Avaliação de Risco de Crédito

Fernando Tsutomu Hara

11/06/2020

% !TEX encoding = UTF-8 Unicode

Avaliação de Risco de Crédito

Este projeto visa construir um modelo preditivo que cálcule a previão de conceder ou não crédito a uma pessoa. Os dados foram extraídos do kaggle de um Banco Alemão e contêm informações sobre 20 variáveis e a classificação se um candidato é considerado um risco de crédito Bom ou Ruim e contém 1000 observações.

Dicionário das Variáveis

Abaixo contém um dicionário com cada variável e o que elas significam:

1. CREDIT.RATING: Coluna indicando se o cliente em questão é um bom cliente para permitir créditos ou não.
   - 1 - sim
   - 0 - não
2. ACOUNT.BALANCE: Montante existente atualmente na conta.
   - 1 - Nenhuma Conta Corrente;
   - 2 - sem saldo ou débito;
   - 3 - até 200;
   - 4 - mais que 200.
3. CREDIT.DURATION.MONTHS: Significa a duração em meses do empréstimo concedido.
4. PREVIOUS.CREDIT.PAYMENT.STATUS: Informações descritivas sobre o histórico financeiro do cliente. Se os créditos antigos dele estão quitados, se ainda está devendo, se até agora os créditos dele estão em bom estado.
   - 0 - pagamento hesitante de créditos anteriores;
   - 1 - conta corrente problemática / há mais créditos em execução, mas em outros bancos;
   - 2 - sem créditos anteriores / reembolsado todos os créditos anteriores;
   - 3 - sem problemas com créditos atuais neste banco;
   - 4 - pagou créditos anteriores neste banco.
5. CREDIT.PURPOSE: Propósito destinado para o crédito concedido.
   - 0 - Outros;
   - 1 - Novo Carro;
   - 2 - Carro Usado;
   - 3 - Itens mobiliários;
   - 4 - Rádio / Televisão
   - 5 - Electrodomésticos;
   - 6 - Reparos;
   - 7 - Educação;
   - 8 - Férias;
   - 9 - Reciclagem;
   - 10 - Negócios.
6. CREDIT.AMOUNT: Montante de crédito requisitado ao banco.

7. SAVINGS: Montante disponível na conta poupança.
    - 1 - não disponível / sem poupança;
    - 2 - menos de 100;
    - 3 - de 100 a 500;
    - 4 - de 500 a 1000;
    - 5 - mais de 1000.
8. EMPLOYMENT.DURATION: Tempo de empregado no atual emprego.
    - 1 - Desempregado;
    - 2 - menos que 1 ano;
    - 3 - 1 a 4 anos;
    - 4 - 4 a 7 anos;
    - 5- mais que 7 anos.
9. INSTALLMENT.RATE: Taxa em % da renda disponível.
    - 1 - mais que 35%;
    - 2 - de 25% a 35%;
    - 3 - de 20% a 25%;
    - 4 - menos que 20%.
10. MARITAL.STATUS: Estado civil e sexo do cliente.
    - 1 - Homem divorciado /vivendo sozinho;
    - 2 - Homem solteiro;
    - 3 - Homem casado / viúvo;
    - 4 - Mulher;
11. GUARANTORS: Tipo de associação em créditos concedidos que já participou.
    - 1 - None;
    - 2 - Co-Requerente;
    - 3 - Fiador;
12. RESIDENCE.DURATION: Tempo de moradia na residência atual.
    - 1 - mais que 1 ano;
    - 2 - de 2 a 4 anos;
    - 3 - de 4 a 7 anos;
    - 4 - mais que 7 anos.
13. CURRENT.ASSETS: Recursos disponíveis mais valiosos
    - 1 - não disponível / sem ativos;
    - 2 - Carro / Outros;
    - 3 - Contrato de poupança com uma sociedade de construção / Seguro de vida;
    - 4 - Proprietário de casa ou terreno.
14. AGE: Idade
15. OTHERS.CREDITS: Mais créditos em execução.
    - 1 - em outro banco;
    - 2 - na loja de departamento ou na casa de pedidos por correio;
    - 3 - sem créditos em execução;
16. APARTMENT.TYPE: Tipo de propriedade da residência.
    - 1 - apartamento alugado;
    - 2 - apartamento ocupado pelo proprietário;
    - 3 - apartamento livre.
17. NUMBER OF EXISTING CREDITS AT THIS BANK: Número de créditos já concedidos no banco.
    - 1 - 1;
    - 2 - 2 a 3;
    - 3 - 4 a 5;
    - 4 - 6 ou mais;
18. OCCUPATION: Estado do trabalho atual
    - 1 - desempregado / não qualificado sem residência permanente;
    - 2 - não qualificado com residência permanente;
    - 3 - trabalhador qualificado / empregado qualificado / funcionário público menor;

- 4 - executivo / autônomo / funcionário superior;

19. DEPENDENTS: Total de dependentes.
    - 1 - 1 a 2;
    - 2 - 3 ou mais.
20. TELEPHONE: Indicativo se o cliente possui telefone ou não.
    - 1 - não
    - 2 - sim
21. FOREIGN.WORKER: Indicando se o cliente é de outra cidade ou se trabalha na mesma cidade do trabalho.
    - 1 - sim
    - 2 - não

Leitura do Data Frame

Aqui faremos a leitura do data frame, e verificar algumas informações iniciais sobre ele.

```
df <- read.csv("credit_dataset.csv", header = TRUE, sep = ",")

str(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 ...
```

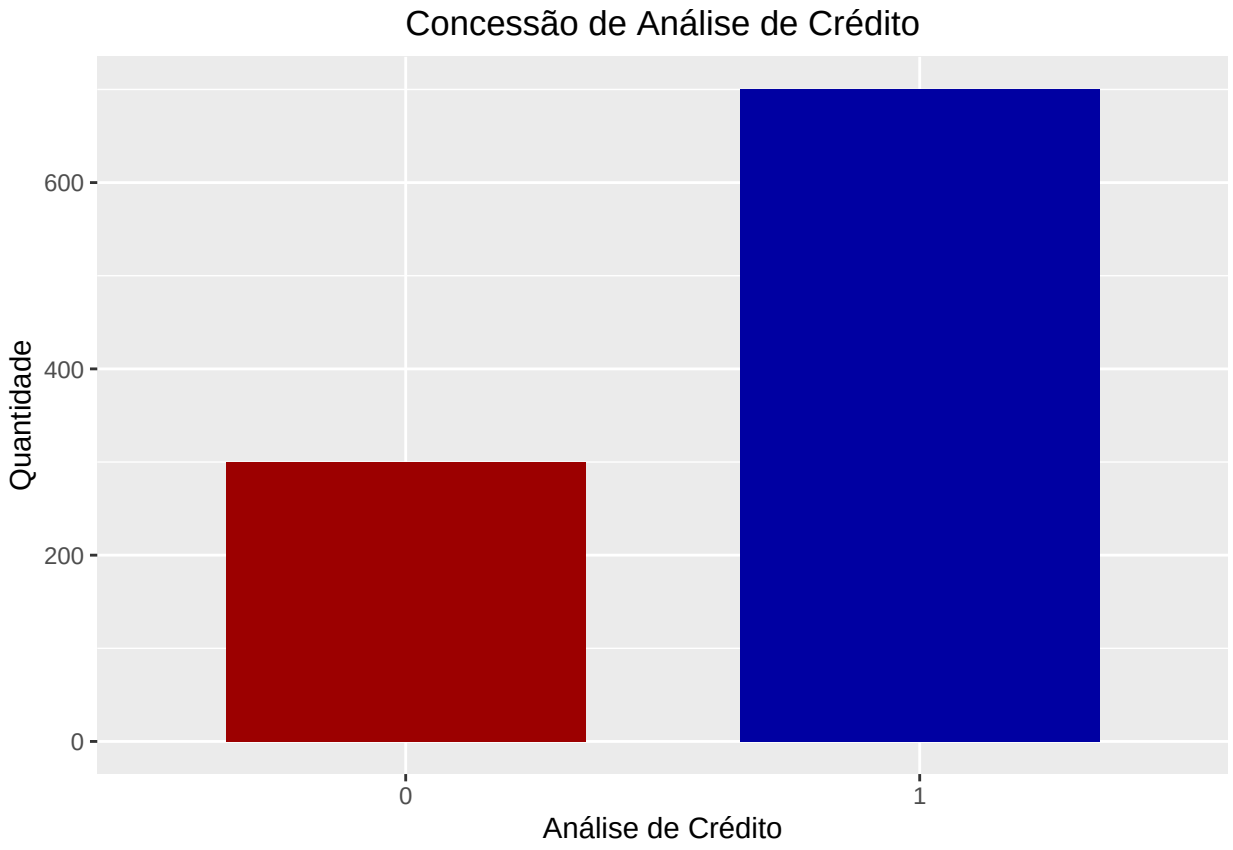
```
# O data frame foi lido com todas a variáveis como numéricas, mas a maioria delas
# é categóricas, vou mudando essas variáveis ao longo da análise exploratória.

# Verificação de valores missing
df[, is.na(df) == TRUE]
```

```
## data frame with 0 columns and 1000 rows
```

Exploração e limpeza dos dados.

Agora faremos a exploração dos dados inciando com a variável target, que é a que nós queremos prever. Logo após irei analiar variável por variável para ver quais são mais relevantes em nosso modelo preditivo.

```r
# Analisando a variável target (credit.racing)
library(ggplot2)

df$credit.rating <- as.factor(df$credit.rating)

ggplot(df, aes(x=credit.rating)) +
  geom_bar(stat="count", width=0.7, fill=c("#9c0000", "#0000a2"))+
  labs(x = "Análise de Crédito", y = "Quantidade",
       title = "Concessão de Análise de Crédito") +
  theme(plot.title = element_text(hjust = 0.5))
```




```r
table(df$credit.rating)
```

```
##
##   0   1
## 300 700
```

```r
prop.table(table(df$credit.rating))
```

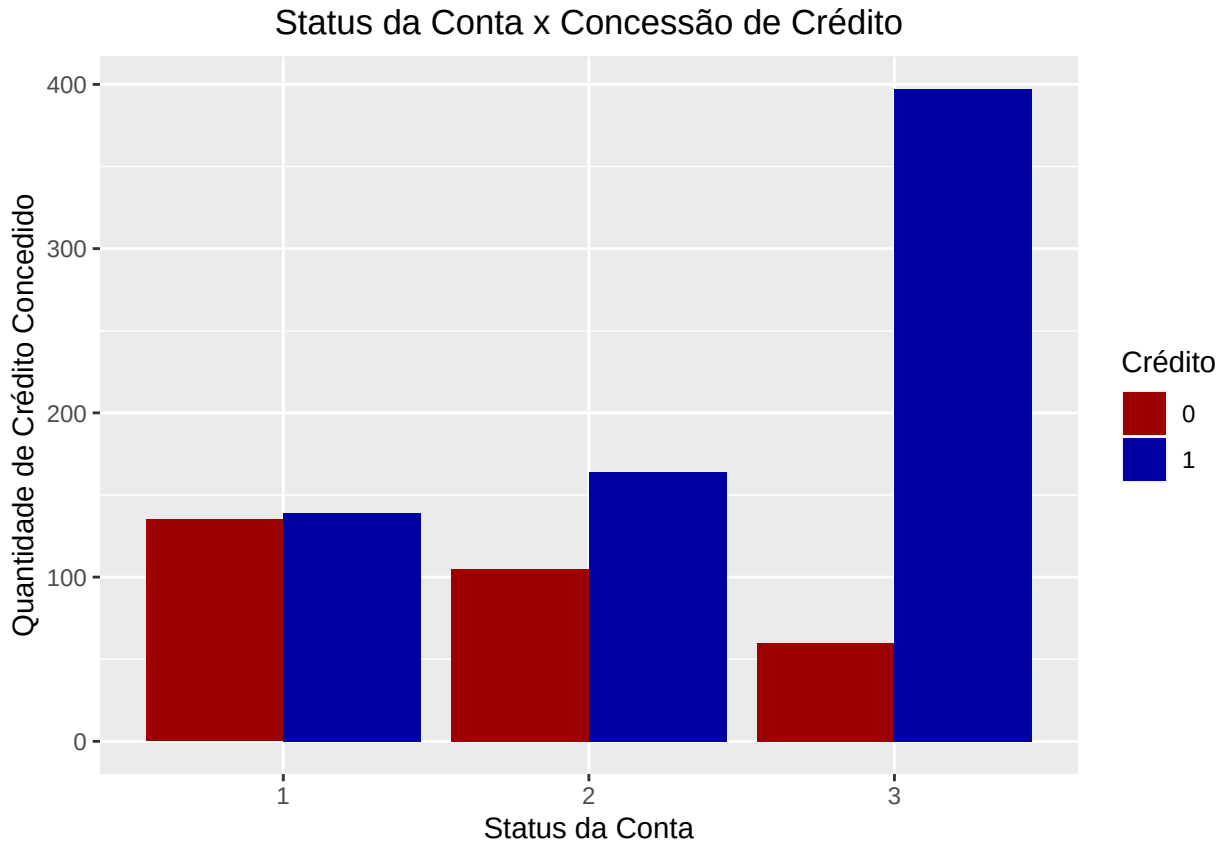
```
##
##   0   1
## 0.3 0.7
```

```r
# Como podemos ver os dados da variável target estão bem desbalanceados, sendo 30% amostras de crédito

# Análise das variáveis para a predição.

# Acount.balance
```

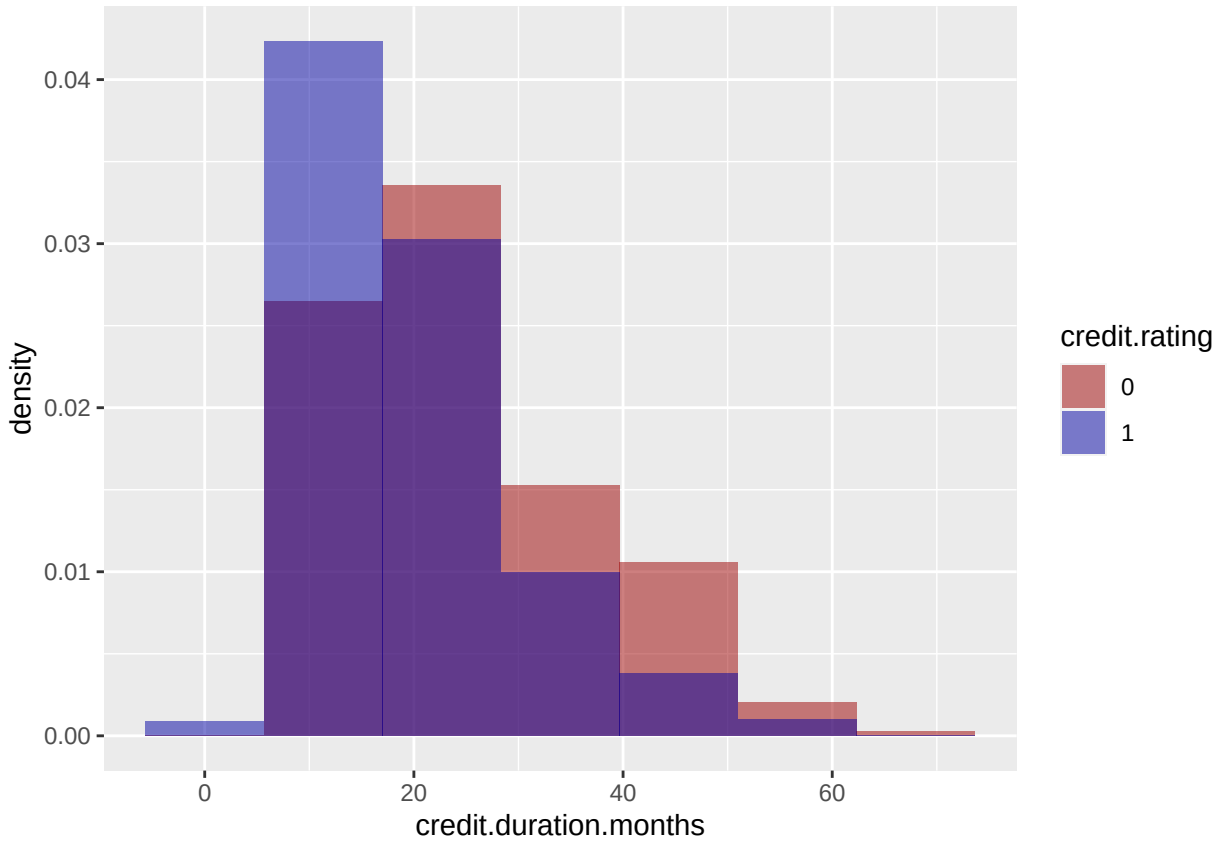
```r
# Tranformando a variável de numérica para fator
df$account.balance <- as.factor(df$account.balance)

# Quantidade de cada fator nessa variável
table(df$account.balance)
```

```
##
##   1   2   3
## 274 269 457
```

