Analise-Risco-Credito

dgoalmeida - projeto do curso da DSA

2021-10-08

% !TEX encoding = UTF-8 Unicode

Estudo sobre classificação para análise de risco de credito

Objetivo desse mini projeto é avaliar o risco de concessão de credito a clientes em uma instituição financeira

```
library(ggplot2)
library(randomForest)
```

```
## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
library(stringr)
library(e1071)
library(caret)
```

```
## Loading required package: lattice
```

Obtendo os dados que serão analisados.

```
credit_df = read.csv('/Users/dgoalmeida/Documents/datascience/data/credit_dataset.csv')
head(credit_df)
```

```
##   credit.rating account.balance credit.duration.months
## 1             1               1                     18
## 2             1               1                      9
## 3             1               2                     12
## 4             1               1                     12
```

```
## 5            1               1                     12
## 6            1               1                     10
##   previous.credit.payment.status credit.purpose credit.amount savings
## 1                              3              2          1049       1
## 2                              3              4          2799       1
## 3                              2              4           841       2
## 4                              3              4          2122       1
## 5                              3              4          2171       1
## 6                              3              4          2241       1
##   employment.duration installment.rate marital.status guarantor
## 1                   1                4              1         1
## 2                   2                2              3         1
## 3                   3                2              1         1
## 4                   2                3              3         1
## 5                   2                4              3         1
## 6                   1                1              3         1
##   residence.duration current.assets age other.credits apartment.type
## 1                  4              2  21             2              1
## 2                  2              1  36             2              1
## 3                  4              1  23             2              1
## 4                  2              1  39             2              1
## 5                  4              2  38             1              2
## 6                  3              1  48             2              1
##   bank.credits occupation dependents telephone foreign.worker
## 1            1          3          1         1              1
## 2            2          3          2         1              1
## 3            1          2          1         1              1
## 4            2          2          2         1              2
## 5            2          2          1         1              2
## 6            2          2          2         1              2
```

```
str(credit_df)
```

```
## 'data.frame':    1000 obs. of  21 variables:
##  $ credit.rating                 : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ account.balance               : int  1 1 2 1 1 1 1 1 3 2 ...
##  $ credit.duration.months        : int  18 9 12 12 12 10 8 6 18 24 ...
##  $ previous.credit.payment.status: int  3 3 2 3 3 3 3 3 3 2 ...
##  $ credit.purpose                : int  2 4 4 4 4 4 4 4 3 3 ...
##  $ credit.amount                 : int  1049 2799 841 2122 2171 2241 3398 1361 1098 3758 ...
##  $ savings                       : int  1 1 2 1 1 1 1 1 1 3 ...
##  $ employment.duration           : int  1 2 3 2 2 1 3 1 1 1 ...
##  $ installment.rate              : int  4 2 2 3 4 1 1 2 4 1 ...
##  $ marital.status                : int  1 3 1 3 3 3 3 3 1 1 ...
##  $ guarantor                     : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ residence.duration            : int  4 2 4 2 4 3 4 4 4 4 ...
##  $ current.assets                : int  2 1 1 1 2 1 1 1 3 4 ...
##  $ age                           : int  21 36 23 39 38 48 39 40 65 23 ...
##  $ other.credits                 : int  2 2 2 2 1 2 2 2 2 2 ...
##  $ apartment.type                : int  1 1 1 1 2 1 2 2 2 1 ...
##  $ bank.credits                  : int  1 2 1 2 2 2 2 1 2 1 ...
##  $ occupation                    : int  3 3 2 2 2 2 2 2 1 1 ...
##  $ dependents                    : int  1 2 1 2 1 2 1 2 1 1 ...
##  $ telephone                     : int  1 1 1 1 1 1 1 1 1 1 ...
```

```
##  $ foreign.worker                : int  1 1 1 2 2 2 2 2 1 1 ...
```

```
summary(credit_df)
```

```
##  credit.rating account.balance credit.duration.months
##  Min.   :0.0   Min.   :1.000   Min.   : 4.0          
##  1st Qu.:0.0   1st Qu.:1.000   1st Qu.:12.0          
##  Median :1.0   Median :2.000   Median :18.0          
##  Mean   :0.7   Mean   :2.183   Mean   :20.9          
##  3rd Qu.:1.0   3rd Qu.:3.000   3rd Qu.:24.0          
##  Max.   :1.0   Max.   :3.000   Max.   :72.0          
##  previous.credit.payment.status credit.purpose  credit.amount      savings     
##  Min.   :1.000                  Min.   :1.000   Min.   :  250   Min.   :1.000  
##  1st Qu.:2.000                  1st Qu.:2.000   1st Qu.: 1366   1st Qu.:1.000  
##  Median :2.000                  Median :3.000   Median : 2320   Median :1.000  
##  Mean   :2.292                  Mean   :2.965   Mean   : 3271   Mean   :1.874  
##  3rd Qu.:3.000                  3rd Qu.:4.000   3rd Qu.: 3972   3rd Qu.:3.000  
##  Max.   :3.000                  Max.   :4.000   Max.   :18424   Max.   :4.000  
##  employment.duration installment.rate marital.status    guarantor    
##  Min.   :1.000       Min.   :1.000    Min.   :1.000   Min.   :1.000  
##  1st Qu.:2.000       1st Qu.:2.000    1st Qu.:1.000   1st Qu.:1.000  
##  Median :2.000       Median :3.000    Median :3.000   Median :1.000  
##  Mean   :2.446       Mean   :2.973    Mean   :2.372   Mean   :1.093  
##  3rd Qu.:4.000       3rd Qu.:4.000    3rd Qu.:3.000   3rd Qu.:1.000  
##  Max.   :4.000       Max.   :4.000    Max.   :4.000   Max.   :2.000  
##  residence.duration current.assets       age        other.credits  
##  Min.   :1.000      Min.   :1.000   Min.   :19.00   Min.   :1.000  
##  1st Qu.:2.000      1st Qu.:1.000   1st Qu.:27.00   1st Qu.:2.000  
##  Median :3.000      Median :2.000   Median :33.00   Median :2.000  
##  Mean   :2.845      Mean   :2.358   Mean   :35.54   Mean   :1.814  
##  3rd Qu.:4.000      3rd Qu.:3.000   3rd Qu.:42.00   3rd Qu.:2.000  
##  Max.   :4.000      Max.   :4.000   Max.   :75.00   Max.   :2.000  
##  apartment.type   bank.credits     occupation      dependents   
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000  
##  1st Qu.:2.000   1st Qu.:1.000   1st Qu.:3.000   1st Qu.:1.000  
##  Median :2.000   Median :1.000   Median :3.000   Median :1.000  
##  Mean   :1.928   Mean   :1.367   Mean   :2.904   Mean   :1.155  
##  3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:3.000   3rd Qu.:1.000  
##  Max.   :3.000   Max.   :2.000   Max.   :4.000   Max.   :2.000  
##    telephone     foreign.worker 
##  Min.   :1.000   Min.   :1.000  
##  1st Qu.:1.000   1st Qu.:1.000  
##  Median :1.000   Median :1.000  
##  Mean   :1.404   Mean   :1.037  
##  3rd Qu.:2.000   3rd Qu.:1.000  
##  Max.   :2.000   Max.   :2.000  
```

Analise dos dados

Analisando dataset

credit_df$credit.ratingavaliaodecredito(variaveltarget)credit_df$account.balance possui saldo credit_df$credit.duration.mont$
status do pagamento anterior credit_df$credit.purposepropsitodocreditocredit_df$credit.amount valor do crédito credit_df$savings?credit_df$employment.duration tempo empregado credit_df$installment.ratetaxadeparcelamentocredit_d$
estqado civil credit_df$guarantorpossuifiadorcredit_df$residence.duration tempo na mesma residencia credit_df$current.assetsquantidadeativoscorrentescredit_df$age idade credit_df$other.creditspossuioutroscreditoscredit_df$ap
tipo de apartamento credit_df$bank.creditspossuicreditonobancocredit_df$occupation tipo de ocupação (trabalho) credit_df$dependentsquantidadedependentescredit_df$telephone possui telefone credit_df$foreign.worker trabalha fora

Análise exploratória dos dados

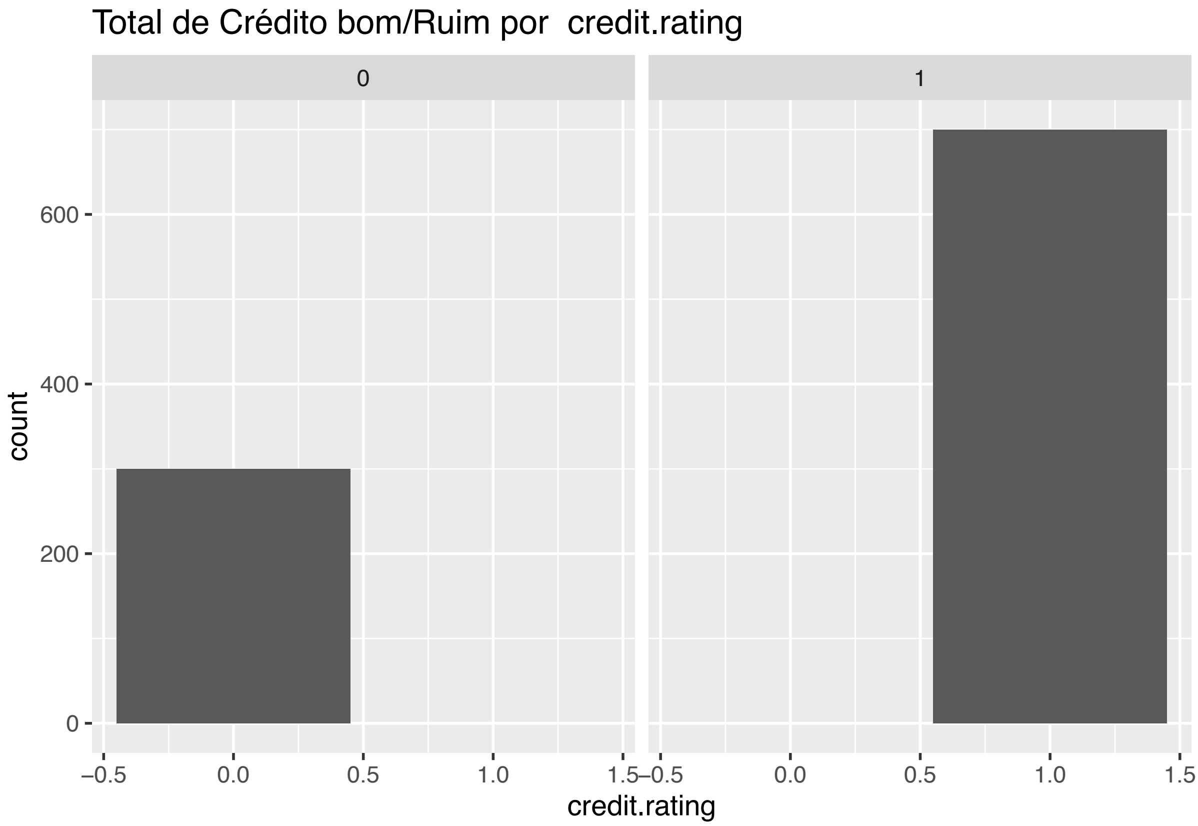
```
#verificando se possuem dados NA
any(is.na(credit_df))
```

```
## [1] FALSE
```

