Projeto Final

Anderson Caio Emerson Tiago Vinicius

2023-07-07

# Projeto Integrado – final

A QuantumFinance está acompanhando um crescimento de inadimplência entre seus clientes e solicita a consultoria para desenvolver uma análise de inadimplência.

Os resultados apresentados pela consultoria mostram que o modelo de credit scoring implementado utilizou 70 variáveis e apresentou uma acurácia de 26% (R2 ajustado = 0.2593609). Consequentemente, a QuantumFinance está oferecendo crédito para mau pagador e deixando de oferecer para o bom pagador.

Para que a QuantumFinance tome decisões mais precisas sobre concessões de crédito, ela precisa aprimorar seu modelo de concessão de crédito.

Desafio: Melhorar a acurácia do modelo preditivo mediante uso do valor target disponível na base de dados train.csv.

## 3ª etapa: Conhecer os dados

A base de dados contém uma coluna chamada id que identifica exclusivamente cada linha, várias colunas identificadas por strings hexadecimais e um destino de coluna que gostaríamos que você previsse. As colunas que contêm hashes SHA-256 para seus valores representam variáveis categóricas, enquanto o restante das variáveis é numérica. O arquivo test.csv tem os mesmos nomes de coluna e tipos de dados que train.csv, mas está faltando a coluna da variável resposta. Não há valores ausentes ou problemas de corrupção de dados em nenhum desses arquivos. Não se preocupe com os significados das variáveis ou dos metadados - este é um conjunto de dados artificial.

| Variável | Tipo da variável (natureza) |
| --- | --- |
| 016399044a | Quantitativa discreta |
| 023c68873b | Qualitativa nominal |
| 0342faceb5 | Quantitativa discreta |
| 04e7268385 | Quantitativa discreta |
| 06888ceac9 | Qualitativa nominal |
| 072b7e8f27 | Quantitativa contínua |
| 087235d61e | Quantitativa discreta |
| 0b846350ef | Quantitativa contínua |
| 0e2ab0831c | Quantitativa contínua |
| 12eda2d982 | Quantitativa contínua |
| 136c1727c3 | Quantitativa contínua |
| 173b6590ae | Quantitativa contínua |
| 174825d438 | Quantitativa discreta |
| 1f222e3669 | Quantitativa contínua |
| 1f3058af83 | Quantitativa discreta |
| 1fa099bb01 | Quantitativa discreta |
| 20f1afc5c7 | Quantitativa contínua |
| 253eb5ef11 | Quantitativa contínua |
| 25bbf0e7e7 | Quantitativa discreta |
| 2719b72c0d | Quantitativa contínua |
| 298ed82b22 | Quantitativa contínua |
| 29bbd86997 | Quantitativa contínua |
| 2a457d15d9 | Quantitativa discreta |
| 2bc6ab42f7 | Quantitativa contínua |
| 2d7fe4693a | Quantitativa contínua |
| 2e874bc151 | Quantitativa contínua |
| 361f93f4d1 | Qualitativa nominal |
| 384bec5dd1 | Qualitativa nominal |
| 3df2300fa2 | Quantitativa contínua |
| 3e200bf766 | Quantitativa discreta |
| 3eb53ae932 | Quantitativa contínua |
| 435dec85e2 | Quantitativa contínua |
| 4468394575 | Quantitativa contínua |
| 49756d8e0f | Quantitativa contínua |
| 4fc17427c8 | Quantitativa contínua |
| 55907cc1de | Quantitativa contínua |
| 55cf3f7627 | Quantitativa contínua |
| 56371466d7 | Quantitativa discreta |
| 5b862c0a8f | Quantitativa discreta |
| 5f360995ef | Quantitativa discreta |
| 60ec1426ce | Quantitativa contínua |
| 63bcf89b1d | Quantitativa contínua |
| 6516422788 | Quantitativa contínua |
| 65aed7dc1f | Quantitativa discreta |
| 6db53d265a | Quantitativa discreta |
| 7734c0c22f | Quantitativa contínua |
| 7743f273c2 | Quantitativa contínua |
| 779d13189e | Quantitativa contínua |
| 77b3b41efa | Quantitativa contínua |
| 7841b6a5b1 | Quantitativa contínua |
| 789b5244a9 | Quantitativa contínua |
| 7925993f42 | Quantitativa contínua |
| 7cb7913148 | Qualitativa ordinal |
| 7fe6cb4c98 | Quantitativa contínua |
| 8311343404 | Quantitativa contínua |
| 87b982928b | Quantitativa contínua |
| 8a21502326 | Quantitativa contínua |
| 8c2e088a3d | Quantitativa discreta |
| 8d0606b150 | Qualitativa nominal |
| 8de0382f02 | Quantitativa discreta |
| 8f5f7c556a | Quantitativa discreta |
| 91145d159d | Qualitativa nominal |
| 96c30c7eef | Quantitativa contínua |
| 96e6f0be58 | Quantitativa contínua |
| 98475257f7 | Quantitativa contínua |
| 99d44111c9 | Quantitativa discreta |
| 9a575e82a4 | Qualitativa nominal |
| 9b6e0b36c2 | Quantitativa contínua |
| a14fd026ce | Quantitativa discreta |
| a24802caa5 | Quantitativa contínua |
| aa69c802b6 | Quantitativa contínua |
| abca7a848f | Quantitativa discreta |
| ac826f0013 | Quantitativa contínua |
| ae08d2297e | Quantitativa discreta |
| aee1e4fc85 | Quantitativa contínua |
| b4112a94a6 | Quantitativa contínua |
| b709f75447 | Quantitativa contínua |
| b835dfe10f | Qualitativa nominal |
| b9a487ac3c | Quantitativa contínua |
| ba54a2a637 | Quantitativa contínua |
| bdf934caa7 | Quantitativa contínua |
| beb6e17af1 | Quantitativa discreta |
| c0c3df65b1 | Quantitativa discreta |
| c1b8ce2354 | Quantitativa discreta |
| c58f611921 | Quantitativa contínua |
| d035af6ffa | Quantitativa discreta |
| d2c775fa99 | Quantitativa discreta |
| d4d6566f9c | Quantitativa discreta |
| dcfcbc2ea1 | Quantitativa discreta |
| e0a0772df0 | Quantitativa contínua |
| e16e640635 | Qualitativa nominal |
| e5efa4d39a | Quantitativa contínua |
| e7ee22effb | Quantitativa discreta |
| e86a2190c1 | Quantitativa discreta |
| ea0f4a32e3 | Quantitativa contínua |
| ed7e658a27 | Quantitativa discreta |
| ee2ac696ff | Quantitativa contínua |
| f013b60e50 | Quantitativa contínua |
| f0a0febd35 | Quantitativa contínua |
| f1f0984934 | Qualitativa nominal |
| f66b98dd69 | Quantitativa contínua |
| fbf66c8021 | Quantitativa contínua |
| fdf8628ca7 | Quantitativa discreta |
| fe0318e273 | Quantitativa contínua |
| fe8cdd80ba | Quantitativa contínua |
| ffd1cdcfc1 | Quantitativa contínua |
| id | Id |
| target | Quantitativa contínua |

## 4ª etapa: Preencher o quadro conceitual estatístico.

| Variável | Tipo da variável (natureza) |
| --- | --- |
| 1. Plano Básico de Análise | 1. Importação dos dados. |
|  | 2. Limpeza e pré-processamento dos dados. |
|  | 3. Análise exploratória dos dados. |
|  | 4. Criação de Graficos e visualizações. |
|  | 5. Teste de hipóteses ou comparação de grupos. |
|  | 6. Ajuste de modelos estatísticos. |
|  | 7. Avaliação da qualidade do modelo. |
|  | 8. Interpretação dos resultados. |
|  | 9. Comunicação dos resultados de forma clara e concisa. |

# nao mostrar os resultados na notacao cientifica  
options(scipen = 999)

## 5ª etapa: Faça a análise descritiva das variáveis. Apresente os Graficos e as medidas resumos.

### Import de pacotes utilizados

library(tidyverse)  
library(ggplot2)  
library(summarytools)  
library(gmodels)  
library(dplyr)  
library(fastDummies)

### Leitura dos dados train.csv

library(readr)  
df <- read\_csv("./data/train.csv")

Os dados apresentam nomes de colunas dificeis de serem identificadas, então o primeiro passo foi converter todas as colunas das variáveis preditoras para x1,x2,x3,…,xn e criar um dicionario pra identificalas.

# Nomes das colunas que você quer renomear  
colunas\_para\_renomear <- setdiff(names(df), c("id", "target"))  
  
# Número de colunas para renomear  
num\_colunas <- length(colunas\_para\_renomear)  
  
# Inicializando o dicionário  
dicionario <- list()  
  
# Renomear as colunas para x1, x2, x3, ... e criar o dicionario  
for (i in 1:num\_colunas) {  
 colname <- colunas\_para\_renomear[i]  
 new\_colname <- paste("x", i, sep="")  
   
 # Adicionando ao dicionário  
 dicionario[[new\_colname]] <- colname  
   
 print(paste(colname, "->", new\_colname))  
   
 # Renomeando a coluna  
 names(df)[names(df) == colname] <- new\_colname  
}

## [1] "016399044a -> x1"  
## [1] "023c68873b -> x2"  
## [1] "0342faceb5 -> x3"  
## [1] "04e7268385 -> x4"  
## [1] "06888ceac9 -> x5"  
## [1] "072b7e8f27 -> x6"  
## [1] "087235d61e -> x7"  
## [1] "0b846350ef -> x8"  
## [1] "0e2ab0831c -> x9"  
## [1] "12eda2d982 -> x10"  
## [1] "136c1727c3 -> x11"  
## [1] "173b6590ae -> x12"  
## [1] "174825d438 -> x13"  
## [1] "1f222e3669 -> x14"  
## [1] "1f3058af83 -> x15"  
## [1] "1fa099bb01 -> x16"  
## [1] "20f1afc5c7 -> x17"  
## [1] "253eb5ef11 -> x18"  
## [1] "25bbf0e7e7 -> x19"  
## [1] "2719b72c0d -> x20"  
## [1] "298ed82b22 -> x21"  
## [1] "29bbd86997 -> x22"  
## [1] "2a457d15d9 -> x23"  
## [1] "2bc6ab42f7 -> x24"  
## [1] "2d7fe4693a -> x25"  
## [1] "2e874bc151 -> x26"  
## [1] "361f93f4d1 -> x27"  
## [1] "384bec5dd1 -> x28"  
## [1] "3df2300fa2 -> x29"  
## [1] "3e200bf766 -> x30"  
## [1] "3eb53ae932 -> x31"  
## [1] "435dec85e2 -> x32"  
## [1] "4468394575 -> x33"  
## [1] "49756d8e0f -> x34"  
## [1] "4fc17427c8 -> x35"  
## [1] "55907cc1de -> x36"  
## [1] "55cf3f7627 -> x37"  
## [1] "56371466d7 -> x38"  
## [1] "5b862c0a8f -> x39"  
## [1] "5f360995ef -> x40"  
## [1] "60ec1426ce -> x41"  
## [1] "63bcf89b1d -> x42"  
## [1] "6516422788 -> x43"  
## [1] "65aed7dc1f -> x44"  
## [1] "6db53d265a -> x45"  
## [1] "7734c0c22f -> x46"  
## [1] "7743f273c2 -> x47"  
## [1] "779d13189e -> x48"  
## [1] "77b3b41efa -> x49"  
## [1] "7841b6a5b1 -> x50"  
## [1] "789b5244a9 -> x51"  
## [1] "7925993f42 -> x52"  
## [1] "7cb7913148 -> x53"  
## [1] "7fe6cb4c98 -> x54"  
## [1] "8311343404 -> x55"  
## [1] "87b982928b -> x56"  
## [1] "8a21502326 -> x57"  
## [1] "8c2e088a3d -> x58"  
## [1] "8d0606b150 -> x59"  
## [1] "8de0382f02 -> x60"  
## [1] "8f5f7c556a -> x61"  
## [1] "91145d159d -> x62"  
## [1] "96c30c7eef -> x63"  
## [1] "96e6f0be58 -> x64"  
## [1] "98475257f7 -> x65"  
## [1] "99d44111c9 -> x66"  
## [1] "9a575e82a4 -> x67"  
## [1] "9b6e0b36c2 -> x68"  
## [1] "a14fd026ce -> x69"  
## [1] "a24802caa5 -> x70"  
## [1] "aa69c802b6 -> x71"  
## [1] "abca7a848f -> x72"  
## [1] "ac826f0013 -> x73"  
## [1] "ae08d2297e -> x74"  
## [1] "aee1e4fc85 -> x75"  
## [1] "b4112a94a6 -> x76"  
## [1] "b709f75447 -> x77"  
## [1] "b835dfe10f -> x78"  
## [1] "b9a487ac3c -> x79"  
## [1] "ba54a2a637 -> x80"  
## [1] "bdf934caa7 -> x81"  
## [1] "beb6e17af1 -> x82"  
## [1] "c0c3df65b1 -> x83"  
## [1] "c1b8ce2354 -> x84"  
## [1] "c58f611921 -> x85"  
## [1] "d035af6ffa -> x86"  
## [1] "d2c775fa99 -> x87"  
## [1] "d4d6566f9c -> x88"  
## [1] "dcfcbc2ea1 -> x89"  
## [1] "e0a0772df0 -> x90"  
## [1] "e16e640635 -> x91"  
## [1] "e5efa4d39a -> x92"  
## [1] "e7ee22effb -> x93"  
## [1] "e86a2190c1 -> x94"  
## [1] "ea0f4a32e3 -> x95"  
## [1] "ed7e658a27 -> x96"  
## [1] "ee2ac696ff -> x97"  
## [1] "f013b60e50 -> x98"  
## [1] "f0a0febd35 -> x99"  
## [1] "f1f0984934 -> x100"  
## [1] "f66b98dd69 -> x101"  
## [1] "fbf66c8021 -> x102"  
## [1] "fdf8628ca7 -> x103"  
## [1] "fe0318e273 -> x104"  
## [1] "fe8cdd80ba -> x105"  
## [1] "ffd1cdcfc1 -> x106"

Durante nossa analise identificamos todas as variaveis qualitativas e as transformamos em variaveis do tipo factor.

df$x2 <- factor(df$x2)  
df$x5 <- factor(df$x5)  
df$x27 <- factor(df$x27)  
df$x28 <- factor(df$x28)  
df$x53 <- factor(df$x53, ordered=TRUE)  
df$x59 <- factor(df$x59)  
df$x62 <- factor(df$x62)  
df$x67 <- factor(df$x67)  
df$x78 <- factor(df$x78)  
df$x91 <- factor(df$x91)  
df$x100 <- factor(df$x100)

Com as variaveis devidamente transformadas fazemos a analise descritiva(Conhecer as variaveis):

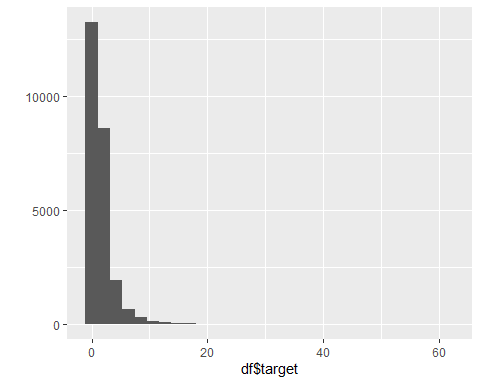
# medidas resumo  
summary(df)