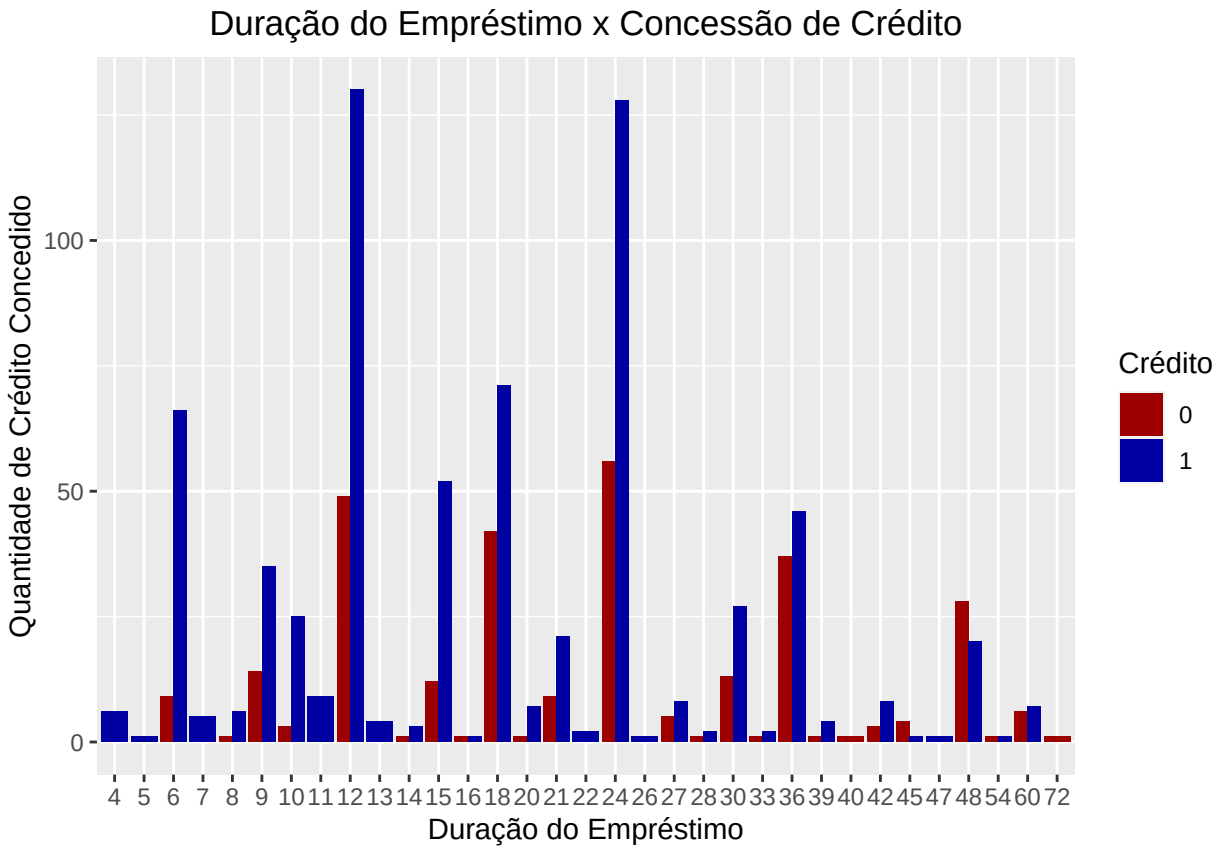
```r
# Gráfico da contagem da variável por crédito financeiro concedido ou não.
ggplot(df, aes(account.balance, ..count..)) +
  geom_bar(aes(fill = credit.rating), position = "dodge") +
  scale_fill_manual(values = c("#9c0000", "#0000a2") ) +
  labs(x = "Status da Conta", y = "Quantidade de Crédito Concedido",
       fill = "Crédito", title = "Status da Conta x Concessão de Crédito") +
  theme(plot.title = element_text(hjust = 0.5))
```




```r
# Como podemos ver pelo gráfico, pessoas que tem algum montante na conta corrente têm mais chances de c

# credit.duration.months

# Histograma da duração do empréstimo separado por créditos iagual a sim e não.
ggplot(df, aes(credit.duration.months, fill = credit.rating)) +
  geom_histogram(alpha = 0.5, aes(y = ..density..), position = 'identity',
                 bins=7) +
  scale_fill_manual(values = c("#9c0000", "#0000a2") )
```




```
# Gráfico da contagem da variável por crédito financeiro concedido ou não.
ggplot(df, aes(as.factor(credit.duration.months), ..count..)) +
  geom_bar(aes(fill = credit.rating), position = "dodge") +
  scale_fill_manual(values = c("#9c0000", "#0000a2") ) +
  labs(x = "Duração do Empréstimo", y = "Quantidade de Crédito Concedido",
       fill = "Crédito", title = "Duração do Empréstimo x Concessão de Crédito") +
  theme(plot.title = element_text(hjust = 0.5))
```




```
# Como podemos ver nos 2 gráficos, a decisão de concessão de crédito financeiro muda e acordo com tempo

# A partir dessa variável será criada uma nova variável que irá juntar os valores acima em grupos, que

# Criando a nova variável
df$fact.credit.duration.months<-findInterval(df$credit.duration.months,
                                             c(0,6,12,18,24,30,36))

# Dando nomes as observações da variável.
library(dplyr)
```

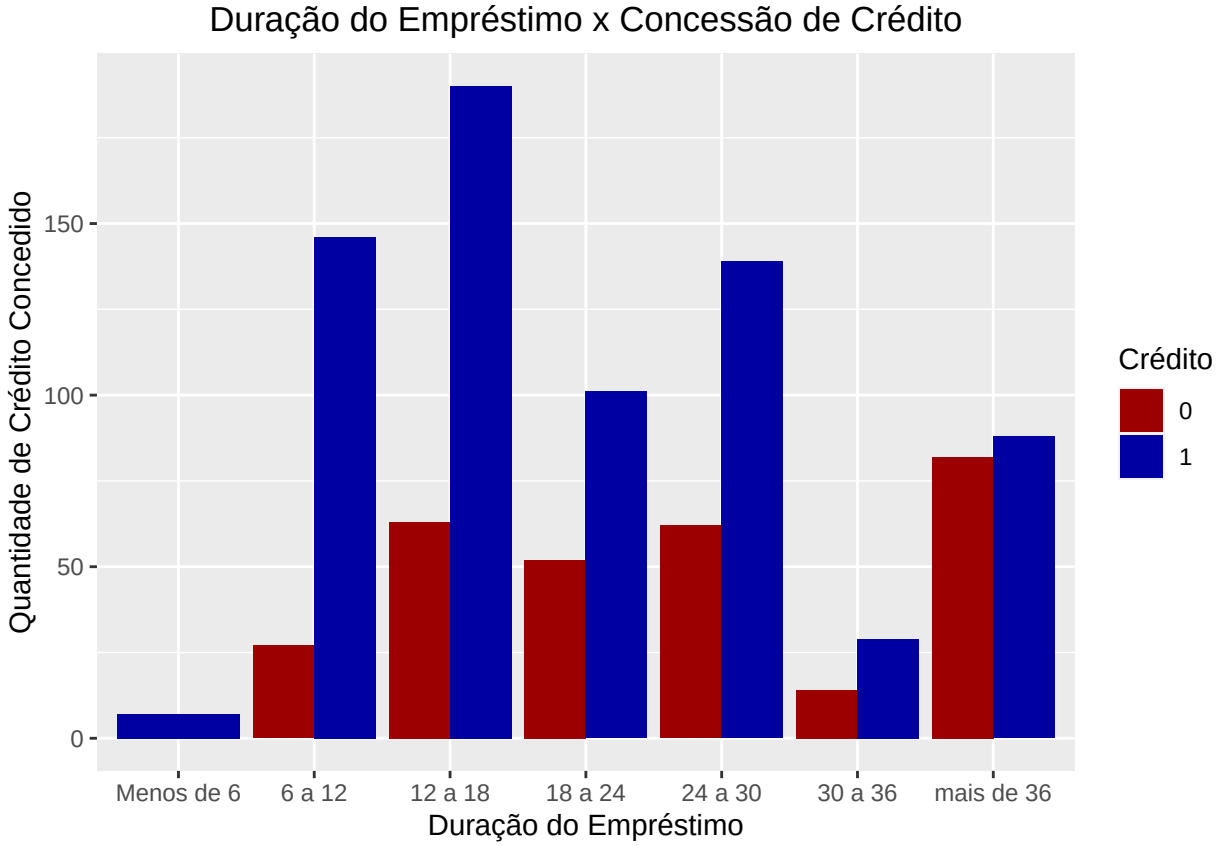
```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
df<-df %>%
  mutate(fact.credit.duration.months=as.factor(fact.credit.duration.months))
levels(df$fact.credit.duration.months)<-c("Menos de 6","6 a 12", "12 a 18",
                                          "18 a 24", "24 a 30", "30 a 36",
                                          "mais de 36")

# Visualizando os primeiros dados das duas colunas.
```

```r
head(df[, c("credit.duration.months", "fact.credit.duration.months")])
```

```
##   credit.duration.months fact.credit.duration.months
## 1                     18                    18 a 24
## 2                      9                     6 a 12
## 3                     12                    12 a 18
## 4                     12                    12 a 18
## 5                     12                    12 a 18
## 6                     10                     6 a 12
```

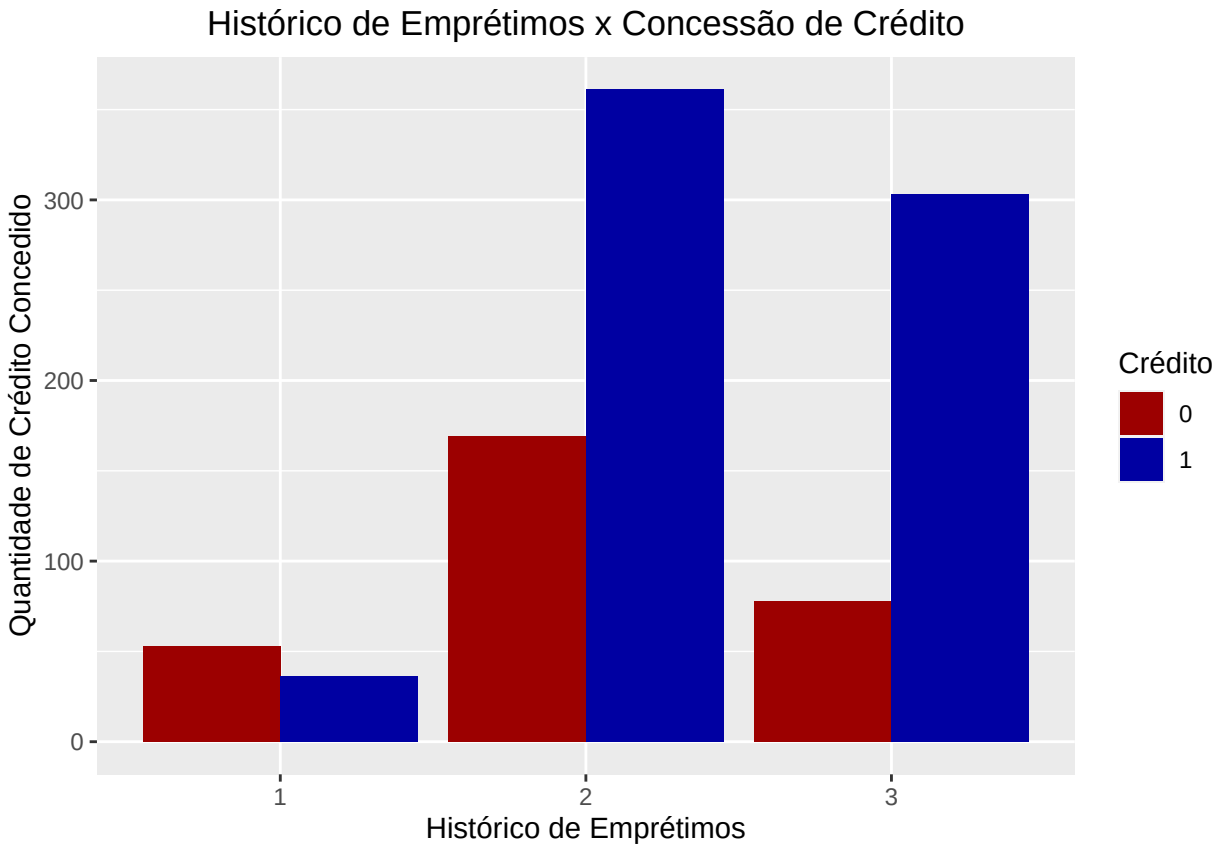
```r
#Análise Gráfica

ggplot(df, aes(fact.credit.duration.months, ..count..)) +
  geom_bar(aes(fill = credit.rating), position = "dodge") +
  scale_fill_manual(values = c("#9c0000", "#0000a2") ) +
  labs(x = "Duração do Empréstimo", y = "Quantidade de Crédito Concedido",
       fill = "Crédito", title = "Duração do Empréstimo x Concessão de Crédito") +
  theme(plot.title = element_text(hjust = 0.5))
```




```r
# Agora podemos visualizar melhor a conclusão dita acima.

# previous.credit.payment.status

# Tranformando a variável de numérica para fator
df$previous.credit.payment.status <- as.factor(df$previous.credit.payment.status)

# Quantidade de cada fator nessa variável
table(df$previous.credit.payment.status)
```

```
##
##   1   2   3
##  89 530 381
```

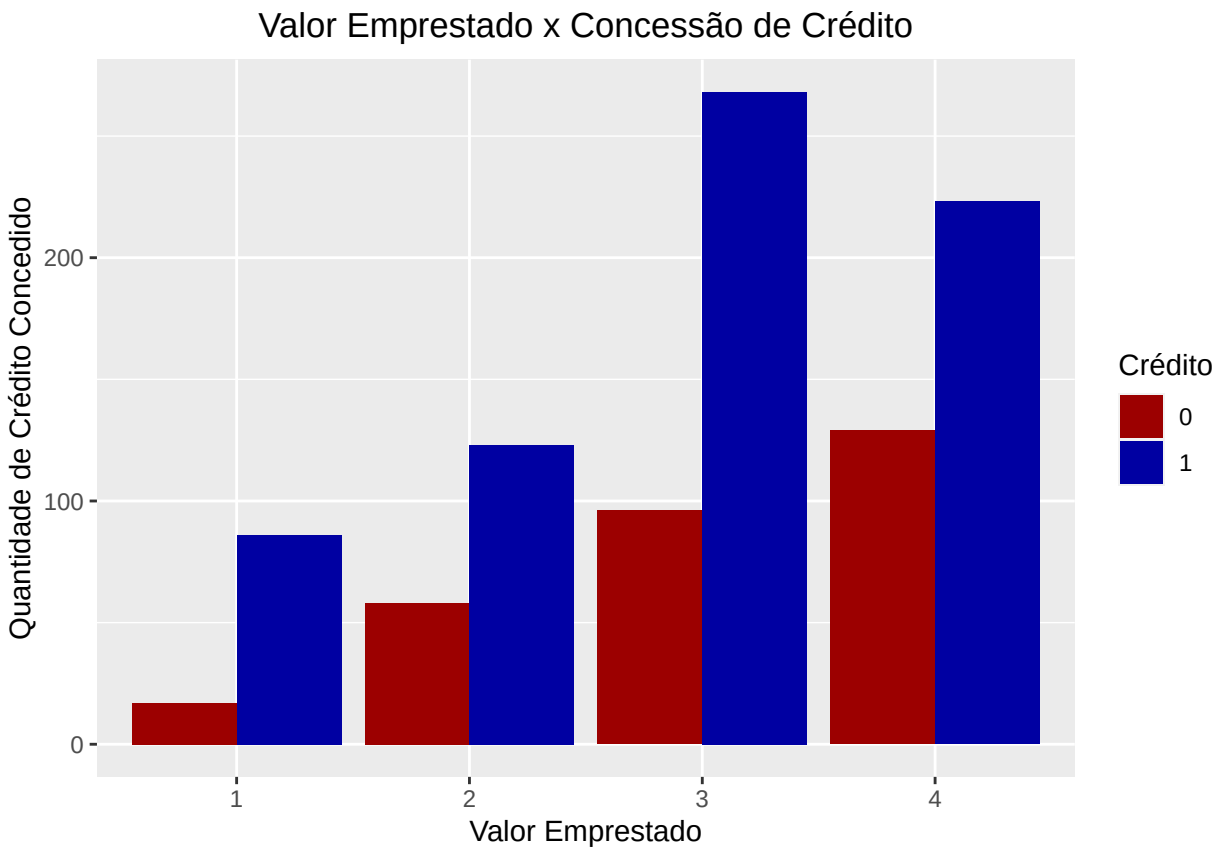
```
# Gráfico da contagem da variável por crédito financeiro concedido ou não.
ggplot(df, aes(previous.credit.payment.status, ..count..)) +
  geom_bar(aes(fill = credit.rating), position = "dodge") +
  scale_fill_manual(values = c("#9c0000", "#0000a2") ) +
  labs(x = "Histórico de Emprétimos", y = "Quantidade de Crédito Concedido",
       fill = "Crédito", title = "Histórico de Emprétimos x Concessão de Crédito") +
  theme(plot.title = element_text(hjust = 0.5))
```