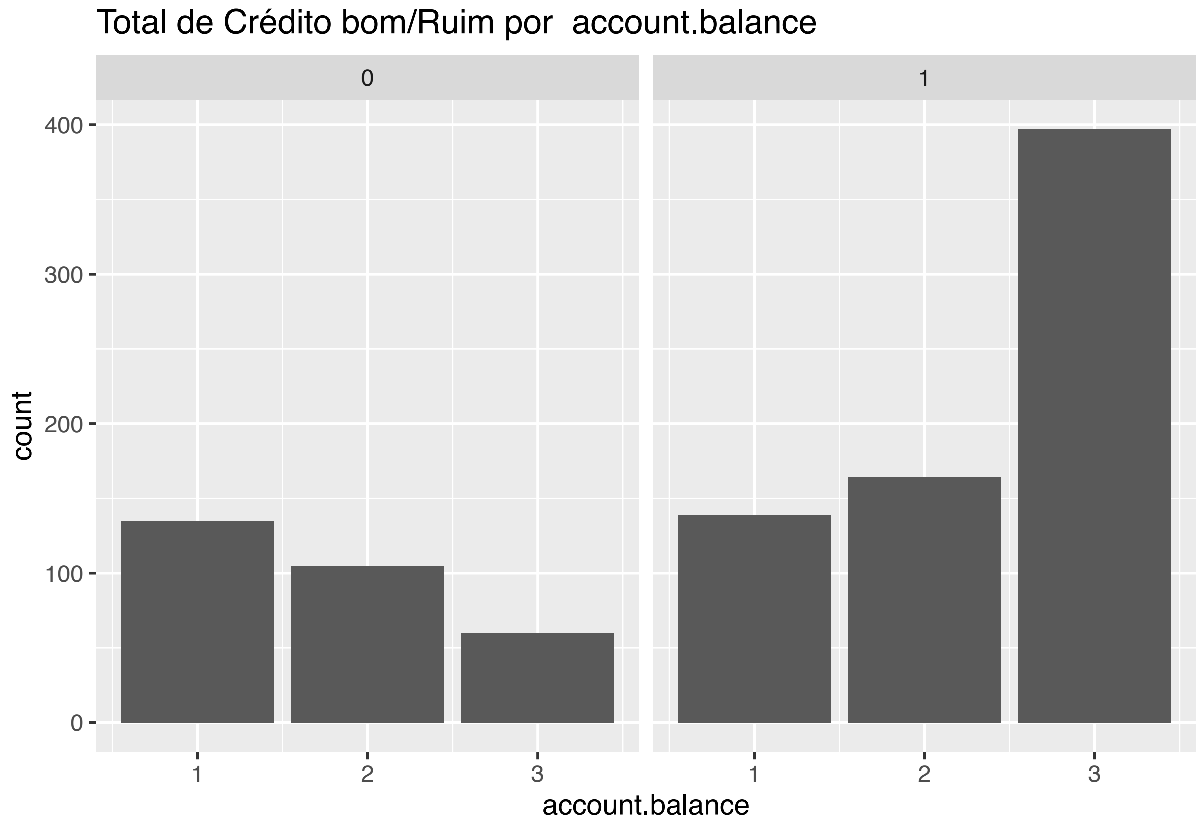
```
# executando a função lapply  que recebe um vetor com o nome das features e  uma função  para plotar um
lapply(colnames(credit_df), function(x){
    ggplot(credit_df, aes_string(x)) +
      geom_bar() +
      facet_grid(. ~ credit.rating) +
      ggtitle(paste("Total de Crédito bom/Ruim por ", x))

})
```

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

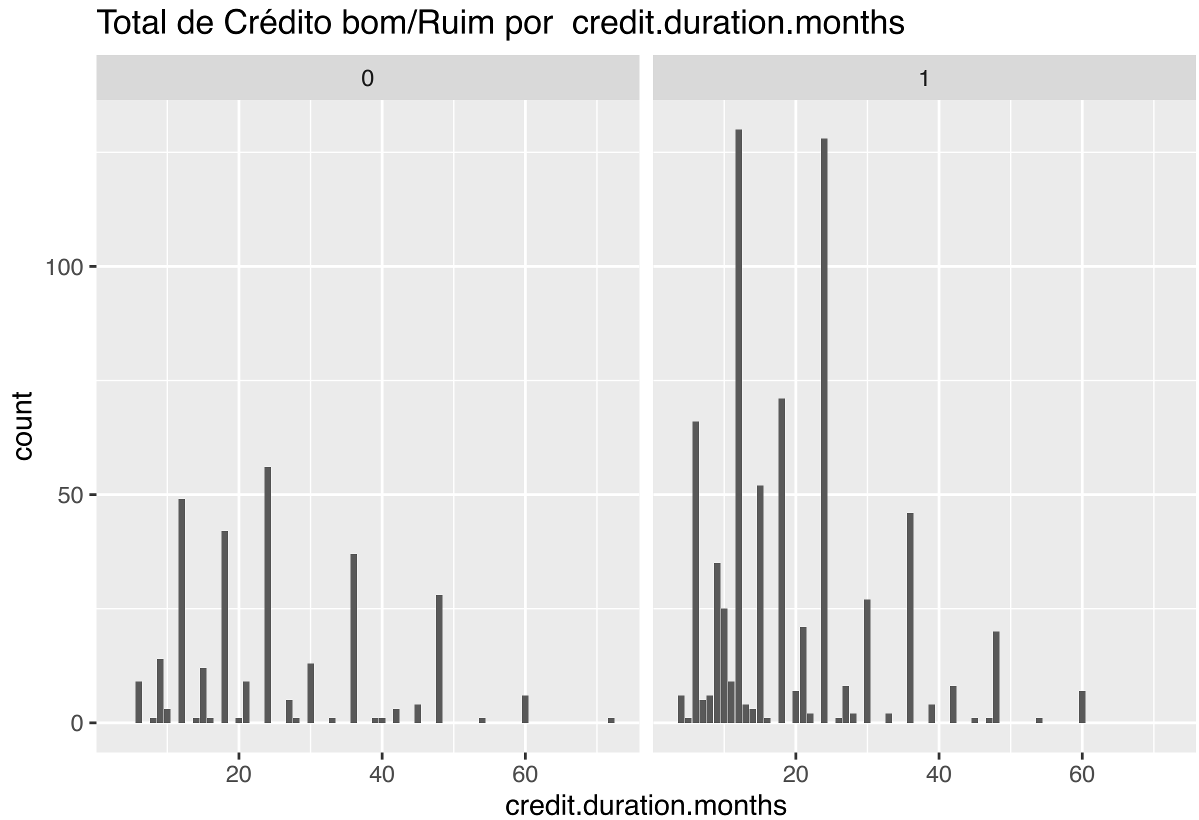

Total de Crédito bom/Ruim por credit.rating
0
1
count
600
400
200
0
–0.5
0.0
0.5
1.0
1.5
credit.rating

```
##
## [[2]]
```

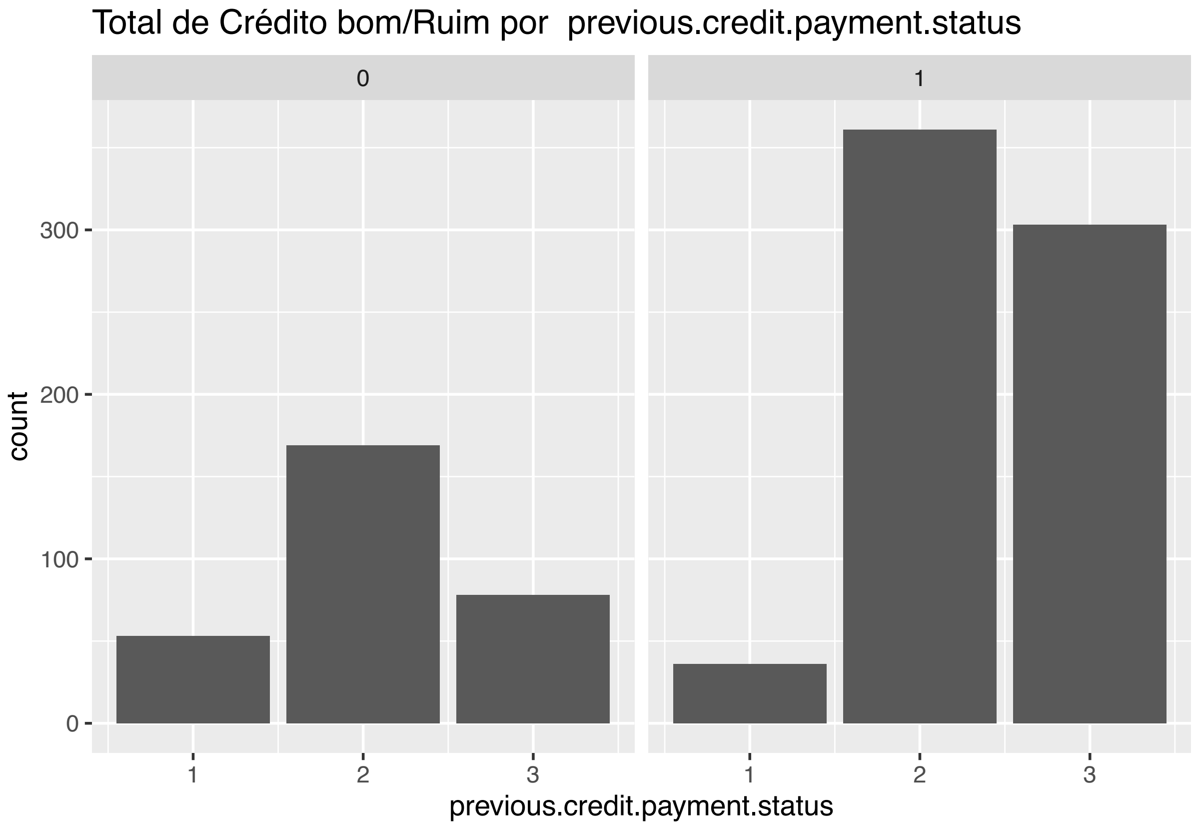


Total de Crédito bom/Ruim por account.balance
0
1
count
400
300
200
100
0
1
2
3
1
2
3
account.balance


```
##
## [[3]]
```