## x1 x2 x3   
## Min. :6129 369c07e037bdf9071badfa06c0f6ef4e: 59 Min. :5040   
## 1st Qu.:6398 c0f379d9af329b0213f319e69a97c4a6: 57 1st Qu.:5292   
## Median :6452 8f41b2d9ac870e4e2211b7f8c77b8015: 56 Median :5339   
## Mean :6452 a61b6349dc22d541cb2bf767cba9a84b: 56 Mean :5339   
## 3rd Qu.:6507 ce46ea842bf36508eb7405ee3241a4c5: 55 3rd Qu.:5385   
## Max. :6805 58586eae2b39d07886d0906440df8c02: 54 Max. :5595   
## (Other) :24639   
## x4 x5 x6 x7   
## Min. :35132 0: 3505 Min. :-131.5 Min. :-33.00   
## 1st Qu.:35717 1:21471 1st Qu.: 279.5 1st Qu.: 10.00   
## Median :35843 Median : 573.0 Median : 24.00   
## Mean :35844 Mean : 789.3 Mean : 37.87   
## 3rd Qu.:35971 3rd Qu.:1074.7 3rd Qu.: 65.00   
## Max. :36617 Max. :7683.0 Max. :155.00   
##   
## x8 x9 x10 x11   
## Min. :-534285.1 Min. :-4801.982 Min. : 6.357 Min. : 0.210   
## 1st Qu.: -1.0 1st Qu.: -0.973 1st Qu.: 32.500 1st Qu.: 1.837   
## Median : 0.0 Median : 0.008 Median : 42.081 Median : 2.707   
## Mean : -21.2 Mean : -0.147 Mean : 44.045 Mean : 3.216   
## 3rd Qu.: 1.0 3rd Qu.: 1.006 3rd Qu.: 53.661 3rd Qu.: 4.016   
## Max. : 8511.2 Max. : 3263.127 Max. :136.076 Max. :31.790   
##   
## x12 x13 x14 x15   
## Min. :0.2129 Min. : 4 Min. :15.23 Min. :34686   
## 1st Qu.:0.8380 1st Qu.:12398 1st Qu.:21.87 1st Qu.:35379   
## Median :0.9485 Median :25072 Median :28.33 Median :35506   
## Mean :0.9349 Mean :25004 Mean :28.35 Mean :35507   
## 3rd Qu.:1.0465 3rd Qu.:37493 3rd Qu.:34.91 3rd Qu.:35634   
## Max. :1.3967 Max. :49952 Max. :41.37 Max. :36323   
##   
## x16 x17 x18 x19   
## Min. :-64.00 Min. : 0.0002 Min. :0.0000029 Min. :-91.00   
## 1st Qu.: 16.00 1st Qu.: 2.4376 1st Qu.:0.2247037 1st Qu.: 17.00   
## Median : 28.00 Median : 5.8543 Median :0.4995968 Median : 23.00   
## Mean : 41.42 Mean : 8.4452 Mean :0.5002357 Mean : 42.27   
## 3rd Qu.: 63.00 3rd Qu.:11.7095 3rd Qu.:0.7757522 3rd Qu.: 66.00   
## Max. :212.00 Max. :86.1792 Max. :0.9999775 Max. :223.00   
##   
## x20 x21 x22 x23   
## Min. : 127.2 Min. :0.003121 Min. :0.002308 Min. :-116.00   
## 1st Qu.: 15913.4 1st Qu.:0.321315 1st Qu.:0.315105 1st Qu.: 17.00   
## Median : 63080.5 Median :0.503397 Median :0.495207 Median : 21.00   
## Mean : 62287.1 Mean :0.500901 Mean :0.499155 Mean : 36.18   
## 3rd Qu.:108226.1 3rd Qu.:0.680018 3rd Qu.:0.683253 3rd Qu.: 54.00   
## Max. :123826.3 Max. :0.997309 Max. :0.999233 Max. : 212.00   
##   
## x24 x25 x26   
## Min. : 24.66 Min. : 0.03204 Min. :0.07276   
## 1st Qu.: 92.60 1st Qu.: 1.99667 1st Qu.:0.68785   
## Median : 131.57 Median : 3.34972 Median :0.89379   
## Mean : 173.87 Mean : 3.91553 Mean :0.89808   
## 3rd Qu.: 202.79 3rd Qu.: 5.21012 3rd Qu.:1.10587   
## Max. :3254.31 Max. :20.37858 Max. :2.03634   
##   
## x27 x28 x29   
## f06a110f1992f4b2ebb6bca6a484978a: 58 0: 2897 Min. : 58.05   
## ee9fc69b78ba54e1eb43e25d099bbe31: 53 1:22079 1st Qu.: 142.35   
## 0be236dfcfb635f559f232d7ca65f93c: 51 Median : 206.88   
## e4332210c6788b369e8cc7f54808dabb: 51 Mean : 281.52   
## 61e66ed9cf26784e514e6536dbf791c5: 49 3rd Qu.: 329.49   
## e55e3904c3dd4cae319de4913fad7004: 49 Max. :5538.49   
## (Other) :24665   
## x30 x31 x32 x33   
## Min. :37674 Min. :1.538 Min. :0.0000369 Min. : 3.062   
## 1st Qu.:38298 1st Qu.:2.468 1st Qu.:0.0737484 1st Qu.:13.876   
## Median :38431 Median :2.715 Median :0.1794271 Median :17.680   
## Mean :38431 Mean :2.743 Mean :0.2560269 Mean :18.338   
## 3rd Qu.:38565 3rd Qu.:2.988 3rd Qu.:0.3522771 3rd Qu.:22.026   
## Max. :39270 Max. :4.854 Max. :2.8260357 Max. :52.560   
##   
## x34 x35 x36 x37   
## Min. :0.0004708 Min. :-29454.38 Min. : 0.00014 Min. :0.0000017   
## 1st Qu.:0.3074208 1st Qu.: -0.96 1st Qu.: 2.80040 1st Qu.:1.0022405   
## Median :0.4999183 Median : 0.02 Median : 6.79340 Median :1.0065804   
## Mean :0.4980019 Mean : 0.22 Mean : 9.87359 Mean :0.9351881   
## 3rd Qu.:0.6885438 3rd Qu.: 1.03 3rd Qu.:13.60292 3rd Qu.:1.0142407   
## Max. :0.9995046 Max. : 33372.54 Max. :83.84800 Max. :1.1100283   
##   
## x38 x39 x40 x41   
## Min. :13225 Min. :4802 Min. :431.0 Min. :0.0515   
## 1st Qu.:13573 1st Qu.:5054 1st Qu.:513.0 1st Qu.:0.6988   
## Median :13652 Median :5102 Median :528.0 Median :0.9324   
## Mean :13652 Mean :5102 Mean :528.5 Mean :0.9420   
## 3rd Qu.:13730 3rd Qu.:5150 3rd Qu.:544.0 3rd Qu.:1.1684   
## Max. :14133 Max. :5389 Max. :632.0 Max. :2.3385   
##   
## x42 x43 x44 x45   
## Min. : 0.00014 Min. : 63.28 Min. :-96.00 Min. : 12   
## 1st Qu.: 1.19356 1st Qu.:202.02 1st Qu.: 17.00 1st Qu.:12626   
## Median : 2.88026 Median :227.86 Median : 24.00 Median :25150   
## Mean : 4.20579 Mean :235.18 Mean : 43.66 Mean :25040   
## 3rd Qu.: 5.79478 3rd Qu.:263.03 3rd Qu.: 69.00 3rd Qu.:37581   
## Max. :42.29624 Max. :469.58 Max. :236.00 Max. :49950   
##   
## x46 x47 x48 x49   
## Min. :-23.49 Min. : 0.001 Min. :-11613.256 Min. :-185.790   
## 1st Qu.: 25.88 1st Qu.: 1.993 1st Qu.: -0.986 1st Qu.: 2.912   
## Median : 49.86 Median : 4.796 Median : 0.005 Median : 39.806   
## Mean : 56.15 Mean : 6.920 Mean : 1.422 Mean : 39.999   
## 3rd Qu.: 78.86 3rd Qu.: 9.653 3rd Qu.: 0.996 3rd Qu.: 77.485   
## Max. :316.56 Max. :72.708 Max. : 15196.709 Max. : 253.983   
##   
## x50 x51 x52 x53   
## Min. :-154.9 Min. :0.1356 Min. : 0.00025 1 :9175   
## 1st Qu.: 928.7 1st Qu.:0.7698 1st Qu.: 1.96949 0 :8455   
## Median :2189.5 Median :0.9283 Median : 4.82701 2 :4917   
## Mean :2182.3 Mean :0.9157 Mean : 6.94611 3 :1830   
## 3rd Qu.:3424.0 3rd Qu.:1.0711 3rd Qu.: 9.67463 4 : 481   
## Max. :4508.7 Max. :1.5943 Max. :70.37153 5 : 99   
## (Other): 19   
## x54 x55 x56 x57   
## Min. : 0.00014 Min. : 0.06004 Min. :0.0000003 Min. : 34.29   
## 1st Qu.: 2.73093 1st Qu.: 2.13747 1st Qu.:0.1954873 1st Qu.: 611.57   
## Median : 6.53069 Median : 4.98332 Median :0.5057884 Median : 1250.03   
## Mean : 9.42799 Mean : 7.09229 Mean :0.5011737 Mean : 1652.19   
## 3rd Qu.:13.12237 3rd Qu.: 9.82829 3rd Qu.:0.8036556 3rd Qu.: 2270.48   
## Max. :91.36094 Max. :71.54517 Max. :0.9999999 Max. :12319.00   
##   
## x58 x59 x60   
## Min. :47421 117cd0c0a39f9b93448f6e73302671bd: 59 Min. : 2281   
## 1st Qu.:47475 fd076afc9ab10ead6bbada35a74869c5: 53 1st Qu.: 52250   
## Median :47483 0f5002bbbb17df40b636a4c4c50bd710: 51 Median : 124587   
## Mean :47482 76492e644fe01fb74d86d279ff588f4a: 51 Mean : 178125   
## 3rd Qu.:47491 79367a94cd8ad8a1386c608d0b150d2a: 51 3rd Qu.: 247637   
## Max. :47523 23ccdc7abca6c5cef5f3f5336e82116d: 50 Max. :1887315   
## (Other) :24661   
## x61 x62 x63   
## Min. :-83.00 33073762299a55d2ab54f847c8a7c921: 66 Min. :0.000126   
## 1st Qu.: 13.00 e256327c1844e3666894a63bef12f128: 63 1st Qu.:0.331263   
## Median : 19.00 9af1df22f4087751f683a99501ad6b98: 60 Median :0.721685   
## Mean : 37.43 254d6edd9704920df808f41b13e3549f: 59 Mean :0.968226   
## 3rd Qu.: 60.00 68c724d6abd5cd93b6f531ecad977f74: 59 3rd Qu.:1.346574   
## Max. :209.00 52fb248a94e8650cebfbbfb21a8b6c0b: 58 Max. :7.492790   
## (Other) :24611   
## x64 x65 x66 x67   
## Min. : 0.05532 Min. : 0.000087 Min. :-142657715 0: 1031   
## 1st Qu.: 1.44897 1st Qu.: 0.717958 1st Qu.: 28086 1:23945   
## Median : 2.70032 Median : 1.742367 Median : 38516   
## Mean : 4.16138 Mean : 2.529009 Mean : 33954   
## 3rd Qu.: 5.10120 3rd Qu.: 3.482941 3rd Qu.: 48969   
## Max. :106.57126 Max. :25.173468 Max. : 32543875   
##   
## x68 x69 x70 x71   
## Min. :-50247.20 Min. :-54.0 Min. :-5339.624 Min. :-26342.955   
## 1st Qu.: -1.01 1st Qu.: 13.0 1st Qu.: -1.038 1st Qu.: -0.996   
## Median : -0.01 Median : 25.0 Median : -0.009 Median : -0.002   
## Mean : -2.13 Mean : 39.4 Mean : 0.734 Mean : -0.750   
## 3rd Qu.: 0.98 3rd Qu.: 65.0 3rd Qu.: 0.992 3rd Qu.: 0.984   
## Max. : 3400.30 Max. :216.0 Max. :28989.428 Max. : 8367.628   
##   
## x72 x73 x74 x75   
## Min. : 4 Min. :0.9085 Min. : 1 Min. :0.0000   
## 1st Qu.:12618 1st Qu.:2.3719 1st Qu.:12298 1st Qu.:0.1415   
## Median :25091 Median :2.8712 Median :24920 Median :0.4902   
## Mean :25035 Mean :2.9878 Mean :24867 Mean :0.4971   
## 3rd Qu.:37528 3rd Qu.:3.4751 3rd Qu.:37432 3rd Qu.:0.8536   
## Max. :49951 Max. :9.8836 Max. :49952 Max. :1.0000   
##   
## x76 x77   
## Min. : 0.000428 Min. :-40148.62   
## 1st Qu.: 0.801365 1st Qu.: -0.98   
## Median : 1.962001 Median : 0.02   
## Mean : 2.821592 Mean : 1.52   
## 3rd Qu.: 3.937126 3rd Qu.: 1.01   
## Max. :30.254508 Max. : 84416.90   
##   
## x78 x79 x80   
## c6f6b3061e7ee6f986caf12f7beef509: 224 Min. :0.08084 Min. :-68361.97   
## 673390b6aa6ebef46d851ebbfd7c19a6: 221 1st Qu.:0.74711 1st Qu.: -1.01   
## 68ab2c192c6fd62d9de503be78cbd3fe: 219 Median :0.91691 Median : -0.01   
## 34b29495d27155762785160c1d44beaa: 215 Mean :0.90959 Mean : -1.84   
## 4661209cb3adaa375abf500a72b87574: 214 3rd Qu.:1.07929 3rd Qu.: 1.00   
## 9a2d2cb82d158f83ab721a316c899698: 209 Max. :1.74156 Max. : 33213.56   
## (Other) :23674   
## x81 x82 x83 x84   
## Min. : 0.0798 Min. : 1 Min. : 1 Min. : 45778999   
## 1st Qu.: 1.5496 1st Qu.:12748 1st Qu.:12895 1st Qu.: 73438011   
## Median : 2.7376 Median :25322 Median :25340 Median : 80764553   
## Mean : 3.9679 Mean :25167 Mean :25232 Mean : 81618138   
## 3rd Qu.: 4.8719 3rd Qu.:37621 3rd Qu.:37691 3rd Qu.: 88890375   
## Max. :67.1302 Max. :49952 Max. :49952 Max. :144378835   
##   
## x85 x86 x87 x88   
## Min. :-13521.150 Min. :-1269 Min. :2776 Min. : 670   
## 1st Qu.: -0.983 1st Qu.: 1532 1st Qu.:2941 1st Qu.: 164445   
## Median : 0.006 Median : 1984 Median :2975 Median : 397960   
## Mean : -0.426 Mean : 1987 Mean :2976 Mean : 567152   
## 3rd Qu.: 0.992 3rd Qu.: 2446 3rd Qu.:3010 3rd Qu.: 779489   
## Max. : 3080.032 Max. :13949 Max. :3179 Max. :6244488   
##   
## x89 x90 x91   
## Min. :38744 Min. :0.0000011 dfaeacc0ced1ca04fc8e0815cdb78525: 184   
## 1st Qu.:39349 1st Qu.:0.2103642 f949431d6f18e3b26f6fde1b20f070ec: 184   
## Median :39487 Median :0.5020075 876b7bc4a2ec297062ee531d9ba4a24f: 179   
## Mean :39485 Mean :0.5007384 250e6488c3a4b403cadb87b2866a50d8: 174   
## 3rd Qu.:39619 3rd Qu.:0.7893820 515ac2ee6867f341f68fa79372d17679: 174   
## Max. :40304 Max. :0.9999998 3e1273ad0cf5a5f27270b9540e046211: 171   
## (Other) :23910   
## x92 x93 x94 x95   
## Min. : -129.3 Min. :-56.0 Min. : 1 Min. :0.1411   
## 1st Qu.:13909.5 1st Qu.: 13.0 1st Qu.:12395 1st Qu.:0.7995   
## Median :21861.3 Median : 21.0 Median :24977 Median :0.9369   
## Mean :22035.8 Mean : 32.8 Mean :24912 Mean :0.9254   
## 3rd Qu.:30173.5 3rd Qu.: 51.0 3rd Qu.:37442 3rd Qu.:1.0612   
## Max. :46660.3 Max. :158.0 Max. :49952 Max. :1.5225   
##   
## x96 x97 x98 x99   
## Min. : 2514 Min. :0.001317 Min. :0.2541 Min. :-188.18   
## 1st Qu.:10312 1st Qu.:1.003929 1st Qu.:0.8434 1st Qu.: 25.61   
## Median :13186 Median :1.005198 Median :0.9534 Median : 69.13   
## Mean :14843 Mean :0.968233 Mean :0.9380 Mean : 69.24   
## 3rd Qu.:17814 3rd Qu.:1.006697 3rd Qu.:1.0466 3rd Qu.: 112.90   
## Max. :62001 Max. :1.017143 Max. :1.3503 Max. : 311.19   
##   
## x100 x101 x102   
## 6f2a574469db65c13448b693ad5aca1c: 106 Min. :0.0000022 Min. :-101.38   
## ac57568a3a435501ac356cad77a4d957: 103 1st Qu.:0.1444816 1st Qu.: 17.06   
## 7d1ab1e7b26fcbd91a972154f6fa860e: 100 Median :0.5829451 Median : 40.68   
## b1d448724055697b94a122878757a54a: 100 Mean :0.5755702 Mean : 40.71   
## c0e421345f4d099c646992fad48af409: 97 3rd Qu.:1.0025573 3rd Qu.: 64.86   
## dd19fd8cab2ba69b1d2271cbd0817ca0: 97 Max. :1.1458164 Max. : 179.20   
## (Other) :24373   
## x103 x104 x105 x106   
## Min. : 846.0 Min. :0.000259 Min. : 0.236 Min. : 0.3315   
## 1st Qu.: 977.0 1st Qu.:1.000519 1st Qu.:14.890 1st Qu.: 1.8503   
## Median : 997.0 Median :1.000597 Median :26.957 Median : 2.6993   
## Mean : 996.9 Mean :0.973738 Mean :27.027 Mean : 3.1630   
## 3rd Qu.:1017.0 3rd Qu.:1.000674 3rd Qu.:39.215 3rd Qu.: 3.9484   
## Max. :1111.0 Max. :1.001109 Max. :61.403 Max. :24.3726   
##   
## id target   
## Min. : 0 Min. :-0.2398   
## 1st Qu.: 6244 1st Qu.: 0.5031   
## Median :12488 Median : 0.9838   
## Mean :12488 Mean : 1.6609   
## 3rd Qu.:18731 3rd Qu.: 1.9680   
## Max. :24975 Max. :61.3261   
##

### Grafico univariado da target

# grafico de uma variavel  
qplot(x=df$target)

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

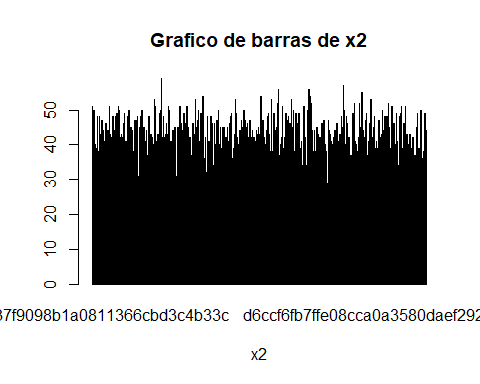


#### Separamos as variaveis quantitativas, qualitativas e a variavel target para fazermos manipulações de maneira mais facil

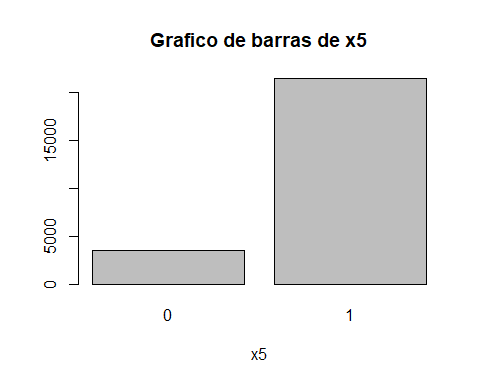
dadosQuali <- df %>%  
 select(x2,x5,x27,x28,x53,x59,x62,x67,x78,x91,x100)  
  
dadosQuant <- df %>%  
 select(-c(id,target,x2,x5,x27,x28,x53,x59,x62,x67,x78,x91,x100))  
  
target <- df$target

### Análise bivariada das variáveis qualitativas

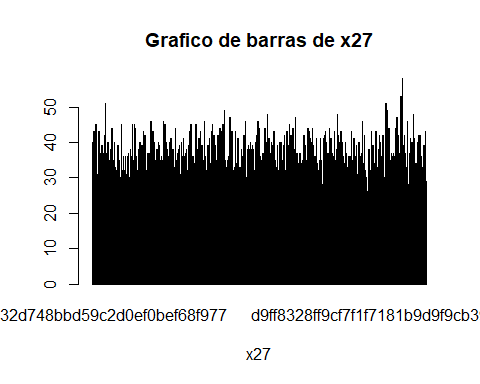
# Graficos de barras para x2 e x5  
barplot(table(df$x2), main="Grafico de barras de x2", xlab="x2")



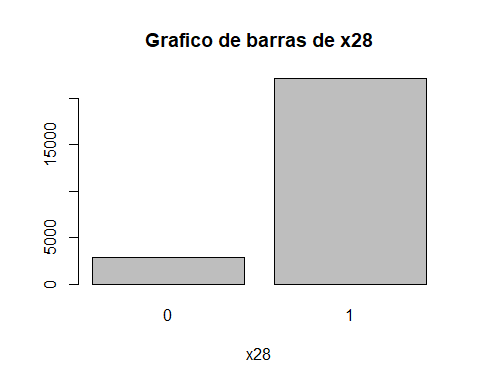
barplot(table(df$x5), main="Grafico de barras de x5", xlab="x5")



