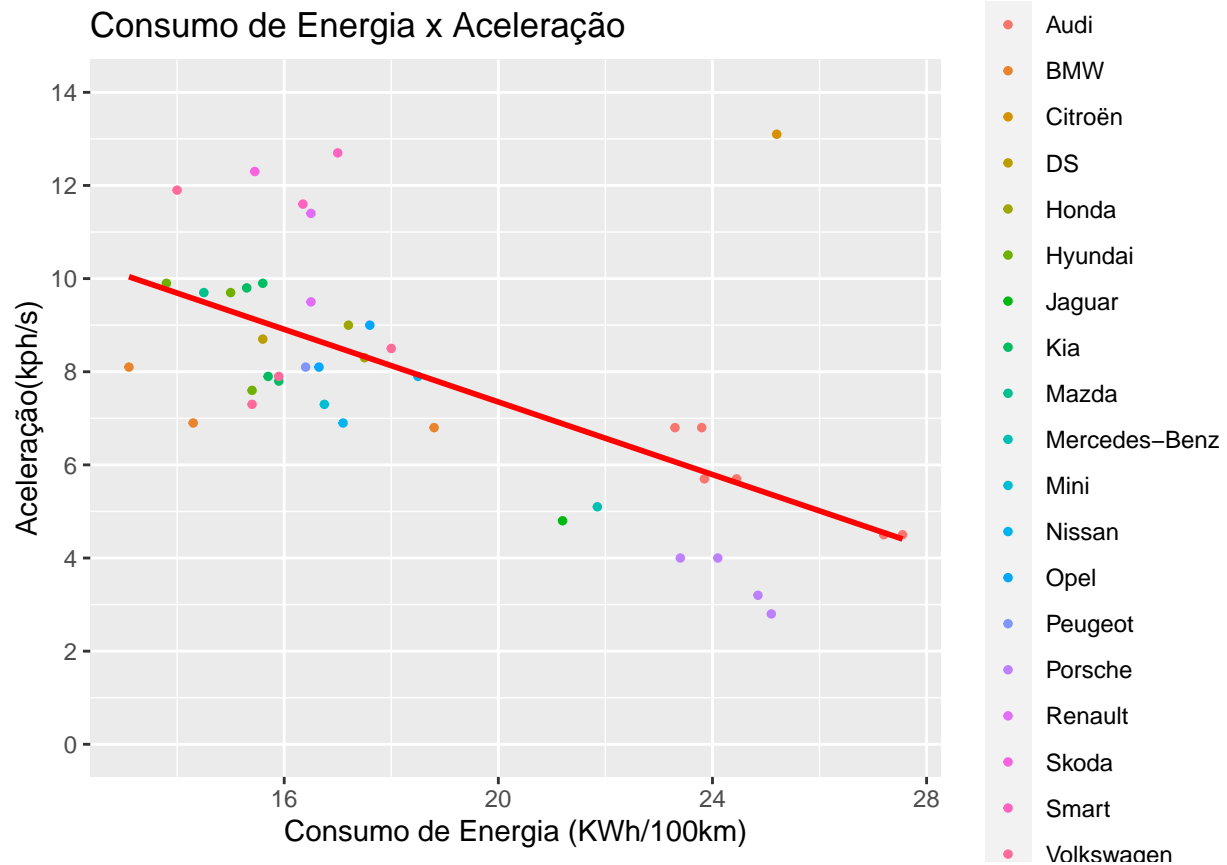
Etapa 13.5: Gráfico 3 - Consumo de Energia x Aceleração

```
summary(df2$`Aceleracao_0_100(kph)(s)`)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    2.800   6.800   7.900   7.893   9.650  13.100
```

```
ggplot(df2, aes(x = `Consumo_Energia(KWh/100km)`, y = `Aceleracao_0_100(kph)(s)`,
               colour = (Fabricante))) + geom_point(size = 1) +
  scale_y_continuous(limits = c(0,14), breaks = seq(0,14,2)) +
  geom_smooth(method = lm, color = "red", se = FALSE) +
  ggtitle("Consumo de Energia x Aceleração") +
  theme(legend.position = "right") +
  xlab ("Consumo de Energia (KWh/100km)") +
  ylab ("Aceleração(kph/s)")
```

```
## 'geom_smooth()' using formula 'y ~ x'
```



Conclusão Gráfico 3:

O gráfico mostra que quanto maior a aceleração do veículo menor é o consumo de

energia. Quanto maior a massa, menor é a aceleração e consequentemente a

velocidade.