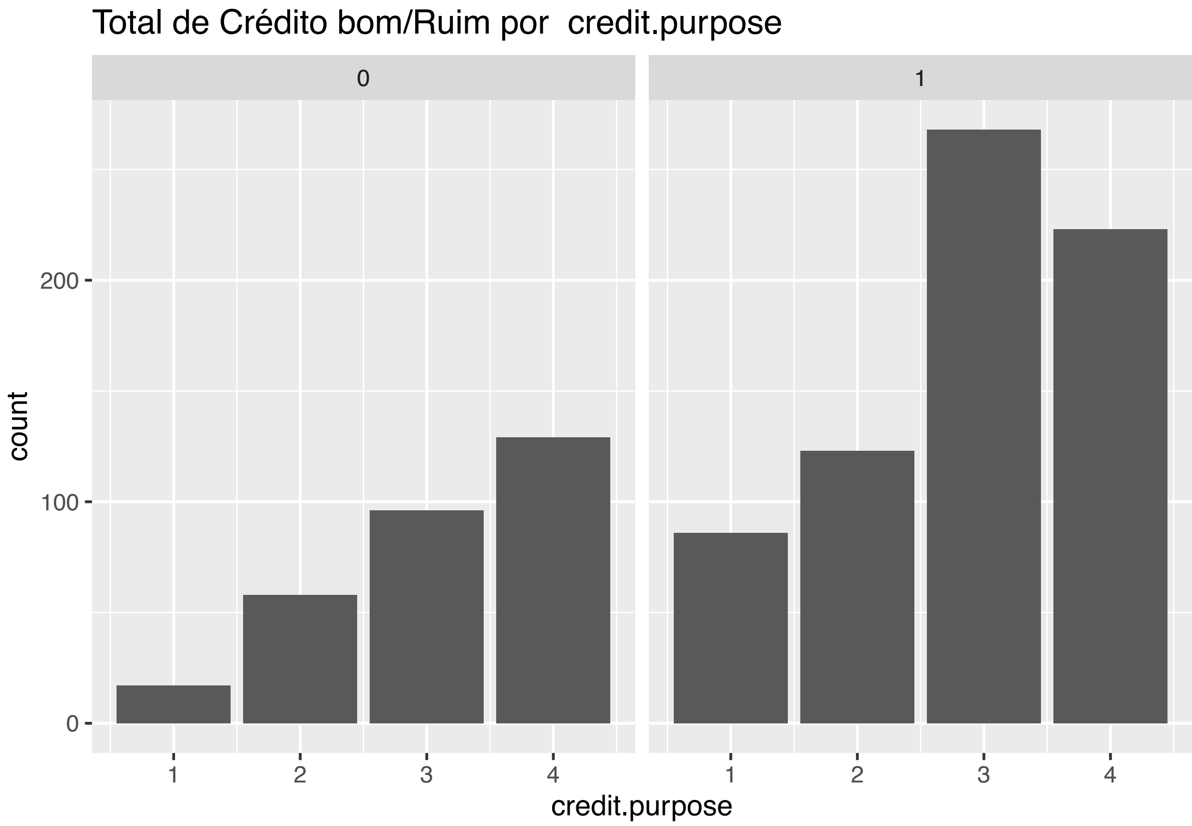

Total de Crédito bom/Ruim por credit.duration.months
0
1
count
100
50
0
20
40
60
20
40
60
credit.duration.months

```
##
## [[4]]
```

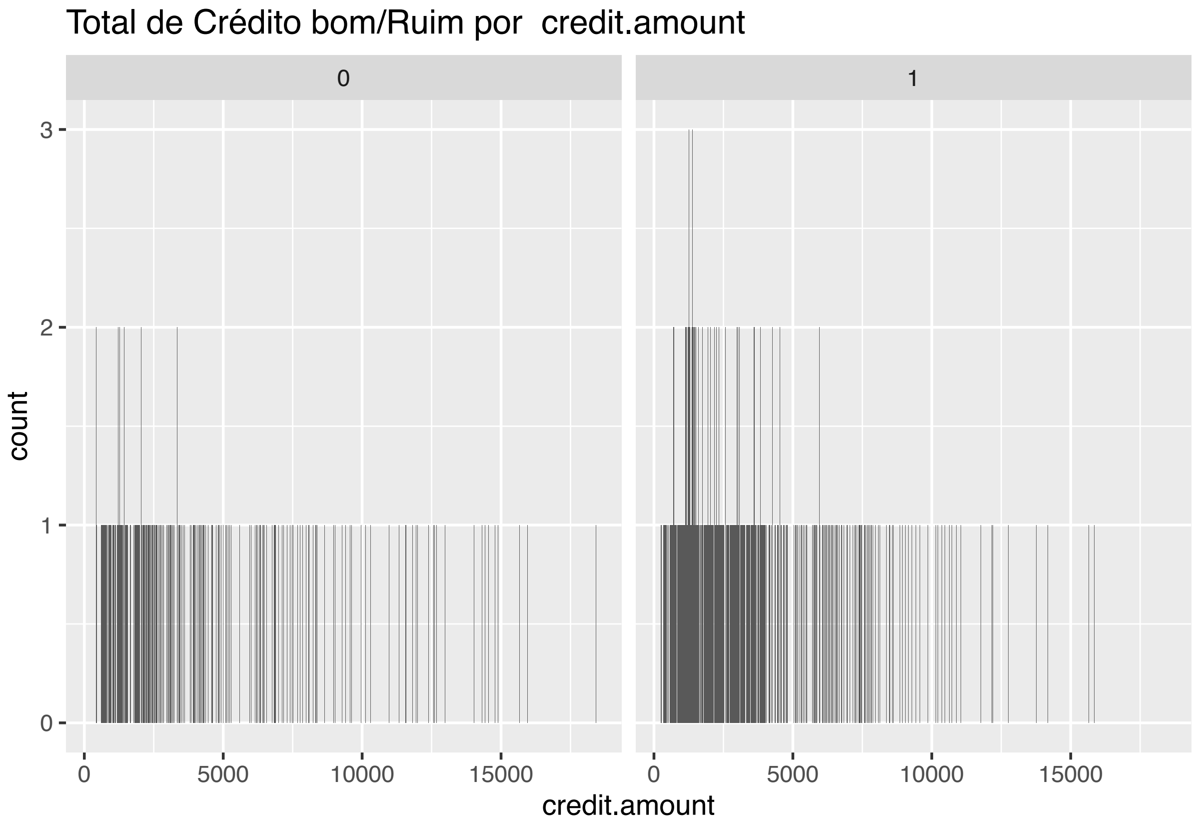


Total de Crédito bom/Ruim por previous.credit.payment.status
0
1
300
200
100
0
count
1
2
3
1
2
3
previous.credit.payment.status

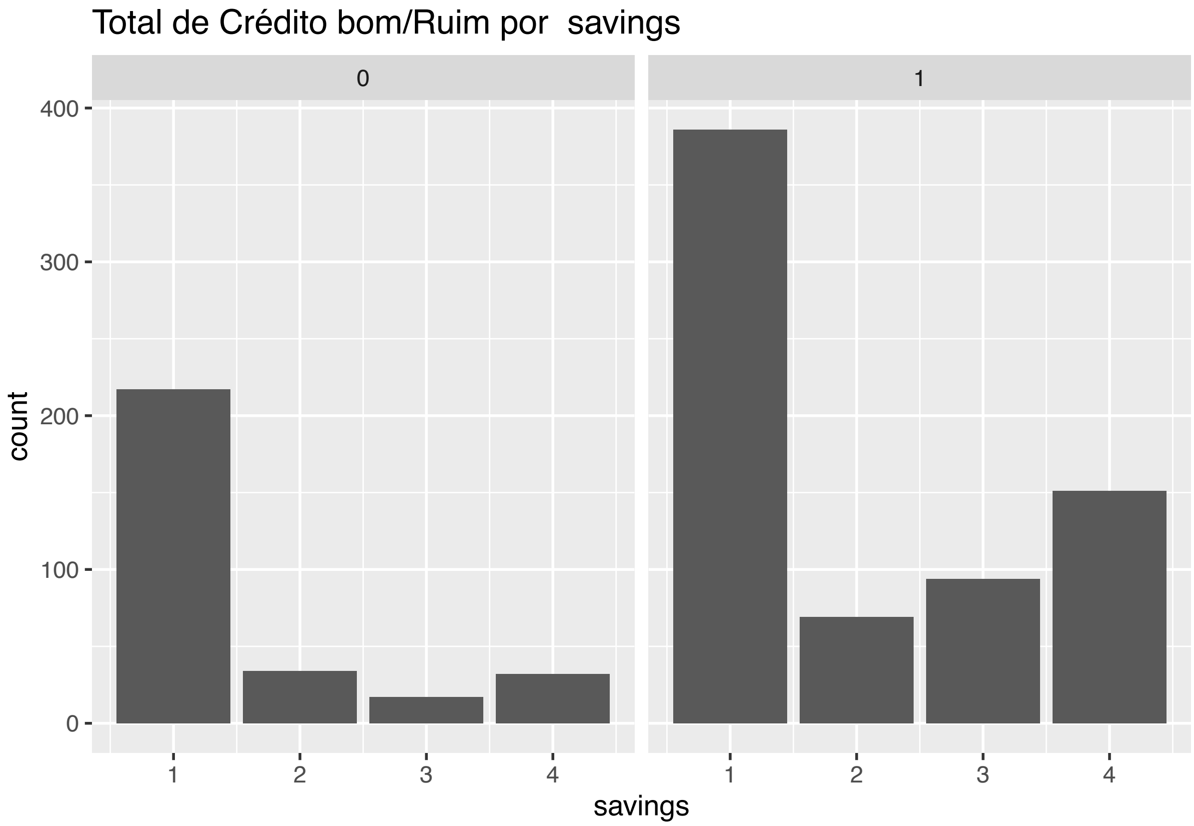

```
##
## [[5]]
```


Total de Crédito bom/Ruim por credit.purpose
0
1
count
200
100
0
1
2
3
4
1
2
3
4
credit.purpose

```
##
## [[6]]
```