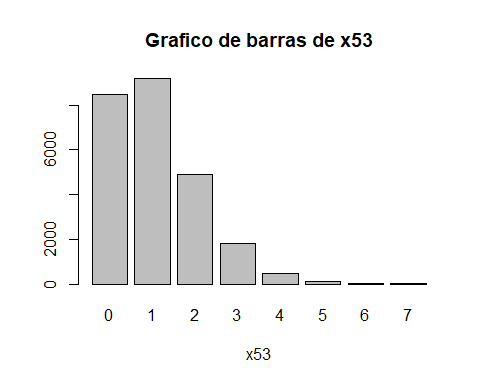
barplot(table(df$x27), main="Grafico de barras de x27", xlab="x27")



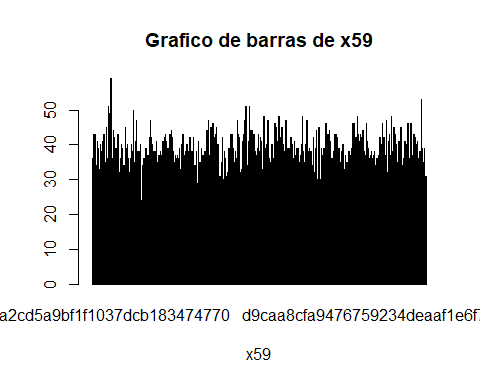
barplot(table(df$x28), main="Grafico de barras de x28", xlab="x28")



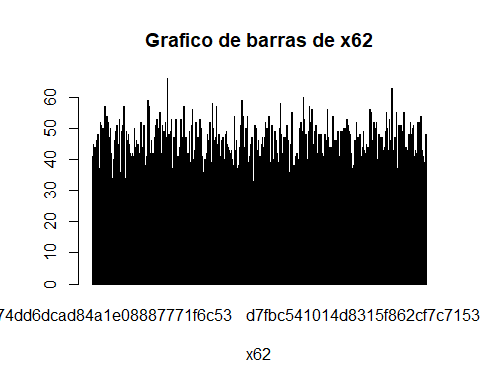
barplot(table(df$x53), main="Grafico de barras de x53", xlab="x53")



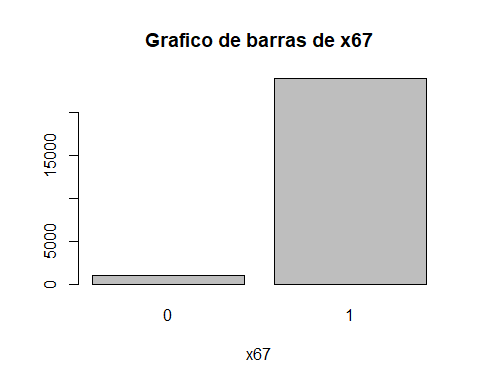
barplot(table(df$x59), main="Grafico de barras de x59", xlab="x59")



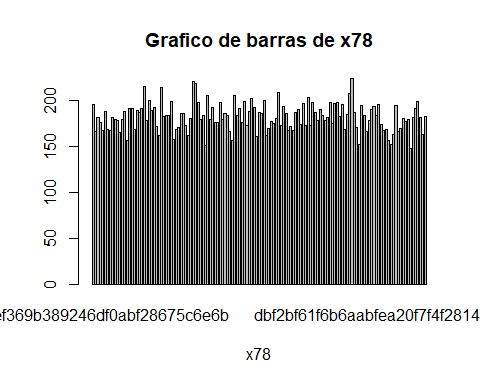
barplot(table(df$x62), main="Grafico de barras de x62", xlab="x62")



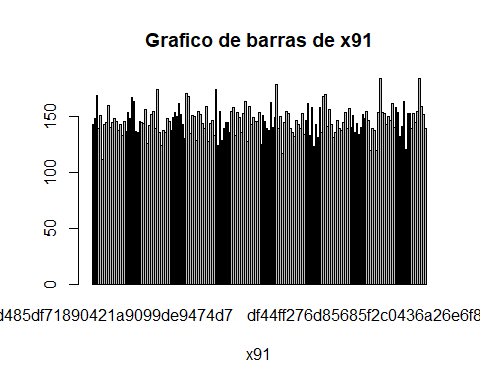
barplot(table(df$x67), main="Grafico de barras de x67", xlab="x67")



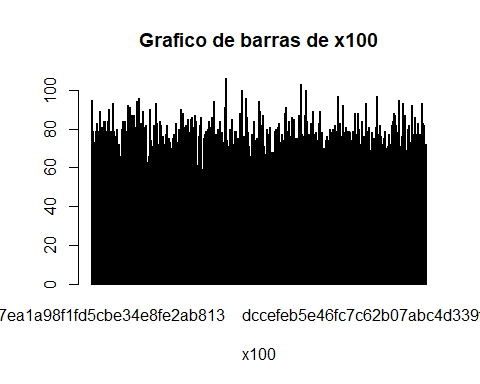
barplot(table(df$x78), main="Grafico de barras de x78", xlab="x78")



barplot(table(df$x91), main="Grafico de barras de x91", xlab="x91")



barplot(table(df$x100), main="Grafico de barras de x100", xlab="x100")



# 6ª etapa: Faça a análise bivariada das variáveis qualitativas.

Analise da variavel target vs x53, x5, x28 e x67

Primeiro passo: transformar a variavel quantitativa (target) em uma qualitativa (faixa de target)

Criterio: fórmula de Sturges

# Calcular o número de bins usando a fórmula de Sturges  
n <- length(df$target)  
k <- round(1 + 3.322 \* log10(n))

#### Para análise bivariada das variáveis qualitativas criamos as faixas de valores para variavel target.

# Criar a variável faixa de target  
df1 <- df  
  
# Usando cut para criar as faixas e labels apropriadas  
df1$fxtarget\_cat <- cut(df$target, breaks = k, include.lowest = TRUE, dig.lab = 10)  
  
# Converter para um fator ordenado  
df1$fxtarget\_cat <- factor(df1$fxtarget\_cat, ordered = TRUE)  
  
# Mostrando as frequências das faixas  
freq(df1$fxtarget\_cat)

## Frequencies   
## df1$fxtarget\_cat   
## Type: Ordered Factor   
##   
## Freq % Valid % Valid Cum. % Total % Total Cum.  
## --------------------------------- ------- --------- -------------- --------- --------------  
## [-0.3014040834,3.608034849] 22429 89.802 89.802 89.802 89.802  
## (3.608034849,7.455907814] 1953 7.820 97.622 7.820 97.622  
## (7.455907814,11.30378078] 397 1.590 99.211 1.590 99.211  
## (11.30378078,15.15165374] 105 0.420 99.632 0.420 99.632  
## (15.15165374,18.99952671] 40 0.160 99.792 0.160 99.792  
## (18.99952671,22.84739967] 25 0.100 99.892 0.100 99.892  
## (22.84739967,26.69527264] 13 0.052 99.944 0.052 99.944  
## (26.69527264,30.5431456] 2 0.008 99.952 0.008 99.952  
## (30.5431456,34.39101857] 2 0.008 99.960 0.008 99.960  
## (34.39101857,38.23889153] 3 0.012 99.972 0.012 99.972  
## (38.23889153,42.0867645] 2 0.008 99.980 0.008 99.980  
## (42.0867645,45.93463746] 1 0.004 99.984 0.004 99.984  
## (45.93463746,49.78251043] 3 0.012 99.996 0.012 99.996  
## (57.47825636,61.38769529] 1 0.004 100.000 0.004 100.000  
## <NA> 0 0.000 100.000  
## Total 24976 100.000 100.000 100.000 100.000

### a) Tabela de frequência bivariada

### b) Teste Qui-quadrado.

Como resultado do CrossTable temos uma tabela de frequência bivariada que compara duas variáveis categóricas e tambem ja nos apresenta o teste Qui-quadrado como output.

Statistics for All Table Factors: Nesta seção, temos o resultado do teste qui-quadrado de Pearson que testa a independência entre as duas variáveis categóricas.

* Chi^2: O valor da estatística qui-quadrado.
* d.f.: Os graus de liberdade para o teste.
* p: O valor-p associado ao teste qui-quadrado. quando este valor é extremamente próximo de 0, sugere que podemos rejeitar a hipótese nula de que as duas variáveis são independentes.

CrossTable (x53 e target)

df1$quali\_x53 <- factor(df1$x53)  
  
CrossTable(df1$fxtarget\_cat,df1$quali\_x53, prop.r = FALSE, prop.c = FALSE, prop.t = FALSE,  
 prop.chisq = FALSE,chisq = TRUE)

##   
##   
## Cell Contents  
## |-------------------------|  
## | N |  
## |-------------------------|  
##   
##   
## Total Observations in Table: 24976   
##   
##   
## | df1$quali\_x53   
## df1$fxtarget\_cat | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Row Total |   
## ----------------------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|  
## [-0.3014040834,3.608034849] | 7602 | 8249 | 4409 | 1640 | 423 | 89 | 15 | 2 | 22429 |   
## ----------------------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|  
## (3.608034849,7.455907814] | 657 | 708 | 381 | 152 | 44 | 9 | 2 | 0 | 1953 |   
## ----------------------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|  
## (7.455907814,11.30378078] | 129 | 139 | 91 | 28 | 10 | 0 | 0 | 0 | 397 |   
## ----------------------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|  
## (11.30378078,15.15165374] | 46 | 36 | 14 | 7 | 1 | 1 | 0 | 0 | 105 |   
## ----------------------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|  
## (15.15165374,18.99952671] | 13 | 16 | 7 | 2 | 2 | 0 | 0 | 0 | 40 |   
## ----------------------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|  
## (18.99952671,22.84739967] | 5 | 12 | 7 | 0 | 1 | 0 | 0 | 0 | 25 |   
## ----------------------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|  
## (22.84739967,26.69527264] | 2 | 9 | 2 | 0 | 0 | 0 | 0 | 0 | 13 |   
## ----------------------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|  
## (26.69527264,30.5431456] | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 2 |   
## ----------------------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|  
## (30.5431456,34.39101857] | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 2 |   
## ----------------------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|  
## (34.39101857,38.23889153] | 1 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 3 |   
## ----------------------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|  
## (38.23889153,42.0867645] | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 2 |   
## ----------------------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|  
## (42.0867645,45.93463746] | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |   
## ----------------------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|  
## (45.93463746,49.78251043] | 0 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 3 |   
## ----------------------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|  
## (57.47825636,61.38769529] | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |   
## ----------------------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|  
## Column Total | 8455 | 9175 | 4917 | 1830 | 481 | 99 | 17 | 2 | 24976 |   
## ----------------------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|  
##   
##   
## Statistics for All Table Factors  
##   
##   
## Pearson's Chi-squared test   
## ------------------------------------------------------------  
## Chi^2 = 56.49066 d.f. = 91 p = 0.9982963   
##   
##   
##

CrossTable (x5 e target)

df1$quali\_x5 <- factor(df1$x5)  
  
CrossTable(df1$fxtarget\_cat,df1$quali\_x5, prop.r = FALSE, prop.c = FALSE, prop.t = FALSE,  
 prop.chisq = FALSE,chisq = TRUE)

##   
##   
## Cell Contents  
## |-------------------------|  
## | N |  
## |-------------------------|  
##   
##   
## Total Observations in Table: 24976   
##   
##   
## | df1$quali\_x5   
## df1$fxtarget\_cat | 0 | 1 | Row Total |   
## ----------------------------|-----------|-----------|-----------|  
## [-0.3014040834,3.608034849] | 2901 | 19528 | 22429 |   
## ----------------------------|-----------|-----------|-----------|  
## (3.608034849,7.455907814] | 446 | 1507 | 1953 |   
## ----------------------------|-----------|-----------|-----------|  
## (7.455907814,11.30378078] | 96 | 301 | 397 |   
## ----------------------------|-----------|-----------|-----------|  
## (11.30378078,15.15165374] | 32 | 73 | 105 |   
## ----------------------------|-----------|-----------|-----------|  
## (15.15165374,18.99952671] | 12 | 28 | 40 |   
## ----------------------------|-----------|-----------|-----------|  
## (18.99952671,22.84739967] | 8 | 17 | 25 |   
## ----------------------------|-----------|-----------|-----------|  
## (22.84739967,26.69527264] | 4 | 9 | 13 |   
## ----------------------------|-----------|-----------|-----------|  
## (26.69527264,30.5431456] | 2 | 0 | 2 |   
## ----------------------------|-----------|-----------|-----------|  
## (30.5431456,34.39101857] | 2 | 0 | 2 |   
## ----------------------------|-----------|-----------|-----------|  
## (34.39101857,38.23889153] | 2 | 1 | 3 |   
## ----------------------------|-----------|-----------|-----------|  
## (38.23889153,42.0867645] | 0 | 2 | 2 |   
## ----------------------------|-----------|-----------|-----------|  
## (42.0867645,45.93463746] | 0 | 1 | 1 |   
## ----------------------------|-----------|-----------|-----------|  
## (45.93463746,49.78251043] | 0 | 3 | 3 |   
## ----------------------------|-----------|-----------|-----------|  
## (57.47825636,61.38769529] | 0 | 1 | 1 |   
## ----------------------------|-----------|-----------|-----------|  
## Column Total | 3505 | 21471 | 24976 |   
## ----------------------------|-----------|-----------|-----------|  
##   
##   
## Statistics for All Table Factors  
##   
##   
## Pearson's Chi-squared test   
## ------------------------------------------------------------  
## Chi^2 = 256.037 d.f. = 13 p = 0.00000000000000000000000000000000000000000000003564283   
##   
##   
##

CrossTable (x28 e target)

df1$quali\_x28 <- factor(df1$x28)  
  
CrossTable(df1$fxtarget\_cat,df1$quali\_x28, prop.r = FALSE, prop.c = FALSE, prop.t = FALSE,  
 prop.chisq = FALSE,chisq = TRUE)

##   
##   
## Cell Contents  
## |-------------------------|  
## | N |  
## |-------------------------|  
##   
##   
## Total Observations in Table: 24976   
##   
##   
## | df1$quali\_x28   
## df1$fxtarget\_cat | 0 | 1 | Row Total |   
## ----------------------------|-----------|-----------|-----------|  
## [-0.3014040834,3.608034849] | 2618 | 19811 | 22429 |   
## ----------------------------|-----------|-----------|-----------|  
## (3.608034849,7.455907814] | 222 | 1731 | 1953 |   
## ----------------------------|-----------|-----------|-----------|  
## (7.455907814,11.30378078] | 37 | 360 | 397 |   
## ----------------------------|-----------|-----------|-----------|  
## (11.30378078,15.15165374] | 12 | 93 | 105 |   
## ----------------------------|-----------|-----------|-----------|  
## (15.15165374,18.99952671] | 3 | 37 | 40 |   
## ----------------------------|-----------|-----------|-----------|  
## (18.99952671,22.84739967] | 1 | 24 | 25 |   
## ----------------------------|-----------|-----------|-----------|  
## (22.84739967,26.69527264] | 2 | 11 | 13 |   
## ----------------------------|-----------|-----------|-----------|  
## (26.69527264,30.5431456] | 0 | 2 | 2 |   
## ----------------------------|-----------|-----------|-----------|  
## (30.5431456,34.39101857] | 1 | 1 | 2 |   
## ----------------------------|-----------|-----------|-----------|  
## (34.39101857,38.23889153] | 1 | 2 | 3 |   
## ----------------------------|-----------|-----------|-----------|  
## (38.23889153,42.0867645] | 0 | 2 | 2 |   
## ----------------------------|-----------|-----------|-----------|  
## (42.0867645,45.93463746] | 0 | 1 | 1 |   
## ----------------------------|-----------|-----------|-----------|  
## (45.93463746,49.78251043] | 0 | 3 | 3 |   
## ----------------------------|-----------|-----------|-----------|  
## (57.47825636,61.38769529] | 0 | 1 | 1 |   
## ----------------------------|-----------|-----------|-----------|  
## Column Total | 2897 | 22079 | 24976 |   
## ----------------------------|-----------|-----------|-----------|  
##   
##   
## Statistics for All Table Factors  
##   
##   
## Pearson's Chi-squared test   
## ------------------------------------------------------------  
## Chi^2 = 9.918561 d.f. = 13 p = 0.7005799   
##   
##   
##

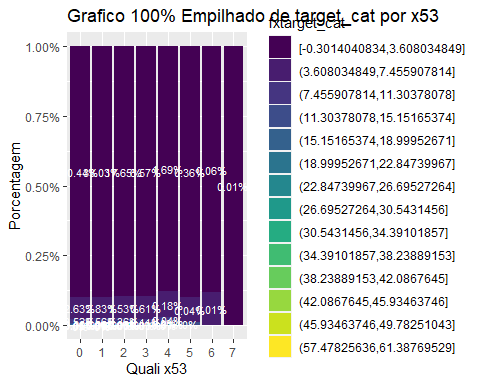
CrossTable (x67 e target)

df1$quali\_x67 <- factor(df1$x67)  
  
CrossTable(df1$fxtarget\_cat,df1$quali\_x67, prop.r = FALSE, prop.c = FALSE, prop.t = FALSE,  
 prop.chisq = FALSE,chisq = TRUE)