Etapa 13.5: Gráficos Auxiliares para ajudar a entender a distribuição dos dados

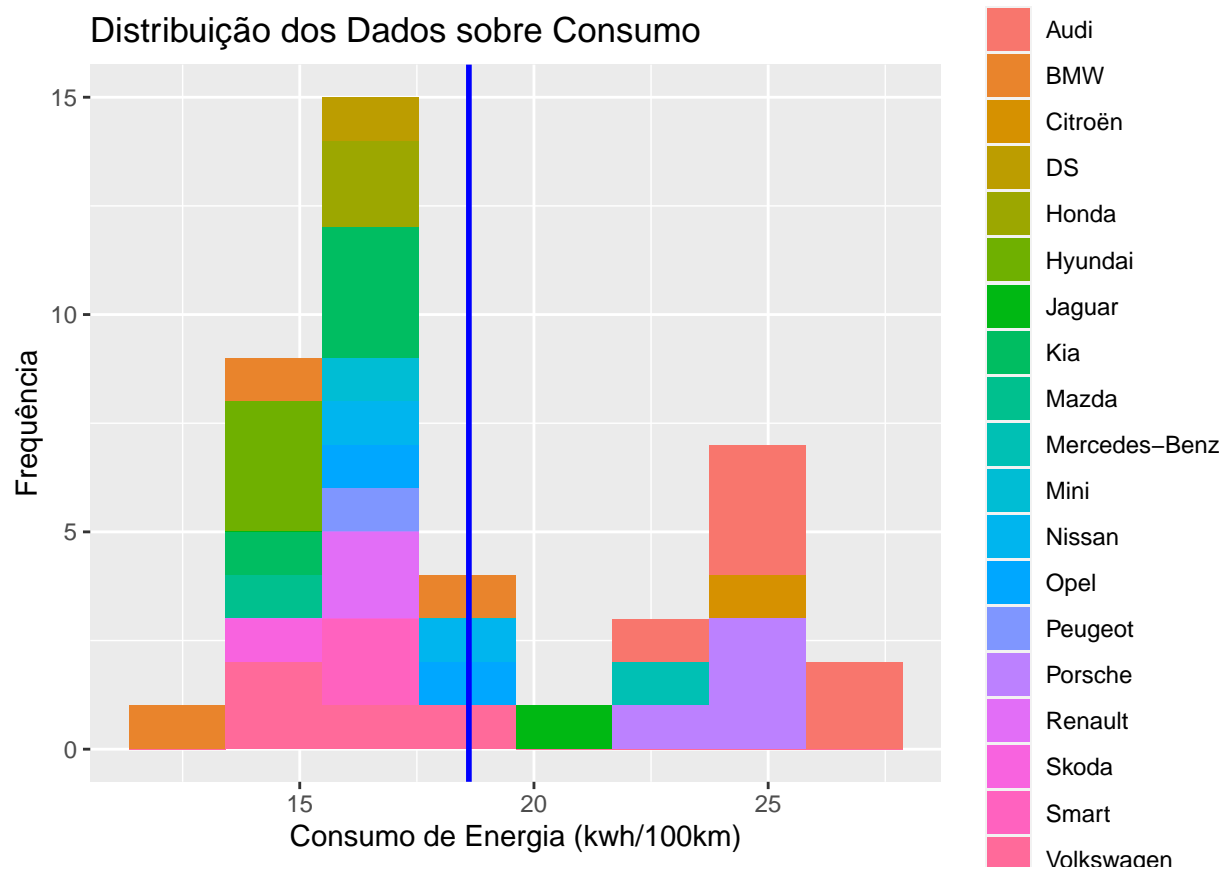
na variável target.

```
summary(df2$`Consumo_Energia(KWh/100km)`)
```


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

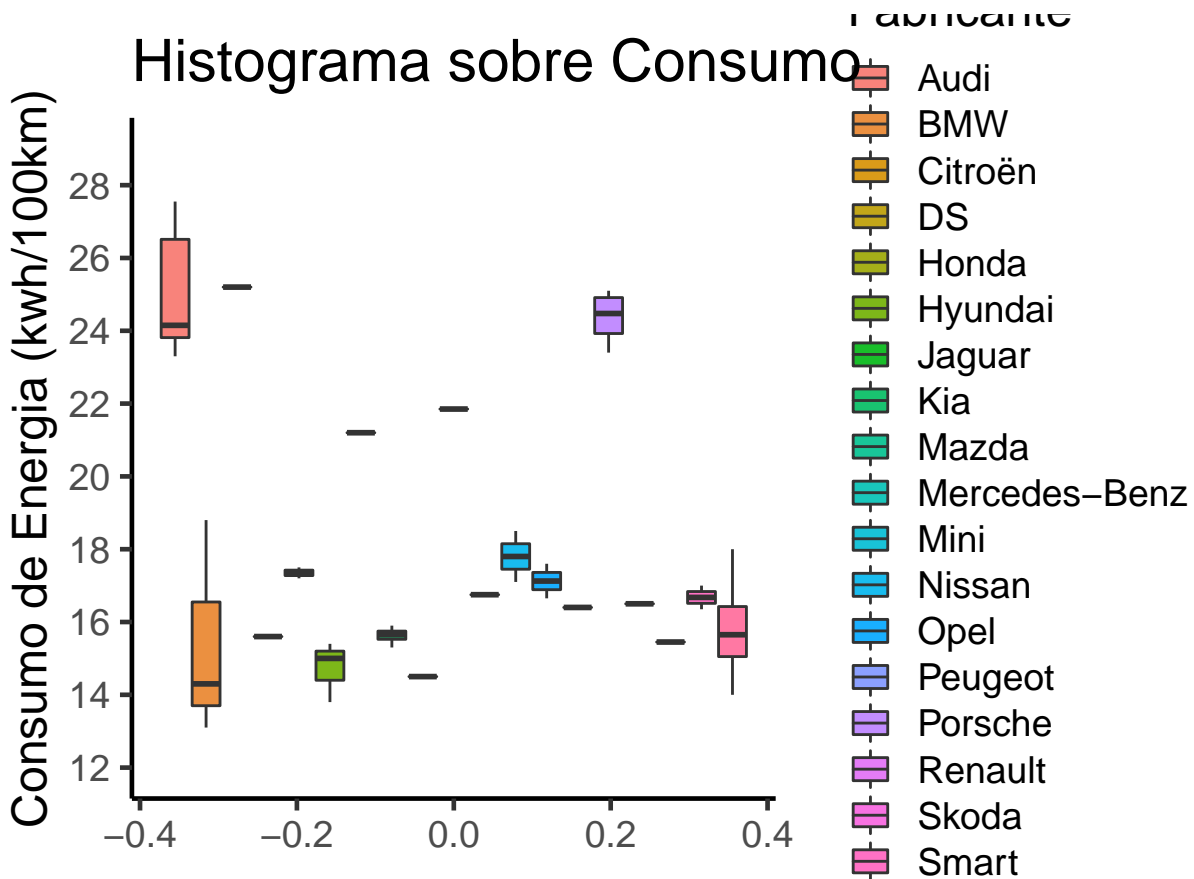
Etapa 13.5.1: Gráfico 4 - Histograma de Distribuição

```
ggplot(df2, aes(x = `Consumo_Energia(KWh/100km)`, fill = Fabricante)) +
  geom_histogram(bins = 8) +
  geom_vline(aes(xintercept=mean(`Consumo_Energia(KWh/100km)`)), color="blue", linetype="solid", size=1) +
  ggtitle("Distribuição dos Dados sobre Consumo") +
  xlab("Consumo de Energia (kwh/100km)") + ylab("Frequência")
```