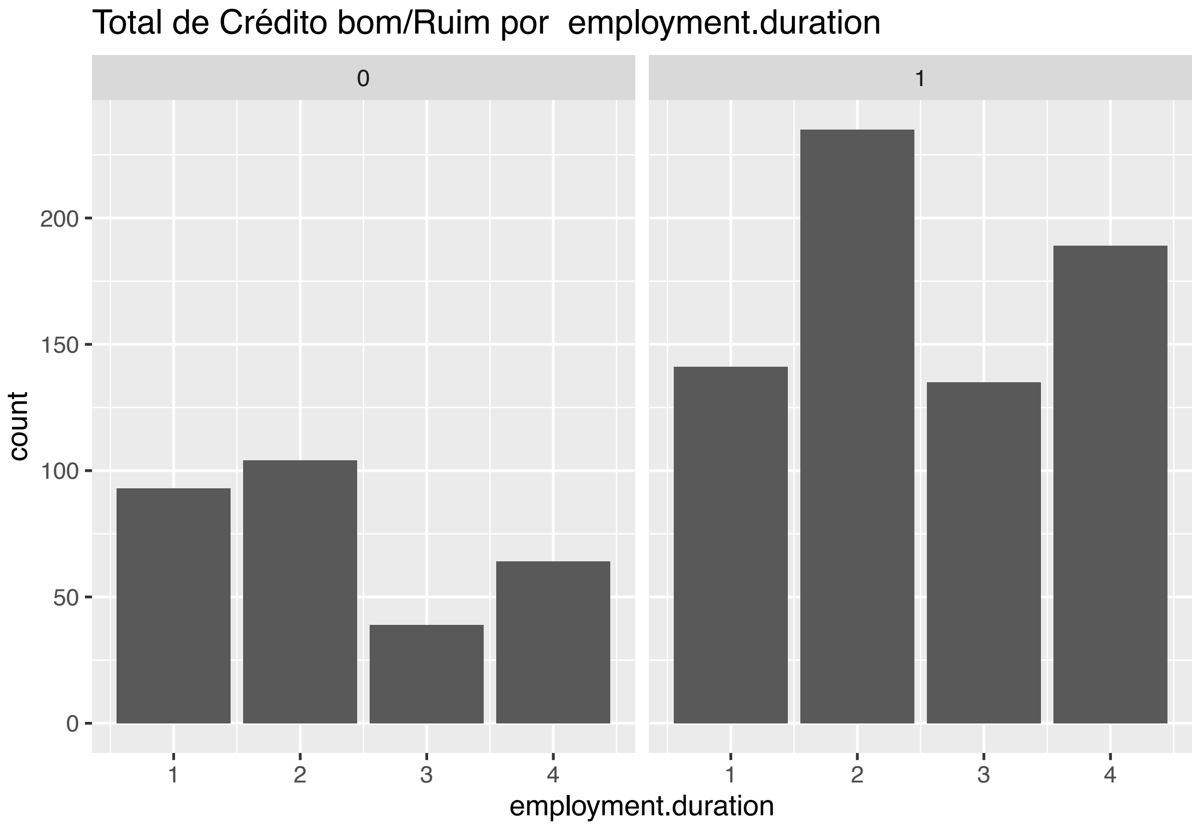

Total de Crédito bom/Ruim por credit.amount
0
1
count
3
2
1
0
0
5000
10000
15000
credit.amount

```
##
## [[7]]
```

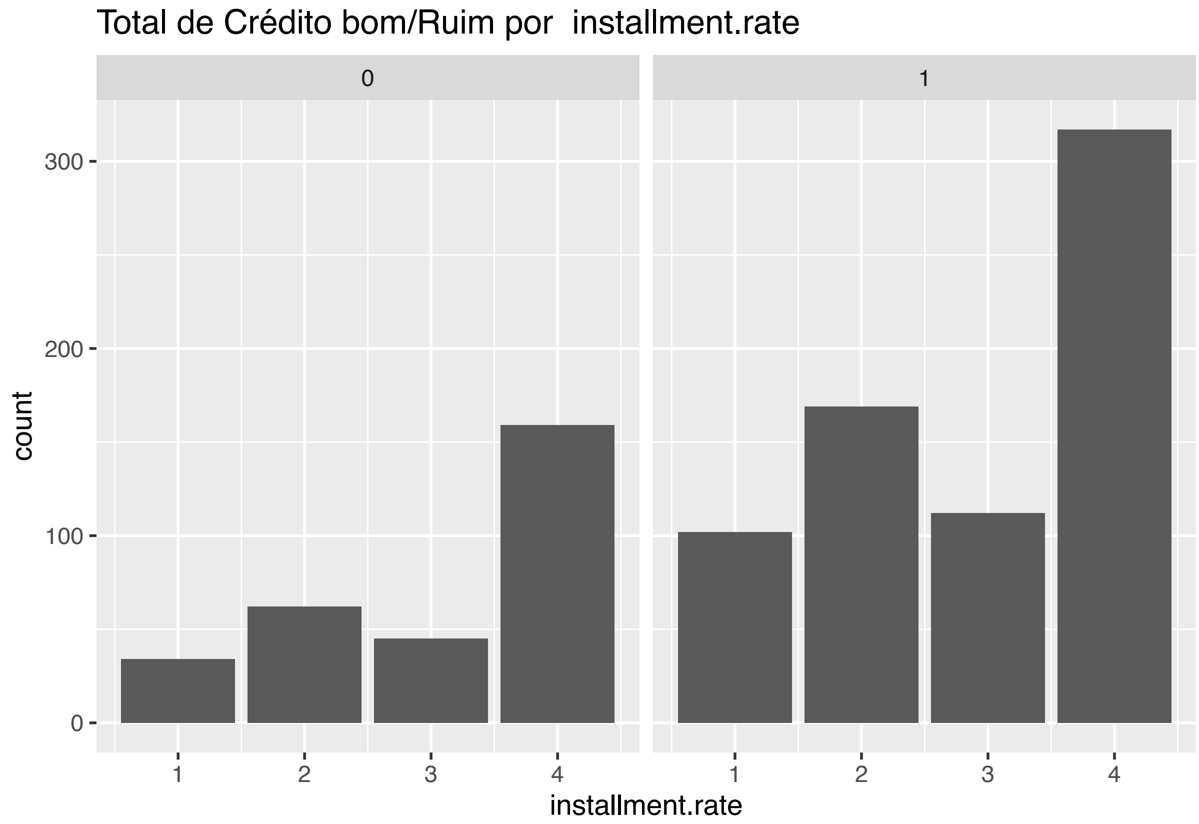


Total de Crédito bom/Ruim por savings
0
1
400
300
200
100
0
count
1
2
3
4
1
2
3
4
savings


```
##
## [[8]]
```

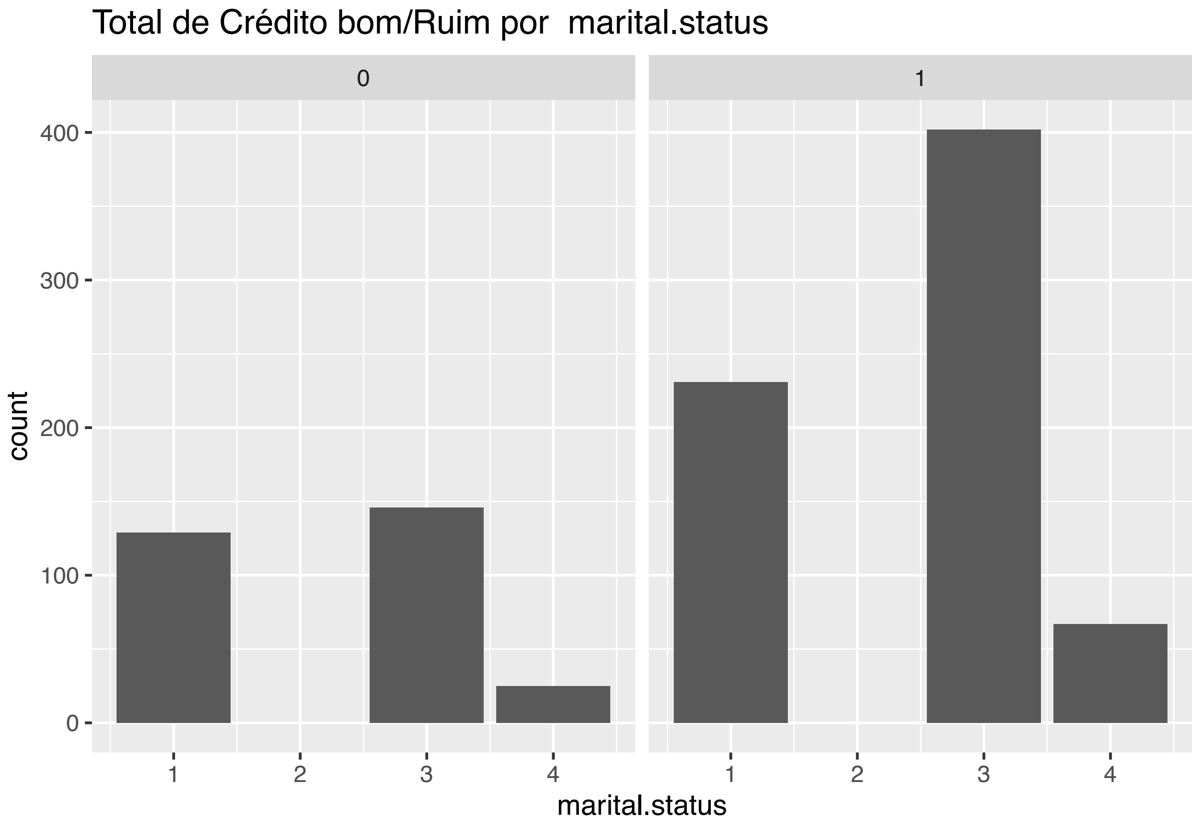

Total de Crédito bom/Ruim por employment.duration
0
1
count
200
150
100
50
0
1
2
3
4
employment.duration

```
##
## [[9]]
```