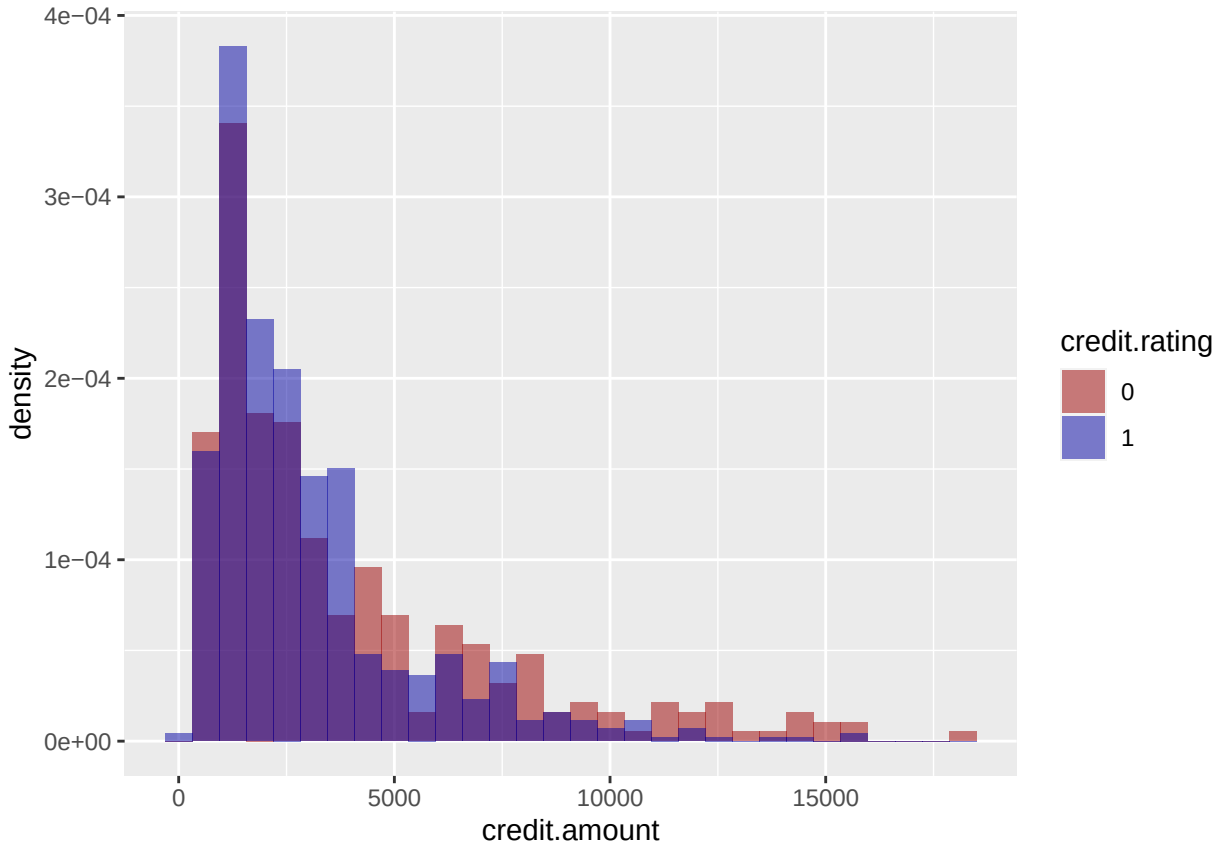

```
# Como podemos ver pelo gráfico quem tem mais problemas históricos com empréstimos tende a ter menos ch

# credit.purpose

# Tranformando a variável de numérica para fator
df$credit.purpose <- as.factor(df$credit.purpose)

# Quantidade de cada fator nessa variável
table(df$credit.purpose)
```

```
##
##   1   2   3   4
## 103 181 364 352
```

```
# Gráfico da contagem da variável por crédito financeiro concedido ou não.
ggplot(df, aes(credit.purpose, ..count..)) +
  geom_bar(aes(fill = credit.rating), position = "dodge") +
  scale_fill_manual(values = c("#9c0000", "#0000a2")) +
  labs(x = "Valor Emprestado", y = "Quantidade de Crédito Concedido",
       fill = "Crédito", title = "Valor Emprestado x Concessão de Crédito") +
  theme(plot.title = element_text(hjust = 0.5))
```




```
# Me parece que essa variável também influencia na concessão do crédito, mas acredito que não seja muit

# credit.amount

# Histograma do valor do emprétimo separado por créditos iaguais a sim e não.
ggplot(df, aes(credit.amount, fill = credit.rating)) +
  geom_histogram(alpha = 0.5, aes(y = ..density..), position = 'identity',
                 bins=30) +
  scale_fill_manual(values = c("#9c0000", "#0000a2") )
```

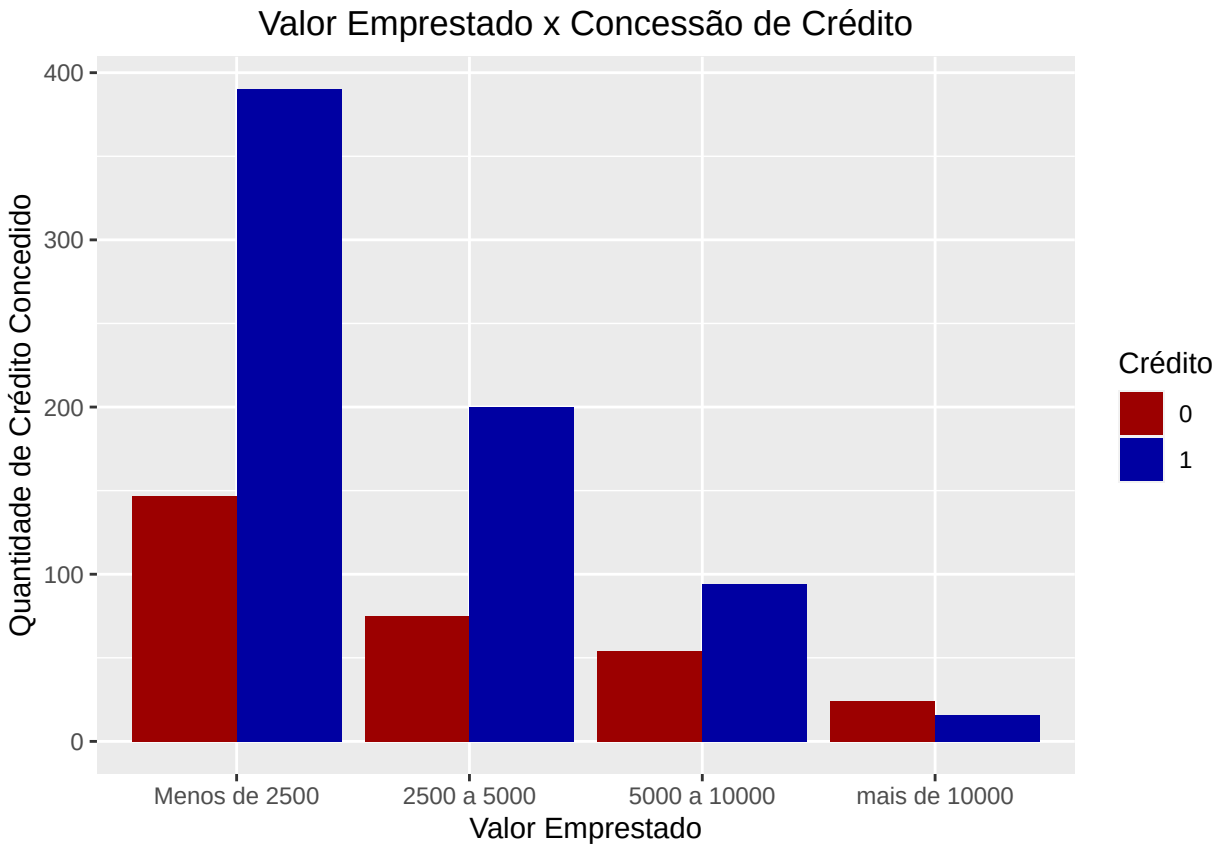



```
# Não consigo tirar uma idéia muito clara dessa variável, mas acredito que ela faz uma pequena diferenç

# Para tentar visualizar melhor vou usar a mesma estratégia empregada na variável numérica anterior. Di

#Criando a variável fact.credit.amount
df$fact.credit.amount<-findInterval(df$credit.amount,
                                    c(0, 2500, 5000, 10000))
df<-df %>%
  mutate(fact.credit.amount=as.factor(fact.credit.amount))
levels(df$fact.credit.amount) <- c("Menos de 2500","2500 a 5000",
                                   "5000 a 10000", "mais de 10000")

# Visualizando os primeiros dados das duas colunas.
head(df[, c("credit.amount", "fact.credit.amount")])
```

```
##   credit.amount fact.credit.amount
## 1          1049      Menos de 2500
## 2          2799        2500 a 5000
## 3           841      Menos de 2500
## 4          2122      Menos de 2500
## 5          2171      Menos de 2500
## 6          2241      Menos de 2500
```

```
# Análise Gráfica
ggplot(df, aes(fact.credit.amount, ..count..)) +
  geom_bar(aes(fill = credit.rating), position = "dodge") +
  scale_fill_manual(values = c("#9c0000", "#0000a2")) +
  labs(x = "Valor Emprestado", y = "Quantidade de Crédito Concedido",
```

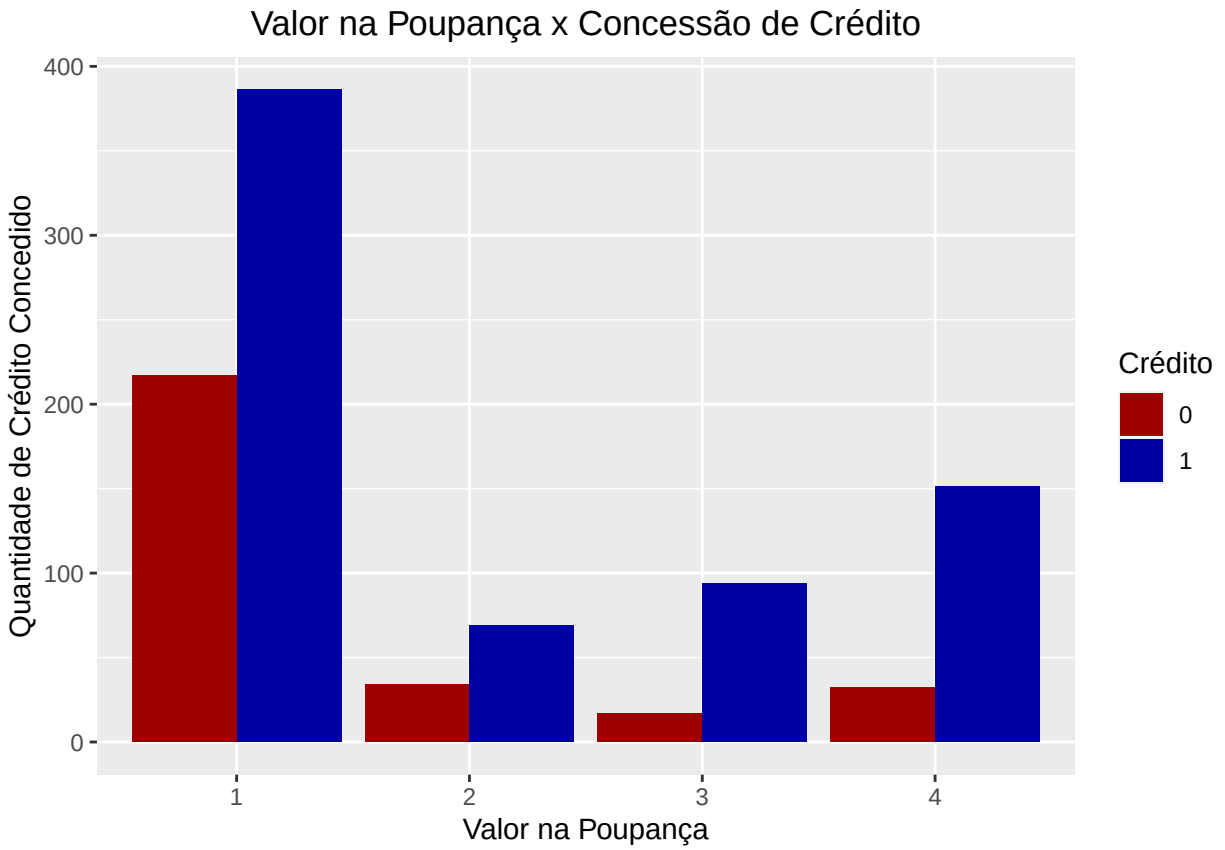
```
      fill = "Crédito", title = "Valor Emprestado x Concessão de Crédito") +
  theme(plot.title = element_text(hjust = 0.5))
```




```
# Agora sim podemos ter uma visão mais clara sobre essa variável. Quem solicita menos dinheiro, tem mai

# savings

# Tranformando a variável de numérica para fator
df$savings <- as.factor(df$savings)

# Quantidade de cada fator nessa variável
table(df$savings)
```

```
##
##   1   2   3   4
## 603 103 111 183
```

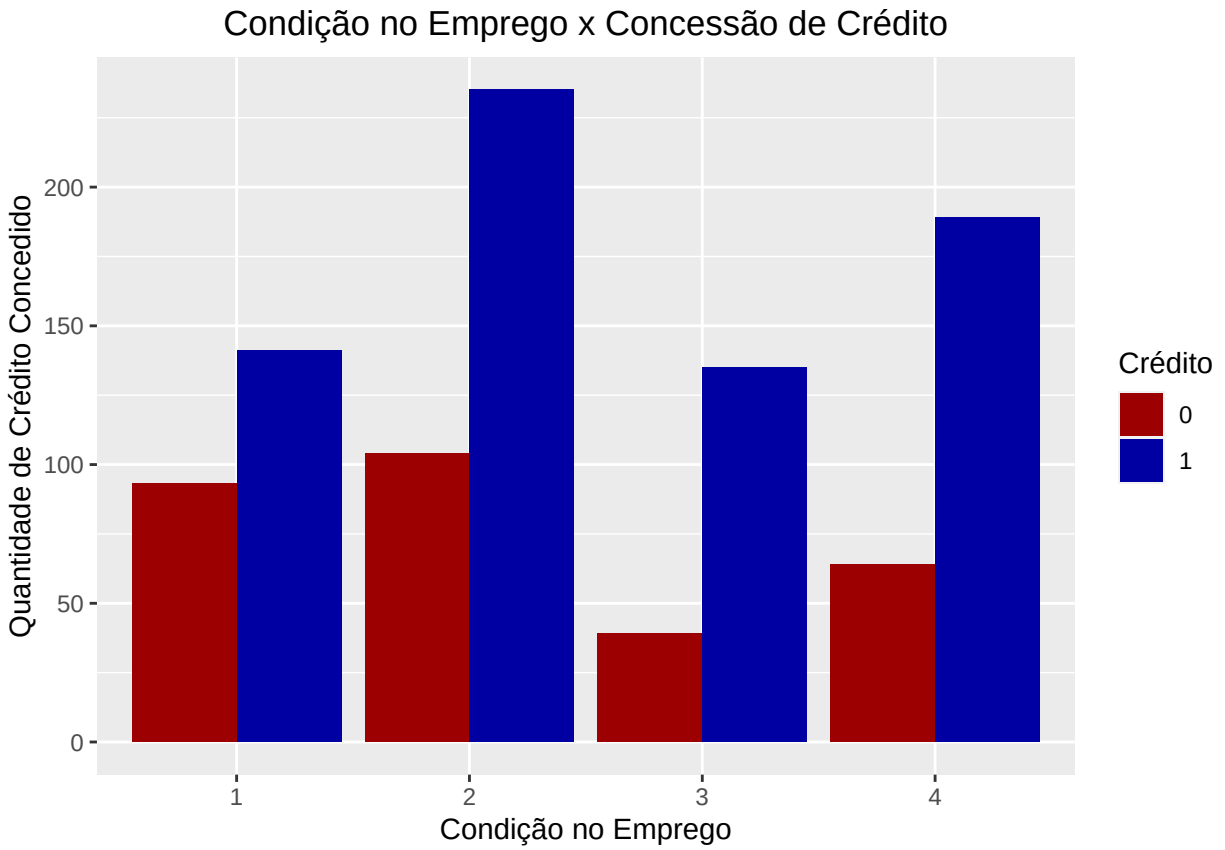
```
# Gráfico da contagem da variável por crédito financeiro concedido ou não.
ggplot(df, aes(savings, ..count..)) +
  geom_bar(aes(fill = credit.rating), position = "dodge") +
  scale_fill_manual(values = c("#9c0000", "#0000a2")) +
  labs(x = "Valor na Poupança", y = "Quantidade de Crédito Concedido",
       fill = "Crédito", title = "Valor na Poupança x Concessão de Crédito") +
  theme(plot.title = element_text(hjust = 0.5))
```




```
# Como esperávamos quem tem mais dinheiro na poupança tem mais chance de ter o crédito concedido.

# employment.duration

# Tranformando a variável de numérica para fator
df$employment.duration <- as.factor(df$employment.duration)

# Quantidade de cada fator nessa variável
table(df$employment.duration)
```

```
##
##   1   2   3   4
## 234 339 174 253
```

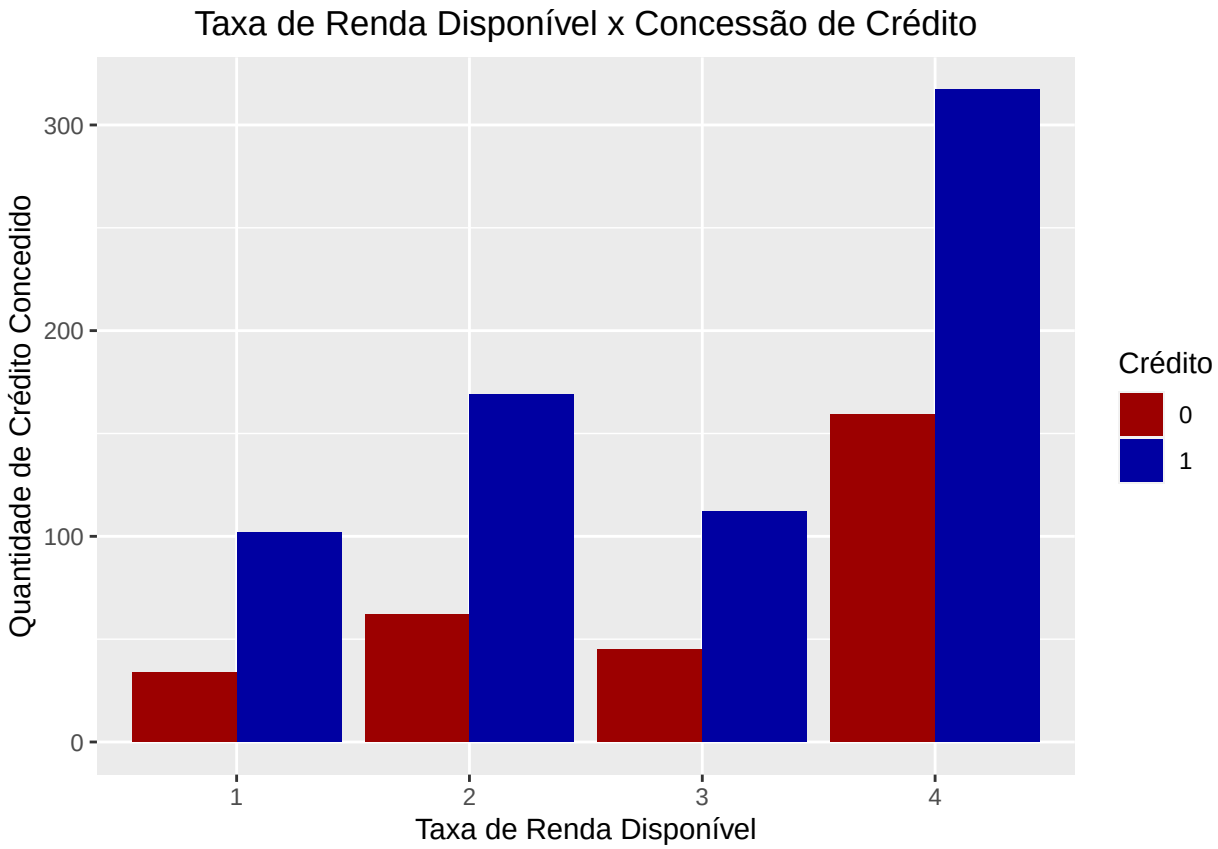
```
# Gráfico da contagem da variável por crédito financeiro concedido ou não.
ggplot(df, aes(employment.duration, ..count..)) +
  geom_bar(aes(fill = credit.rating), position = "dodge") +
  scale_fill_manual(values = c("#9c0000", "#0000a2")) +
  labs(x = "Condição no Emprego", y = "Quantidade de Crédito Concedido",
       fill = "Crédito", title = "Condição no Emprego x Concessão de Crédito") +
  theme(plot.title = element_text(hjust = 0.5))
```




```
# Essa variável também influencia diretamente em na variável target. Quem está desempregado, ou a menos

# installment.rate

# Tranformando a variável de numérica para fator
df$installment.rate <- as.factor(df$installment.rate)

# Quantidade de cada fator nessa variável
table(df$installment.rate)
```

```
##
##   1   2   3   4
## 136 231 157 476
```