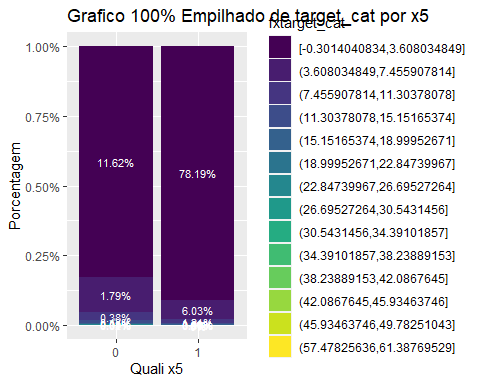
##   
##   
## Cell Contents  
## |-------------------------|  
## | N |  
## |-------------------------|  
##   
##   
## Total Observations in Table: 24976   
##   
##   
## | df1$quali\_x67   
## df1$fxtarget\_cat | 0 | 1 | Row Total |   
## ----------------------------|-----------|-----------|-----------|  
## [-0.3014040834,3.608034849] | 764 | 21665 | 22429 |   
## ----------------------------|-----------|-----------|-----------|  
## (3.608034849,7.455907814] | 185 | 1768 | 1953 |   
## ----------------------------|-----------|-----------|-----------|  
## (7.455907814,11.30378078] | 52 | 345 | 397 |   
## ----------------------------|-----------|-----------|-----------|  
## (11.30378078,15.15165374] | 14 | 91 | 105 |   
## ----------------------------|-----------|-----------|-----------|  
## (15.15165374,18.99952671] | 6 | 34 | 40 |   
## ----------------------------|-----------|-----------|-----------|  
## (18.99952671,22.84739967] | 4 | 21 | 25 |   
## ----------------------------|-----------|-----------|-----------|  
## (22.84739967,26.69527264] | 3 | 10 | 13 |   
## ----------------------------|-----------|-----------|-----------|  
## (26.69527264,30.5431456] | 0 | 2 | 2 |   
## ----------------------------|-----------|-----------|-----------|  
## (30.5431456,34.39101857] | 0 | 2 | 2 |   
## ----------------------------|-----------|-----------|-----------|  
## (34.39101857,38.23889153] | 1 | 2 | 3 |   
## ----------------------------|-----------|-----------|-----------|  
## (38.23889153,42.0867645] | 0 | 2 | 2 |   
## ----------------------------|-----------|-----------|-----------|  
## (42.0867645,45.93463746] | 1 | 0 | 1 |   
## ----------------------------|-----------|-----------|-----------|  
## (45.93463746,49.78251043] | 1 | 2 | 3 |   
## ----------------------------|-----------|-----------|-----------|  
## (57.47825636,61.38769529] | 0 | 1 | 1 |   
## ----------------------------|-----------|-----------|-----------|  
## Column Total | 1031 | 23945 | 24976 |   
## ----------------------------|-----------|-----------|-----------|  
##   
##   
## Statistics for All Table Factors  
##   
##   
## Pearson's Chi-squared test   
## ------------------------------------------------------------  
## Chi^2 = 342.7846 d.f. = 13 p = 0.00000000000000000000000000000000000000000000000000000000000000002553164   
##   
##   
##

## c) Grafico 100% empilhado

ggplot(df1, aes(fill = fxtarget\_cat, x = quali\_x53)) +  
 geom\_bar(position = "fill") +  
 geom\_text(  
 aes(label = paste0(round(100 \* ..count../sum(..count..), 2), "%"),  
 y = ..count..),  
 position = position\_fill(vjust = 0.5),  
 stat = "count",  
 size = 3,  
 color = "white" # Definir a cor do label como branca  
 ) +  
 labs(y = "Porcentagem", x = "Quali x53", title = "Grafico 100% Empilhado de target\_cat por x53") +  
 scale\_y\_continuous(labels = scales::percent\_format(scale = 1))



ggplot(df1, aes(fill = fxtarget\_cat, x = quali\_x5)) +  
 geom\_bar(position = "fill") +  
 geom\_text(  
 aes(label = paste0(round(100 \* ..count../sum(..count..), 2), "%"),  
 y = ..count..),  
 position = position\_fill(vjust = 0.5),  
 stat = "count",  
 size = 3,  
 color = "white" # Definir a cor do label como branca  
 ) +  
 labs(y = "Porcentagem", x = "Quali x5", title = "Grafico 100% Empilhado de target\_cat por x5") +  
 scale\_y\_continuous(labels = scales::percent\_format(scale = 1))



## 7ª etapa: Faça a análise bivariada das variáveis quantitativas.

### e) Análise de correlação de Pearson.

Selecionando as top 10 variaveis preditoras para fazermos a análise bivariada das variáveis quantitativas utilizando a correlação de pearson

1º passo - Aplicar correlação de pearson nas variaveis quantitativas

df\_quant\_corr <- cbind(dadosQuant,target)  
  
# Calculando a matriz de correlação de pearson  
correlacao <- cor(df\_quant\_corr)  
head(correlacao)

## x1 x3 x4 x6 x7  
## x1 1.0000000000 -0.0008321963 -0.003624797 0.0019983860 0.002808924  
## x3 -0.0008321963 1.0000000000 0.006272639 -0.0008101303 0.009103130  
## x4 -0.0036247970 0.0062726392 1.000000000 0.0014666091 -0.002300537  
## x6 0.0019983860 -0.0008101303 0.001466609 1.0000000000 0.002267395  
## x7 0.0028089243 0.0091031298 -0.002300537 0.0022673949 1.000000000  
## x8 0.0058303507 -0.0103189499 0.003445743 0.0058203056 0.006100992  
## x8 x9 x10 x11 x12  
## x1 0.005830351 0.00246511743 -0.0037266397 -0.0033185100 0.001733434  
## x3 -0.010318950 -0.00546138311 0.0009018064 0.0035956387 0.003357122  
## x4 0.003445743 0.01061788503 0.0013006223 -0.0062222042 0.001243228  
## x6 0.005820306 -0.00422931570 -0.0012971817 0.0076626340 0.003560071  
## x7 0.006100992 -0.00250054618 -0.0010237875 0.0006392020 -0.004895799  
## x8 1.000000000 0.00007731053 -0.0038687790 0.0003792376 -0.009975825  
## x13 x14 x15 x16 x17  
## x1 -0.004842906 -0.0031592557 0.0087029717 -0.0026129316 0.006419444  
## x3 0.002919105 0.0029147949 -0.0025357446 -0.0008546989 0.001497567  
## x4 0.001858086 -0.0006099921 0.0065605744 -0.0084681219 0.018826962  
## x6 0.010031673 0.0016472830 -0.0038515600 0.0102763446 -0.004083805  
## x7 -0.003102100 0.0011078452 -0.0009946081 -0.0066364170 -0.003159171  
## x8 -0.001574464 -0.0106802559 0.0018128831 -0.0105260891 -0.003207388  
## x18 x19 x20 x21 x22  
## x1 0.002893706 0.002808261 0.005728961 0.0047423092 -0.005882706  
## x3 0.004485556 -0.001955048 0.002593750 0.0123941360 0.006120782  
## x4 0.005601628 -0.001153513 0.001124931 -0.0039129029 -0.007614048  
## x6 -0.003484412 -0.001048058 -0.006666357 0.0013134625 -0.004016844  
## x7 0.007156813 0.001882257 -0.001737479 0.0007169492 0.004134451  
## x8 0.001887464 0.003072182 0.002812181 0.0114691307 0.006813752  
## x23 x24 x25 x26 x29  
## x1 -0.00618961361 -0.0002203695 -0.0036067783 0.001251859 0.000506659  
## x3 -0.00005792338 0.0048442317 0.0075284310 -0.005856561 0.004763369  
## x4 0.00458324444 0.0112850686 0.0016697137 0.008994933 0.012549746  
## x6 -0.00562415551 -0.0090347534 -0.0006650095 -0.001207830 -0.009878492  
## x7 -0.00586981201 -0.0019424113 0.0043411741 0.003227073 -0.001112208  
## x8 -0.00867553262 -0.0083400423 0.0027382845 -0.004047562 -0.008719404  
## x30 x31 x32 x33 x34  
## x1 0.0030276575 -0.010529916 -0.001701896 0.001559493 -0.004327196  
## x3 0.0115602799 0.003167645 -0.014326758 -0.001929161 -0.004382633  
## x4 0.0033141957 -0.004270402 0.006871476 0.004851186 -0.003698816  
## x6 -0.0084951775 -0.005873805 -0.002582066 -0.007628624 0.001026681  
## x7 -0.0006882836 -0.001989727 -0.012007137 0.010875615 -0.008787134  
## x8 -0.0050290387 0.006653834 0.006330461 -0.015709310 0.001016285  
## x35 x36 x37 x38 x39  
## x1 0.00308614245 -0.005783486 -0.0038567059 -0.004706491 0.0007501352  
## x3 -0.00731974888 -0.006177452 -0.0166773632 -0.006870303 0.0073117333  
## x4 0.00642468350 -0.002098038 -0.0006112108 0.010533176 -0.0050111355  
## x6 0.00618238693 -0.012941480 0.0029298372 -0.005847306 0.0005386599  
## x7 0.00377998667 -0.002715281 -0.0025724177 -0.007092081 -0.0012461480  
## x8 0.00005789121 -0.005353304 -0.0020436260 -0.002343033 -0.0073621540  
## x40 x41 x42 x43 x44  
## x1 -0.0004712121 -0.0081490900 0.009695202 -0.007098942 -0.002145869  
## x3 0.0014912145 0.0030252023 0.005875041 0.003915858 -0.009628626  
## x4 -0.0016313569 0.0068179353 0.007814309 -0.008704613 0.010096677  
## x6 -0.0026751756 0.0050924912 -0.003169041 0.001883428 -0.001990032  
## x7 -0.0041827497 -0.0021779142 -0.005539887 -0.001834709 -0.008970774  
## x8 -0.0005207988 -0.0002563174 0.003110559 0.007167571 0.004992579  
## x45 x46 x47 x48 x49  
## x1 -0.009555359 -0.008504359 0.0005812961 0.0100849628 0.001514076  
## x3 -0.003914188 -0.002491229 -0.0044654404 -0.0025680956 0.007587439  
## x4 0.002885670 -0.005283155 -0.0064948990 0.0003214721 0.001834658  
## x6 0.013419416 0.004857831 -0.0051501117 0.0021003842 -0.003573276  
## x7 -0.001932564 0.005242542 0.0060255001 0.0064457949 -0.002491669  
## x8 -0.010923298 -0.017318634 0.0010818414 -0.0003376472 -0.003635405  
## x50 x51 x52 x54 x55  
## x1 0.006281987 -0.004730117 -0.009238474 -0.004661734 -0.001732875  
## x3 -0.013974770 -0.001568446 -0.001910831 0.017074875 -0.003061413  
## x4 0.007599485 0.002572090 0.001122037 0.005399226 0.006942913  
## x6 -0.002578155 0.004954875 0.004794571 0.009755553 -0.001961848  
## x7 0.010777355 0.009655216 0.002541081 -0.003575714 0.002506427  
## x8 0.009622798 0.003422034 -0.018876754 0.006108780 -0.002141618  
## x56 x57 x58 x60 x61  
## x1 0.0062506583 -0.007897791 0.0040286949 0.0026488090 -0.001720756  
## x3 0.0011380353 0.003106812 -0.0190298494 -0.0005431027 -0.001236833  
## x4 -0.0008913798 0.003084881 -0.0030643087 0.0016383706 0.002266884  
## x6 0.0030666755 0.001386677 0.0130012335 0.9958192886 -0.000773330  
## x7 0.0061067021 0.004516274 0.0004124984 0.0025788790 0.002841411  
## x8 -0.0039920534 0.004319637 0.0027403233 0.0057695173 0.004886082  
## x63 x64 x65 x66 x68  
## x1 -0.007870773 0.0003431034 -0.004339014 0.0061586959 0.0109842814  
## x3 0.002924673 0.0046297694 -0.007859324 -0.0039121018 -0.0076182693  
## x4 0.002933345 0.0126851995 -0.002278321 0.0014906324 -0.0060016792  
## x6 0.001514620 -0.0093715621 -0.001603947 0.0063870950 0.0039732276  
## x7 0.004398375 -0.0011540880 0.003719297 0.0081809790 -0.0096067380  
## x8 0.004345173 -0.0087803853 0.005518413 0.0000214719 -0.0001182105  
## x69 x70 x71 x72 x73  
## x1 0.00451317989 -0.003460667973 -0.00439320754 0.002852343 -0.004277236  
## x3 -0.00265077468 0.006871526512 0.00712128087 -0.012997324 0.002308342  
## x4 -0.00115244285 0.011382057455 -0.00719855514 -0.005582851 -0.008931097  
## x6 0.00004729663 -0.001291190825 0.00585647049 0.005557912 -0.004276294  
## x7 0.01394313885 0.003555301401 0.00453423599 -0.006977432 -0.003329816  
## x8 -0.00445445887 0.000008868438 -0.00003014197 -0.001831280 -0.008374251  
## x74 x75 x76 x77 x79  
## x1 -0.008887404 -0.010222324 0.0026442500 0.0073603837 -0.00632287524  
## x3 -0.004785572 -0.002835517 -0.0005420775 -0.0106988305 -0.00490190023  
## x4 0.001211356 0.002908049 0.0016404973 0.0012698568 0.00007580089  
## x6 -0.002589661 0.008480675 0.9958191329 -0.0008658012 0.00416514940  
## x7 -0.002581162 0.003834897 0.0025772145 0.0007590789 0.00456621558  
## x8 -0.002624125 -0.007375458 0.0057688476 0.0000571431 0.00140217691  
## x80 x81 x82 x83 x84  
## x1 0.00033712995 -0.002892216 -0.0054231349 -0.00381751980 -0.010529866  
## x3 -0.00319982553 0.001341551 -0.0123381211 -0.00210767433 0.003167687  
## x4 -0.00801282265 -0.011482855 -0.0008768976 -0.00002180066 -0.004270398  
## x6 -0.00470130297 0.014847821 -0.0046198386 0.00235109464 -0.005873807  
## x7 -0.00029064278 0.007772878 -0.0037080706 -0.00534468710 -0.001989798  
## x8 -0.00005192247 0.002504110 0.0094254836 0.00043917358 0.006653785  
## x85 x86 x87 x88 x89  
## x1 0.00615735656 -0.00147534799 -0.0054446487 -0.001700805 0.002656186  
## x3 -0.00391334125 0.00104551514 0.0009676961 -0.014332404 0.008635814  
## x4 0.00149053248 -0.00005760458 0.0052178798 0.006879588 -0.006770334  
## x6 0.00638621373 0.00010687579 0.0081036976 -0.002583531 -0.005998693  
## x7 0.00818011251 0.00231378205 -0.0117025129 -0.012004888 0.001518865  
## x8 0.00002208646 -0.01054890700 -0.0056554395 0.006325840 0.003547063  
## x90 x92 x93 x94 x95  
## x1 0.006237485 -0.005756647 0.001836736 -0.007116918 -0.009335684  
## x3 -0.013689483 0.005934469 -0.002796536 0.011556037 0.010748565  
## x4 0.007600083 -0.007457578 -0.008545151 0.001012697 0.003883422  
## x6 -0.002830790 -0.004053671 -0.005244134 0.005022798 -0.001778632  
## x7 0.010957136 0.004163486 -0.001362471 0.012460978 -0.005780594  
## x8 0.009585710 0.006811346 -0.004974000 -0.005484287 0.005067956  
## x96 x97 x98 x99 x101  
## x1 -0.008930784 0.0148930101 0.0011519440 0.009412425 0.005747033  
## x3 0.002556928 -0.0008665307 -0.0013283040 -0.004889929 0.002590851  
## x4 0.003666497 0.0005427453 -0.0053349515 -0.002654247 0.001120647  
## x6 0.002675791 -0.0037540255 0.0022329444 -0.008778385 -0.006658601  
## x7 0.006909660 -0.0139560257 0.0001710532 0.003241088 -0.001747026  
## x8 0.005066854 -0.0012472445 0.0048782685 0.004995720 0.002807669  
## x102 x103 x104 x105 x106  
## x1 0.0052599698 0.0084329252 -0.0006424986 0.002872365 -0.004468945  
## x3 -0.0004619518 0.0101212431 -0.0114479541 0.004655969 0.005078409  
## x4 0.0111674075 -0.0001789051 0.0088099422 0.005957455 0.005798902  
## x6 -0.0066419905 0.0038405281 0.0026474702 -0.002841993 -0.001651108  
## x7 0.0103721065 0.0093300214 0.0006673214 0.008618914 -0.007056624  
## x8 -0.0037147151 0.0036996254 -0.0012010807 0.001730469 -0.008638824  
## target  
## x1 -0.027028720  
## x3 0.003440220  
## x4 -0.013537320  
## x6 -0.004892913  
## x7 0.037973797  
## x8 0.004259164

### f) Matriz de correlação de Pearson.

head(correlacao)