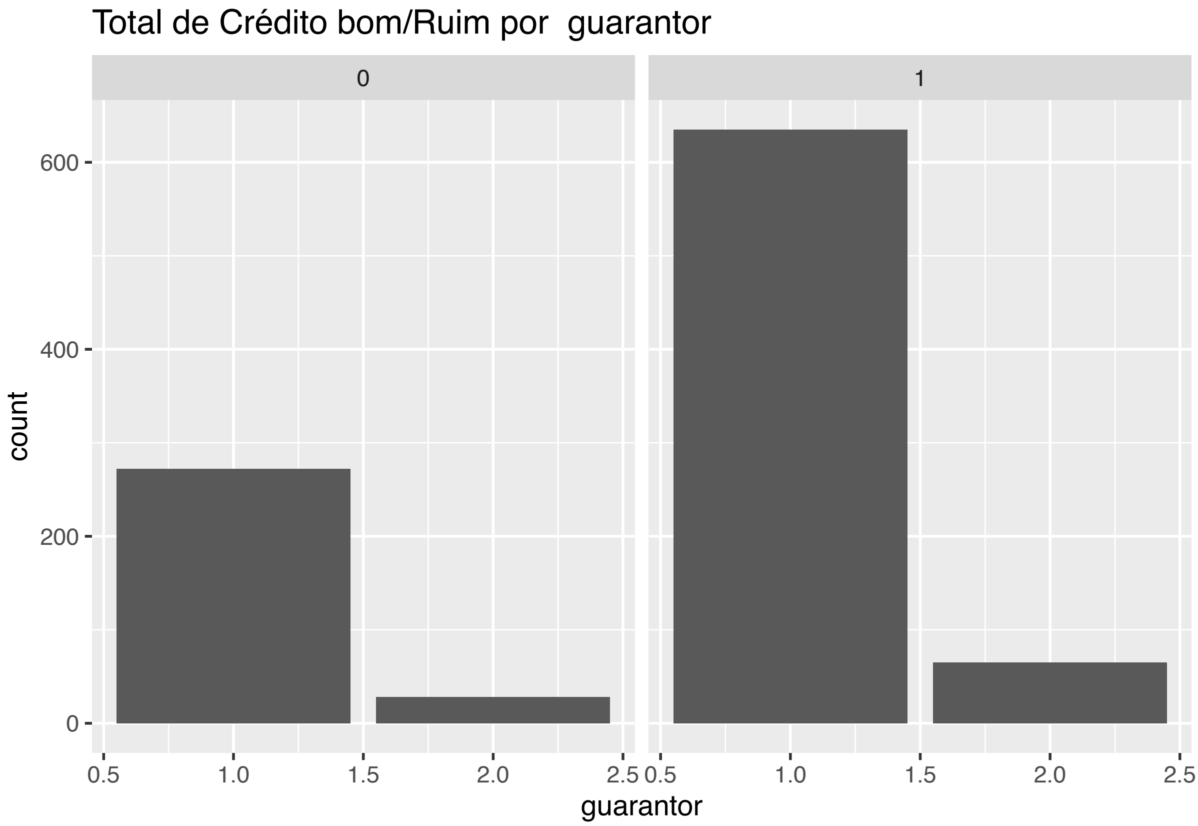
Total de Crédito bom/Ruim por installment.rate
0
1
300
200
100
0
count
1
2
3
4
1
2
3
4
installment.rate


```
##
## [[10]]
```

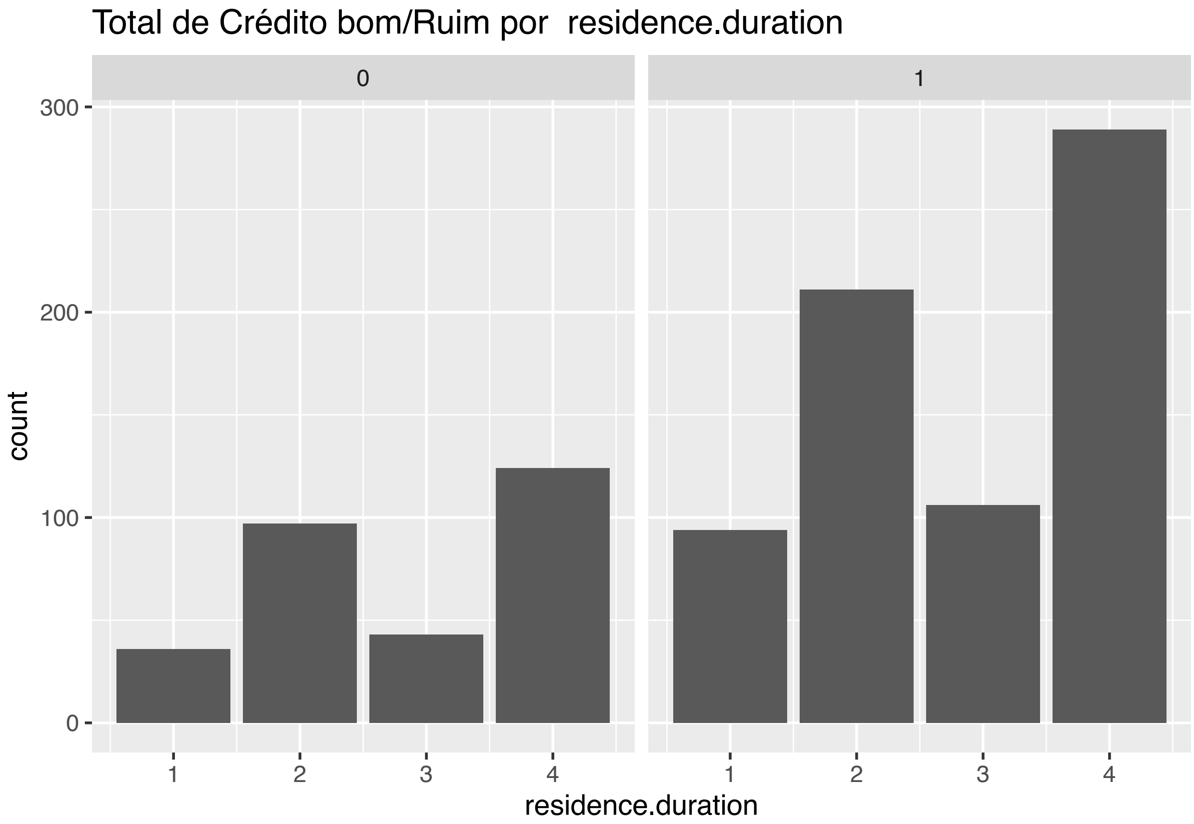


Total de Crédito bom/Ruim por marital.status
0
1
count
400
300
200
100
0
1
2
3
4
marital.status

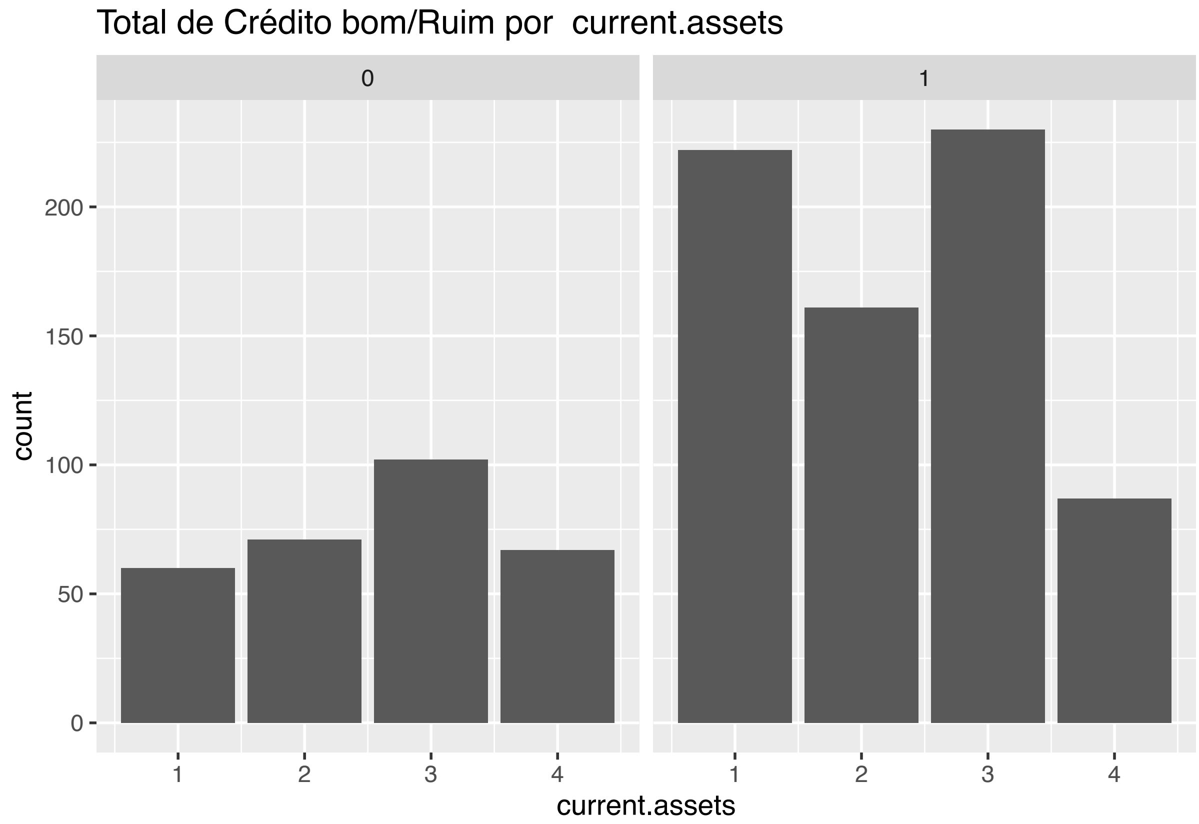

```
##
## [[11]]
```


Total de Crédito bom/Ruim por guarantor
0
1
600
400
200
0
count
0.5
1.0
1.5
2.0
2.5
guarantor

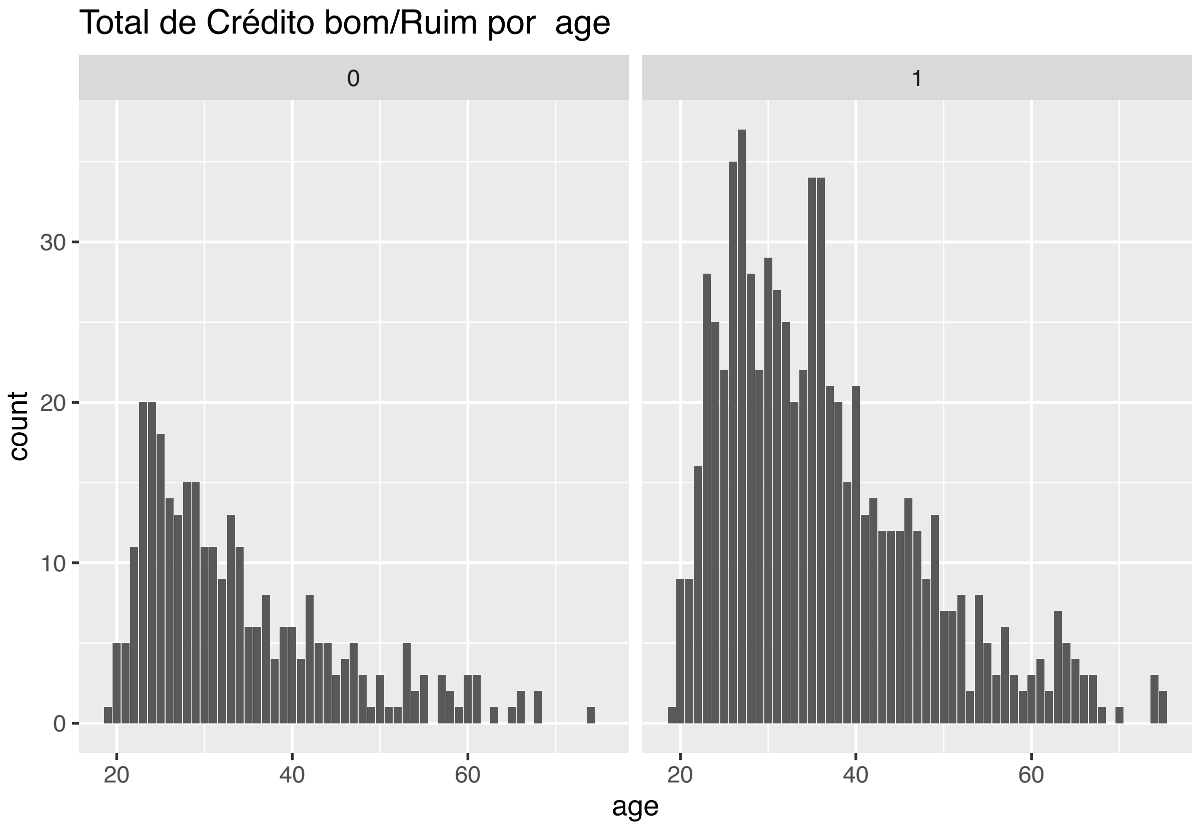
```
##
## [[12]]
```


Total de Crédito bom/Ruim por residence.duration
0
1
300
200
100
0
count
1
2
3
4
1
2
3
4
residence.duration

```
##
## [[13]]
```