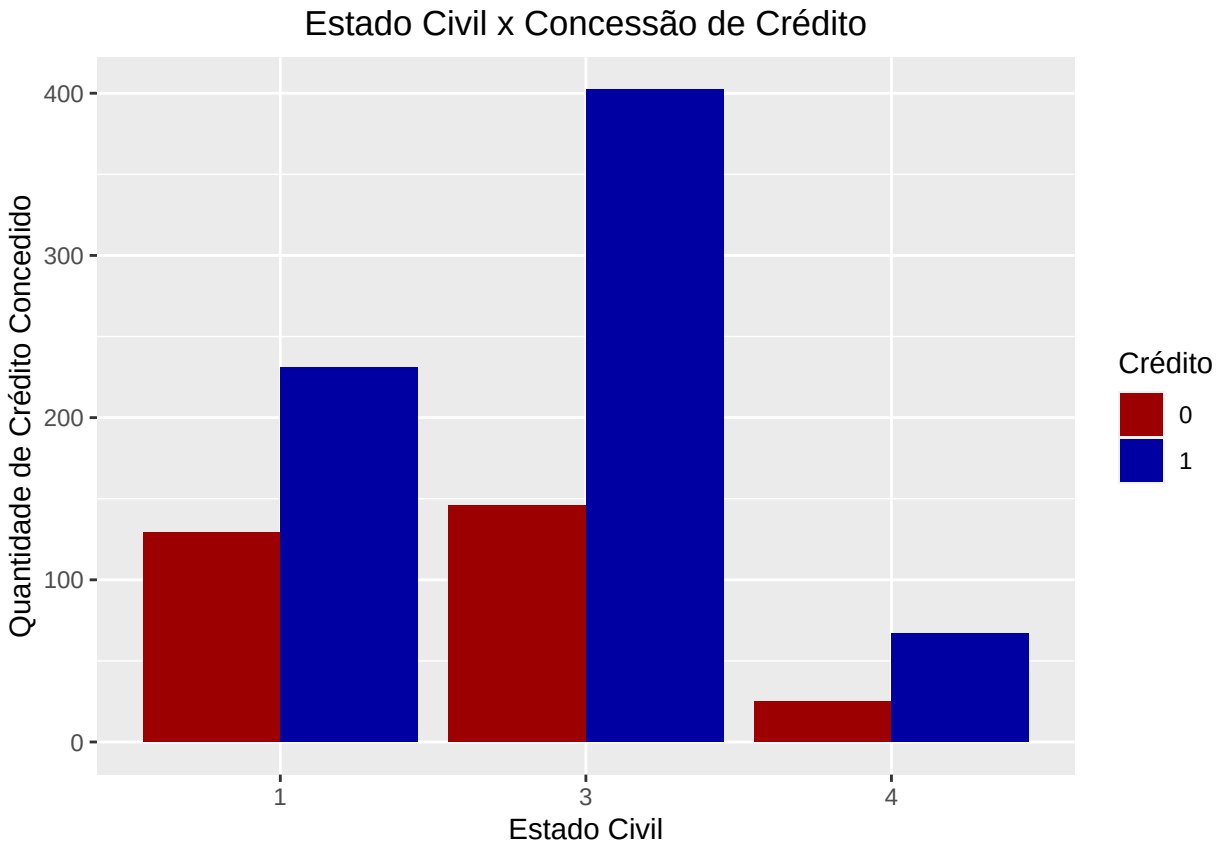
```
# Gráfico da contagem da variável por crédito financeiro concedido ou não.
ggplot(df, aes(installment.rate, ..count..)) +
  geom_bar(aes(fill = credit.rating), position = "dodge") +
  scale_fill_manual(values = c("#9c0000", "#0000a2")) +
  labs(x = "Taxa de Renda Disponível", y = "Quantidade de Crédito Concedido",
       fill = "Crédito",
       title = "Taxa de Renda Disponível x Concessão de Crédito") +
  theme(plot.title = element_text(hjust = 0.5))
```




```
# Esta é outra variável interessante, pois como vemos, quem tem mais renda disponível, tem tem mais cha

# marital.status

# Tranformando a variável de numérica para fator
df$marital.status <- as.factor(df$marital.status)

# Quantidade de cada fator nessa variável
table(df$marital.status)
```

```
##
##   1   3   4
## 360 548  92
```

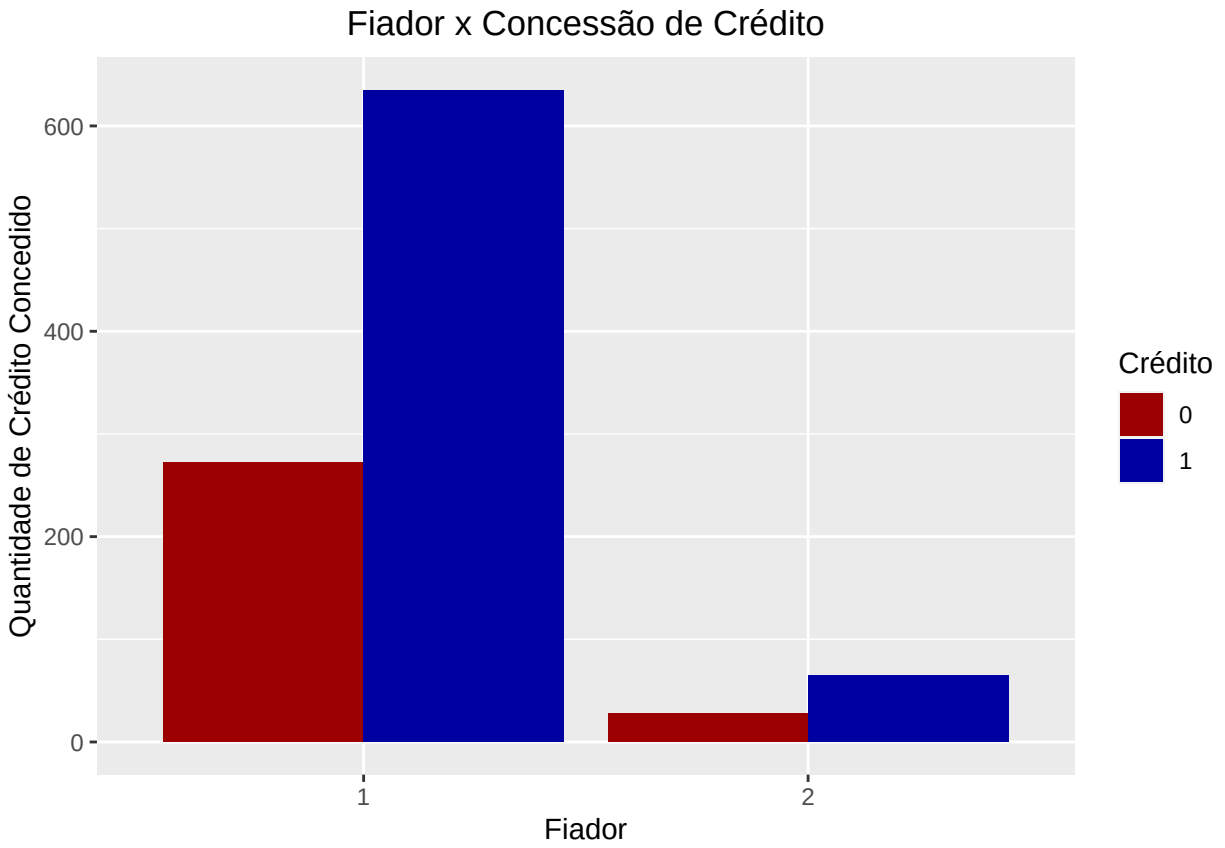
```
# Gráfico da contagem da variável por crédito financeiro concedido ou não.
ggplot(df, aes(marital.status, ..count..)) +
  geom_bar(aes(fill = credit.rating), position = "dodge") +
  scale_fill_manual(values = c("#9c0000", "#0000a2")) +
  labs(x = "Estado Civil", y = "Quantidade de Crédito Concedido",
       fill = "Crédito", title = "Estado Civil x Concessão de Crédito") +
  theme(plot.title = element_text(hjust = 0.5))
```




```
# Podemos ver nesse gráfico que homens casados / viúvos têm mais chance de ter o crédito concedido do q

# guarantor

# Tranformando a variável de numérica para fator
df$guarantor <- as.factor(df$guarantor)

# Quantidade de cada fator nessa variável
table(df$guarantor)
```

```
##
##   1   2
## 907  93
```

```
# Gráfico da contagem da variável por crédito financeiro concedido ou não.
ggplot(df, aes(guarantor, ..count..)) +
  geom_bar(aes(fill = credit.rating), position = "dodge") +
  scale_fill_manual(values = c("#9c0000", "#0000a2")) +
  labs(x = "Fiador", y = "Quantidade de Crédito Concedido", fill = "Crédito", title = "Fiador x Concessã
  theme(plot.title = element_text(hjust = 0.5))
```




```
# Essa variável não parece fazer muita diferença para o modelo preditivo, pois além de termos poucas ob

perct.guarantor <- group_by(df, guarantor) %>%
  mutate(group_size = n()) %>%
  group_by(guarantor, credit.rating) %>%
  summarise(perc = (n()/max(group_size)*100))
perct.guarantor
```

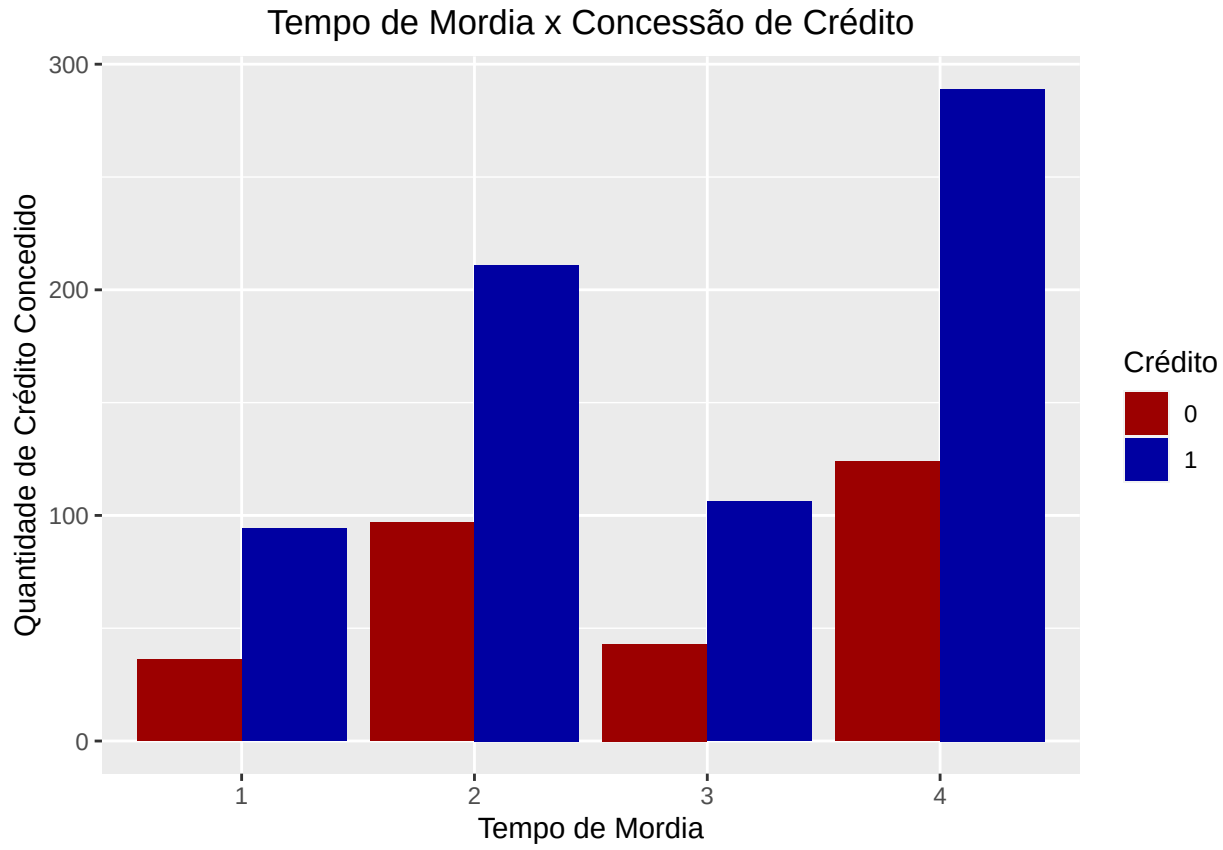
```
## # A tibble: 4 x 3
## # Groups:   guarantor [2]
##   guarantor credit.rating  perc
##   <fct>     <fct>         <dbl>
## 1 1         0              30.0
## 2 1         1              70.0
## 3 2         0              30.1
## 4 2         1              69.9
```

```
# Apenas retificar o que eu disse acima fiz essa tabela para vermos como esses dados estão distribuidos

# residence.duration

# Tranformando a variável de numérica para fator
df$residence.duration <- as.factor(df$residence.duration)

# Quantidade de cada fator nessa variável
table(df$residence.duration)
```

```
##
```

```
##   1   2   3   4
## 130 308 149 413
```

```
# Gráfico da contagem da variável por crédito financeiro concedido ou não.
ggplot(df, aes(residence.duration, ..count..)) +
  geom_bar(aes(fill = credit.rating), position = "dodge") +
  scale_fill_manual(values = c("#9c0000", "#0000a2")) +
  labs(x = "Tempo de Mordia", y = "Quantidade de Crédito Concedido",
       fill = "Crédito", title = "Tempo de Mordia x Concessão de Crédito") +
  theme(plot.title = element_text(hjust = 0.5))
```




```
# Verificando o percentual das variáveis
perct.residence.duration <- group_by(df, residence.duration) %>%
  mutate(group_size = n()) %>%
  group_by(residence.duration, credit.rating) %>%
  summarise(perc = (n()/max(group_size)*100))
perct.residence.duration
```

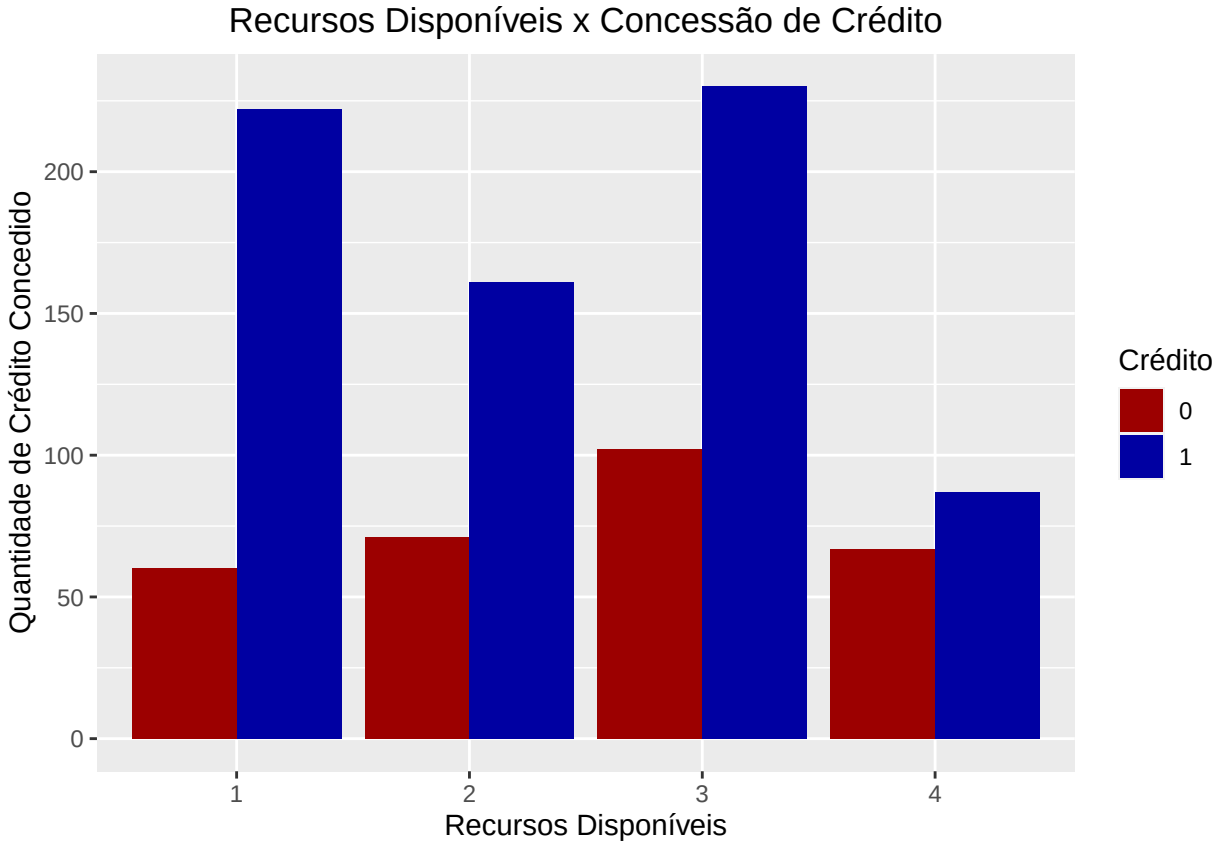
```
## # A tibble: 8 x 3
## # Groups:   residence.duration [4]
##   residence.duration credit.rating  perc
##   <fct>              <fct>         <dbl>
## 1 1                  0              27.7
## 2 1                  1              72.3
## 3 2                  0              31.5
## 4 2                  1              68.5
## 5 3                  0              28.9
## 6 3                  1              71.1
```

```
## 7 4                   0              30.0
## 8 4                   1              70.0
```

```
# Conforme podemos ver no gráfico e na tabela, o percentual de chance de conseguir ou não crédito é qua

# current.assets

# Quantidade de cada fator nessa variável
df$current.assets <- as.factor(df$current.assets)

# Quantidade de cada fator nessa variável
table(df$current.assets)
```

```
##
##   1   2   3   4
## 282 232 332 154
```

```
# Gráfico da contagem da variável por crédito financeiro concedido ou não.
ggplot(df, aes(current.assets, ..count..)) +
  geom_bar(aes(fill = credit.rating), position = "dodge") +
  scale_fill_manual(values = c("#9c0000", "#0000a2")) +
  labs(x = "Recursos Disponíveis", y = "Quantidade de Crédito Concedido",
       fill = "Crédito", title = "Recursos Disponíveis x Concessão de Crédito") +
  theme(plot.title = element_text(hjust = 0.5))
```