## x1 x3 x4 x6 x7  
## x1 1.0000000000 -0.0008321963 -0.003624797 0.0019983860 0.002808924  
## x3 -0.0008321963 1.0000000000 0.006272639 -0.0008101303 0.009103130  
## x4 -0.0036247970 0.0062726392 1.000000000 0.0014666091 -0.002300537  
## x6 0.0019983860 -0.0008101303 0.001466609 1.0000000000 0.002267395  
## x7 0.0028089243 0.0091031298 -0.002300537 0.0022673949 1.000000000  
## x8 0.0058303507 -0.0103189499 0.003445743 0.0058203056 0.006100992  
## x8 x9 x10 x11 x12  
## x1 0.005830351 0.00246511743 -0.0037266397 -0.0033185100 0.001733434  
## x3 -0.010318950 -0.00546138311 0.0009018064 0.0035956387 0.003357122  
## x4 0.003445743 0.01061788503 0.0013006223 -0.0062222042 0.001243228  
## x6 0.005820306 -0.00422931570 -0.0012971817 0.0076626340 0.003560071  
## x7 0.006100992 -0.00250054618 -0.0010237875 0.0006392020 -0.004895799  
## x8 1.000000000 0.00007731053 -0.0038687790 0.0003792376 -0.009975825  
## x13 x14 x15 x16 x17  
## x1 -0.004842906 -0.0031592557 0.0087029717 -0.0026129316 0.006419444  
## x3 0.002919105 0.0029147949 -0.0025357446 -0.0008546989 0.001497567  
## x4 0.001858086 -0.0006099921 0.0065605744 -0.0084681219 0.018826962  
## x6 0.010031673 0.0016472830 -0.0038515600 0.0102763446 -0.004083805  
## x7 -0.003102100 0.0011078452 -0.0009946081 -0.0066364170 -0.003159171  
## x8 -0.001574464 -0.0106802559 0.0018128831 -0.0105260891 -0.003207388  
## x18 x19 x20 x21 x22  
## x1 0.002893706 0.002808261 0.005728961 0.0047423092 -0.005882706  
## x3 0.004485556 -0.001955048 0.002593750 0.0123941360 0.006120782  
## x4 0.005601628 -0.001153513 0.001124931 -0.0039129029 -0.007614048  
## x6 -0.003484412 -0.001048058 -0.006666357 0.0013134625 -0.004016844  
## x7 0.007156813 0.001882257 -0.001737479 0.0007169492 0.004134451  
## x8 0.001887464 0.003072182 0.002812181 0.0114691307 0.006813752  
## x23 x24 x25 x26 x29  
## x1 -0.00618961361 -0.0002203695 -0.0036067783 0.001251859 0.000506659  
## x3 -0.00005792338 0.0048442317 0.0075284310 -0.005856561 0.004763369  
## x4 0.00458324444 0.0112850686 0.0016697137 0.008994933 0.012549746  
## x6 -0.00562415551 -0.0090347534 -0.0006650095 -0.001207830 -0.009878492  
## x7 -0.00586981201 -0.0019424113 0.0043411741 0.003227073 -0.001112208  
## x8 -0.00867553262 -0.0083400423 0.0027382845 -0.004047562 -0.008719404  
## x30 x31 x32 x33 x34  
## x1 0.0030276575 -0.010529916 -0.001701896 0.001559493 -0.004327196  
## x3 0.0115602799 0.003167645 -0.014326758 -0.001929161 -0.004382633  
## x4 0.0033141957 -0.004270402 0.006871476 0.004851186 -0.003698816  
## x6 -0.0084951775 -0.005873805 -0.002582066 -0.007628624 0.001026681  
## x7 -0.0006882836 -0.001989727 -0.012007137 0.010875615 -0.008787134  
## x8 -0.0050290387 0.006653834 0.006330461 -0.015709310 0.001016285  
## x35 x36 x37 x38 x39  
## x1 0.00308614245 -0.005783486 -0.0038567059 -0.004706491 0.0007501352  
## x3 -0.00731974888 -0.006177452 -0.0166773632 -0.006870303 0.0073117333  
## x4 0.00642468350 -0.002098038 -0.0006112108 0.010533176 -0.0050111355  
## x6 0.00618238693 -0.012941480 0.0029298372 -0.005847306 0.0005386599  
## x7 0.00377998667 -0.002715281 -0.0025724177 -0.007092081 -0.0012461480  
## x8 0.00005789121 -0.005353304 -0.0020436260 -0.002343033 -0.0073621540  
## x40 x41 x42 x43 x44  
## x1 -0.0004712121 -0.0081490900 0.009695202 -0.007098942 -0.002145869  
## x3 0.0014912145 0.0030252023 0.005875041 0.003915858 -0.009628626  
## x4 -0.0016313569 0.0068179353 0.007814309 -0.008704613 0.010096677  
## x6 -0.0026751756 0.0050924912 -0.003169041 0.001883428 -0.001990032  
## x7 -0.0041827497 -0.0021779142 -0.005539887 -0.001834709 -0.008970774  
## x8 -0.0005207988 -0.0002563174 0.003110559 0.007167571 0.004992579  
## x45 x46 x47 x48 x49  
## x1 -0.009555359 -0.008504359 0.0005812961 0.0100849628 0.001514076  
## x3 -0.003914188 -0.002491229 -0.0044654404 -0.0025680956 0.007587439  
## x4 0.002885670 -0.005283155 -0.0064948990 0.0003214721 0.001834658  
## x6 0.013419416 0.004857831 -0.0051501117 0.0021003842 -0.003573276  
## x7 -0.001932564 0.005242542 0.0060255001 0.0064457949 -0.002491669  
## x8 -0.010923298 -0.017318634 0.0010818414 -0.0003376472 -0.003635405  
## x50 x51 x52 x54 x55  
## x1 0.006281987 -0.004730117 -0.009238474 -0.004661734 -0.001732875  
## x3 -0.013974770 -0.001568446 -0.001910831 0.017074875 -0.003061413  
## x4 0.007599485 0.002572090 0.001122037 0.005399226 0.006942913  
## x6 -0.002578155 0.004954875 0.004794571 0.009755553 -0.001961848  
## x7 0.010777355 0.009655216 0.002541081 -0.003575714 0.002506427  
## x8 0.009622798 0.003422034 -0.018876754 0.006108780 -0.002141618  
## x56 x57 x58 x60 x61  
## x1 0.0062506583 -0.007897791 0.0040286949 0.0026488090 -0.001720756  
## x3 0.0011380353 0.003106812 -0.0190298494 -0.0005431027 -0.001236833  
## x4 -0.0008913798 0.003084881 -0.0030643087 0.0016383706 0.002266884  
## x6 0.0030666755 0.001386677 0.0130012335 0.9958192886 -0.000773330  
## x7 0.0061067021 0.004516274 0.0004124984 0.0025788790 0.002841411  
## x8 -0.0039920534 0.004319637 0.0027403233 0.0057695173 0.004886082  
## x63 x64 x65 x66 x68  
## x1 -0.007870773 0.0003431034 -0.004339014 0.0061586959 0.0109842814  
## x3 0.002924673 0.0046297694 -0.007859324 -0.0039121018 -0.0076182693  
## x4 0.002933345 0.0126851995 -0.002278321 0.0014906324 -0.0060016792  
## x6 0.001514620 -0.0093715621 -0.001603947 0.0063870950 0.0039732276  
## x7 0.004398375 -0.0011540880 0.003719297 0.0081809790 -0.0096067380  
## x8 0.004345173 -0.0087803853 0.005518413 0.0000214719 -0.0001182105  
## x69 x70 x71 x72 x73  
## x1 0.00451317989 -0.003460667973 -0.00439320754 0.002852343 -0.004277236  
## x3 -0.00265077468 0.006871526512 0.00712128087 -0.012997324 0.002308342  
## x4 -0.00115244285 0.011382057455 -0.00719855514 -0.005582851 -0.008931097  
## x6 0.00004729663 -0.001291190825 0.00585647049 0.005557912 -0.004276294  
## x7 0.01394313885 0.003555301401 0.00453423599 -0.006977432 -0.003329816  
## x8 -0.00445445887 0.000008868438 -0.00003014197 -0.001831280 -0.008374251  
## x74 x75 x76 x77 x79  
## x1 -0.008887404 -0.010222324 0.0026442500 0.0073603837 -0.00632287524  
## x3 -0.004785572 -0.002835517 -0.0005420775 -0.0106988305 -0.00490190023  
## x4 0.001211356 0.002908049 0.0016404973 0.0012698568 0.00007580089  
## x6 -0.002589661 0.008480675 0.9958191329 -0.0008658012 0.00416514940  
## x7 -0.002581162 0.003834897 0.0025772145 0.0007590789 0.00456621558  
## x8 -0.002624125 -0.007375458 0.0057688476 0.0000571431 0.00140217691  
## x80 x81 x82 x83 x84  
## x1 0.00033712995 -0.002892216 -0.0054231349 -0.00381751980 -0.010529866  
## x3 -0.00319982553 0.001341551 -0.0123381211 -0.00210767433 0.003167687  
## x4 -0.00801282265 -0.011482855 -0.0008768976 -0.00002180066 -0.004270398  
## x6 -0.00470130297 0.014847821 -0.0046198386 0.00235109464 -0.005873807  
## x7 -0.00029064278 0.007772878 -0.0037080706 -0.00534468710 -0.001989798  
## x8 -0.00005192247 0.002504110 0.0094254836 0.00043917358 0.006653785  
## x85 x86 x87 x88 x89  
## x1 0.00615735656 -0.00147534799 -0.0054446487 -0.001700805 0.002656186  
## x3 -0.00391334125 0.00104551514 0.0009676961 -0.014332404 0.008635814  
## x4 0.00149053248 -0.00005760458 0.0052178798 0.006879588 -0.006770334  
## x6 0.00638621373 0.00010687579 0.0081036976 -0.002583531 -0.005998693  
## x7 0.00818011251 0.00231378205 -0.0117025129 -0.012004888 0.001518865  
## x8 0.00002208646 -0.01054890700 -0.0056554395 0.006325840 0.003547063  
## x90 x92 x93 x94 x95  
## x1 0.006237485 -0.005756647 0.001836736 -0.007116918 -0.009335684  
## x3 -0.013689483 0.005934469 -0.002796536 0.011556037 0.010748565  
## x4 0.007600083 -0.007457578 -0.008545151 0.001012697 0.003883422  
## x6 -0.002830790 -0.004053671 -0.005244134 0.005022798 -0.001778632  
## x7 0.010957136 0.004163486 -0.001362471 0.012460978 -0.005780594  
## x8 0.009585710 0.006811346 -0.004974000 -0.005484287 0.005067956  
## x96 x97 x98 x99 x101  
## x1 -0.008930784 0.0148930101 0.0011519440 0.009412425 0.005747033  
## x3 0.002556928 -0.0008665307 -0.0013283040 -0.004889929 0.002590851  
## x4 0.003666497 0.0005427453 -0.0053349515 -0.002654247 0.001120647  
## x6 0.002675791 -0.0037540255 0.0022329444 -0.008778385 -0.006658601  
## x7 0.006909660 -0.0139560257 0.0001710532 0.003241088 -0.001747026  
## x8 0.005066854 -0.0012472445 0.0048782685 0.004995720 0.002807669  
## x102 x103 x104 x105 x106  
## x1 0.0052599698 0.0084329252 -0.0006424986 0.002872365 -0.004468945  
## x3 -0.0004619518 0.0101212431 -0.0114479541 0.004655969 0.005078409  
## x4 0.0111674075 -0.0001789051 0.0088099422 0.005957455 0.005798902  
## x6 -0.0066419905 0.0038405281 0.0026474702 -0.002841993 -0.001651108  
## x7 0.0103721065 0.0093300214 0.0006673214 0.008618914 -0.007056624  
## x8 -0.0037147151 0.0036996254 -0.0012010807 0.001730469 -0.008638824  
## target  
## x1 -0.027028720  
## x3 0.003440220  
## x4 -0.013537320  
## x6 -0.004892913  
## x7 0.037973797  
## x8 0.004259164

2º passo - Filtrar as top 10 pegando os coeficientes de correlação entre as variaveis preditoras e a variável target

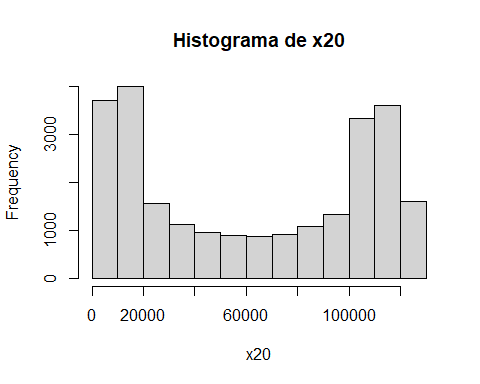
# Pegando os coeficientes de correlação para a variável target  
correlacao\_com\_target <- correlacao[,"target"]  
  
# Removendo a correlação da variável target com ela mesma  
correlacao\_com\_target <- correlacao\_com\_target[-which(names(correlacao\_com\_target) == "target")]  
  
# Ordenando em valor absoluto  
ordenado <- sort(abs(correlacao\_com\_target), decreasing = TRUE)  
  
# Selecionando as top 10  
top\_10 <- ordenado[1:10]  
top\_10

## x20 x101 x75 x63 x57 x94 x96 x18   
## 0.2158847 0.2158470 0.1559728 0.1518013 0.1517786 0.1496535 0.1486564 0.1452504   
## x105 x45   
## 0.1437556 0.1346590

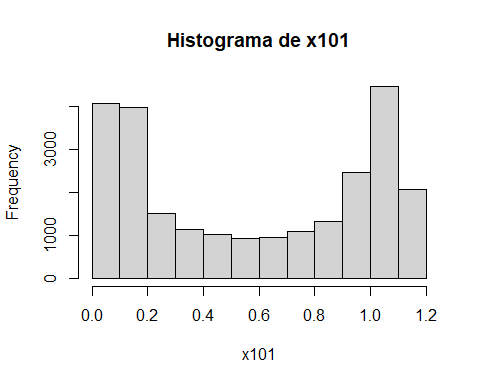
### d) Grafico de dispersão.

3º passo - Fazer a analise bivariada das variáveis quantitativas

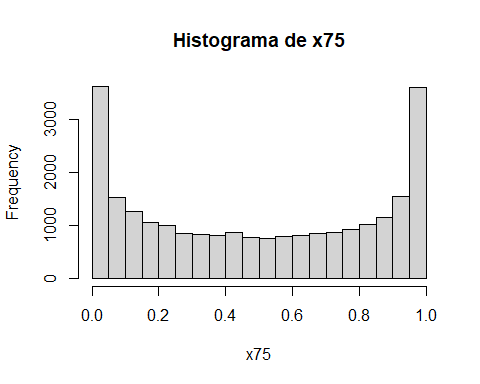
# Histogramas para x1 e x3  
hist(df$x20, main="Histograma de x20", xlab="x20")



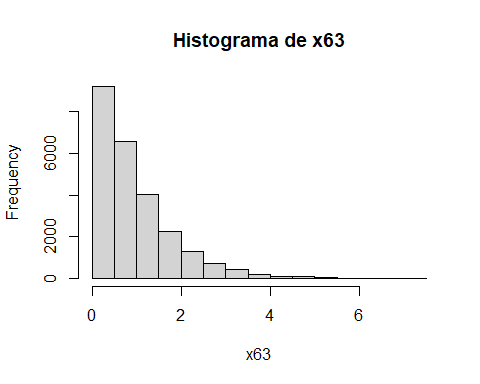
hist(df$x101, main="Histograma de x101", xlab="x101")



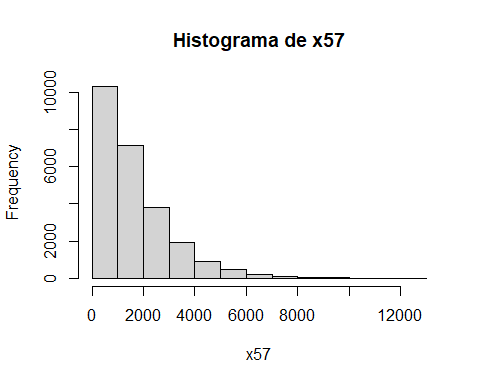
hist(df$x75, main="Histograma de x75", xlab="x75")



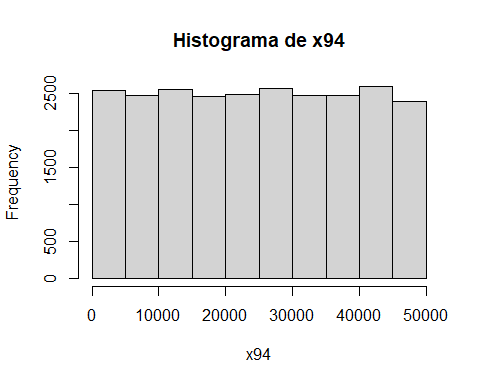
hist(df$x63, main="Histograma de x63", xlab="x63")



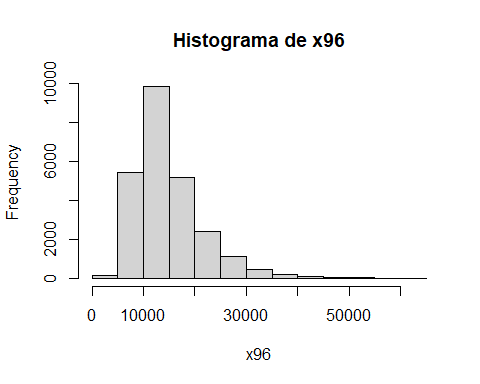
hist(df$x57, main="Histograma de x57", xlab="x57")



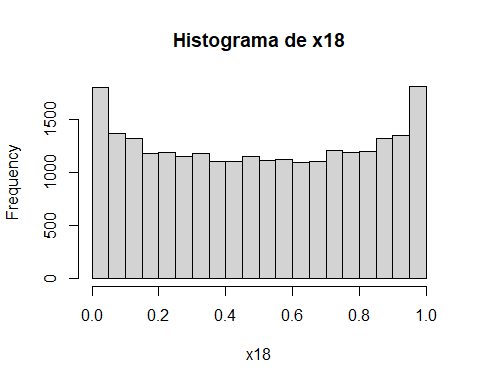
hist(df$x94, main="Histograma de x94", xlab="x94")



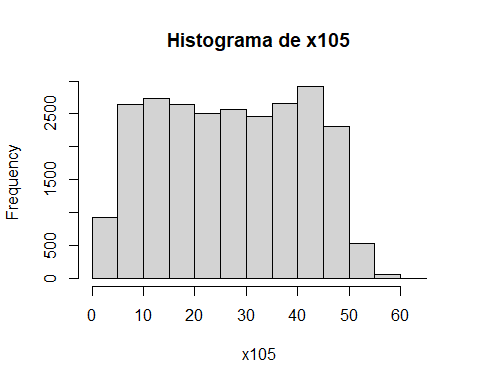
hist(df$x96, main="Histograma de x96", xlab="x96")



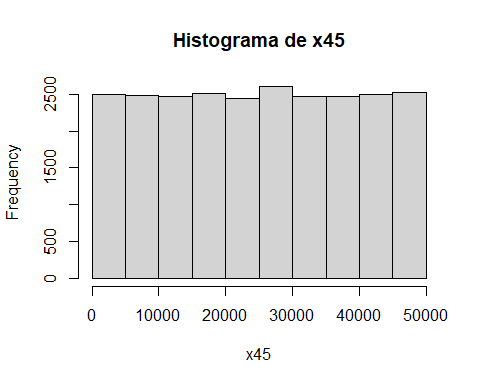
hist(df$x18, main="Histograma de x18", xlab="x18")



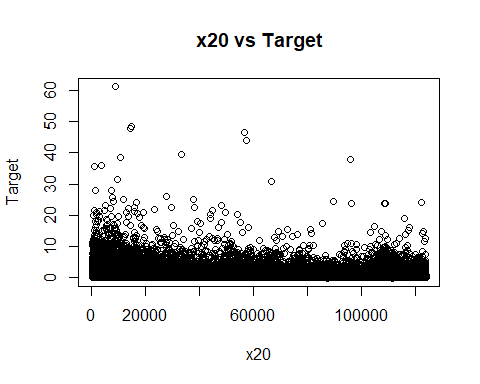
hist(df$x105, main="Histograma de x105", xlab="x105")



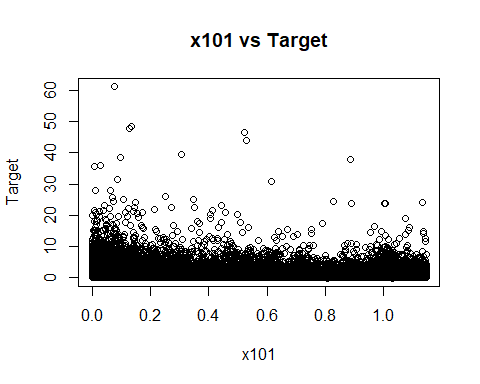
hist(df$x45, main="Histograma de x45", xlab="x45")



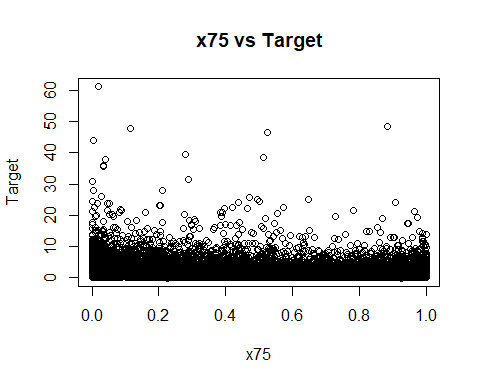
# Graficos de dispersão de x1 e x3 em relação à variável target  
plot(df$x20, df$target, main="x20 vs Target", xlab="x20", ylab="Target")