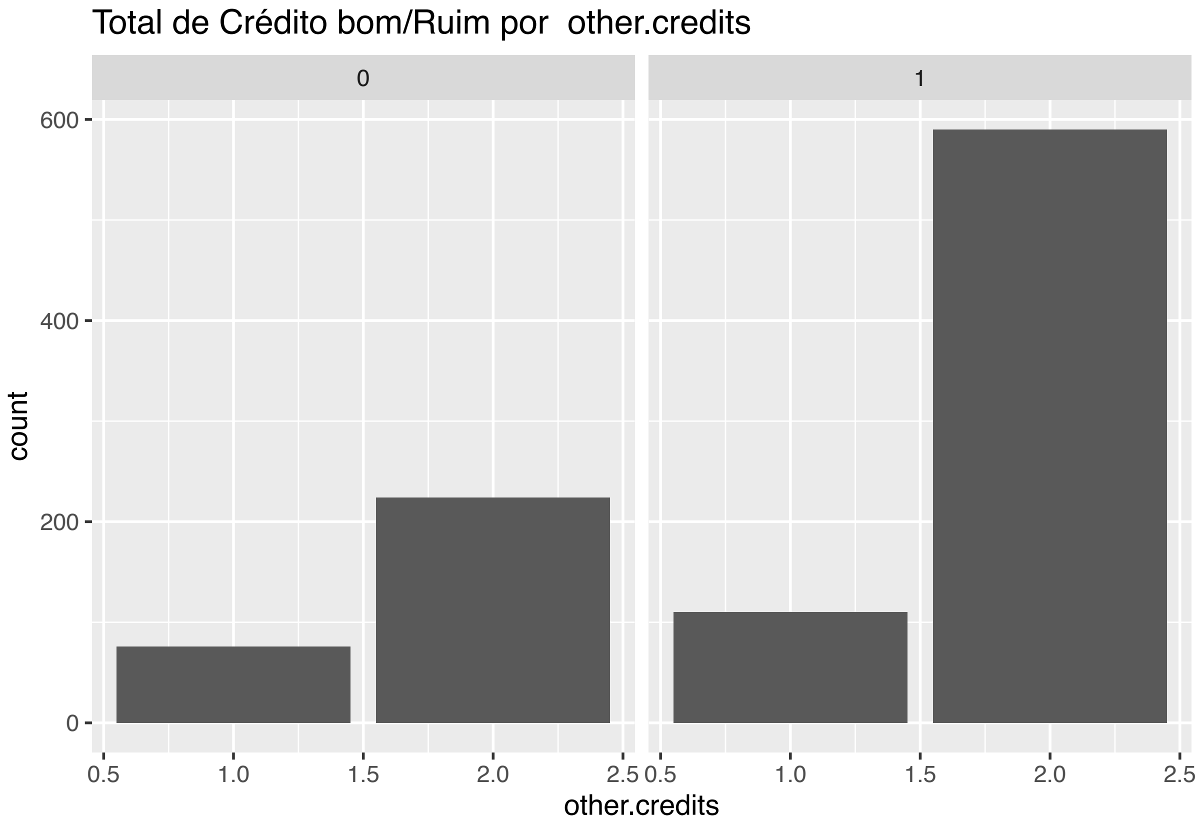

Total de Crédito bom/Ruim por current.assets
0
1
count
200
150
100
50
0
1
2
3
4
current.assets

```
##
## [[14]]
```



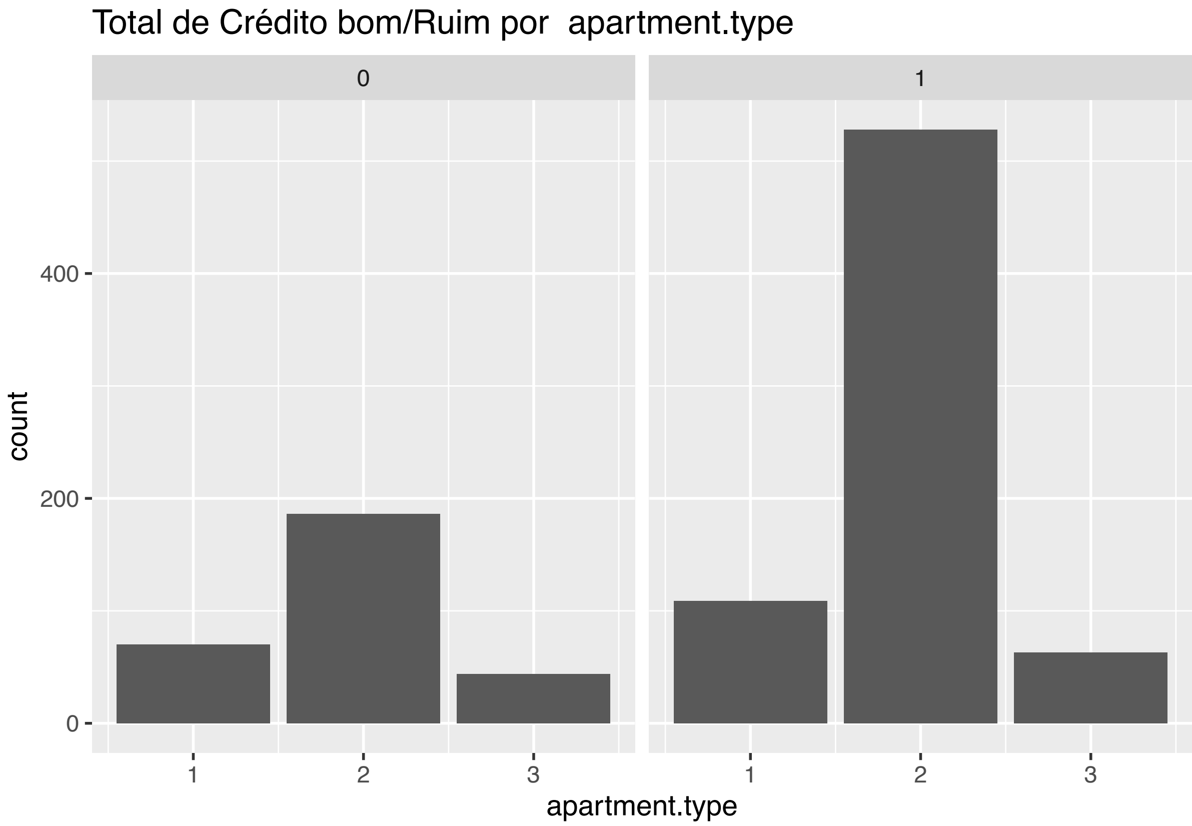
Total de Crédito bom/Ruim por age
0
1
count
30
20
10
0
20
40
60
age


```
##
## [[15]]
```

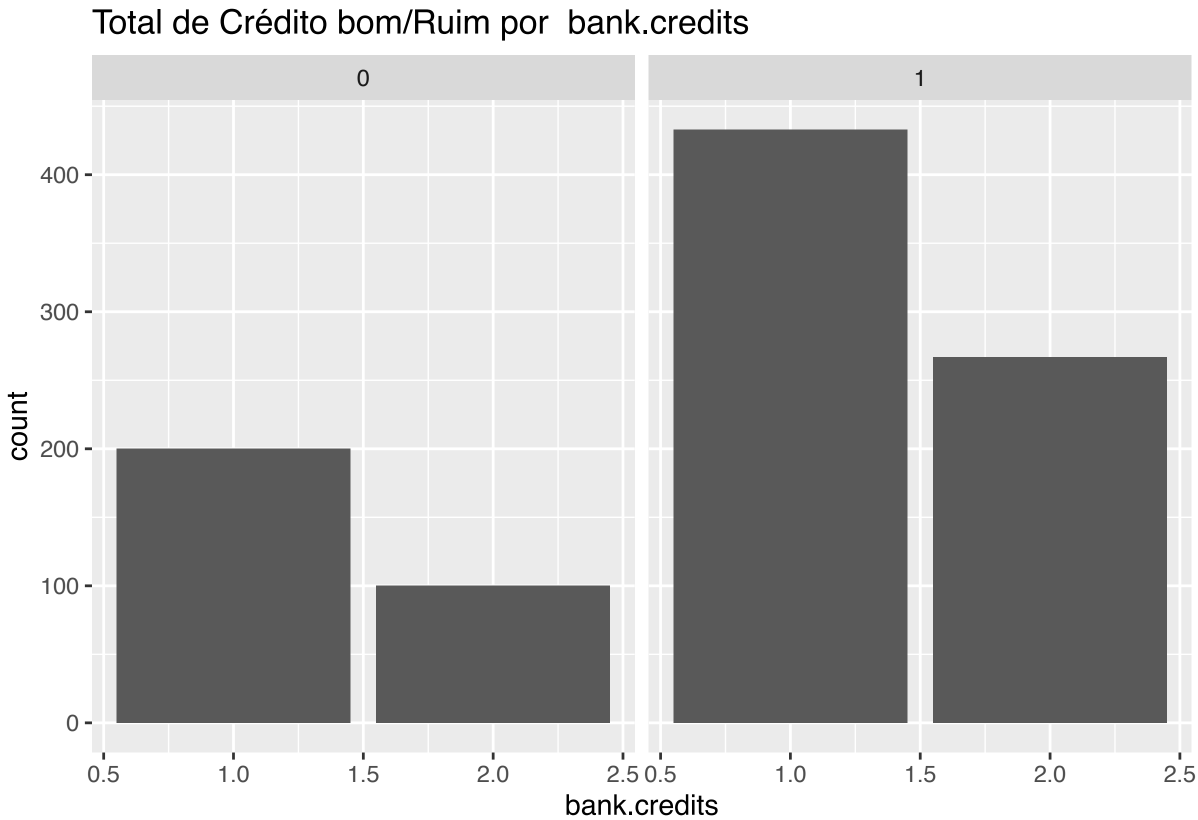


Total de Crédito bom/Ruim por other.credits
0
1
600
400
200
0
count
0.5
1.0
1.5
2.0
2.5
other.credits