Etapa 13.5.2: Gráfico 5 - BoxPlot do Consumo de Energia dos Carros

```
ggplot(df2, aes(y = `Consumo_Energia(KWh/100km)`, fill = Fabricante)) +
  geom_boxplot(show.legend = T, alpha = .9) +
  scale_y_continuous(limits = c(12,29), breaks = seq(12,29,2)) +
  ggtitle("Histograma sobre Consumo") +
  theme_classic(base_size = 18) +
  ylab("Consumo de Energia (kwh/100km)")
```



Análise Preditiva

Etapa 14: Divisão dos dados em Treino e Teste

Observação: Para criação do Modelo, será utilizado o conjunto de dados criado

anteriormente com apenas valores numéricos, “dados_numericos”.

```
View(dados_numericos)
```

Etapa 14.1: Removendo as variáveis com valor de significância mais próximo de zero

```
dados_numericos$`Autonomia(WLTP)(km)` <- NULL
dados_numericos$`Altura(cm)` <- NULL
```

```
dados_numericos$N_Assentos <- NULL
dados_numericos$N_Portas <- NULL
```

```
View(dados_numericos)
head(dados_numericos)
```

```
## # A tibble: 6 x 17
##   'Preco_Minimo(PLN)' 'Potencia_Motor(KM)' 'Torque(Nm)' 'Capacidade_Bateria(kw~'
##           <dbl>           <dbl>           <dbl>           <dbl>
## 1           345700           360           664           95
## 2           308400           313           540           71
## 3           414900           503           973           95
## 4           319700           313           540           71
## 5           357000           360           664           95
## 6           426200           503           973           95
## # ... with 13 more variables: 'Distancia_Eixos(cm)' <dbl>,
## #   'Comprimento(cm)' <dbl>, 'Largura(cm)' <dbl>, 'Peso_Vazio(kg)' <dbl>,
## #   'Peso_Admissivel(kg)' <dbl>, 'Carga_Max(kg)' <dbl>, 'Raio_Pneu(in)' <dbl>,
## #   'Vel_Maxima(kph)' <dbl>, 'Mala(VDA)(L)' <dbl>,
## #   'Aceleracao_0_100(kph)(s)' <dbl>, 'Potencia_Max_Carregamento(kW)' <dbl>,
## #   'Consumo_Energia(KWh/100km)' <dbl>, 'Desempenho(kg/hp)' <dbl>
```

Etapa 14.2: Divisão do dataset em dados de Treino e Testee

```
set.seed(480)
split = sample.split(dados_numericos$`Consumo_Energia(KWh/100km)` , SplitRatio = 0.70)
```

Dados de Treino

```
dados_treino = subset(dados_numericos, split == TRUE)
View(dados_treino)
head(dados_treino)
```

```
## # A tibble: 6 x 17
##   'Preco_Minimo(PLN)' 'Potencia_Motor(KM)' 'Torque(Nm)' 'Capacidade_Bateria(kw~'
##           <dbl>           <dbl>           <dbl>           <dbl>
## 1           345700           360           664           95
## 2           308400           313           540           71
## 3           414900           503           973           95
## 4           319700           313           540           71
## 5           357000           360           664           95
## 6           426200           503           973           95
## # ... with 13 more variables: 'Distancia_Eixos(cm)' <dbl>,
## #   'Comprimento(cm)' <dbl>, 'Largura(cm)' <dbl>, 'Peso_Vazio(kg)' <dbl>,
## #   'Peso_Admissivel(kg)' <dbl>, 'Carga_Max(kg)' <dbl>, 'Raio_Pneu(in)' <dbl>,
## #   'Vel_Maxima(kph)' <dbl>, 'Mala(VDA)(L)' <dbl>,
## #   'Aceleracao_0_100(kph)(s)' <dbl>, 'Potencia_Max_Carregamento(kW)' <dbl>,
## #   'Consumo_Energia(KWh/100km)' <dbl>, 'Desempenho(kg/hp)' <dbl>
```

```
dim(dados_treino)
```

```
## [1] 29 17
```