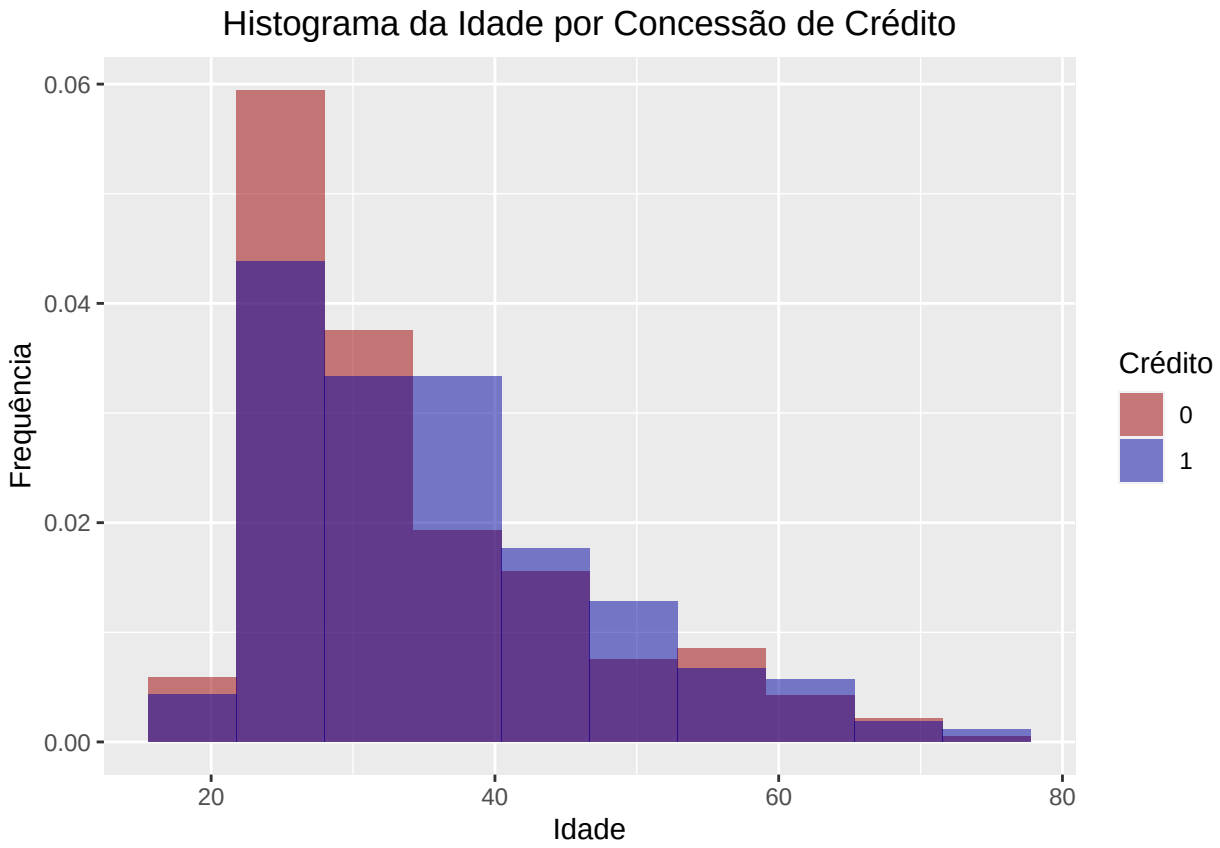



```
# Verificando o percentual das variáveis
perct.current.assets <- group_by(df, current.assets) %>%
  mutate(group_size = n()) %>%
  group_by(current.assets, credit.rating) %>%
```

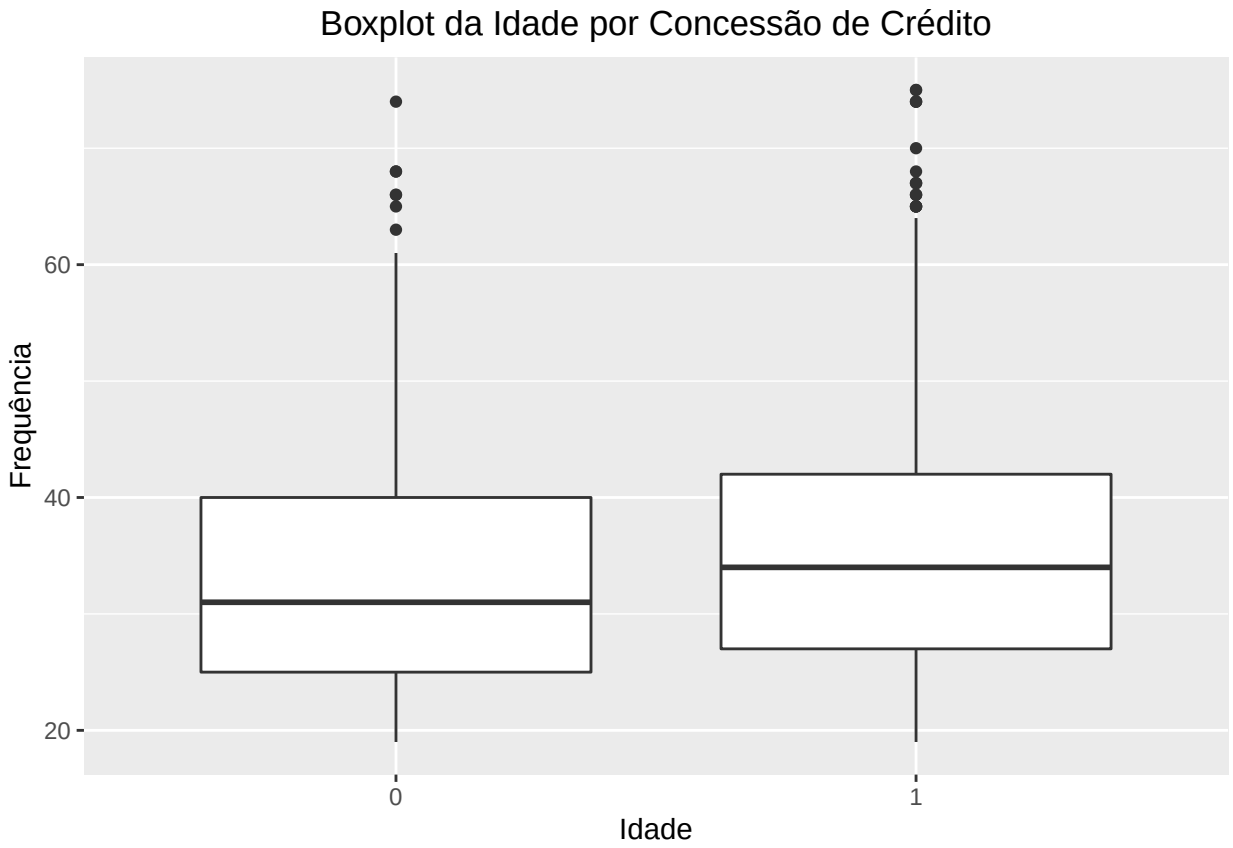
```
  summarise(perc = (n()/max(group_size)*100))
perct.current.assets
```

```
## # A tibble: 8 x 3
## # Groups:   current.assets [4]
##   current.assets credit.rating  perc
##   <fct>          <fct>         <dbl>
## 1 1              0              21.3
## 2 1              1              78.7
## 3 2              0              30.6
## 4 2              1              69.4
## 5 3              0              30.7
## 6 3              1              69.3
## 7 4              0              43.5
## 8 4              1              56.5
```

```
# Aparentemente quem é proprietário de uma casa tem menos chance de conseguir um empréstimo do que os o

# Age

# Histograma da idade separado por créditos igual a sim e não.
ggplot(df, aes(age, fill = credit.rating)) +
  geom_histogram(alpha = 0.5, aes(y = ..density..), position = 'identity',
                 bins=10) +
  scale_fill_manual(values = c("#9c0000", "#0000a2")) +
  labs(x = "Idade", y = "Frequência", fill = "Crédito",
       title = "Histograma da Idade por Concessão de Crédito") +
  theme(plot.title = element_text(hjust = 0.5))
```




```
# Boxplot
ggplot(df, aes(x=credit.rating, y=age)) +
  geom_boxplot() +
  labs(x = "Idade", y = "Frequência",
       fill = "Crédito", title = "Boxplot da Idade por Concessão de Crédito") +
  theme(plot.title = element_text(hjust = 0.5))
```




```
# Podemos ver pelo histograma e pelo boxplot que os mais jovens normalmente têm menos chance de consegu

# Vou criar a coluna de grupo de idade também para poder analisar melhor esses dados.

# Criando a variável fact.age
summary(df$age)
```

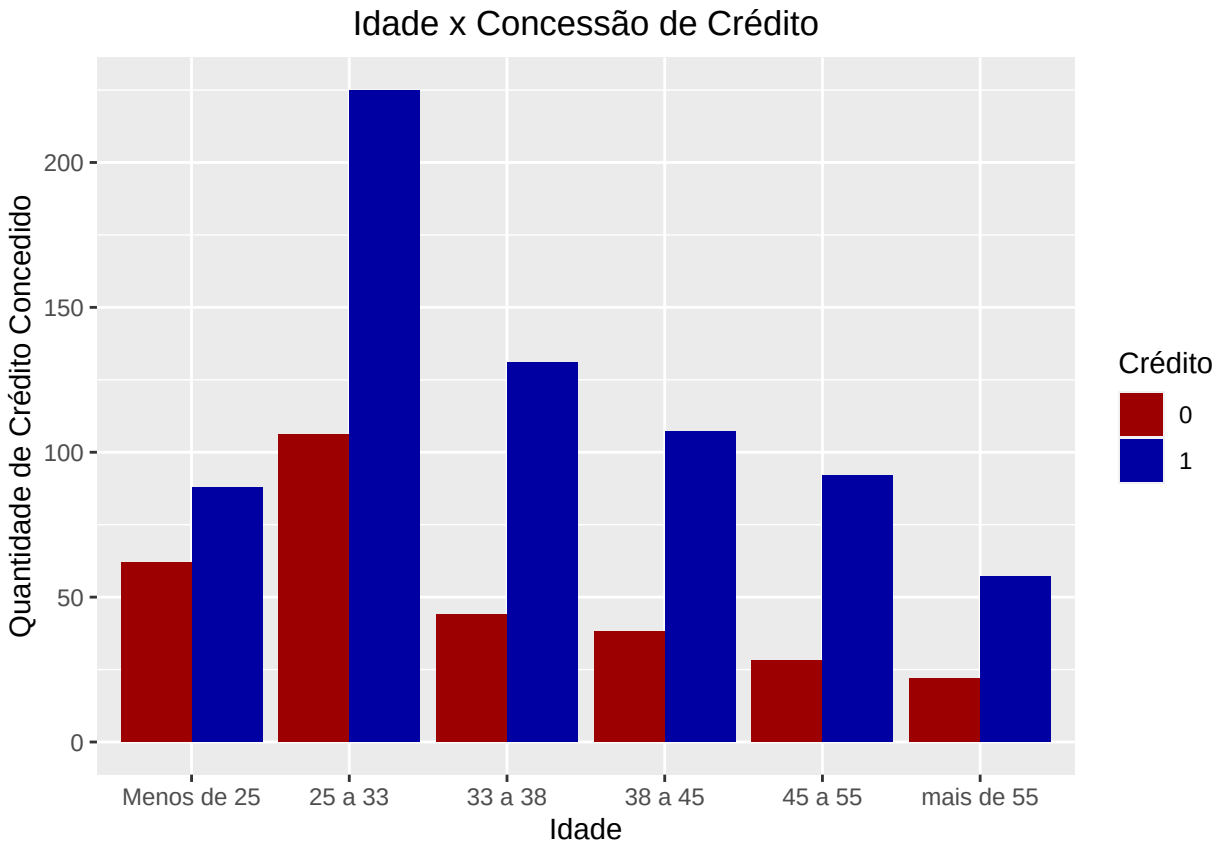
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   19.00   27.00   33.00   35.54   42.00   75.00
```

```
df$fact.age<-findInterval(df$age, c(18, 25, 33, 38, 45, 55))
df<-df %>%
  mutate(fact.age=as.factor(fact.age))
levels(df$fact.age) <- c("Menos de 25", "25 a 33", "33 a 38", "38 a 45", "45 a 55",
                         "mais de 55")

# Visualizando os primeiros dados das duas colunas.
head(df[, c("age", "fact.age")])
```

```
##   age    fact.age
## 1  21 Menos de 25
## 2  36     33 a 38
## 3  23 Menos de 25
## 4  39     38 a 45
## 5  38     38 a 45
## 6  48     45 a 55
```

```
# Análise Gráfica
ggplot(df, aes(fact.age, ..count..)) +
```

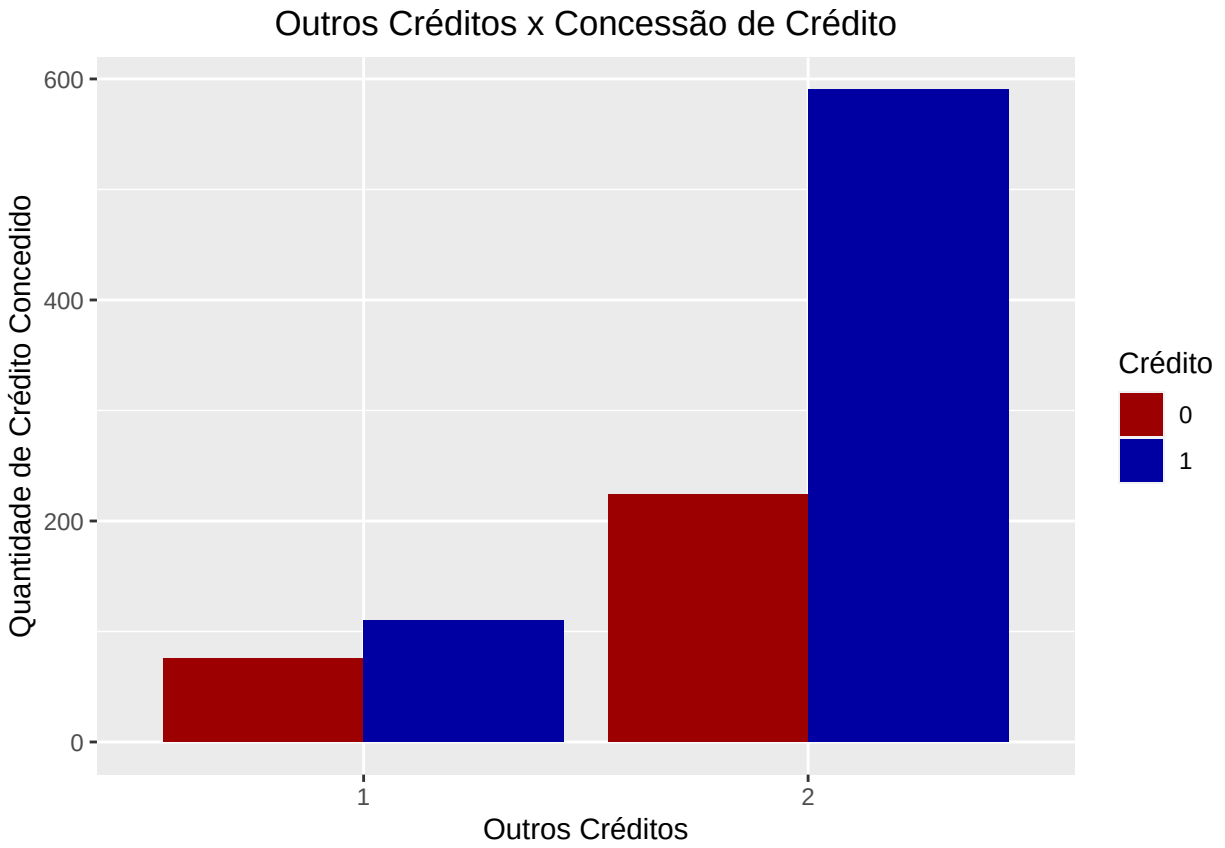
```
  geom_bar(aes(fill = credit.rating), position = "dodge") +
  scale_fill_manual(values = c("#9c0000", "#0000a2")) +
  labs(x = "Idade", y = "Quantidade de Crédito Concedido",
       fill = "Crédito", title = "Idade x Concessão de Crédito") +
  theme(plot.title = element_text(hjust = 0.5))
```




```
# Podemos perceber a relação entre a idade e a concessão de crédito de uma forma mais clara agora.