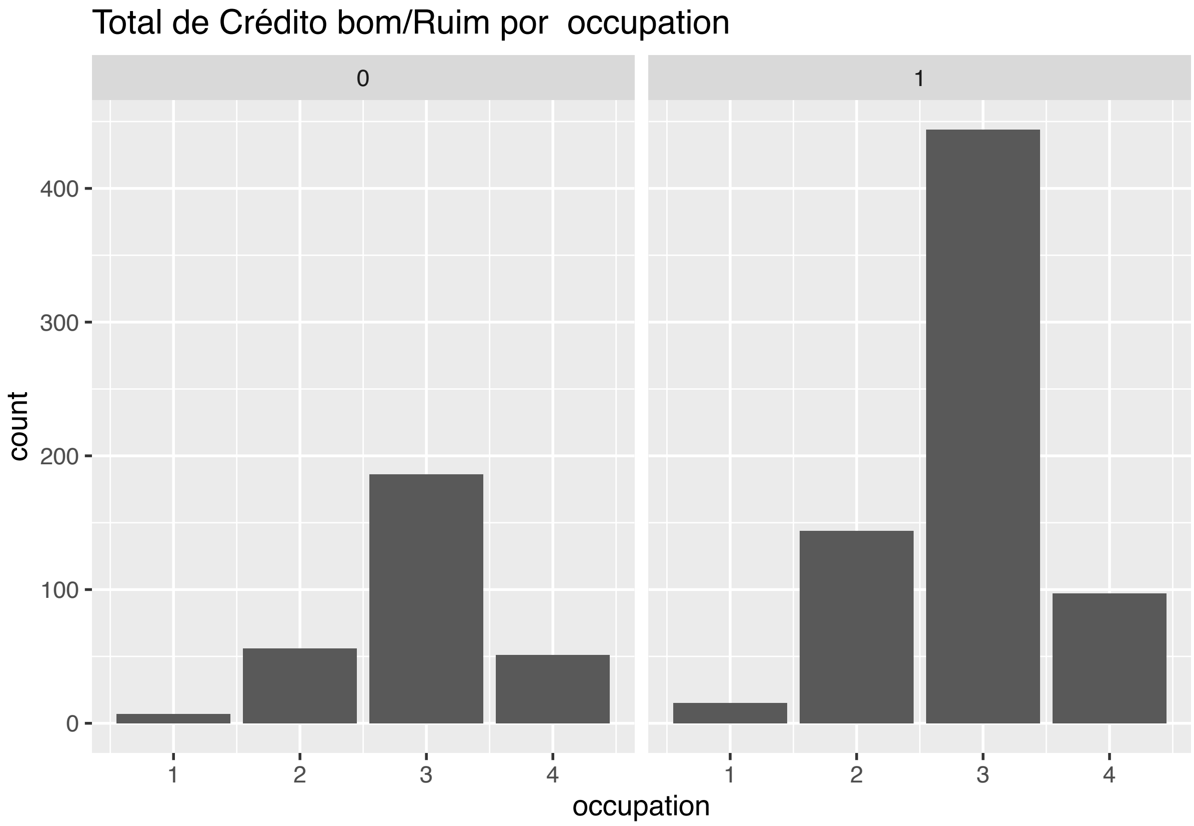

```
##
## [[16]]
```


Total de Crédito bom/Ruim por apartment.type
0
1
count
400
200
0
1
2
3
1
2
3
apartment.type

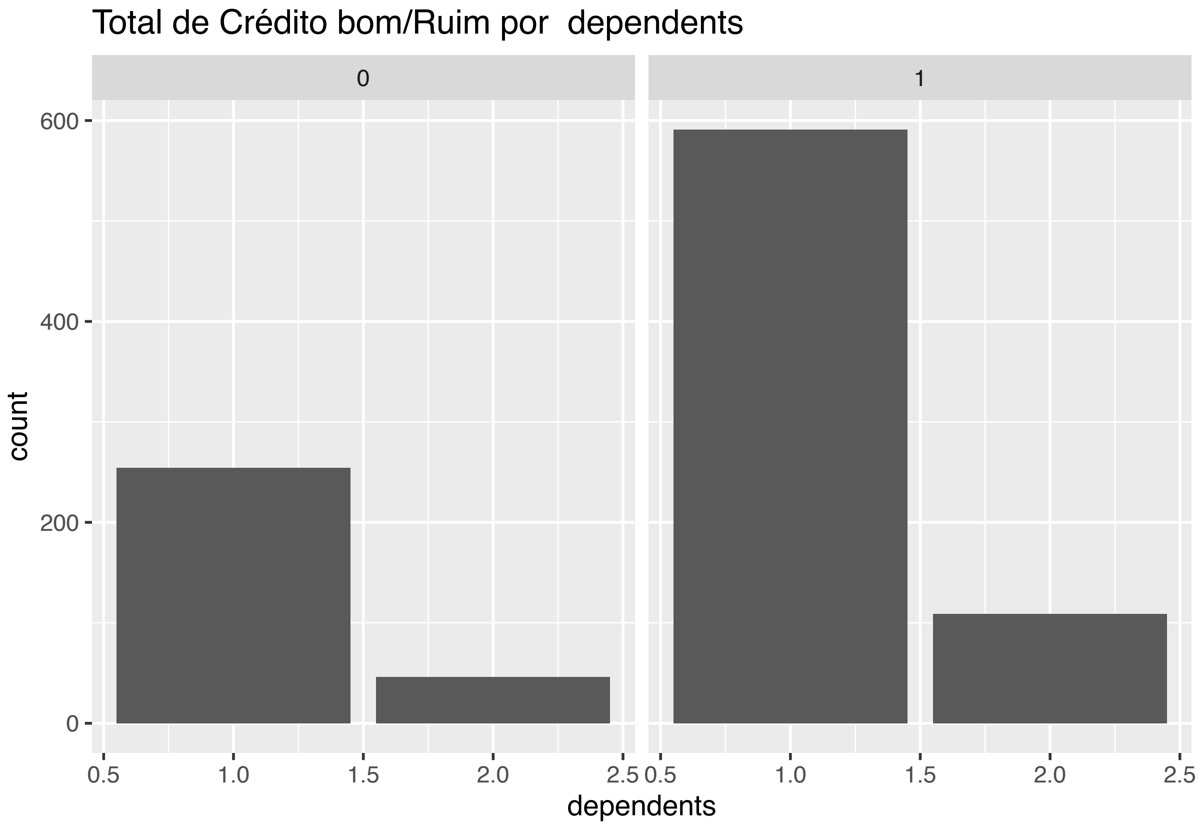
```
##
## [[17]]
```


Total de Crédito bom/Ruim por bank.credits
0
1
count
400
300
200
100
0
0.5
1.0
1.5
2.0
2.5
bank.credits

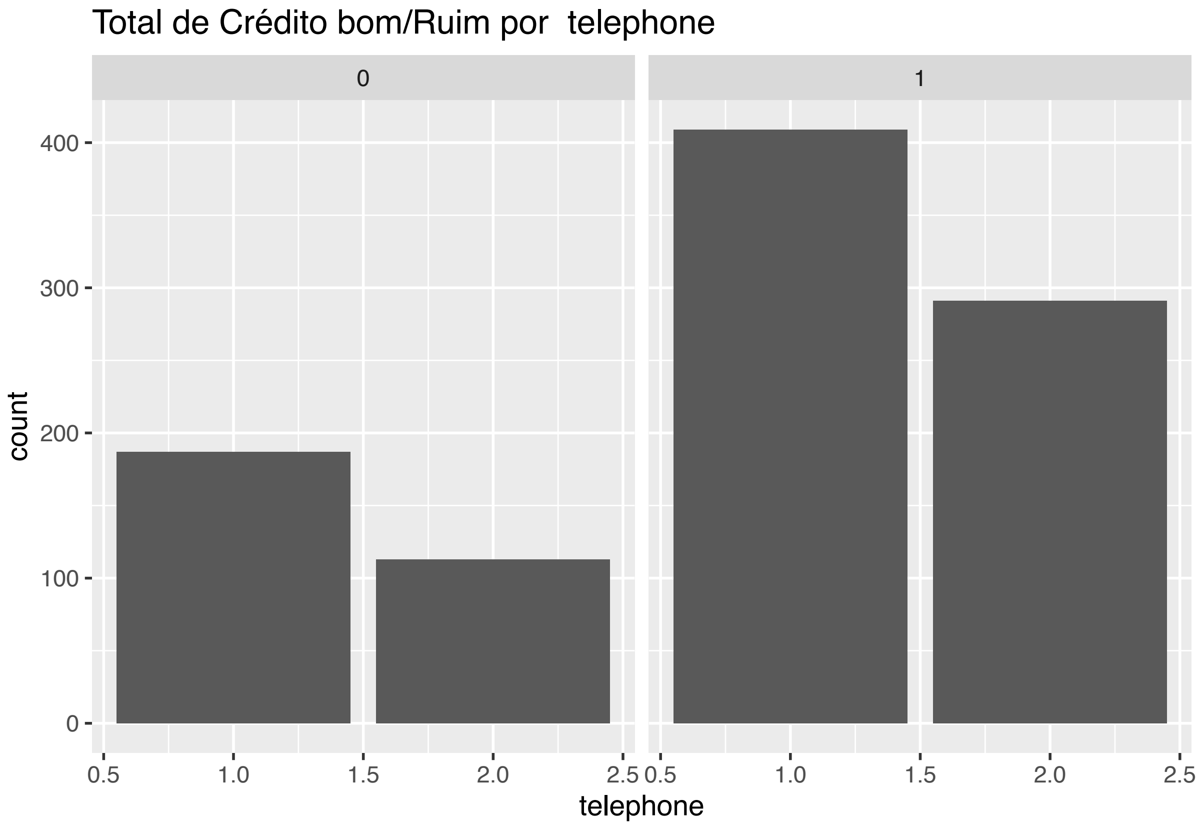
```
##
## [[18]]
```


Total de Crédito bom/Ruim por occupation
0
1
count
400
300
200
100
0
1
2
3
4
1
2
3
4
occupation

```
##
## [[19]]
```