Dados de Teste

```
dados_teste = subset(dados_numericos, split == FALSE)
View(dados_teste)
head(dados_teste)
```

```
## # A tibble: 6 x 17
##   'Preco_Minimo(PLN)' 'Potencia_Motor(KM)' 'Torque(Nm)' 'Capacidade_Bateria(kw~'
##   <dbl>                <dbl>          <dbl>          <dbl>
## 1      165900           154            315            35.5
## 2      184500           136            295            38.3
## 3      154400           136            395            39.2
## 4      178400           204            395             64
## 5      359500           400            696             90
## 6      167990           204            395             64
## # ... with 13 more variables: 'Distancia_Eixos(cm)' <dbl>,
## #   'Comprimento(cm)' <dbl>, 'Largura(cm)' <dbl>, 'Peso_Vazio(kg)' <dbl>,
## #   'Peso_Admissivel(kg)' <dbl>, 'Carga_Max(kg)' <dbl>, 'Raio_Pneu(in)' <dbl>,
## #   'Vel_Maxima(kph)' <dbl>, 'Mala(VDA)(L)' <dbl>,
## #   'Aceleracao_0_100(kph)(s)' <dbl>, 'Potencia_Max_Carregamento(kW)' <dbl>,
## #   'Consumo_Energia(KWh/100km)' <dbl>, 'Desempenho(kg/hp)' <dbl>
```

```
dim(dados_teste)
```

```
## [1] 13 17
```

Etapa 14.2.1: Criando o Modelo Preditivo, versão 1:

```
modelo_v1 <- lm(`Consumo_Energia(KWh/100km)` ~ ., data = dados_treino)
```

Etapa 14.2.2: Visualizando os coeficientes

```
modelo_v1
```

```
##
## Call:
## lm(formula = 'Consumo_Energia(KWh/100km)' ~ ., data = dados_treino)
##
## Coefficients:
```

```
##              (Intercept)              'Preco_Minimo(PLN)'
##              3.114e+01              2.942e-06
##      'Potencia_Motor(KM)'              'Torque(Nm)'
##              3.513e-02              -3.543e-03
##      'Capacidade_Bateria(kwh)'              'Distancia_Eixos(cm)'
##              -9.506e-02              -1.035e-02
##      'Comprimento(cm)'              'Largura(cm)'
##              -2.164e-02              1.597e-05
##      'Peso_Vazio(kg)'              'Peso_Admissivel(kg)'
##              1.726e-02              -9.567e-03
##      'Carga_Max(kg)'              'Raio_Pneu(in)'
##              1.557e-02              -1.266e+00
##      'Vel_Maxima(kph)'              'Mala(VDA) (L)'
##              -7.050e-03              6.514e-03
##      'Aceleracao_0_100(kph) (s)'      'Potencia_Max_Carregamento(kW)'
##              -1.148e+00              -6.517e-03
##      'Desempenho(kg/hp)'
##              1.564e+00
```

```
summary(modelo_v1) # R-squared: 0.97
```

```
##
## Call:
## lm(formula = 'Consumo_Energia(KWh/100km)' ~ ., data = dados_treino)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.3944 -0.3394  0.1351  0.4976  1.0836
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      3.114e+01  1.259e+01   2.474  0.02929 *
## 'Preco_Minimo(PLN)' 2.942e-06  8.314e-06   0.354  0.72956
## 'Potencia_Motor(KM)' 3.513e-02  1.867e-02   1.881  0.08439 .
## 'Torque(Nm)'      -3.543e-03  5.948e-03  -0.596  0.56250
## 'Capacidade_Bateria(kwh)' -9.506e-02  3.925e-02  -2.422  0.03220 *
## 'Distancia_Eixos(cm)' -1.035e-02  5.147e-02  -0.201  0.84399
## 'Comprimento(cm)'  -2.164e-02  2.945e-02  -0.735  0.47657
## 'Largura(cm)'      1.597e-05  1.585e-02   0.001  0.99921
## 'Peso_Vazio(kg)'    1.726e-02  7.079e-03   2.438  0.03128 *
## 'Peso_Admissivel(kg)' -9.567e-03  5.734e-03  -1.669  0.12105
## 'Carga_Max(kg)'     1.557e-02  5.644e-03   2.760  0.01729 *
## 'Raio_Pneu(in)'    -1.266e+00  3.377e-01  -3.748  0.00278 **
## 'Vel_Maxima(kph)'   -7.050e-03  4.195e-02  -0.168  0.86933
## 'Mala(VDA) (L)'     6.514e-03  7.431e-03   0.877  0.39791
## 'Aceleracao_0_100(kph) (s)' -1.148e+00  6.205e-01  -1.850  0.08906 .
## 'Potencia_Max_Carregamento(kW)' -6.517e-03  1.411e-02  -0.462  0.65239
## 'Desempenho(kg/hp)'  1.564e+00  7.162e-01   2.184  0.04958 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.13 on 12 degrees of freedom
## Multiple R-squared:  0.9704, Adjusted R-squared:  0.9308
## F-statistic: 24.55 on 16 and 12 DF, p-value: 9.737e-07
```

Conclusão do modelo_v1:

O modelo apresentou um nível de precisão alto, alguns outros fatores que ajudam

a explicar isso, além de todo o processo de Limpeza e Transformação executado

durante a Análise Exploratória, são que o conjunto de dados é relativamente

pequeno, e muitos parâmetros com uma alta correlação com a variável target foram

usados para treinar o modelo. Devido a esses fatores vou permanecer somente com

essa versão de modelo de análise de regressão.