# other.credits

# Quantidade de cada fator nessa variável
df$other.credits <- as.factor(df$other.credits)

# Quantidade de cada fator nessa variável
table(df$other.credits)
```

```
##
##   1   2
## 186 814
```

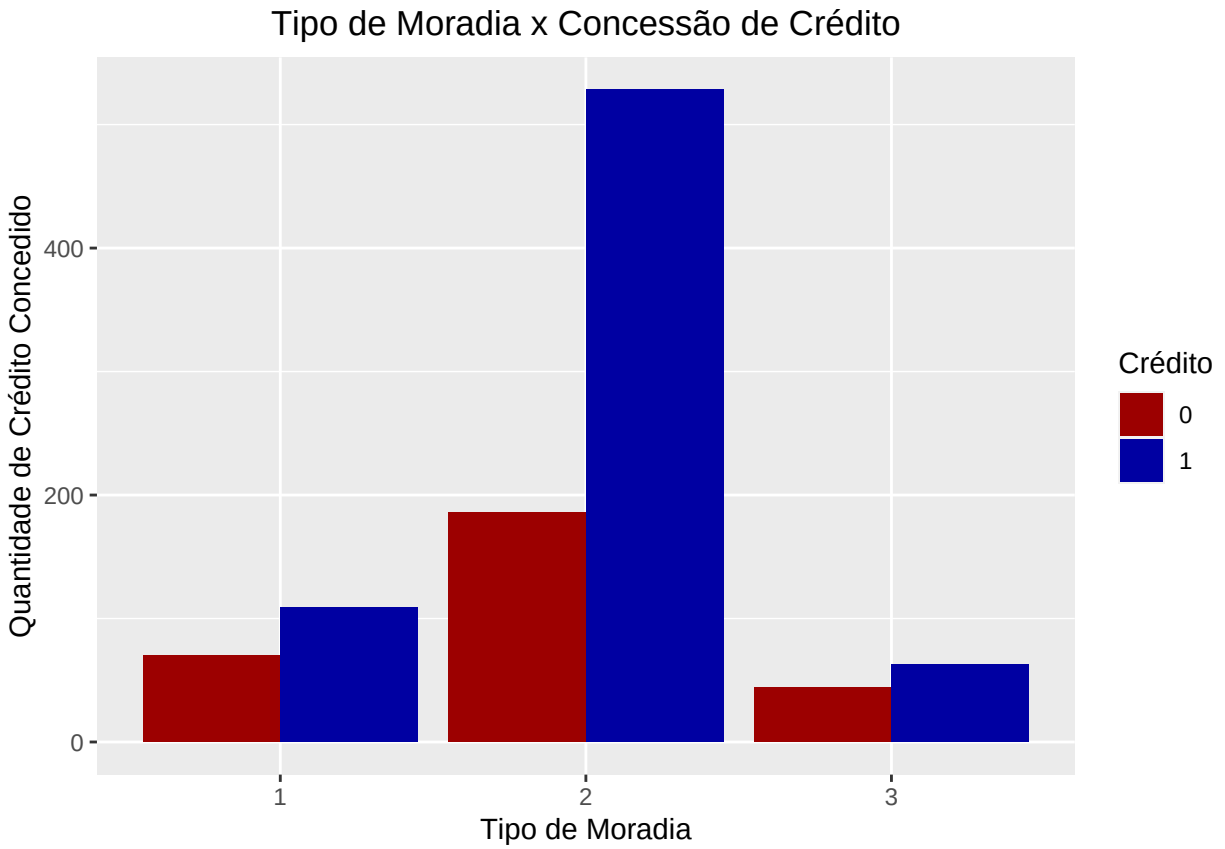
```
# Gráfico da contagem da variável por crédito financeiro concedido ou não.
ggplot(df, aes(other.credits, ..count..)) +
  geom_bar(aes(fill = credit.rating), position = "dodge") +
  scale_fill_manual(values = c("#9c0000", "#0000a2")) +
  labs(x = "Outros Créditos", y = "Quantidade de Crédito Concedido",
       fill = "Crédito", title = "Outros Créditos x Concessão de Crédito") +
  theme(plot.title = element_text(hjust = 0.5))
```




```
# Pessoas que tem créditos em outros bancos têm mais dificuldade em conseguir o crédito.

# apartment.type

# Quantidade de cada fator nessa variável
df$apartment.type <- as.factor(df$apartment.type)

# Quantidade de cada fator nessa variável
table(df$apartment.type)
```

```
##
##   1   2   3
## 179 714 107
```

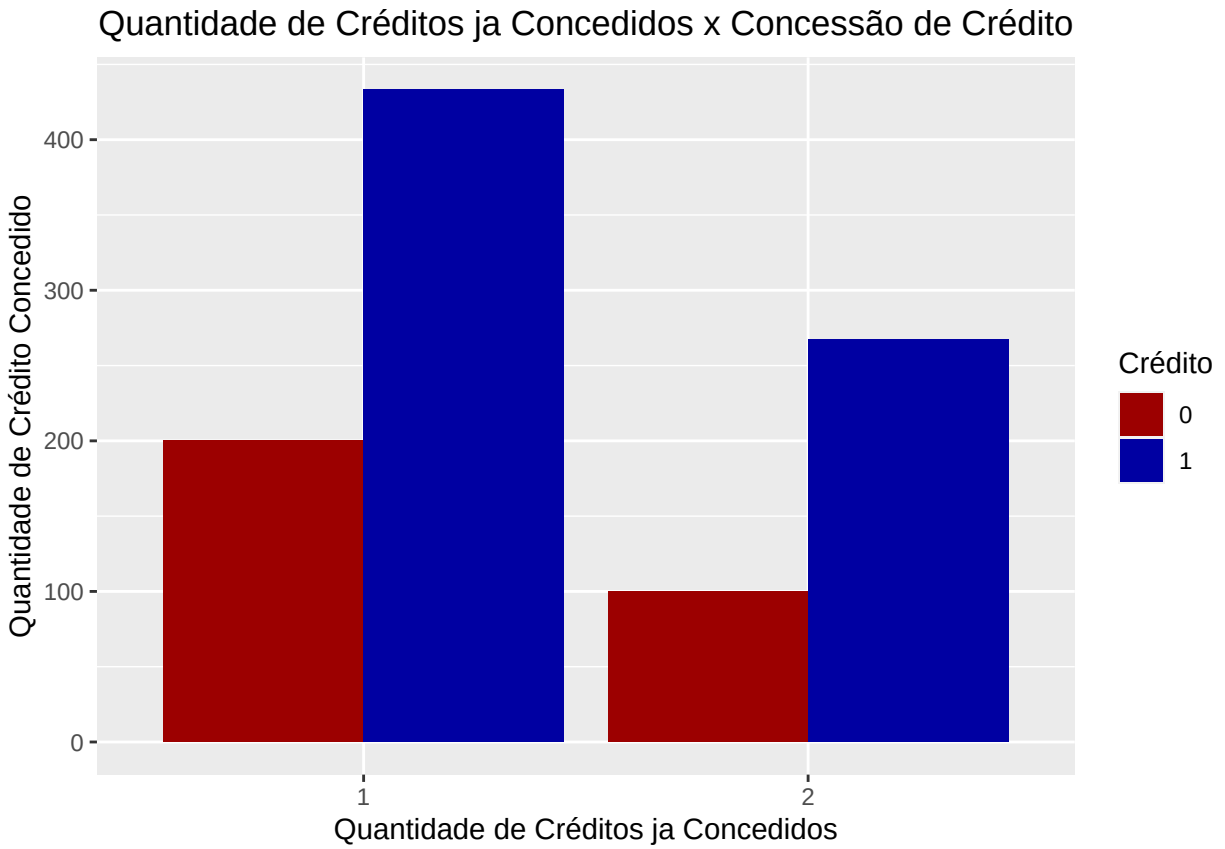
```
# Gráfico da contagem da variável por crédito financeiro concedido ou não.
ggplot(df, aes(apartment.type, ..count..)) +
  geom_bar(aes(fill = credit.rating), position = "dodge") +
  scale_fill_manual(values = c("#9c0000", "#0000a2")) +
  labs(x = "Tipo de Moradia", y = "Quantidade de Crédito Concedido",
       fill = "Crédito", title = "Tipo de Moradia x Concessão de Crédito") +
  theme(plot.title = element_text(hjust = 0.5))
```




```
# Essa variável também irá entrar em nosso modelo preditivo.

# bank.credits

# Quantidade de cada fator nessa variável
df$bank.credits <- as.factor(df$bank.credits)

# Quantidade de cada fator nessa variável
table(df$bank.credits)
```

```
##
##   1   2
## 633 367
```

```
# Gráfico da contagem da variável por crédito financeiro concedido ou não.
ggplot(df, aes(bank.credits, ..count..)) +
  geom_bar(aes(fill = credit.rating), position = "dodge") +
  scale_fill_manual(values = c("#9c0000", "#0000a2")) +
  labs(x = "Quantidade de Créditos ja Concedidos", y = "Quantidade de Crédito Concedido",
       fill = "Crédito", title = "Quantidade de Créditos ja Concedidos x Concessão de Crédito") +
  theme(plot.title = element_text(hjust = 0.5))
```




```
# Verificando o percentual das variáveis
perct.bank.credits <- group_by(df, bank.credits) %>%
  mutate(group_size = n()) %>%
  group_by(bank.credits, credit.rating) %>%
  summarise(perc = (n()/max(group_size)*100))
perct.bank.credits
```

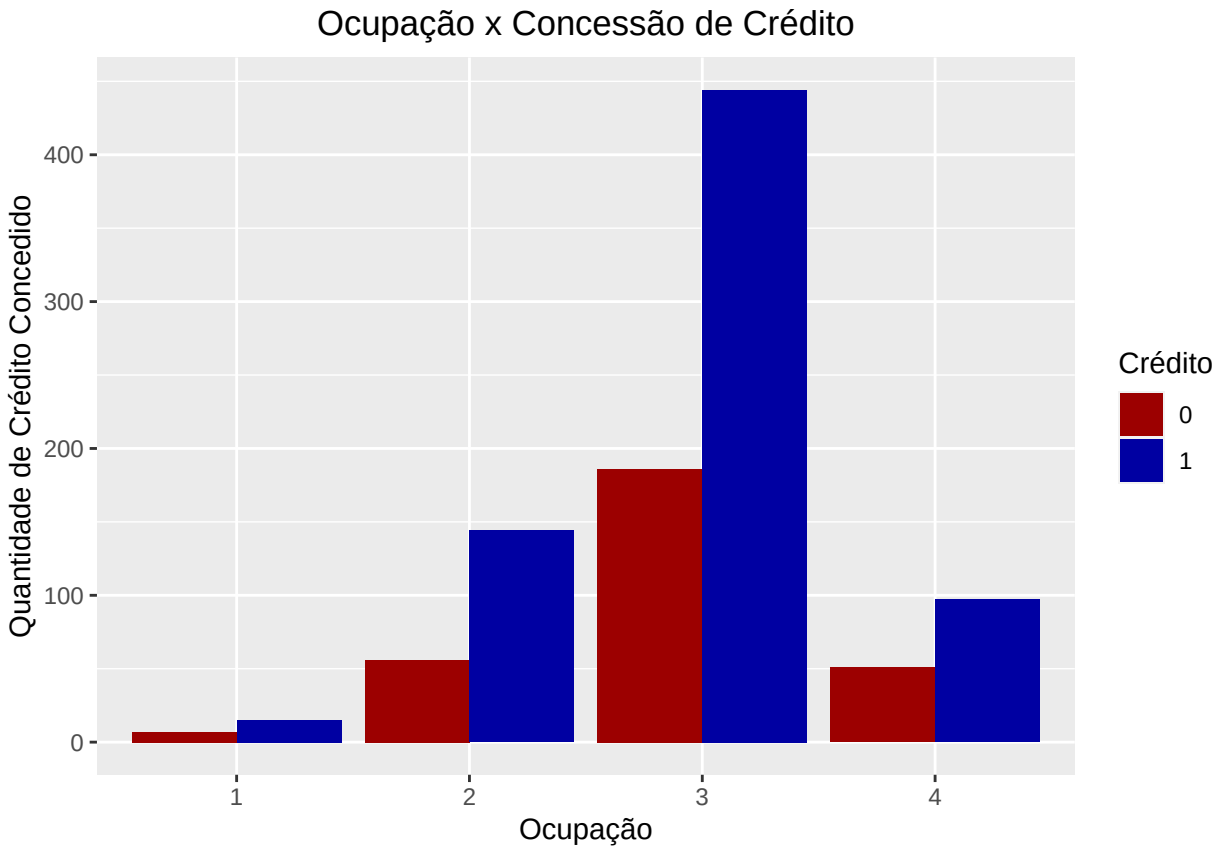
```
## # A tibble: 4 x 3
## # Groups:   bank.credits [2]
##   bank.credits credit.rating  perc
##   <fct>        <fct>         <dbl>
## 1 1            0              31.6
## 2 1            1              68.4
## 3 2            0              27.2
## 4 2            1              72.8
```

```
# A diferença no número de créditos ja concedidos é tão pouca que não vale a pena considerar no modelo

# occupation

# Quantidade de cada fator nessa variável
df$occupation <- as.factor(df$occupation)

# Quantidade de cada fator nessa variável
table(df$occupation)
```

```
##
##   1   2   3   4
```

```
##  22 200 630 148
```

```
# Gráfico da contagem da variável por crédito financeiro concedido ou não.
ggplot(df, aes(occupation, ..count..)) +
  geom_bar(aes(fill = credit.rating), position = "dodge") +
  scale_fill_manual(values = c("#9c0000", "#0000a2")) +
  labs(x = "Ocupação", y = "Quantidade de Crédito Concedido",
       fill = "Crédito", title = "Ocupação x Concessão de Crédito") +
  theme(plot.title = element_text(hjust = 0.5))
```




```
# Verificando o percentual das variáveis
perct.occupation <- group_by(df, occupation) %>%
  mutate(group_size = n()) %>%
  group_by(occupation, credit.rating) %>%
  summarise(perc = (n()/max(group_size)*100))
perct.occupation
```

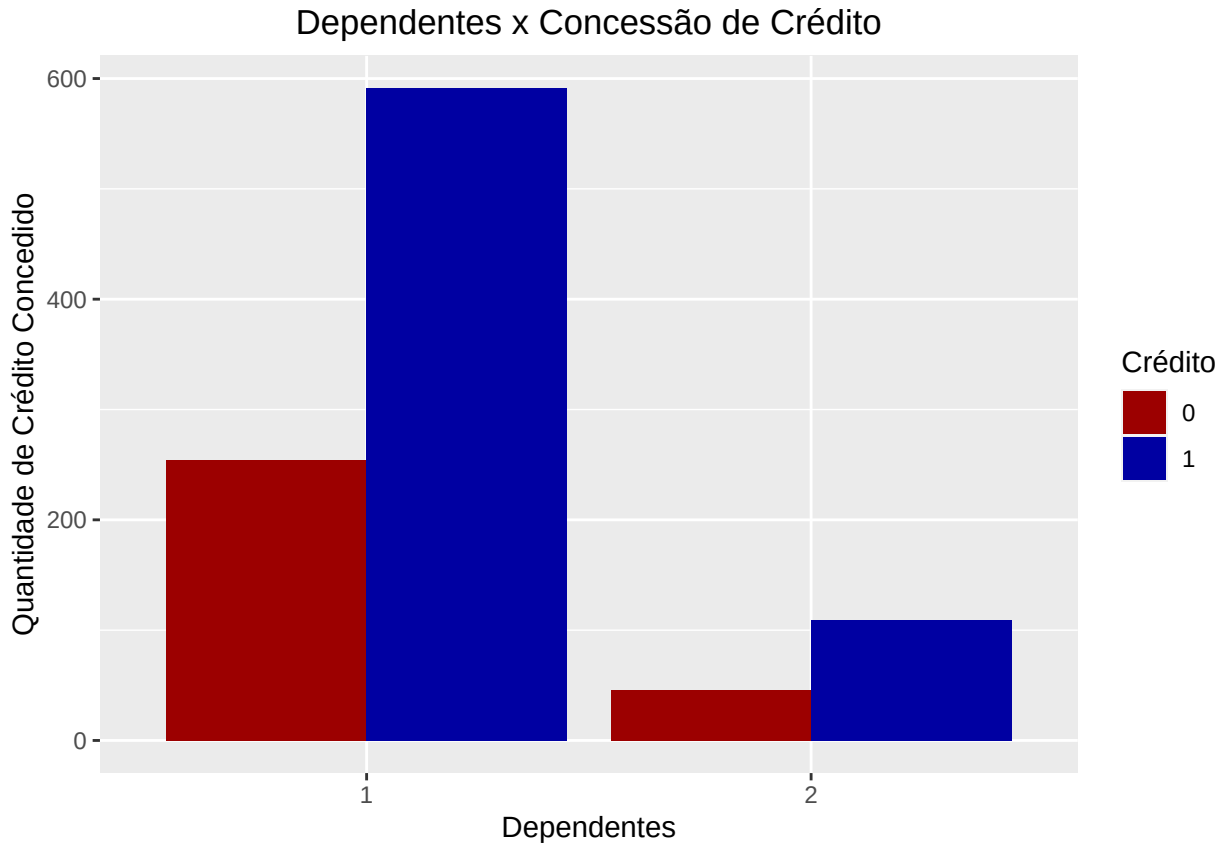
```
## # A tibble: 8 x 3
## # Groups:   occupation [4]
##   occupation credit.rating  perc
##   <fct>      <fct>         <dbl>
## 1 1          0              31.8
## 2 1          1              68.2
## 3 2          0              28.
## 4 2          1              72
## 5 3          0              29.5
## 6 3          1              70.5
## 7 4          0              34.5
```