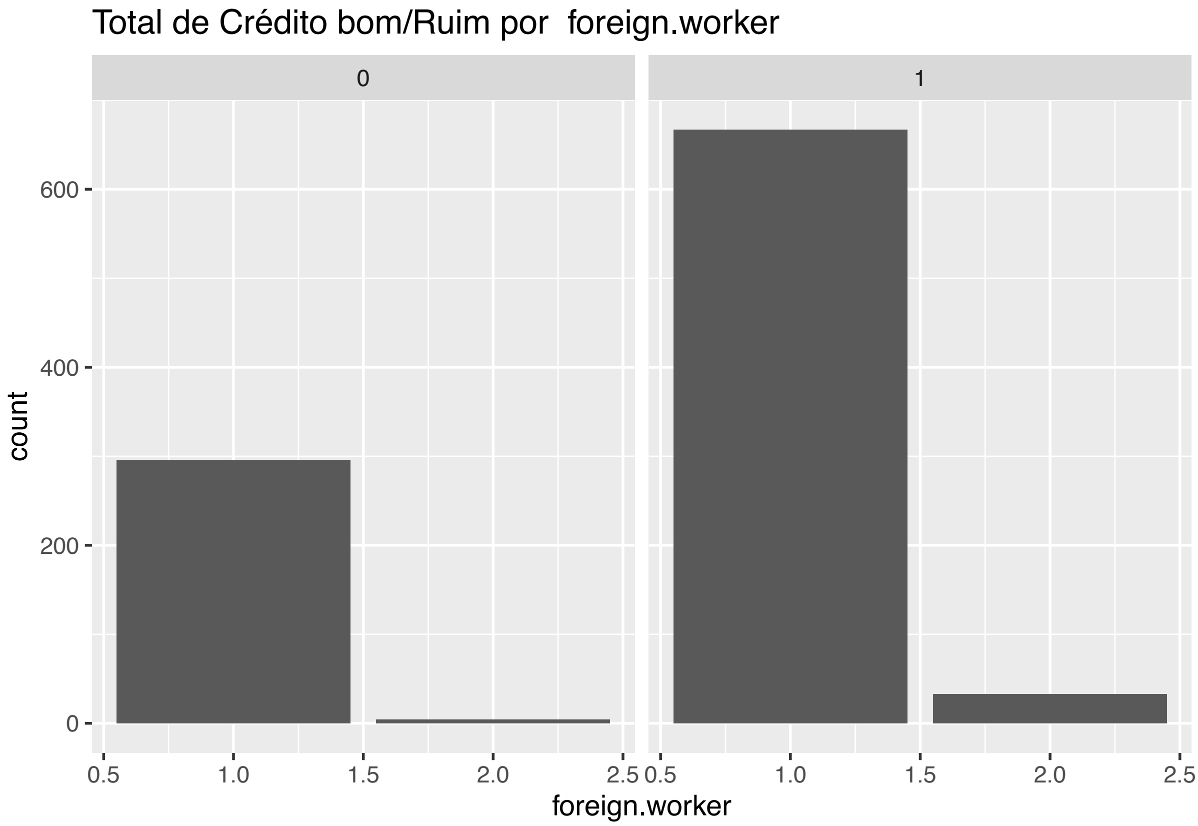

Total de Crédito bom/Ruim por dependents
0
1
600
400
200
0
count
0.5
1.0
1.5
2.0
2.5
dependents

```
##
## [[20]]
```

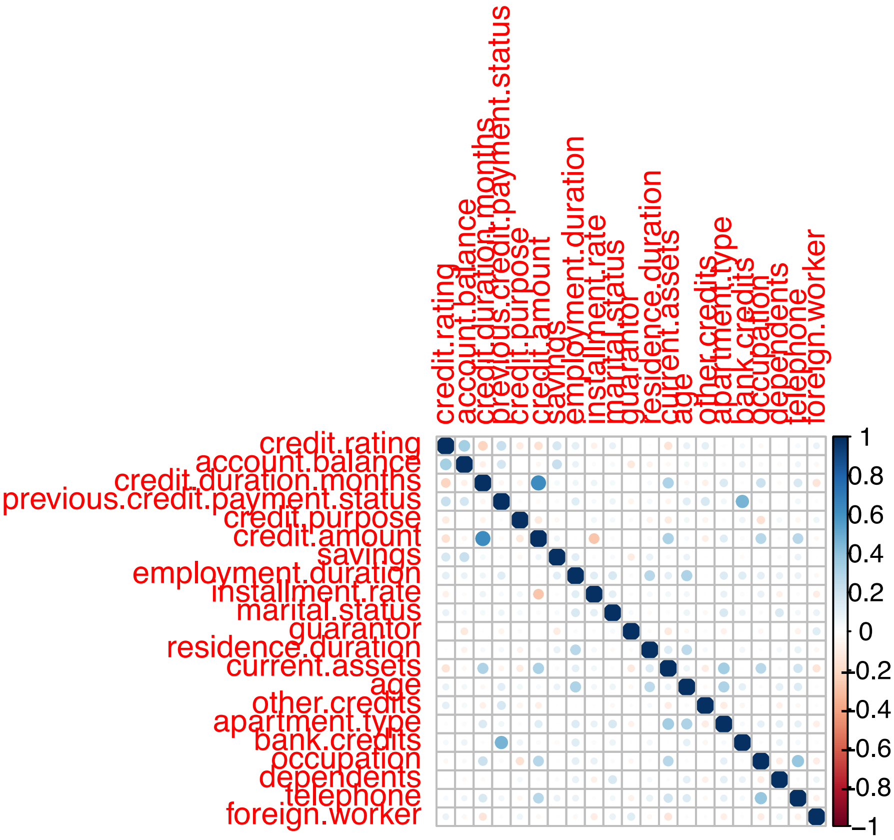


Total de Crédito bom/Ruim por telephone
0
1
count
400
300
200
100
0
0.5
1.0
1.5
2.0
2.5
telephone


```
##
## [[21]]
```




```
# criando um grafico de correlação
corrplot::corrplot(cor(credit_df))
```




```r
# convertendo a variavel target em fator (como vou usar algoritmos de suport vector machine, ele precis
#target seja  um fator para executar como modelo de classificação)
credit_df$credit.rating = as.factor(credit_df$credit.rating)

#Normalizando dados que estão com escala diferente
credit_df$credit.duration.months = scale(credit_df$credit.duration.months, center = T, scale = T)
credit_df$age = scale(credit_df$age, center = T, scale = T)
credit_df$credit.amount = scale(credit_df$credit.amount, center = T, scale = T)
```

efetuando seleção de variaveis. verifica quais são mais importantes para o modelo

usar randomForest para verogicar variaveis mais relevantes definindo numero de arvores igual a 100 definindo tamanho dos nos = 10 definindo importance = TRUE para retornar grau de importancia de uma variavel

```r
modelo = randomForest(
  credit.rating ~ .,
  data = credit_df,
  ntree = 100,
  nodesize = 10, importance = T)


importance = as.data.frame(modelo$importance)

# pegando apenas as variaveis mais relevantes para o experimento, de acordo com a
```

```
row_names = rownames(importance[importance$MeanDecreaseAccuracy >= mean(importance$MeanDecreaseAccuracy
library(stringr)

formula_ = as.formula(paste('credit.rating ~',str_c(row_names, collapse = ' + ')))
```

```
# função para gerar dados de treino e de test
splitData = function(dataframe, seed = NULL){
  if(!is.null(seed)) set.seed(seed)
  index = 1:nrow(dataframe)
  trinindex = sample(index, trunc(length(index)/2))
  trainset = dataframe[trinindex,]
  testset = dataframe[-trinindex,]
  list(trainset = trainset, testset = testset)
}

# gerando dados de treino e test
split = splitData(credit_df, seed = 808)

#separando dados
dados_treino = split$trainset
dados_test = split$testset
```

```
# Usando modelo de classificação SVM
svm_model = svm(formula_, data = dados_treino)

################################################################################
# Treinando modelo
pred = predict(svm_model, newdata = dados_test,  type = 'prob')

# Analisando resultado do treinamento
table(dados_test$credit.rating, pred)
```

```
##    pred
##       0   1
##   0  58  99
##   1  26 317
```

```
round(prop.table(table(dados_test$credit.rating, pred)
) * 100, digits = 1)
```

```
##    pred
##        0    1
##   0 11.6 19.8
##   1  5.2 63.4
```

```
# verificando acuracia
mean(pred == dados_test$credit.rating)
```

```
## [1] 0.75
```

```r
#Gerando confusion matrix com library caret
caret::confusionMatrix(as.factor(dados_test$credit.rating), as.factor(pred))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0  58  99
##          1  26 317
##
##                Accuracy : 0.75
##                  95% CI : (0.7096, 0.7874)
##     No Information Rate : 0.832
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.336
##
##  Mcnemar's Test P-Value : 1.196e-10
##
##             Sensitivity : 0.6905
##             Specificity : 0.7620
##          Pos Pred Value : 0.3694
##          Neg Pred Value : 0.9242
##              Prevalence : 0.1680
##          Detection Rate : 0.1160
##    Detection Prevalence : 0.3140
##       Balanced Accuracy : 0.7262
##
##        'Positive' Class : 0
##
```

```r
# Usando modelo de classificação randonforest
modelo_r_forest = randomForest(formula_, data = dados_treino)

# Treinando modelo
pred_forest = predict(modelo_r_forest, newdata = dados_test)


# Analisando resultado do treinamento
table(dados_test$credit.rating, pred_forest)
```

```
##    pred_forest
##       0   1
##   0  75  82
##   1  44 299
```

```r
round(prop.table(table(dados_test$credit.rating, pred_forest)
) * 100, digits = 1)
```

```
##    pred_forest
##        0    1
##   0 15.0 16.4
##   1  8.8 59.8
```

```r
# verificando acuracia
mean(pred_forest == dados_test$credit.rating)
```

```
## [1] 0.748
```

```r
#Gerando confusion matrix com library caret
#library(caret)
caret::confusionMatrix(as.factor(dados_test$credit.rating), as.factor(pred_forest))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0  75  82
##          1  44 299
##
##                Accuracy : 0.748
##                  95% CI : (0.7075, 0.7855)
##     No Information Rate : 0.762
##     P-Value [Acc > NIR] : 0.7855530
##
##                   Kappa : 0.374
##
##  Mcnemar's Test P-Value : 0.0009799
##
##             Sensitivity : 0.6303
##             Specificity : 0.7848
##          Pos Pred Value : 0.4777
##          Neg Pred Value : 0.8717
##              Prevalence : 0.2380
##          Detection Rate : 0.1500
##    Detection Prevalence : 0.3140
##       Balanced Accuracy : 0.7075
##
##        'Positive' Class : 0
##
```