Etapa 15: Prevendo o Consumo com os dados de teste:

```
previsao_v1 <- predict(modelo_v1, dados_teste)
previsao_v1
```

```
##      1      2      3      4      5      6      7      8
## 15.76445 16.87019 15.15874 15.21028 20.75292 15.94504 18.04080 15.92902
##      9     10     11     12     13
## 16.16417 26.63509 16.32933 16.79481 19.10013
```

Etapa 15.1: Visualizando os valores previstos e observados para “previsao_v1”

```
resultados_v1 <- cbind(previsao_v1, dados_teste$`Consumo_Energia(KWh/100km)`)
```

```
colnames(resultados_v1) <- c('Previsto', 'Real')
```

```
resultados_v1 <- as.data.frame(resultados_v1)
```

```
View(resultados_v1)
head(resultados_v1)
```

```
##   Previsto Real
## 1 15.76445 17.5
## 2 16.87019 13.8
## 3 15.15874 15.0
## 4 15.21028 15.4
## 5 20.75292 21.2
## 6 15.94504 15.9
```

Etapa 16: Para fins de estudo e entender mais como o modelo criado acima se

comporta, abaixo vamos utilizar os dados que foram removidos do Dataset original

na Etapa 9, como um exemplo de validação para o modelo_v1.

```
View(removed_col)
head(removed_col)
```

```
## # A tibble: 6 x 25
##   Fabricante Modelo   'Preco_Minimo(~' 'Potencia_Moto~' 'Torque(Nm)' Tipo_Freio
##   <chr>         <chr>         <dbl>         <dbl>         <dbl> <chr>
## 1 Citroën      ë-C4             125000         136           260 disc (fro~
## 2 Peugeot      e-2008           149400         136           260 disc (fro~
## 3 Tesla        Model 3 ~         195490         285           450 disc (fro~
## 4 Tesla        Model 3 ~         235490         372           510 disc (fro~
## 5 Tesla        Model 3 ~         260490         480           639 disc (fro~
## 6 Tesla        Model S ~         368990         525           755 disc (fro~
## # ... with 19 more variables: Transmissao <chr>,
## #   'Capacidade_Bateria(kwh)' <dbl>, 'Autonomia(WLTP)(km)' <dbl>,
## #   'Distancia_Eixos(cm)' <dbl>, 'Comprimento(cm)' <dbl>, 'Largura(cm)' <dbl>,
## #   'Altura(cm)' <dbl>, 'Peso_Vazio(kg)' <dbl>, 'Peso_Admissivel(kg)' <dbl>,
## #   'Carga_Max(kg)' <dbl>, N_Assentos <dbl>, N_Portas <dbl>,
## #   'Raio_Pneu(in)' <dbl>, 'Vel_Maxima(kph)' <dbl>, 'Mala(VDA)(L)' <dbl>,
## #   'Aceleracao_0_100(kph)(s)' <dbl>, ...
```

Etapa 16.1: Criando uma nova variável

```
validacao <- removed_col
```

Etapa 16.2: Removendo o objeto anterior para liberar memória RAM

```
rm(removed_col)
```

Etapa 16.3: Selecionando apenas as colunas numéricas

```
variaveis_numericas_2 <- sapply(validacao, is.numeric)
validacao_numericos <- validacao[variaveis_numericas_2]

View(validacao_numericos)
head(validacao_numericos)
```

```
## # A tibble: 6 x 21
##   'Preco_Minimo(PLN)' 'Potencia_Motor(KM)' 'Torque(Nm)' 'Capacidade_Bateria(kw-'
##   <dbl> <dbl> <dbl> <dbl>
## 1 125000 136 260 50
## 2 149400 136 260 50
## 3 195490 285 450 54
## 4 235490 372 510 75
## 5 260490 480 639 75
## 6 368990 525 755 100
## # ... with 17 more variables: 'Autonomia(WLTP)(km)' <dbl>,
## # 'Distancia_Eixos(cm)' <dbl>, 'Comprimento(cm)' <dbl>, 'Largura(cm)' <dbl>,
## # 'Altura(cm)' <dbl>, 'Peso_Vazio(kg)' <dbl>, 'Peso_Admissivel(kg)' <dbl>,
## # 'Carga_Max(kg)' <dbl>, N_Assentos <dbl>, N_Portas <dbl>,
## # 'Raio_Pneu(in)' <dbl>, 'Vel_Maxima(kph)' <dbl>, 'Mala(VDA)(L)' <dbl>,
## # 'Aceleracao_0_100(kph)(s)' <dbl>, 'Potencia_Max_Carregamento(kW)' <dbl>,
## # 'Consumo_Energia(KWh/100km)' <dbl>, 'Desempenho(kg/hp)' <dbl>
```