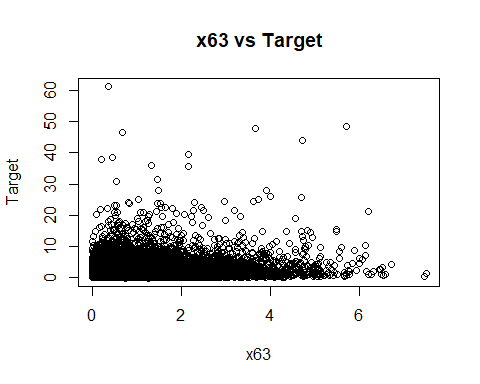
plot(df$x101, df$target, main="x101 vs Target", xlab="x101", ylab="Target")



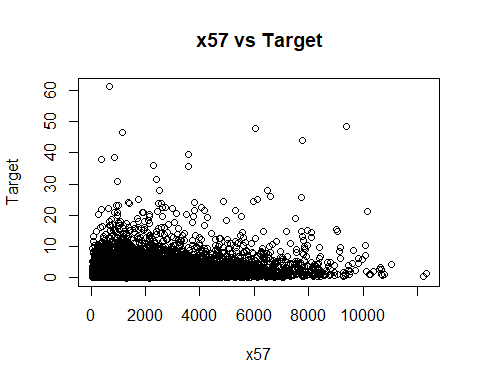
plot(df$x75, df$target, main="x75 vs Target", xlab="x75", ylab="Target")



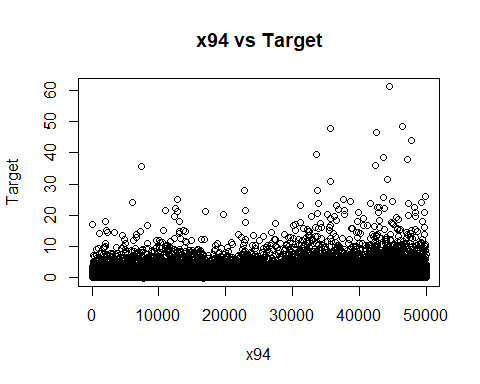
plot(df$x63, df$target, main="x63 vs Target", xlab="x63", ylab="Target")



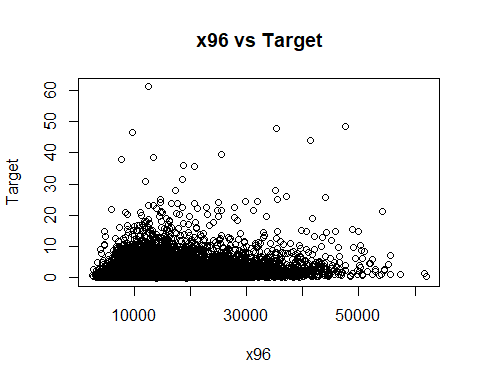
plot(df$x57, df$target, main="x57 vs Target", xlab="x57", ylab="Target")



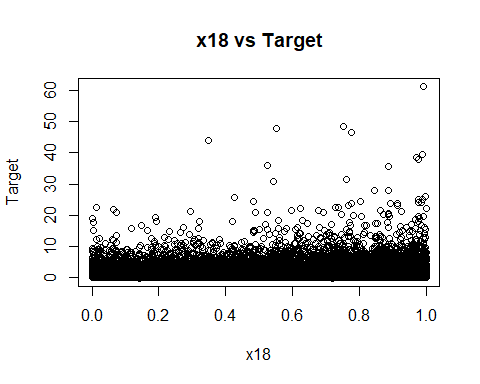
plot(df$x94, df$target, main="x94 vs Target", xlab="x94", ylab="Target")



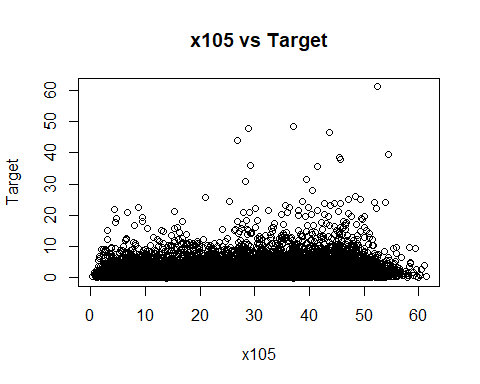
plot(df$x96, df$target, main="x96 vs Target", xlab="x96", ylab="Target")



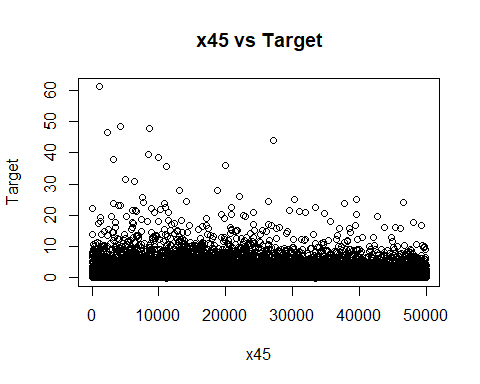
plot(df$x18, df$target, main="x18 vs Target", xlab="x18", ylab="Target")



plot(df$x105, df$target, main="x105 vs Target", xlab="x105", ylab="Target")



plot(df$x45, df$target, main="x45 vs Target", xlab="x45", ylab="Target")



## 9ª etapa: Construção do modelo preditivo.

### Variáveis quantitativas

Transformando as variáveis preditoras quantitativas em qualitativas. Nesta função utilizamos o método do Intervalo Interquartil (IQR), todas as colunas que apresentaram outliers são convertidas para qualitativas. Após a conversão criamos intervalos para cada uma delas, e o número de bins é definido com base na fórmula de Sturges.

# Função para detectar se uma coluna tem outliers  
detect\_outliers <- function(column) {  
 Q1 <- quantile(column, 0.25)  
 Q3 <- quantile(column, 0.75)  
 IQR <- Q3 - Q1  
 lower\_bound <- Q1 - 1.5 \* IQR  
 upper\_bound <- Q3 + 1.5 \* IQR  
 return(any(column < lower\_bound | column > upper\_bound))  
}  
  
# Função para transformar uma coluna numérica em categorias  
transform\_to\_categories <- function(column) {  
 n <- length(column)  
 k <- round(1 + 3.322 \* log10(n)) # número de bins com base na fórmula de Sturges  
 return(cut(column, breaks = k, labels = FALSE))  
}

Aplicando a função para todos os dados quantitativos

# dadosQuant1 = dadosQuant  
col\_quant\_qual <- c()  
# Processando cada coluna do dataset  
for (col\_name in names(dadosQuant)) {  
 if (is.numeric(dadosQuant[[col\_name]])) {  
 if (detect\_outliers(dadosQuant[[col\_name]])) {  
 col\_quant\_qual <- c(col\_quant\_qual, col\_name)  
   
 dadosQuali[[col\_name]] <- as.factor(transform\_to\_categories(dadosQuant[[col\_name]]))  
 dadosQuant[[col\_name]] <- NULL  
 cat(sprintf("Coluna '%s' transformada em categorica.\n", col\_name))  
 }  
 }  
}

## Coluna 'x1' transformada em categorica.  
## Coluna 'x3' transformada em categorica.  
## Coluna 'x4' transformada em categorica.  
## Coluna 'x6' transformada em categorica.  
## Coluna 'x7' transformada em categorica.  
## Coluna 'x8' transformada em categorica.  
## Coluna 'x9' transformada em categorica.  
## Coluna 'x10' transformada em categorica.  
## Coluna 'x11' transformada em categorica.  
## Coluna 'x12' transformada em categorica.  
## Coluna 'x15' transformada em categorica.  
## Coluna 'x16' transformada em categorica.  
## Coluna 'x17' transformada em categorica.  
## Coluna 'x19' transformada em categorica.  
## Coluna 'x23' transformada em categorica.  
## Coluna 'x24' transformada em categorica.  
## Coluna 'x25' transformada em categorica.  
## Coluna 'x26' transformada em categorica.  
## Coluna 'x29' transformada em categorica.  
## Coluna 'x30' transformada em categorica.  
## Coluna 'x31' transformada em categorica.  
## Coluna 'x32' transformada em categorica.  
## Coluna 'x33' transformada em categorica.  
## Coluna 'x35' transformada em categorica.  
## Coluna 'x36' transformada em categorica.  
## Coluna 'x37' transformada em categorica.  
## Coluna 'x38' transformada em categorica.  
## Coluna 'x39' transformada em categorica.  
## Coluna 'x40' transformada em categorica.  
## Coluna 'x41' transformada em categorica.  
## Coluna 'x42' transformada em categorica.  
## Coluna 'x43' transformada em categorica.  
## Coluna 'x44' transformada em categorica.  
## Coluna 'x46' transformada em categorica.  
## Coluna 'x47' transformada em categorica.  
## Coluna 'x48' transformada em categorica.  
## Coluna 'x49' transformada em categorica.  
## Coluna 'x51' transformada em categorica.  
## Coluna 'x52' transformada em categorica.  
## Coluna 'x54' transformada em categorica.  
## Coluna 'x55' transformada em categorica.  
## Coluna 'x57' transformada em categorica.  
## Coluna 'x58' transformada em categorica.  
## Coluna 'x60' transformada em categorica.  
## Coluna 'x61' transformada em categorica.  
## Coluna 'x63' transformada em categorica.  
## Coluna 'x64' transformada em categorica.  
## Coluna 'x65' transformada em categorica.  
## Coluna 'x66' transformada em categorica.  
## Coluna 'x68' transformada em categorica.  
## Coluna 'x69' transformada em categorica.  
## Coluna 'x70' transformada em categorica.  
## Coluna 'x71' transformada em categorica.  
## Coluna 'x73' transformada em categorica.  
## Coluna 'x76' transformada em categorica.  
## Coluna 'x77' transformada em categorica.  
## Coluna 'x79' transformada em categorica.  
## Coluna 'x80' transformada em categorica.  
## Coluna 'x81' transformada em categorica.  
## Coluna 'x84' transformada em categorica.  
## Coluna 'x85' transformada em categorica.  
## Coluna 'x86' transformada em categorica.  
## Coluna 'x87' transformada em categorica.  
## Coluna 'x88' transformada em categorica.  
## Coluna 'x89' transformada em categorica.  
## Coluna 'x93' transformada em categorica.  
## Coluna 'x95' transformada em categorica.  
## Coluna 'x96' transformada em categorica.  
## Coluna 'x97' transformada em categorica.  
## Coluna 'x98' transformada em categorica.  
## Coluna 'x99' transformada em categorica.  
## Coluna 'x102' transformada em categorica.  
## Coluna 'x103' transformada em categorica.  
## Coluna 'x104' transformada em categorica.  
## Coluna 'x106' transformada em categorica.

Normalizando os dados quantitativos que não apresentaram outliers para o treinamento do modelo

# Normalização para o intervalo [0, 1]  
df\_norm <- dadosQuant %>%  
 # select(-target) %>%  
 mutate(across(everything(), ~(. - min(.)) / (max(.) - min(.))))

Análise de correlacao de Pearson para variaveis quantitativas que não possuem outlier

# selecionar somente as variaveis quantitativas  
  
df\_corr <- cbind(df\_norm,target)  
  
correlacao <- cor(df\_corr)  
correlacao

## x13 x14 x18 x20 x21  
## x13 1.00000000000 -0.00598272551 -0.0037602424 -0.0022639848 -0.004596051  
## x14 -0.00598272551 1.00000000000 -0.0042285731 -0.0029249558 0.002036772  
## x18 -0.00376024238 -0.00422857312 1.0000000000 -0.0054865605 -0.001034281  
## x20 -0.00226398483 -0.00292495581 -0.0054865605 1.0000000000 0.001606259  
## x21 -0.00459605084 0.00203677191 -0.0010342814 0.0016062593 1.000000000  
## x22 -0.00008053166 0.00016023451 -0.0118438760 -0.0047334351 0.014536422  
## x34 -0.00953778523 0.00534132459 0.0073717893 -0.0009417172 -0.002492874  
## x45 -0.00903454197 0.00746630224 -0.0002073913 0.0065096791 0.006165096  
## x50 -0.00788729475 -0.00260211289 0.0016405970 0.0077699132 -0.009060700  
## x56 0.00590460432 -0.00340278568 0.0088211426 0.0041904956 -0.001386876  
## x72 0.00169984059 0.00800765914 0.0056380057 -0.0146491428 -0.003797914  
## x74 -0.00520809340 -0.00007245535 -0.0099645502 0.0070827686 0.015737275  
## x75 -0.00432361410 -0.00464059215 0.0104857950 0.1152482919 -0.012584604  
## x82 -0.01631469054 0.00573777528 0.0018167358 -0.0014314216 0.008069647  
## x83 -0.00165416904 -0.00993980638 0.0041580712 0.0024250334 -0.007503694  
## x90 -0.00853386638 -0.00187285765 0.0015919383 0.0072413220 -0.009166028  
## x92 -0.00025690679 -0.00008627035 -0.0117490951 -0.0048173844 0.014755992  
## x94 -0.00258902431 -0.00343299960 -0.0031578495 0.0031112847 -0.004372723  
## x101 -0.00226838870 -0.00291378800 -0.0054679966 0.9999987980 0.001626191  
## x105 -0.00228853719 -0.00291321011 0.9800415672 -0.0055062632 0.001122327  
## target -0.06715429390 0.00790445806 0.1452504090 -0.2158846790 -0.012575161  
## x22 x34 x45 x50 x56  
## x13 -0.00008053166 -0.0095377852 -0.0090345420 -0.0078872947 0.00590460432  
## x14 0.00016023451 0.0053413246 0.0074663022 -0.0026021129 -0.00340278568  
## x18 -0.01184387598 0.0073717893 -0.0002073913 0.0016405970 0.00882114261  
## x20 -0.00473343511 -0.0009417172 0.0065096791 0.0077699132 0.00419049558  
## x21 0.01453642176 -0.0024928735 0.0061650959 -0.0090607002 -0.00138687647  
## x22 1.00000000000 -0.0094208876 0.0041934812 -0.0063576533 -0.00302503110  
## x34 -0.00942088765 1.0000000000 -0.0048237239 0.0025220260 0.00447409226  
## x45 0.00419348118 -0.0048237239 1.0000000000 -0.0048712593 0.00262203861  
## x50 -0.00635765329 0.0025220260 -0.0048712593 1.0000000000 0.00047677425  
## x56 -0.00302503110 0.0044740923 0.0026220386 0.0004767743 1.00000000000  
## x72 -0.00095757771 -0.0010341108 -0.0029852720 0.0006367332 -0.00001298569  
## x74 0.00022795502 0.0030486205 0.0159292408 -0.0027103459 0.00795898854  
## x75 0.00597101367 -0.0106605471 0.0016876846 -0.0036602641 0.00399255164  
## x82 -0.00879022669 0.0010324911 -0.0016566013 -0.0017651893 0.00140213836  
## x83 0.00686237553 0.0051527378 -0.0025616123 0.0047557437 0.00477399282  
## x90 -0.00642709697 0.0025243147 -0.0053147533 0.9985249883 0.00068894189  
## x92 0.99963861574 -0.0096493777 0.0042182701 -0.0064517063 -0.00310447615  
## x94 -0.00334997976 -0.0062194707 0.0006898639 -0.0090652502 -0.00478078074  
## x101 -0.00473764814 -0.0009344124 0.0065085650 0.0077687677 0.00422788260  
## x105 -0.01078186241 0.0089623676 0.0009217907 0.0002719618 0.00969426392  
## target 0.01167759203 0.0148289084 -0.1346590433 -0.0873657660 0.03729793409  
## x72 x74 x75 x82 x83  
## x13 0.00169984059 -0.00520809340 -0.004323614 -0.016314691 -0.0016541690  
## x14 0.00800765914 -0.00007245535 -0.004640592 0.005737775 -0.0099398064  
## x18 0.00563800567 -0.00996455018 0.010485795 0.001816736 0.0041580712  
## x20 -0.01464914276 0.00708276858 0.115248292 -0.001431422 0.0024250334  
## x21 -0.00379791352 0.01573727519 -0.012584604 0.008069647 -0.0075036943  
## x22 -0.00095757771 0.00022795502 0.005971014 -0.008790227 0.0068623755  
## x34 -0.00103411083 0.00304862049 -0.010660547 0.001032491 0.0051527378  
## x45 -0.00298527195 0.01592924081 0.001687685 -0.001656601 -0.0025616123  
## x50 0.00063673321 -0.00271034590 -0.003660264 -0.001765189 0.0047557437  
## x56 -0.00001298569 0.00795898854 0.003992552 0.001402138 0.0047739928  
## x72 1.00000000000 -0.00116963418 0.003685356 -0.003697962 0.0030294716  
## x74 -0.00116963418 1.00000000000 -0.001823967 0.004718025 0.0002341786  
## x75 0.00368535632 -0.00182396747 1.000000000 -0.008772800 0.0067373176  
## x82 -0.00369796175 0.00471802487 -0.008772800 1.000000000 0.0114305186  
## x83 0.00302947159 0.00023417864 0.006737318 0.011430519 1.0000000000  
## x90 0.00128709978 -0.00211409964 -0.003428911 -0.001822815 0.0045040888  
## x92 -0.00118396614 0.00003180002 0.006065699 -0.008998138 0.0066496635  
## x94 0.00313814445 -0.00863650824 0.004916130 0.004127002 -0.0065767353  
## x101 -0.01466961350 0.00708863885 0.115240394 -0.001429513 0.0024346589  
## x105 0.00580264588 -0.00927725635 0.009450146 -0.001656447 0.0071582652  
## target 0.01188435440 -0.01720474421 -0.155972836 -0.105159738 0.0304525124  
## x90 x92 x94 x101 x105  
## x13 -0.0085338664 -0.00025690679 -0.0025890243 -0.0022683887 -0.0022885372  
## x14 -0.0018728576 -0.00008627035 -0.0034329996 -0.0029137880 -0.0029132101  
## x18 0.0015919383 -0.01174909514 -0.0031578495 -0.0054679966 0.9800415672  
## x20 0.0072413220 -0.00481738438 0.0031112847 0.9999987980 -0.0055062632  
## x21 -0.0091660280 0.01475599194 -0.0043727232 0.0016261905 0.0011223270  
## x22 -0.0064270970 0.99963861574 -0.0033499798 -0.0047376481 -0.0107818624  
## x34 0.0025243147 -0.00964937767 -0.0062194707 -0.0009344124 0.0089623676  
## x45 -0.0053147533 0.00421827011 0.0006898639 0.0065085650 0.0009217907  
## x50 0.9985249883 -0.00645170628 -0.0090652502 0.0077687677 0.0002719618  
## x56 0.0006889419 -0.00310447615 -0.0047807807 0.0042278826 0.0096942639  
## x72 0.0012870998 -0.00118396614 0.0031381445 -0.0146696135 0.0058026459  
## x74 -0.0021140996 0.00003180002 -0.0086365082 0.0070886388 -0.0092772564  
## x75 -0.0034289114 0.00606569944 0.0049161305 0.1152403936 0.0094501465  
## x82 -0.0018228145 -0.00899813781 0.0041270022 -0.0014295125 -0.0016564474  
## x83 0.0045040888 0.00664966354 -0.0065767353 0.0024346589 0.0071582652  
## x90 1.0000000000 -0.00653577516 -0.0092887035 0.0072404216 0.0001420905  
## x92 -0.0065357752 1.00000000000 -0.0034806790 -0.0048211111 -0.0107083671  
## x94 -0.0092887035 -0.00348067902 1.0000000000 0.0031129304 -0.0061978896  
## x101 0.0072404216 -0.00482111111 0.0031129304 1.0000000000 -0.0054868242  
## x105 0.0001420905 -0.01070836711 -0.0061978896 -0.0054868242 1.0000000000  
## target -0.0875410773 0.01156129594 0.1496535008 -0.2158469989 0.1437556049  
## target  
## x13 -0.067154294  
## x14 0.007904458  
## x18 0.145250409  
## x20 -0.215884679  
## x21 -0.012575161  
## x22 0.011677592  
## x34 0.014828908  
## x45 -0.134659043  
## x50 -0.087365766  
## x56 0.037297934  
## x72 0.011884354  
## x74 -0.017204744  
## x75 -0.155972836  
## x82 -0.105159738  
## x83 0.030452512  
## x90 -0.087541077  
## x92 0.011561296  
## x94 0.149653501  
## x101 -0.215846999  
## x105 0.143755605  
## target 1.000000000