```
## 8 4          1              65.5
```

```
# A diferença percentual de cada categoria é bem pouca, acho que não compensa utilizar essa variável em

# dependents

# Quantidade de cada fator nessa variável
df$dependents <- as.factor(df$dependents)

# Quantidade de cada fator nessa variável
table(df$dependents)
```

```
##
##   1   2
## 845 155
```

```
# Gráfico da contagem da variável por crédito financeiro concedido ou não.
ggplot(df, aes(dependents, ..count..)) +
  geom_bar(aes(fill = credit.rating), position = "dodge") +
  scale_fill_manual(values = c("#9c0000", "#0000a2")) +
  labs(x = "Dependentes", y = "Quantidade de Crédito Concedido",
       fill = "Crédito", title = "Dependentes x Concessão de Crédito") +
  theme(plot.title = element_text(hjust = 0.5))
```




```
# Verificando o percentual das variáveis
perct.dependents <- group_by(df, dependents) %>%
  mutate(group_size = n()) %>%
  group_by(dependents, credit.rating) %>%
  summarise(perc = (n()/max(group_size)*100))
```

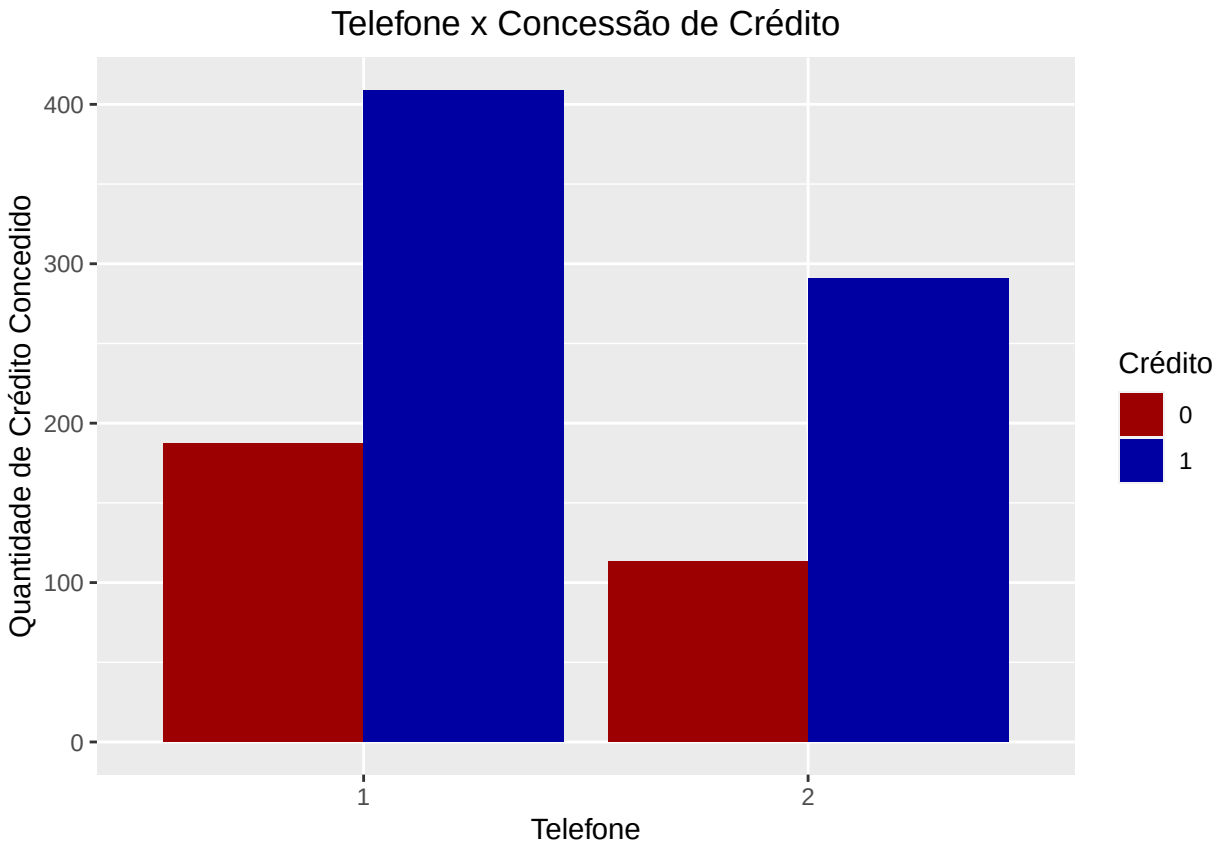
```
perct.dependents
```

```
## # A tibble: 4 x 3
## # Groups:   dependents [2]
##   dependents credit.rating  perc
##   <fct>      <fct>         <dbl>
## 1 1          0              30.1
## 2 1          1              69.9
## 3 2          0              29.7
## 4 2          1              70.3
```

```
# Essa variável também não compensa se utilizada em nosso modelo preditivo.

# telephone

# Quantidade de cada fator nessa variável
df$telephone <- as.factor(df$telephone)

# Quantidade de cada fator nessa variável
table(df$telephone)
```

```
##
##   1   2
## 596 404
```

```
# Gráfico da contagem da variável por crédito financeiro concedido ou não.
ggplot(df, aes(telephone, ..count..)) +
  geom_bar(aes(fill = credit.rating), position = "dodge") +
  scale_fill_manual(values = c("#9c0000", "#0000a2")) +
  labs(x = "Telefone", y = "Quantidade de Crédito Concedido",
       fill = "Crédito", title = "Telefone x Concessão de Crédito") +
  theme(plot.title = element_text(hjust = 0.5))
```




```
# Verificando o percentual das variáveis
perct.telephone <- group_by(df, telephone) %>%
  mutate(group_size = n()) %>%
  group_by(telephone, credit.rating) %>%
  summarise(perc = (n()/max(group_size)*100))
perct.telephone
```

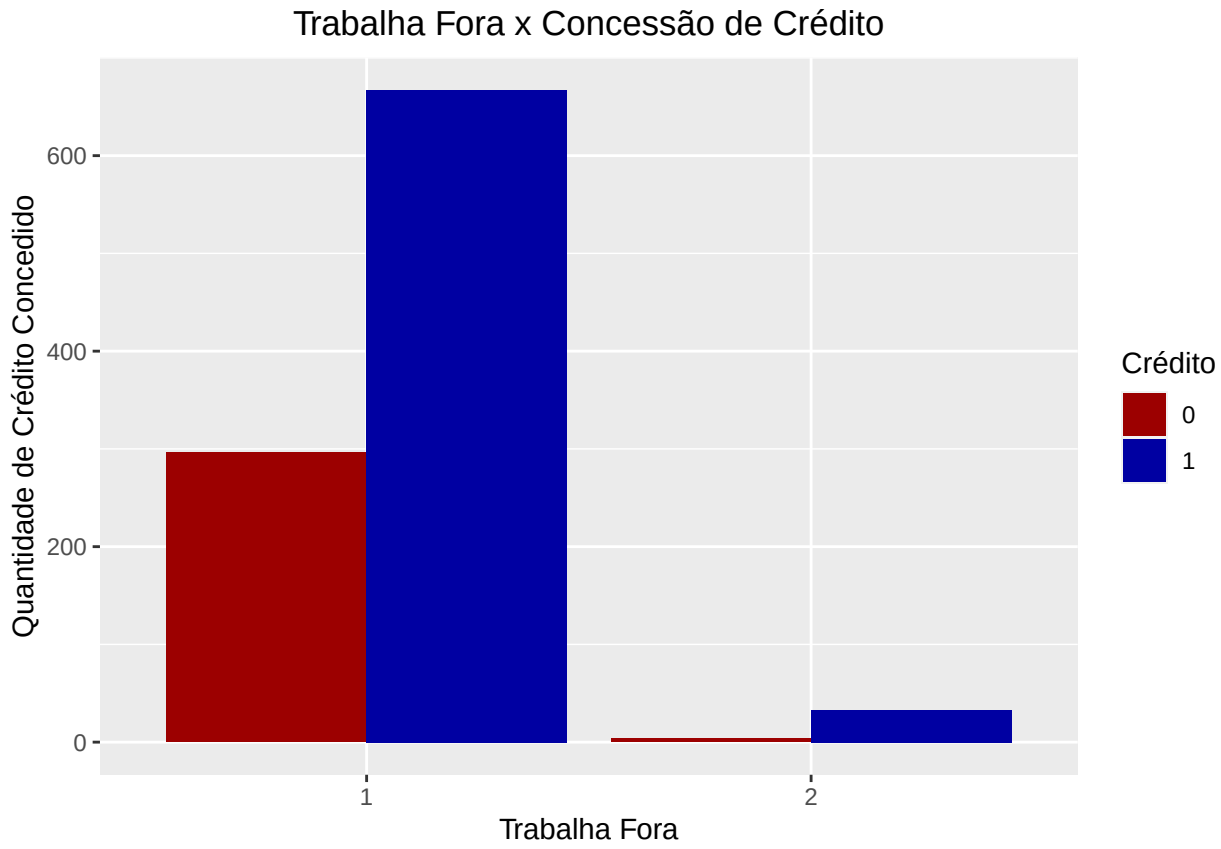
```
## # A tibble: 4 x 3
## # Groups:   telephone [2]
##   telephone credit.rating  perc
##   <fct>     <fct>         <dbl>
## 1 1         0              31.4
## 2 1         1              68.6
## 3 2         0              28.0
## 4 2         1              72.0
```

```
# A existência de telefone também não muda muito em relação à concessão ou não de crédito.

# foreign.worker

# Quantidade de cada fator nessa variável
df$foreign.worker <- as.factor(df$foreign.worker)

# Quantidade de cada fator nessa variável
table(df$foreign.worker)
```

```
##
##   1   2
```

```
## 963  37
```

```
# Gráfico da contagem da variável por crédito financeiro concedido ou não.
ggplot(df, aes(foreign.worker, ..count..)) +
  geom_bar(aes(fill = credit.rating), position = "dodge") +
  scale_fill_manual(values = c("#9c0000", "#0000a2")) +
  labs(x = "Trabalha Fora", y = "Quantidade de Crédito Concedido",
       fill = "Crédito", title = "Trabalha Fora x Concessão de Crédito") +
  theme(plot.title = element_text(hjust = 0.5))
```




```
# Verificando o percentual das variáveis
perct.foreign.worker <- group_by(df, foreign.worker) %>%
  mutate(group_size = n()) %>%
  group_by(foreign.worker, credit.rating) %>%
  summarise(perc = (n()/max(group_size)*100))
perct.foreign.worker
```

```
## # A tibble: 4 x 3
## # Groups:   foreign.worker [2]
##   foreign.worker credit.rating  perc
##   <fct>          <fct>         <dbl>
## 1 1              0              30.7
## 2 1              1              69.3
## 3 2              0              10.8
## 4 2              1              89.2
```

```
# Como podemos ver pessoas que trabalham e moram na mesma cidade têm mais chances de conseguir um empré
```

Treino e Teste

Agora iremos criar as variáveis de treino e teste para iniciar a predição do nosso data frame.

```
require(caTools)
```

```
## Loading required package: caTools
```

```
#Criando uma seed
set.seed(123)
#Dividindo o data frame em treino e teste.
sample = sample.split(df, SplitRatio = 0.70)
train = subset(df, sample ==TRUE)
test = subset(df, sample==FALSE)

#Verificando o número de linhas de cada data frame
nrow(train)
```

```
## [1] 667
```

```
nrow(test)
```

```
## [1] 333
```

Primeiros modelos preditivos

Nesta etapa iremos rodar os algoritmos de regressão logística, suport vector machine, árvore de decisão, random forest e Naive Bayes para ver qual deles se comportam melhor com as variáveis escolhidas por mim.

```
# Agora é chegou a hora de rodar os algoritmos de modelo preditivo.
# Vamos começar treinando o modelo com as variáveis que eu achei mais interessantes
# durante a fase de análise.
formula_v1 <- as.formula('credit.rating ~ account.balance +
                         fact.credit.duration.months +
                         previous.credit.payment.status + credit.purpose +
                         fact.credit.amount + savings + employment.duration +
                         installment.rate + marital.status + current.assets +
                         fact.age + other.credits + apartment.type +
                         foreign.worker')

# Treinando o modelo com o algoritmo de regressão logística
model_glm_v1 <- glm(formula = formula_v1, data = train, family = "binomial")

# Verificando alguns resultados do modelo treinado
summary(model_glm_v1)
```

```
##
## Call:
## glm(formula = formula_v1, family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5913  -0.6494   0.4095   0.7080   2.0969
##
## Coefficients:
```

```
##                                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)                          12.42380  553.34136   0.022 0.982087
## account.balance2                      0.31610    0.26265   1.204 0.228770
## account.balance3                      1.56185    0.26764   5.836 5.36e-09 ***
## fact.credit.duration.months6 a 12   -12.17017  553.34116  -0.022 0.982453
## fact.credit.duration.months12 a 18  -12.38685  553.34116  -0.022 0.982140
## fact.credit.duration.months18 a 24  -13.01325  553.34117  -0.024 0.981237
## fact.credit.duration.months24 a 30  -12.99020  553.34120  -0.023 0.981271
## fact.credit.duration.months30 a 36  -13.20855  553.34138  -0.024 0.980956
## fact.credit.duration.monthsmais de 36 -13.53145  553.34126  -0.024 0.980490
## previous.credit.payment.status2       1.26640    0.38521   3.288 0.001011 **
## previous.credit.payment.status3       1.82232    0.39592   4.603 4.17e-06 ***
## credit.purpose2                      -1.40585    0.50225  -2.799 0.005124 **
## credit.purpose3                      -1.54871    0.48656  -3.183 0.001458 **
## credit.purpose4                      -1.77371    0.47703  -3.718 0.000201 ***
## fact.credit.amount2500 a 5000         0.26942    0.28640   0.941 0.346851
## fact.credit.amount5000 a 10000       -0.18965    0.39044  -0.486 0.627162
## fact.credit.amountmais de 10000      -0.49981    0.64610  -0.774 0.439181
## savings2                              0.74472    0.36566   2.037 0.041687 *
## savings3                              0.62936    0.40994   1.535 0.124723
## savings4                              0.63560    0.31590   2.012 0.044217 *
## employment.duration2                  0.17722    0.26711   0.663 0.507017
## employment.duration3                  0.94378    0.35242   2.678 0.007407 **
## employment.duration4                  0.29166    0.30994   0.941 0.346705
## installment.rate2                    -0.04057    0.38122  -0.106 0.915247
## installment.rate3                    -0.47854    0.40707  -1.176 0.239771
## installment.rate4                    -0.31940    0.36480  -0.876 0.381266
## marital.status3                       0.19046    0.23938   0.796 0.426244
## marital.status4                       0.70821    0.42461   1.668 0.095337 .
## current.assets2                      -0.50747    0.30296  -1.675 0.093932 .
## current.assets3                      -0.52429    0.28215  -1.858 0.063142 .
## current.assets4                      -0.96489    0.49069  -1.966 0.049253 *
## fact.age25 a 33                       0.04576    0.31872   0.144 0.885825
## fact.age33 a 38                       0.46731    0.39003   1.198 0.230864
## fact.age38 a 45                       0.27874    0.39797   0.700 0.483669
## fact.age45 a 55                       0.08003    0.42527   0.188 0.850726
## fact.agemais de 55                    0.12234    0.51251   0.239 0.811329
## other.credits2                        0.27822    0.26942   1.033 0.301750
## apartment.type2                       0.36027    0.29051   1.240 0.214928
## apartment.type3                       0.72353    0.56634   1.278 0.201412
## foreign.worker2                       1.04471    0.77476   1.348 0.177517
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 816.41  on 666  degrees of freedom
## Residual deviance: 605.67  on 627  degrees of freedom
## AIC: 685.67
##
## Number of Fisher Scoring iterations: 14
```

```
# Realizando a predição com o modelo treinado
pred_glm_v1 <- predict(model_glm_v1, test, type="response")
```

```r
# Arredondando para 0 ou 1
pred_glm_v1 <- round(pred_glm_v1)

#Confusion Matrix da predição.
library(caret)
```

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

```r
confusionMatrix(table(data = pred_glm_v1, reference = test$credit.rating),
                positive = '1')
```

```
## Confusion Matrix and Statistics
##
##     reference
## data   0   1
##    0  41  30
##    1  58 204
##
##                Accuracy : 0.7357
##                  95% CI : (0.6849, 0.7823)
##     No Information Rate : 0.7027
##     P-Value [Acc > NIR] : 0.103103
##
##                   Kappa : 0.3113
##
##  Mcnemar's Test P-Value : 0.003999
##
##             Sensitivity : 0.8718
##             Specificity : 0.4141
##          Pos Pred Value : 0.7786
##          Neg Pred Value : 0.5775
##              Prevalence : 0.7027
##          Detection Rate : 0.6126
##    Detection Prevalence : 0.7868
##       Balanced Accuracy : 0.6430
##
##        'Positive' Class : 1
##
```

```r
# A regressão logística nos entregou um bom resultado, mas vamos verificar como esses dados se comporta

## Criando o modelo com o algoritmo Árvore de Decissão
library(C50)
modelo_tree_v1 = C5.0(formula_v1, data = train)

# Previsões nos dados de teste
pred_tree_v1 = predict(modelo_tree_v1, test, type='class')

# Confusion Matrix
confusionMatrix(test$credit.rating, pred_tree_v1, positive = '1')
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
```

```
##          0  45  54
##          1  40 194
##
##                Accuracy : 0.7177
##                  95% CI : (0.6661, 0.7654)
##     No Information Rate : 0.7447
##     P-Value [Acc > NIR] : 0.883
##
##                   Kappa : 0.2957
##
##  Mcnemar's Test P-Value : 0.180
##
##             Sensitivity : 0.7823
##             Specificity : 0.5294
##          Pos Pred Value : 0.8291
##          Neg Pred Value : 0.4545
##              Prevalence : 0.7447
##          Detection Rate : 0.5826
##    Detection Prevalence : 0.7027
##       Balanced Accuracy : 0.6558
##
##        'Positive' Class : 1
##
```

```r
# Este modelo teve um desempenho pouco pior do que o modelo de regessão
# logística

# Criando o modelo com o algoritmo SVM (Suport Vector Machine)
library(e1071)

modelo_svm_v1 <- svm(formula_v1, data = train,
                     type = 'C-classification', kernel = 'radial')

# Previsões nos dados de teste
pred_svm_v1 = predict(modelo_svm_v1, test)

# Confusion Matrix
confusionMatrix(test$credit.rating, pred_svm_v1, positive = '1')
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0  12  87
##          1   6 228
##
##                Accuracy : 0.7207
##                  95% CI : (0.6692, 0.7683)
##     No Information Rate : 0.9459
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.1251
##
##  Mcnemar's Test P-Value : <2e-16
##
```

```
##             Sensitivity : 0.7238
##             Specificity : 0.6667
##          Pos Pred Value : 0.9744
##          Neg Pred Value : 0.1212
##              Prevalence : 0.9459
##          Detection Rate : 0.6847
##    Detection Prevalence : 0.7027
##       Balanced Accuracy : 0.6952
##
##        'Positive' Class : 1
##
```

```r
# O modelo de regressão logística ainda está se saindo melhor por enquanto.