Etapa 16.2: Removendo o objeto anterior para liberar memória RAM

```
rm(validacao)
```

Etapa 16.4: Excluindo a variável Target, a mesma será prevista pelo modelo

```
validacao_numericos$`Consumo_Energia(KWh/100km)` <- NULL

View(validacao_numericos)
head(validacao_numericos)
```



```
## # A tibble: 6 x 20
##   'Preco_Minimo(PLN)' 'Potencia_Motor(KM)' 'Torque(Nm)' 'Capacidade_Bateria(kw~'
##           <dbl>           <dbl>           <dbl>           <dbl>
## 1           125000           136           260           50
## 2           149400           136           260           50
## 3           195490           285           450           54
## 4           235490           372           510           75
## 5           260490           480           639           75
## 6           368990           525           755          100
## # ... with 16 more variables: 'Autonomia(WLTP)(km)' <dbl>,
## #   'Distancia_Eixos(cm)' <dbl>, 'Comprimento(cm)' <dbl>, 'Largura(cm)' <dbl>,
## #   'Altura(cm)' <dbl>, 'Peso_Vazio(kg)' <dbl>, 'Peso_Admissivel(kg)' <dbl>,
## #   'Carga_Max(kg)' <dbl>, N_Assentos <dbl>, N_Portas <dbl>,
## #   'Raio_Pneu(in)' <dbl>, 'Vel_Maxima(kph)' <dbl>, 'Mala(VDA)(L)' <dbl>,
## #   'Aceleracao_0_100(kph)(s)' <dbl>, 'Potencia_Max_Carregamento(kW)' <dbl>,
## #   'Desempenho(kg/hp)' <dbl>
```

Etapa 16.5: Função que realiza a Imputação do valor da média, de cada coluna

com seus respectivos valores ausêntes

```
imputacao <- function(x){
  for (i in 1:ncol(x)) {
    medias <- mean(as.numeric(unlist(x[,i])), na.rm = TRUE)
    for (j in 1:nrow(x)){
      if (is.na(x[j,i]))
        x[j,i] <- medias
    }
  }
  return(x)
}
```

Etapa 16.6: Fazendo a chamada da função “Imputação” no Dataset

```
validacao_numericos <- imputacao(validacao_numericos)
class(validacao_numericos)
```

```
## [1] "tbl_df"      "tbl"        "data.frame"
```

```
View(validacao_numericos)
head(validacao_numericos)
```

```
## # A tibble: 6 x 20
##   'Preco_Minimo(PLN)' 'Potencia_Motor(KM)' 'Torque(Nm)' 'Capacidade_Bateria(kw~'
##           <dbl>           <dbl>           <dbl>           <dbl>
```

```
## 1      125000      136      260      50
## 2      149400      136      260      50
## 3      195490      285      450      54
## 4      235490      372      510      75
## 5      260490      480      639      75
## 6      368990      525      755     100
## # ... with 16 more variables: 'Autonomia(WLTP)(km)' <dbl>,
## #   'Distancia_Eixos(cm)' <dbl>, 'Comprimento(cm)' <dbl>, 'Largura(cm)' <dbl>,
## #   'Altura(cm)' <dbl>, 'Peso_Vazio(kg)' <dbl>, 'Peso_Admissivel(kg)' <dbl>,
## #   'Carga_Max(kg)' <dbl>, N_Assentos <dbl>, N_Portas <dbl>,
## #   'Raio_Pneu(in)' <dbl>, 'Vel_Maxima(kph)' <dbl>, 'Mala(VDA)(L)' <dbl>,
## #   'Aceleracao_0_100(kph)(s)' <dbl>, 'Potencia_Max_Carregamento(kW)' <dbl>,
## #   'Desempenho(kg/hp)' <dbl>
```

Etapa 16.7: Fazendo a Previsão no Dataset de “Validação”

```
previsao_v2 <- predict(modelo_v1, validacao_numericos)
previsao_v2
```

```
##      1      2      3      4      5      6      7      8
## 15.86255 19.87574 13.15940 18.70200 19.33314 30.50250 35.70793 30.45438
##      9     10     11
## 39.35111 32.69633 31.54377
```