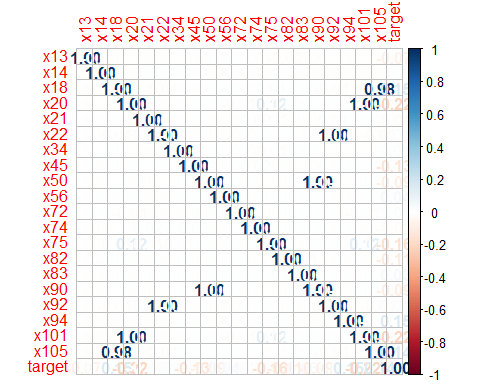
mc <- correlacao

Matriz de correlação das variaveis quantitativas com a target

library(corrplot)

## corrplot 0.92 loaded

#corrplot(mc)  
corrplot(mc, type="full", method="number")



### Variáveis qualitativas

Criando as variaveis dummy

# Criar uma lista de todas as colunas categóricas, exceto 'x5'  
# columns\_to\_dummy <- setdiff(names(dadosQuali), c("x5", "x28", "x67"))  
  
# criar variaveis dummies das variaveis preditoras qualitativas  
var\_dummies <- dummy\_cols(dadosQuali, select\_columns = names(dadosQuali),  
 remove\_first\_dummy = TRUE,  
 remove\_selected\_columns = TRUE)

df\_modelo <- cbind(df\_norm,var\_dummies,target)

### g) Selecionar as variáveis preditoras.

Após o tratamento dos dados temos um dataframe com 4261 variáveis preditoras.

paste("Preditoras:",length(names(df\_modelo))-1)

## [1] "Preditoras: 4261"

### h) Definir a variável resposta.

Após o tratamento dos dados temos um dataframe com 1 coluna target.

paste("Target",length(names(df\_modelo))-4261)

## [1] "Target 1"

### i) Rodar o modelo de Regressão Linear Múltipla.

Dividir a amostra em treino e validação (70/30 %)

#Dividir em duas amostras  
set.seed(1010)  
train <- sample(nrow(df\_modelo), 0.7\*nrow(df\_modelo), replace = FALSE)  
TrainSet <- df\_modelo[train,]  
ValidSet <- df\_modelo[-train,]

Comparar a variável resposta nas duas amostras

summary(TrainSet$target)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -0.2398 0.5005 0.9821 1.6508 1.9589 48.5965

summary(ValidSet$target)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -0.1290 0.5129 0.9862 1.6842 1.9905 61.3261

## Modelo inicial

modelo\_inicial <- lm(target ~ ., data = TrainSet)  
summary\_modelo\_inicial <- summary(modelo\_inicial)

Resultado modelo inicial

# Imprimir o erro padrão residual  
cat("Residual standard error: ", summary\_modelo\_inicial$sigma, "\n")

Residual standard error: 1.60537

# Imprimir R-squared e R-squared ajustado  
cat("Multiple R-squared: ", summary\_modelo\_inicial$r.squared, "\n")

Multiple R-squared: 0.6034719

cat("Adjusted R-squared: ", summary\_modelo\_inicial$adj.r.squared, "\n")

Adjusted R-squared: 0.4791807

# Calcular e imprimir a estatística F e o p-value  
f\_statistic <- summary\_modelo\_inicial$fstatistic  
p\_valor <- pf(f\_statistic[1], f\_statistic[2], f\_statistic[3], lower.tail = FALSE)  
cat("F-statistic: ", f\_statistic[1], " on ", f\_statistic[2], " and ", f\_statistic[3], " DF, p-value: ", p\_valor, "\n")

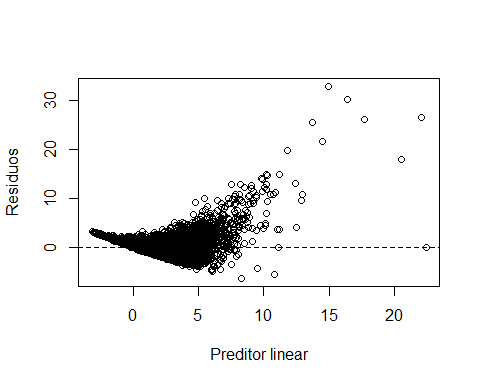
F-statistic: 4.855308 on 4172 and 13310 DF, p-value: 0

### j) Análise de resíduos

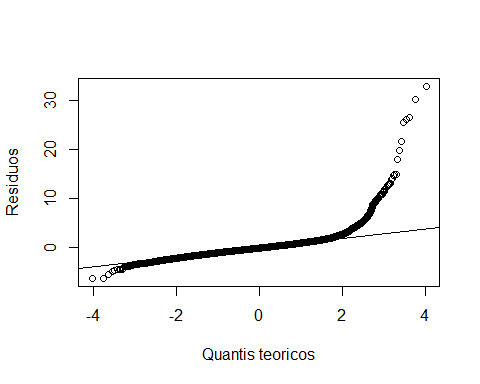
Calculando os resíduos padronizados do modelo

TrainSet$Val\_pred <- predict(modelo\_inicial)   
TrainSet$residuo <- resid(modelo\_inicial)  
TrainSet$rp <- rstandard(modelo\_inicial)

plot(predict(modelo\_inicial),TrainSet$residuo, xlab = "Preditor linear",ylab = "Residuos")  
abline(h = 0, lty = 2)



qqnorm(residuals(modelo\_inicial), ylab="Residuos",xlab="Quantis teoricos",main="")  
qqline(residuals(modelo\_inicial))



Excluindo os outliers

#Excluir os outliers  
TrainSet\_1 <-filter(TrainSet,TrainSet$rp>=-2&TrainSet$rp<=2)   
  
#Pre-processamento dos dados  
# Apaga a coluna   
TrainSet\_1$Val\_pred = NULL  
TrainSet\_1$residuo = NULL  
TrainSet\_1$rp = NULL

### k) Calcular as medidas de erros do modelo na amostra train.csv.

Erro quadratico medio na amostra de treino

mse <- mean((TrainSet$target - TrainSet$Val\_pred)^2)  
sqrt(mse)

## [1] 1.400736

## Modelo1

Treino do modelo1 sem os outliers

modelo1 <- lm(target ~ ., data = TrainSet\_1)  
summary\_modelo1 <- summary(modelo1)

# Imprimir o erro padrão residual  
cat("Residual standard error: ", summary\_modelo1$sigma, "\n")

Residual standard error: 0.8735002

# Imprimir R-squared e R-squared ajustado  
cat("Multiple R-squared: ", summary\_modelo1$r.squared, "\n")

Multiple R-squared: 0.7091969

cat("Adjusted R-squared: ", summary\_modelo1$adj.r.squared, "\n")

Adjusted R-squared: 0.6157998

# Calcular e imprimir a estatística F e o p-value  
f\_statistic <- summary\_modelo1$fstatistic  
p\_valor <- pf(f\_statistic[1], f\_statistic[2], f\_statistic[3], lower.tail = FALSE)  
cat("F-statistic: ", f\_statistic[1], " on ", f\_statistic[2], " and ", f\_statistic[3], " DF, p-value: ", p\_valor, "\n")

F-statistic: 7.59335 on 4119 and 12825 DF, p-value: 0

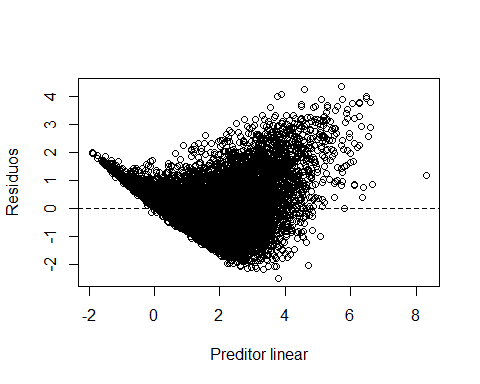
### j) Análise de resíduos

Calculando os resíduos padronizados do modelo1

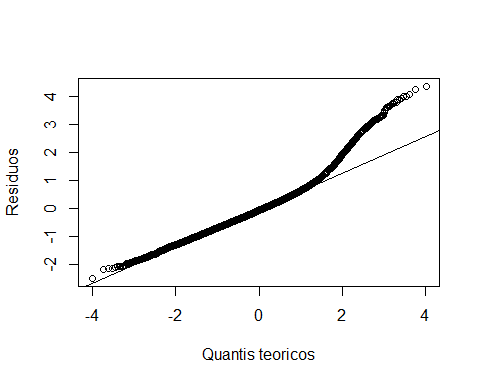
TrainSet\_1$Val\_pred <- predict(modelo1)   
TrainSet\_1$residuo <- resid(modelo1)  
TrainSet\_1$rp <- rstandard(modelo1)

Análise de resíduos modelo1

plot(predict(modelo1),TrainSet\_1$residuo, xlab = "Preditor linear",ylab = "Residuos")  
abline(h = 0, lty = 2)



qqnorm(residuals(modelo1), ylab="Residuos",xlab="Quantis teoricos",main="")  
qqline(residuals(modelo1))



Excluindo os outliers modelo1

#Excluir os outliers  
TrainSet\_2 <-filter(TrainSet\_1,TrainSet\_1$rp>=-2&TrainSet\_1$rp<=2)   
  
#Pre-processamento dos dados  
# Apaga a coluna   
TrainSet\_2$Val\_pred = NULL  
TrainSet\_2$residuo = NULL  
TrainSet\_2$rp = NULL

### k) Calcular as medidas de erros do modelo na amostra train.csv.

Erro quadratico medio na amostra de treino modelo1

# Erro quadratico medio na amostra de treino  
mse1 <- mean((TrainSet\_1$target - TrainSet\_1$Val\_pred)^2)  
sqrt(mse1)

## [1] 0.7599253

## Modelo2

Treino do modelo2 sem os outliers

modelo2 <- lm(target ~ ., data = TrainSet\_2)  
summary\_modelo2 <- summary(modelo2)

# Imprimir o erro padrão residual  
cat("Residual standard error: ", summary\_modelo2$sigma, "\n")

Residual standard error: 0.6249221

# Imprimir R-squared e R-squared ajustado  
cat("Multiple R-squared: ", summary\_modelo2$r.squared, "\n")

Multiple R-squared: 0.7588375

cat("Adjusted R-squared: ", summary\_modelo2$adj.r.squared, "\n")

Adjusted R-squared: 0.6765441

# Calcular e imprimir a estatística F e o p-value  
f\_statistic <- summary\_modelo2$fstatistic  
p\_valor <- pf(f\_statistic[1], f\_statistic[2], f\_statistic[3], lower.tail = FALSE)  
cat("F-statistic: ", f\_statistic[1], " on ", f\_statistic[2], " and ", f\_statistic[3], " DF, p-value: ", p\_valor, "\n")

F-statistic: 9.221124 on 4102 and 12021 DF, p-value: 0

### j) Análise de resíduos

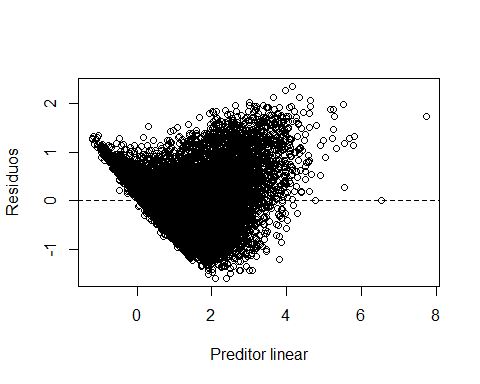
Calculando os resíduos padronizados do modelo2

TrainSet\_2$Val\_pred <- predict(modelo2)   
TrainSet\_2$residuo <- resid(modelo2)  
TrainSet\_2$rp <- rstandard(modelo2)

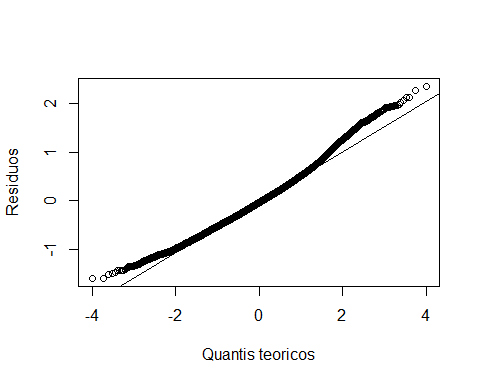
df\_analise2 <- TrainSet\_2 %>%  
 select(target, Val\_pred, residuo, rp)

Análise de resíduos modelo2

plot(predict(modelo2),TrainSet\_2$residuo, xlab = "Preditor linear",ylab = "Residuos")  
abline(h = 0, lty = 2)



qqnorm(residuals(modelo2), ylab="Residuos",xlab="Quantis teoricos",main="")  
qqline(residuals(modelo2))



Excluindo os outliers modelo2

#Excluir os outliers  
TrainSet\_3 <-filter(TrainSet\_2,TrainSet\_2$rp>=-2&TrainSet\_2$rp<=2)   
  
#Pre-processamento dos dados  
# Apaga a coluna   
TrainSet\_3$Val\_pred = NULL  
TrainSet\_3$residuo = NULL  
TrainSet\_3$rp = NULL

### k) Calcular as medidas de erros do modelo na amostra train.csv.

Erro quadratico medio na amostra de treino modelo2

# Erro quadratico medio na amostra de treino  
mse2 <- mean((TrainSet\_2$target - TrainSet\_2$Val\_pred)^2)  
sqrt(mse2)

## [1] 0.5395849

## Modelo3

Treino do modelo3 sem os outliers

modelo3 <- lm(target ~ ., data = TrainSet\_3)  
summary\_modelo3 <- summary(modelo3)

# Imprimir o erro padrão residual  
cat("Residual standard error: ", summary\_modelo3$sigma, "\n")

Residual standard error: 0.4950465

# Imprimir R-squared e R-squared ajustado  
cat("Multiple R-squared: ", summary\_modelo3$r.squared, "\n")

Multiple R-squared: 0.7922323

cat("Adjusted R-squared: ", summary\_modelo3$adj.r.squared, "\n")

Adjusted R-squared: 0.7164532

# Calcular e imprimir a estatística F e o p-value  
f\_statistic <- summary\_modelo3$fstatistic  
p\_valor <- pf(f\_statistic[1], f\_statistic[2], f\_statistic[3], lower.tail = FALSE)  
cat("F-statistic: ", f\_statistic[1], " on ", f\_statistic[2], " and ", f\_statistic[3], " DF, p-value: ", p\_valor, "\n")

F-statistic: 10.45449 on 4093 and 11222 DF, p-value: 0

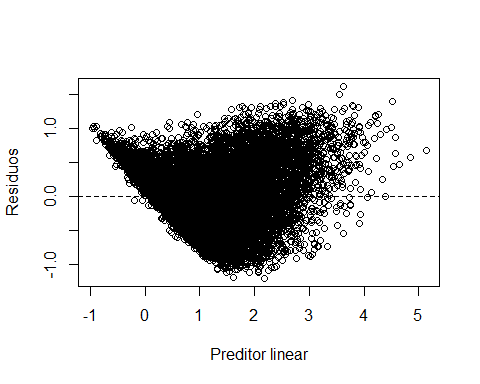
### j) Análise de resíduos

Calculando os resíduos padronizados do modelo3

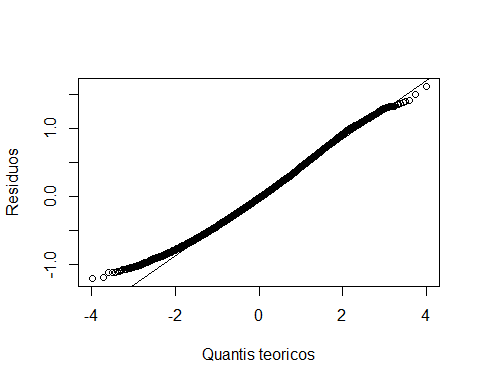
TrainSet\_3$Val\_pred <- predict(modelo3)   
TrainSet\_3$residuo <- resid(modelo3)  
TrainSet\_3$rp <- rstandard(modelo3)

Análise de resíduos modelo3

plot(predict(modelo3),TrainSet\_3$residuo, xlab = "Preditor linear",ylab = "Residuos")  
abline(h = 0, lty = 2)



qqnorm(residuals(modelo3), ylab="Residuos",xlab="Quantis teoricos",main="")  
qqline(residuals(modelo3))



### k) Calcular as medidas de erros do modelo na amostra train.csv.

Erro quadratico medio na amostra de treino modelo3

# Erro quadratico medio na amostra de treino  
mse3 <- mean((TrainSet\_3$target - TrainSet\_3$Val\_pred)^2)  
sqrt(mse3)

## [1] 0.4237488

# AMOSTRA DE VALIDACAO

Função para calcular o R² MSE e RMSE das amostras de validação

# Função para detectar se uma coluna tem outliers  
avaliacao\_modelo <- function(dataset, predictions) {  
 # Calculando o MSE  
 mse <- mean((dataset$target - predictions)^2)  
   