# Criando o modelo com o algoritmo Random Forest
library(rpart)
modelo_rf_v1 = rpart(formula_v1, data = train, control = rpart.control(cp = .0005))
# Previsões nos dados de teste
pred_rf_v1 = predict(modelo_rf_v1, test, type='class')

# Confusion Matrix
confusionMatrix(test$credit.rating, pred_rf_v1, positive = '1')
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0  34  65
##          1  42 192
##
##                Accuracy : 0.6787
##                  95% CI : (0.6256, 0.7285)
##     No Information Rate : 0.7718
##     P-Value [Acc > NIR] : 0.99996
##
##                   Kappa : 0.1757
##
##  Mcnemar's Test P-Value : 0.03344
##
##             Sensitivity : 0.7471
##             Specificity : 0.4474
##          Pos Pred Value : 0.8205
##          Neg Pred Value : 0.3434
##              Prevalence : 0.7718
##          Detection Rate : 0.5766
##    Detection Prevalence : 0.7027
##       Balanced Accuracy : 0.5972
##
##        'Positive' Class : 1
##
```

```r
# Esse foi o pior resultado até o momento.


# Criando o modelo com o algoritmo Naive Bayes
```

```
model_nb_v1 = naiveBayes(formula_v1, data=train)

# Previsões nos dados de teste
pred_nb_v1 <- predict(model_nb_v1, newdata=test)

# Confusion Matrix
confusionMatrix(test$credit.rating, pred_nb_v1, positive = '1')
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0  53  46
##          1  38 196
##
##                Accuracy : 0.7477
##                  95% CI : (0.6975, 0.7935)
##     No Information Rate : 0.7267
##     P-Value [Acc > NIR] : 0.213
##
##                   Kappa : 0.3819
##
##  Mcnemar's Test P-Value : 0.445
##
##             Sensitivity : 0.8099
##             Specificity : 0.5824
##          Pos Pred Value : 0.8376
##          Neg Pred Value : 0.5354
##              Prevalence : 0.7267
##          Detection Rate : 0.5886
##    Detection Prevalence : 0.7027
##       Balanced Accuracy : 0.6962
##
##        'Positive' Class : 1
##
```

```
# Este foi o melhor resultado encontrado.
```

Feature Selection.

Como podemos ver o algoritmo de Naive Bayes foi o que obteve o melhor resultado na minha primeira tentativa. Agora vamos tentar melhorar o algoritmo utilizando feature selection com o algoritmo Random Forest.

```
#Antes temos que normalizar os dados numéricos que ainda não foram utilizados.
normaliza_dados <- function(df, var){
  for(v in var)
    df[[v]] <- scale(df[[v]], center=T, scale=T)
  return(df)
}
var <- c('credit.duration.months', 'credit.amount', 'age')
df<- normaliza_dados(df, var)

# Atualizando train e test
train = subset(df, sample ==TRUE)
```

```
test = subset(df, sample==FALSE)

# Feature Selection com o Random Forest
require(randomForest)
```

```
## Loading required package: randomForest
```

```
## randomForest 4.6-14
```

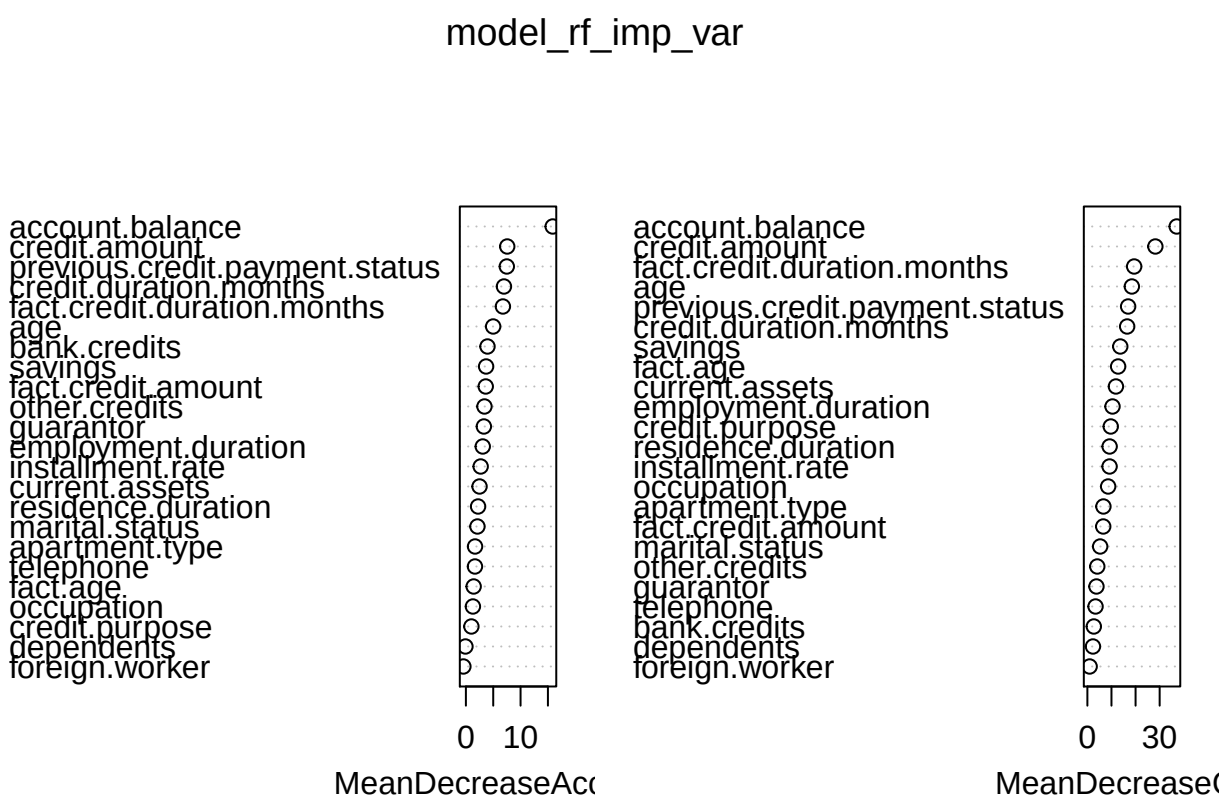
```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

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

```
model_rf_imp_var <- randomForest(credit.rating ~ ., data = df, ntree = 100,
                                 nodesize = 10, importance = TRUE)

# Plotando as variáveis por grau de importância
varImpPlot(model_rf_imp_var)
```




```
# Neste gráfico podemos ver as variáveis mais relevantes na predição do modelo.
```

Modelos Preditivos com Feature Selection.

Agora que temos uma ideia melhor de quais variáveis importam mais para nosso agloritmo iremos fazer a predição novamente e comparar os resultados.

```
# Vamos utilizar o modelo as primeiras 10 variaveis do modelo de random forest para treinar o data fram
formula_v2 <- as.formula('credit.rating ~ account.balance  +
                         previous.credit.payment.status + savings +
                         fact.credit.duration.months + credit.duration.months +
                         age + credit.amount + bank.credits + fact.credit.amount +
                         other.credits')

# Criando o modelo com o algoritmo Naive Bayes
model_nb_v2 = naiveBayes(formula_v2, data=train)

# Previsões nos dados de teste
pred_nb_v2 <- predict(model_nb_v2, newdata=test)

# Confusion Matrix
confusionMatrix(test$credit.rating, pred_nb_v2, positive = '1')
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0  45  54
##          1  38 196
##
##                Accuracy : 0.7237
##                  95% CI : (0.6723, 0.7711)
##     No Information Rate : 0.7508
##     P-Value [Acc > NIR] : 0.8848
##
##                   Kappa : 0.3064
##
##  Mcnemar's Test P-Value : 0.1179
##
##             Sensitivity : 0.7840
##             Specificity : 0.5422
##          Pos Pred Value : 0.8376
##          Neg Pred Value : 0.4545
##              Prevalence : 0.7508
##          Detection Rate : 0.5886
##    Detection Prevalence : 0.7027
##       Balanced Accuracy : 0.6631
##
##        'Positive' Class : 1
##
```

```
# O primeiro modelo está com um melhor desempenho até o momento.

#Vamos utilizar as primeiras 15 variáveis agora.
formula_v3 <- as.formula('credit.rating ~ account.balance  +
                         previous.credit.payment.status +  savings +
                         fact.credit.duration.months + credit.duration.months +
                         age + credit.amount + bank.credits + fact.credit.amount +
```

```
                        other.credits + guarantor + employment.duration +
                        installment.rate + current.assets + residence.duration')

# Criando o modelo com o algoritmo Naive Bayes
model_nb_v3 = naiveBayes(formula_v3, data=train)

# Previsões nos dados de teste
pred_nb_v3 <- predict(model_nb_v3, newdata=test)

# Confusion Matrix
confusionMatrix(test$credit.rating, pred_nb_v3, positive = '1')
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0  46  53
##          1  39 195
##
##                Accuracy : 0.7237
##                  95% CI : (0.6723, 0.7711)
##     No Information Rate : 0.7447
##     P-Value [Acc > NIR] : 0.8274
##
##                   Kappa : 0.3107
##
##  Mcnemar's Test P-Value : 0.1753
##
##             Sensitivity : 0.7863
##             Specificity : 0.5412
##          Pos Pred Value : 0.8333
##          Neg Pred Value : 0.4646
##              Prevalence : 0.7447
##          Detection Rate : 0.5856
##    Detection Prevalence : 0.7027
##       Balanced Accuracy : 0.6637
##
##        'Positive' Class : 1
##
```

```
# Obtivemos a mesma acuracia do primeiro modelo, vamos utilizar todas as varuáveis agora para ver como

formula_v4 <- as.formula('credit.rating ~ .')

# Criando o modelo com o algoritmo Naive Bayes
model_nb_v4 = naiveBayes(formula_v4, data=train)

# Previsões nos dados de teste
pred_nb_v4 <- predict(model_nb_v4, newdata=test)

# Confusion Matrix
confusionMatrix(test$credit.rating, pred_nb_v4, positive = '1')
```

```
## Confusion Matrix and Statistics
##
```

```
##           Reference
## Prediction   0   1
##          0  51  48
##          1  37 197
##
##                Accuracy : 0.7447
##                  95% CI : (0.6944, 0.7907)
##     No Information Rate : 0.7357
##     P-Value [Acc > NIR] : 0.3814
##
##                   Kappa : 0.3689
##
##  Mcnemar's Test P-Value : 0.2781
##
##             Sensitivity : 0.8041
##             Specificity : 0.5795
##          Pos Pred Value : 0.8419
##          Neg Pred Value : 0.5152
##              Prevalence : 0.7357
##          Detection Rate : 0.5916
##    Detection Prevalence : 0.7027
##       Balanced Accuracy : 0.6918
##
##        'Positive' Class : 1
##
```

```
# O modelo_v2 de Naive Bayes, com acuracia de 0,7477, foi o que se saiu melhor até o momento, portanto
```

Balanceamento do Data Frame

Agora iremos balancear nosso data frame de modo que tenhamos aproximadamente 50% de variáveis 1 e 50% de variáveis não. Isso pode ajudar a melhorar o desempenho do algoritmo. Para medir a melhora iremos utilizar a curva ROC.

```
# Antes de fazer o balanceamento precisamos transformar as variáveis que foram normalizadqos para o tip
train$credit.duration.months <- as.numeric(train$credit.duration.months)
train$credit.amount  <- as.numeric(train$credit.amount )
train$age <- as.numeric(train$age )
test$credit.duration.months <- as.numeric(test$credit.duration.months)
test$credit.amount  <- as.numeric(test$credit.amount )
test$age <- as.numeric(test$age )


library(ROSE)
```

```
## Loaded ROSE 0.0-3
```

```
# Balanceando os dados
# ROSE nos dados de treino
rose_train <- ROSE(credit.rating ~ ., data = train, seed = 1)$data
prop.table(table(rose_train$credit.rating))
```

```
##
##         1         0
## 0.5172414 0.4827586
```

```
# ROSE nos dados de teste
rose_test <- ROSE(credit.rating ~ ., data = test, seed = 1)$data
prop.table(table(rose_test$credit.rating))
```

```
##
##         1         0
## 0.5345345 0.4654655
```

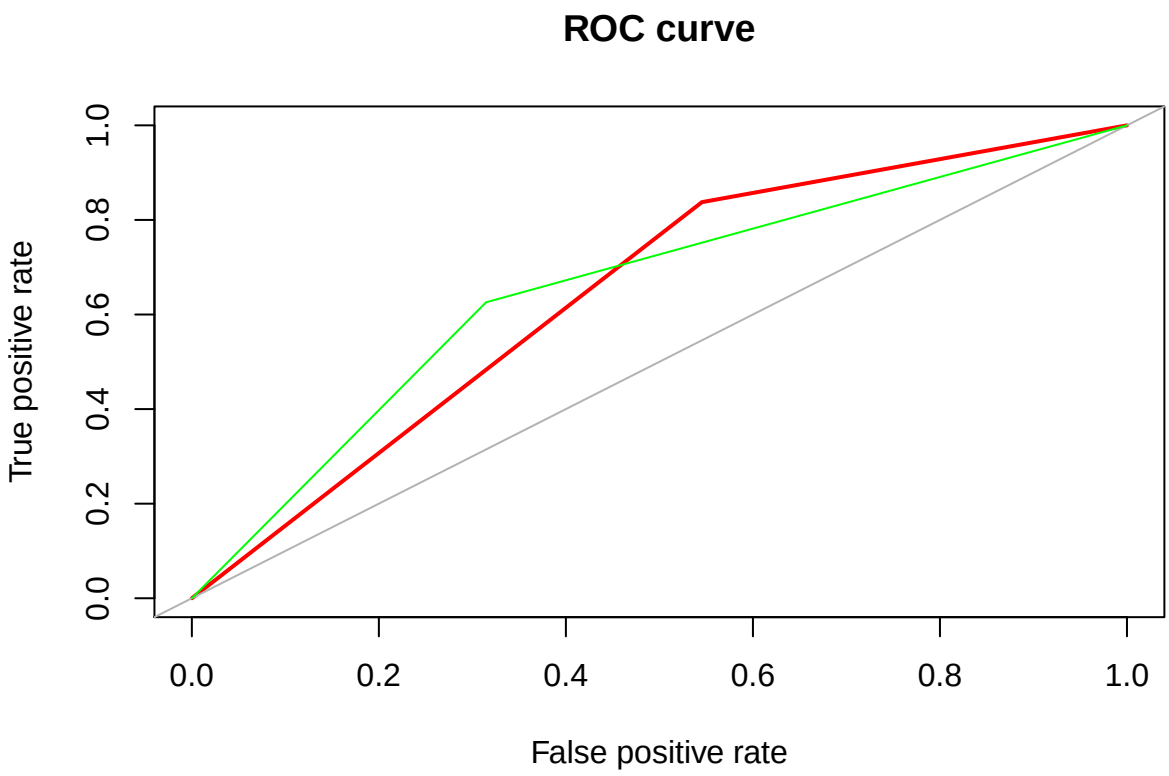
```
# Criando o modelo com o algoritmo Naive Bayes
model_nb_v5 = naiveBayes(formula_v3, data=rose_train)

# Previsões nos dados de teste
pred_nb_v5 <- predict(model_nb_v5, newdata=rose_test)

# Confusion Matrix
confusionMatrix(rose_test$credit.rating, pred_nb_v5, positive = '1')
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   1   0
##          1 122  56
##          0  58  97
##
##                Accuracy : 0.6577
##                  95% CI : (0.604, 0.7085)
##     No Information Rate : 0.5405
##     P-Value [Acc > NIR] : 9.423e-06
##
##                   Kappa : 0.3115
##
##  Mcnemar's Test P-Value : 0.9254
##
##             Sensitivity : 0.6778
##             Specificity : 0.6340
##          Pos Pred Value : 0.6854
##          Neg Pred Value : 0.6258
##              Prevalence : 0.5405
##          Detection Rate : 0.3664
##    Detection Prevalence : 0.5345
##       Balanced Accuracy : 0.6559
##
##        'Positive' Class : 1
##
```

```
# Curva roc para o modelo_v3 (balanceado)
roc.curve(test$credit.rating, pred_nb_v2, plotit = T, col = "red")
```

```
## Area under the curve (AUC): 0.646
```

```
# Curva roc para o modelo_v5 (desbalanceado)
roc.curve(rose_test$credit.rating, pred_nb_v5, plotit = T,
          col = "green", add.roc = T)
```




```
## Area under the curve (AUC): 0.656
```

Conclusão

Podemo ver que curva ROC para o modelo_v5 começa melhor do que o modelo_v2, mas depois de um certo ponto ela piora ficando mais próximo da diagonal central, portanto o modelo_v2 foi o que teve o melhor desempenho nesta análise.

Fim

Fernando Tsutomu Hara