Etapa 16.8: Incluindo a Previsão no Dataset

```
validacao_final <- cbind(validacao_numericos, previsao_v2)
colnames(validacao_final)[21] <- "Consumo_Previsto"
```

```
View(validacao_final)
head(validacao_final)
```

```
##   Preco_Minimo(PLN) Potencia_Motor(KM) Torque(Nm) Capacidade_Bateria(kwh)
## 1      125000      136      260      50
## 2      149400      136      260      50
## 3      195490      285      450      54
## 4      235490      372      510      75
## 5      260490      480      639      75
## 6      368990      525      755     100
##   Autonomia(WLTP)(km) Distancia_Eixos(cm) Comprimento(cm) Largura(cm)
## 1      350      266.7      435.4      180.0
## 2      320      260.5      430.0      177.0
## 3      430      287.5      469.0      193.0
## 4      580      287.5      469.0      193.0
## 5      567      287.5      469.0      193.0
## 6      652      296.0      497.9      196.4
##   Altura(cm) Peso_Vazio(kg) Peso_Admissivel(kg) Carga_Max(kg) N_Assentos
```

```
## 1      152.2      1541      2000.000      459.0000      5
## 2      153.0      1548      2583.333      660.6667      5
## 3      144.0      1626      2583.333      660.6667      5
## 4      144.0      1862      2583.333      660.6667      5
## 5      144.0      1862      2583.333      660.6667      5
## 6      144.5      2391      2583.333      660.6667      5
##      N_Portas Raio_Pneu(in) Vel_Maxima(kph) Mala(VDA)(L) Aceleracao_0_100(kph)(s)
## 1          5          16          150          380          9.5000
## 2          5          16          150          434          4.5625
## 3          5          18          225          425          5.6000
## 4          5          18          233          425          4.4000
## 5          5          20          261          425          3.3000
## 6          5          19          250          745          3.8000
##      Potencia_Max_Carregamento(kW) Desempenho(kg/hp) Consumo_Previsto
## 1                                100          8.449577          15.86255
## 2                                100          8.487959          19.87574
## 3                                150          4.254484          13.15940
## 4                                150          3.732570          18.70200
## 5                                150          2.892742          19.33314
## 6                                150          3.396186          30.50250
```

Exemplo

Abaixo uma tentativa de ilustrar a minha sugestão de como o dado final sobre a

Previsão do Consumo de Energia, poderia ser apresentado forma visual aos

tomadores de decisão.

Etapa 17: Data Frame dados de Teste + Consumo Previsto

```
teste_final <- cbind(dados_teste, previsao_v1)
colnames(teste_final)[18] <- "Consumo_Previsto"
```

```
View(teste_final)
head(teste_final)
```

```
##      Preco_Minimo(PLN) Potencia_Motor(KM) Torque(Nm) Capacidade_Bateria(kwh)
## 1          165900          154          315          35.5
## 2          184500          136          295          38.3
## 3          154400          136          395          39.2
## 4          178400          204          395          64.0
## 5          359500          400          696          90.0
## 6          167990          204          395          64.0
```

	Distancia_Eixos(cm)	Comprimento(cm)	Largura(cm)	Peso_Vazio(kg)		
## 1	253.8	389.4	175.2	1543		
## 2	270.0	447.0	182.0	1527		
## 3	260.0	418.0	180.0	1535		
## 4	260.0	418.0	180.0	1685		
## 5	299.0	468.2	201.1	2208		
## 6	270.0	437.5	180.5	1737		
	Peso_Admissivel(kg)	Carga_Max(kg)	Raio_Pneu(in)	Vel_Maxima(kph)	Mala(VDA)(L)	
## 1	1870	350	17	145	171	
## 2	1970	518	16	165	357	
## 3	2020	485	17	155	332	
## 4	2170	485	17	167	332	
## 5	2670	537	20	200	656	
## 6	2230	493	17	167	451	
	Aceleracao_0_100(kph)(s)	Potencia_Max_Carregamento(kW)				
## 1	8.3	100				
## 2	9.9	100				
## 3	9.7	100				
## 4	7.6	100				
## 5	4.8	100				
## 6	7.8	100				
	Consumo_Energia(KWh/100km)	Desempenho(kg/hp)	Consumo_Previsto			
## 1	17.5	7.471648	15.76445			
## 2	13.8	8.372812	16.87019			
## 3	15.0	8.416678	15.15874			
## 4	15.4	6.159436	15.21028			
## 5	21.2	4.116331	20.75292			
## 6	15.9	6.349520	15.94504			

Etapa 18: Obtendo os fabricantes dos carros com os valores previsto na

previsao_v1

```
previsao_fabricante <- left_join(df, teste_final)
```

```
## Joining, by = c("Preco_Minimo(PLN)", "Potencia_Motor(KM)", "Torque(Nm)",
## "Capacidade_Bateria(kwh)", "Distancia_Eixos(cm)", "Comprimento(cm)",
## "Largura(cm)", "Peso_Vazio(kg)", "Peso_Admissivel(kg)", "Carga_Max(kg)",
## "Raio_Pneu(in)", "Vel_Maxima(kph)", "Mala(VDA)(L)", "Aceleracao_0_100(kph)(s)",
## "Potencia_Max_Carregamento(kW)", "Consumo_Energia(KWh/100km)",
## "Desempenho(kg/hp)")
```

```
previsao_fabricante <- na.omit(previsao_fabricante)
```

```
View(previsao_fabricante)
head(previsao_fabricante)
```

```
## # A tibble: 6 x 26
```

```
## Fabricante Modelo 'Preco_Minimo(~' 'Potencia_Moto~' 'Torque(Nm)' Tipo_Freio
## <chr> <chr> <dbl> <dbl> <dbl> <chr>
## 1 Honda e Advance 165900 154 315 disc (fro~
## 2 Hyundai Ioniq el~ 184500 136 295 disc (fro~
## 3 Hyundai Kona ele~ 154400 136 395 disc (fro~
## 4 Hyundai Kona ele~ 178400 204 395 disc (fro~
## 5 Jaguar I-Pace 359500 400 696 disc (fro~
## 6 Kia e-Niro 6~ 167990 204 395 disc (fro~
## # ... with 20 more variables: Transmissao <chr>,
## # 'Capacidade_Bateria(kwh)' <dbl>, 'Autonomia(WLTP)(km)' <dbl>,
## # 'Distancia_Eixos(cm)' <dbl>, 'Comprimento(cm)' <dbl>, 'Largura(cm)' <dbl>,
## # 'Altura(cm)' <dbl>, 'Peso_Vazio(kg)' <dbl>, 'Peso_Admissivel(kg)' <dbl>,
## # 'Carga_Max(kg)' <dbl>, N_Assentos <dbl>, N_Portas <dbl>,
## # 'Raio_Pneu(in)' <dbl>, 'Vel_Maxima(kph)' <dbl>, 'Mala(VDA)(L)' <dbl>,
## # 'Aceleracao_0_100(kph)(s)' <dbl>, ...
```