 # Calculando o RMSE  
 rmse <- sqrt(mse)  
   
 # Calculando o MAE  
 mae <- mean(abs(dataset$target - predictions))  
   
 # Calcular a soma dos quadrados dos resíduos (SS\_res)  
 SS\_res <- sum((dataset$target - predictions)^2)  
   
 # Calcular a soma total dos quadrados (SS\_tot)  
 SS\_tot <- sum((dataset$target - mean(dataset$target))^2)  
   
 # Calcular o R² nos dados de validação  
 R2\_validation <- 1 - (SS\_res / SS\_tot)  
   
 # Imprimindo os resultados  
 # Imprimir o R²  
 cat("R²:", R2\_validation, "\n")  
 cat("MSE:", mse, "\n")  
 cat("RMSE:", rmse, "\n")  
 # cat("MAE:", mae, "\n")  
 }

Realizando as predições para os dados de validação em todos os modelos para fins de comparação

# Fazer previsões nos dados de validação  
predictions <- predict(modelo\_inicial, newdata = ValidSet)  
predictions1 <- predict(modelo1, newdata = ValidSet)  
predictions2 <- predict(modelo2, newdata = ValidSet)  
predictions3 <- predict(modelo3, newdata = ValidSet)

summary\_modelo\_inicial <- summary(modelo\_inicial)  
summary\_modelo1 <- summary(modelo1)  
summary\_modelo2 <- summary(modelo2)  
summary\_modelo3 <- summary(modelo3)

# Resultado Final

Medidas de erros dos modelos com os dados de treinamento e validação

# Fazer previsões nos dados de validação  
  
cat("\n","-----------Avaliação Modelo Inicial:-----------","\n",sep = "")

cat("\n","Treino:","\n",sep = "")

cat("R² ajustado:", summary\_modelo\_inicial$adj.r.squared, "\n")

cat("R²:", summary\_modelo\_inicial$r.squared, "\n")

cat("MSE:", mse, "\n")

cat("RMSE:", sqrt(mse), "\n")

cat("\n","Validação:","\n",sep = "")

cat("\n","Validação:","\n",sep = "")

avaliacao\_modelo(ValidSet,predictions)

cat("\n","-----------Avaliação Modelo1:-----------","\n",sep = "")

cat("\n","Treino:","\n",sep = "")

cat("R² ajustado:", summary\_modelo1$adj.r.squared, "\n")

cat("R²:", summary\_modelo1$r.squared, "\n")

cat("MSE:", mse1, "\n")

cat("RMSE:", sqrt(mse1), "\n")

cat("\n","Validação:","\n",sep = "")

avaliacao\_modelo(ValidSet,predictions1)

cat("\n","-----------Avaliação Modelo2:-----------","\n",sep = "")

cat("\n","Treino:","\n",sep = "")

cat("R² ajustado:", summary\_modelo2$adj.r.squared, "\n")

cat("R²:", summary\_modelo2$r.squared, "\n")

cat("MSE:", mse2, "\n")

cat("RMSE:", sqrt(mse2), "\n")

cat("\n","Validação:","\n",sep = "")

avaliacao\_modelo(ValidSet,predictions2)

cat("\n","-----------Avaliação Modelo3:-----------","\n",sep = "")

cat("\n","Treino:","\n",sep = "")

cat("R² ajustado:", summary\_modelo3$adj.r.squared, "\n")

cat("R²:", summary\_modelo3$r.squared, "\n")

cat("MSE:", mse3, "\n")

cat("RMSE:", sqrt(mse3), "\n")

cat("\n","Validação:","\n",sep = "")

avaliacao\_modelo(ValidSet,predictions3)

———–Avaliação Modelo Inicial:———–

Treino:

R² ajustado: 0.4791807

R²: 0.6034719

MSE: 1.962061

RMSE: 1.400736

Validação:

R²: 0.2911947

MSE: 3.789938

RMSE: 1.946776

———–Avaliação Modelo1:———–

Treino:

R² ajustado: 0.6157911

R²: 0.7091904

MSE: 0.5774865

RMSE: 0.7599253

Validação:

R²: 0.3650671

MSE: 3.394947

RMSE: 1.842538

———–Avaliação Modelo2:———–

Treino:

R² ajustado: 0.6766769

R²: 0.7589813

MSE: 0.2910977

RMSE: 0.5395347

Validação:

R²: 0.33936

MSE: 3.532402

RMSE: 1.879468

———–Avaliação Modelo3:———–

Treino:

R² ajustado: 0.7164942

R²: 0.7923031

MSE: 0.1795279

RMSE: 0.4237073

Validação:

R²: 0.3069979

MSE: 3.70544

RMSE: 1.924952

## M. Import dos dados da amostra test.csv

df\_test <- read\_csv("./data/test.csv")

Antes de aplicarmos o modelo devemos fazer as mesmas tratativas nos dados, feitas anteriormente nos dados de treinamento.

#### Passo 1

Converter as colunas com os nomes x1,x2,x3,…,xn

# Nomes das colunas que você quer renomear  
colunas\_para\_renomear <- setdiff(names(df\_test), c("id", "target"))  
  
# Número de colunas para renomear  
num\_colunas <- length(colunas\_para\_renomear)  
  
# Inicializando o dicionário  
dicionario <- list()  
  
# Renomear as colunas para x1, x2, x3, ... e criar o dicionario  
for (i in 1:num\_colunas) {  
 colname <- colunas\_para\_renomear[i]  
 new\_colname <- paste("x", i, sep="")  
   
 # Adicionando ao dicionário  
 dicionario[[new\_colname]] <- colname  
   
 print(paste(colname, "->", new\_colname))  
   
 # Renomeando a coluna  
 names(df\_test)[names(df\_test) == colname] <- new\_colname  
}

## [1] "016399044a -> x1"  
## [1] "023c68873b -> x2"  
## [1] "0342faceb5 -> x3"  
## [1] "04e7268385 -> x4"  
## [1] "06888ceac9 -> x5"  
## [1] "072b7e8f27 -> x6"  
## [1] "087235d61e -> x7"  
## [1] "0b846350ef -> x8"  
## [1] "0e2ab0831c -> x9"  
## [1] "12eda2d982 -> x10"  
## [1] "136c1727c3 -> x11"  
## [1] "173b6590ae -> x12"  
## [1] "174825d438 -> x13"  
## [1] "1f222e3669 -> x14"  
## [1] "1f3058af83 -> x15"  
## [1] "1fa099bb01 -> x16"  
## [1] "20f1afc5c7 -> x17"  
## [1] "253eb5ef11 -> x18"  
## [1] "25bbf0e7e7 -> x19"  
## [1] "2719b72c0d -> x20"  
## [1] "298ed82b22 -> x21"  
## [1] "29bbd86997 -> x22"  
## [1] "2a457d15d9 -> x23"  
## [1] "2bc6ab42f7 -> x24"  
## [1] "2d7fe4693a -> x25"  
## [1] "2e874bc151 -> x26"  
## [1] "361f93f4d1 -> x27"  
## [1] "384bec5dd1 -> x28"  
## [1] "3df2300fa2 -> x29"  
## [1] "3e200bf766 -> x30"  
## [1] "3eb53ae932 -> x31"  
## [1] "435dec85e2 -> x32"  
## [1] "4468394575 -> x33"  
## [1] "49756d8e0f -> x34"  
## [1] "4fc17427c8 -> x35"  
## [1] "55907cc1de -> x36"  
## [1] "55cf3f7627 -> x37"  
## [1] "56371466d7 -> x38"  
## [1] "5b862c0a8f -> x39"  
## [1] "5f360995ef -> x40"  
## [1] "60ec1426ce -> x41"  
## [1] "63bcf89b1d -> x42"  
## [1] "6516422788 -> x43"  
## [1] "65aed7dc1f -> x44"  
## [1] "6db53d265a -> x45"  
## [1] "7734c0c22f -> x46"  
## [1] "7743f273c2 -> x47"  
## [1] "779d13189e -> x48"  
## [1] "77b3b41efa -> x49"  
## [1] "7841b6a5b1 -> x50"  
## [1] "789b5244a9 -> x51"  
## [1] "7925993f42 -> x52"  
## [1] "7cb7913148 -> x53"  
## [1] "7fe6cb4c98 -> x54"  
## [1] "8311343404 -> x55"  
## [1] "87b982928b -> x56"  
## [1] "8a21502326 -> x57"  
## [1] "8c2e088a3d -> x58"  
## [1] "8d0606b150 -> x59"  
## [1] "8de0382f02 -> x60"  
## [1] "8f5f7c556a -> x61"  
## [1] "91145d159d -> x62"  
## [1] "96c30c7eef -> x63"  
## [1] "96e6f0be58 -> x64"  
## [1] "98475257f7 -> x65"  
## [1] "99d44111c9 -> x66"  
## [1] "9a575e82a4 -> x67"  
## [1] "9b6e0b36c2 -> x68"  
## [1] "a14fd026ce -> x69"  
## [1] "a24802caa5 -> x70"  
## [1] "aa69c802b6 -> x71"  
## [1] "abca7a848f -> x72"  
## [1] "ac826f0013 -> x73"  
## [1] "ae08d2297e -> x74"  
## [1] "aee1e4fc85 -> x75"  
## [1] "b4112a94a6 -> x76"  
## [1] "b709f75447 -> x77"  
## [1] "b835dfe10f -> x78"  
## [1] "b9a487ac3c -> x79"  
## [1] "ba54a2a637 -> x80"  
## [1] "bdf934caa7 -> x81"  
## [1] "beb6e17af1 -> x82"  
## [1] "c0c3df65b1 -> x83"  
## [1] "c1b8ce2354 -> x84"  
## [1] "c58f611921 -> x85"  
## [1] "d035af6ffa -> x86"  
## [1] "d2c775fa99 -> x87"  
## [1] "d4d6566f9c -> x88"  
## [1] "dcfcbc2ea1 -> x89"  
## [1] "e0a0772df0 -> x90"  
## [1] "e16e640635 -> x91"  
## [1] "e5efa4d39a -> x92"  
## [1] "e7ee22effb -> x93"  
## [1] "e86a2190c1 -> x94"  
## [1] "ea0f4a32e3 -> x95"  
## [1] "ed7e658a27 -> x96"  
## [1] "ee2ac696ff -> x97"  
## [1] "f013b60e50 -> x98"  
## [1] "f0a0febd35 -> x99"  
## [1] "f1f0984934 -> x100"  
## [1] "f66b98dd69 -> x101"  
## [1] "fbf66c8021 -> x102"  
## [1] "fdf8628ca7 -> x103"  
## [1] "fe0318e273 -> x104"  
## [1] "fe8cdd80ba -> x105"  
## [1] "ffd1cdcfc1 -> x106"

#### Passo 2

Transformar as colunas qualitativas em factor

df\_test$x2 <- factor(df\_test$x2)  
df\_test$x5 <- factor(df\_test$x5)  
df\_test$x27 <- factor(df\_test$x27)  
df\_test$x28 <- factor(df\_test$x28)  
df\_test$x53 <- factor(df\_test$x53, ordered=TRUE)  
df\_test$x59 <- factor(df\_test$x59)  
df\_test$x62 <- factor(df\_test$x62)  
df\_test$x67 <- factor(df\_test$x67)  
df\_test$x78 <- factor(df\_test$x78)  
df\_test$x91 <- factor(df\_test$x91)  
df\_test$x100 <- factor(df\_test$x100)

#### Passo 3

Separar as variáveis qualitativas e quantitativas

dadosQuali\_test <- df\_test %>%  
 select(x2,x5,x27,x28,x53,x59,x62,x67,x78,x91,x100)  
  
dadosQuant\_test <- df\_test %>%  
 select(-c(id,x2,x5,x27,x28,x53,x59,x62,x67,x78,x91,x100))

#### Passo 4

Com base na analise feita nos dados de treinamento, criamos um array com os nomes de todas as colunas que foram transformadas em qualitativas para aplicar o mesmo para os dados de teste

# Processando cada coluna do dataset  
for (col\_name in col\_quant\_qual) {  
   
 dadosQuali\_test[[col\_name]] <- as.factor(transform\_to\_categories(dadosQuant\_test[[col\_name]]))  
 dadosQuant\_test[[col\_name]] <- NULL  
  
}

#### Passo 5

Utilizamos o critério de normalização “min-max” para as variáveis quantitativas dos dados de teste. No entanto, considerando que as amostras de dados de teste podem não possuir os mesmos valores mínimos e máximos observados nos dados de treinamento, estabelecemos uma estratégia alternativa. Criamos dois vetores, denominados training\_maxs e training\_mins, que armazenam os valores de máximo e mínimo usados na normalização dos dados quantitativos de treinamento. Esses valores são então aplicados na normalização dos dados de teste, assegurando consistência no tratamento dos dados.

# Calculando os valores mínimos e máximos para cada coluna do dataframe quantitativo  
training\_mins <- apply(dadosQuant, 2, min, na.rm = TRUE)  
training\_maxs <- apply(dadosQuant, 2, max, na.rm = TRUE)  
  
# Normalizando o conjunto de teste  
df\_test\_norm <- dadosQuant\_test # Presumindo que dadosQuant\_test contém as variáveis quantitativas do conjunto de teste  
for (col\_name in names(training\_mins)) {  
 df\_test\_norm[[col\_name]] <- (dadosQuant\_test[[col\_name]] - training\_mins[col\_name]) / (training\_maxs[col\_name] - training\_mins[col\_name])  
}

#### Passo 6

Criando as variaveis dummy

# criar variaveis dummies das variaveis preditoras qualitativas  
var\_dummies\_test <- dummy\_cols(dadosQuali\_test, select\_columns = names(dadosQuali\_test),  
 remove\_first\_dummy = TRUE,  
 remove\_selected\_columns = TRUE)

#### Passo 7

Juntar todos os dados quantitativos e qualitativos ja tratados para um dadaset que sera utilizado para fazer as predições.

df\_test\_modelo <- cbind(df\_test\_norm,var\_dummies\_test)

#### Passo 8

Devido às diferenças de conteúdo entre o dataframe de teste e o dataframe de treinamento, a geração de variáveis dummies resultou em colunas desalinhadas. Neste ponto, foi crucial garantir que ambos os dataframes tivessem as mesmas colunas. Para isso, comparamos as colunas e ajustamos o conteúdo das variáveis dummies. Quando identificamos colunas ausentes no dataframe de testes, preenchemos essas colunas com o valor zero.

# Passo 2: Identificar colunas que estão faltando nos dados de teste  
training\_column\_names <- colnames(df\_modelo)  
missing\_columns <- setdiff(training\_column\_names, colnames(df\_test\_modelo))  
  
# Passo 3: Adicionar colunas faltantes aos dados de teste  
missing\_data <- matrix(0, nrow = nrow(df\_test\_modelo), ncol = length(missing\_columns))  
colnames(missing\_data) <- missing\_columns  
df\_test\_modelo <- cbind(df\_test\_modelo, missing\_data)  
  
# Passo 4: Identificar e remover colunas extras que estão presentes nos dados de teste mas não nos dados de treinamento  
extra\_columns <- setdiff(colnames(df\_test\_modelo), training\_column\_names)  
df\_test\_modelo <- df\_test\_modelo[, !colnames(df\_test\_modelo) %in% extra\_columns]  
  
# Agora, os dados de teste devem ter exatamente a mesma estrutura que os dados de treinamento,  
# e você pode usar seu modelo treinado para fazer previsões no conjunto de dados de teste.  
  
# Isso reordena as colunas na matriz df\_test\_modelo para corresponder à ordem em training\_column\_names  
df\_test\_modelo <- df\_test\_modelo[, training\_column\_names]

## Predições base de teste

Fazendo as predições na base de dados teste.csv e criando a coluna target com as predições.

predictions\_test <- predict(modelo1, newdata = df\_test\_modelo)  
df\_test$target <- predictions\_test  
head(df\_test)

## # A tibble: 6 × 108  
## x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 x11 x12  
## <dbl> <fct> <dbl> <dbl> <fct> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 6456 46fd5e9b… 5274 36039 1 1020. 22 -0.717 -0.660 43.8 2.90 0.821  
## 2 6376 3e1c31c3… 5360 35886 1 2543. 15 0.112 8.37 76.0 3.30 0.902  
## 3 6293 5638e0d1… 5358 35974 1 1392. 67 0.862 1.80 56.1 1.42 0.888  
## 4 6335 d1079afa… 5305 35905 1 418. 6 207. -1.43 46.4 4.23 0.774  
## 5 6514 30dd7387… 5380 35860 1 1071. 70 -1.44 8.45 61.5 1.74 0.987  
## 6 6502 f2b5aeda… 5427 36053 1 339. 76 1.12 -0.589 32.0 2.60 0.839  
## # ℹ 96 more variables: x13 <dbl>, x14 <dbl>, x15 <dbl>, x16 <dbl>, x17 <dbl>,  
## # x18 <dbl>, x19 <dbl>, x20 <dbl>, x21 <dbl>, x22 <dbl>, x23 <dbl>,  
## # x24 <dbl>, x25 <dbl>, x26 <dbl>, x27 <fct>, x28 <fct>, x29 <dbl>,  
## # x30 <dbl>, x31 <dbl>, x32 <dbl>, x33 <dbl>, x34 <dbl>, x35 <dbl>,  
## # x36 <dbl>, x37 <dbl>, x38 <dbl>, x39 <dbl>, x40 <dbl>, x41 <dbl>,  
## # x42 <dbl>, x43 <dbl>, x44 <dbl>, x45 <dbl>, x46 <dbl>, x47 <dbl>,  
## # x48 <dbl>, x49 <dbl>, x50 <dbl>, x51 <dbl>, x52 <dbl>, x53 <ord>, …

# Conclusão

Durante a execução do trabalho tentamos varias abordagens para melhorar o R²

* Normalizamos as variáveis preditoras quantitativas;
* Transformamos as variáveis preditoras quantitativas em qualitativas usando o método de quartis, criando faixas de valores com sturgers
* Filtramos somente as variáveis preditoras com p-value < 0.05, fizemos testes também com 0.1
* Trocamos a semente várias vezes
* Modificamos a divisão de treino em validação em 70/30, 50/50, 90/10
* Aplicamos um algoritmo para remover multicolinearidade nas qualitativas (Cramér’s V) e nas quantitativas utilizamos a própria matriz para identificar pontualmente os casos.

Fizemos uma combinação de várias ações para tentar aumentar a acurácia do modelo obtendo um R² de 0.3650671 para os dados de validação e um R² de 0.7091904 para os dados de treinamento.