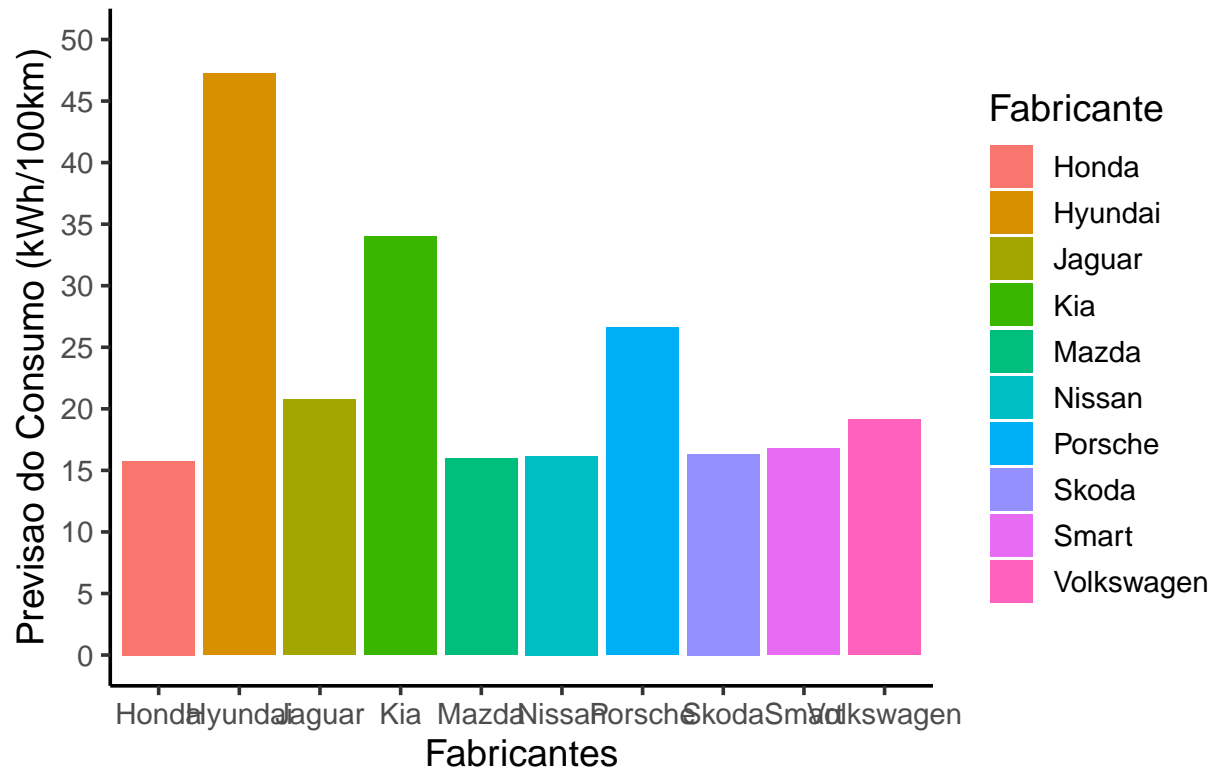
Etapa 19: Gráfico 6 - Consumo_Previsto x Fabricante, para os dados de Teste

```
summary(previsao_fabricante$Consumo_Previsto)
```

```
## Min. 1st Qu. Median Mean 3rd Qu. Max.
## 15.16 15.93 16.33 17.59 18.04 26.64
```

```
ggplot(previsao_fabricante, aes(y = Consumo_Previsto, x = Fabricante, fill = Fabricante)) +
  geom_bar(stat = "identity") + ggtitle("Consumo_Previsto por Fabricante - Dados de Teste") +
  scale_y_continuous(limits = c(0,50), breaks = seq(0,50,5)) +
  theme_classic(base_size = 14) + xlab("Fabricantes") +
  ylab("Previsao do Consumo (kWh/100km)")
```

Consumo_Previsto por Fabricante – Dados de Teste



Etapa 20: Conclusão

Na intenção de ilustrar parte do resultado do trabalho realizado, fiz o

Gráfico 6. O gráfico acima mostra a relação do Consumo de Energia Previsto por

Fabricantes, foram selecionados randomicamente na Etapa 14.2, a ilustração é

específica das previsões realizadas pelo modelo nos dados de Teste. O objetivo

era criar um modelo que fizesse as previsões do Consumo de Energia dos carros

foi criando com sucesso. Todo processo que foi desenvolvido da carga de dados

a previsão final, foi seguindo a metodologia aprendida, e aplicando os conceitos

ensinados nas aulas.

Fim