Estudo de Caso - Czech Bank

Trabalho Final da disciplina ‘Decisões Empresariais e Raciocínio Analítico’
Prof. Gustavo Mirapalheta

MBA Big Data e Business Analytics, FGV-MMURAD
Vitória, 10 de Outubro de 2022

Grupo:
Iago de Carvalho Nunes
Felipe Guimarães Freitas
Fidelis José Coimbra Junior
Marcone Martins Negreiros
Suellen Brandenburg Soares

1 Introdução

Com base no documento descritor do exercício (PKDD’99 Discovery Challenge - Guide to the Financial Data Set), sabe-se que a administração deste hipotético banco na República Tcheca quer, ao mesmo tempo, conhecer melhor seus clientes e tomar ações para melhorar seus serviços. Sabendo da existência de contas bancárias com e sem empréstimo, propõe-se aumentar o volume de empréstimo do banco, oferecendo essa modalidade de crédito para contas sem empréstimo, mas que têm uma maior probabilidade de serem de bons pagadores.

Com uma variável de resposta binária (bons e maus pagadores de empréstimo), o Modelo de Regressão Logística para caso binário é capaz de fornecer esse tipo de informação. Em suma, treina-se o modelo a indicar a probabilidade de uma observação estar em um dos dois grupos de uma variável de resposta. A estimação do modelo é realizada por meio da maximização da função de log-verossimilhança com o suporte de algoritmos de otimização não linear (para mais detalhes, ver Hosmer e Lemeshow [2000]), e com base em um conjunto X = (X1, … , Xk) de variáveis relevantes ao problema e que são pré-selecionadas pelo analista. Assim, o modelo identifica padrões existentes em diversas variáveis e infere a probabilidade de ser bom pagador. As métricas de

desempenho e o limiar γ foram obtidas por meio de k-fold cross-validation com k=10. Após a análise da matriz de confusão da amostra de treino, utiliza-se o modelo para prever a classificação das observações na amostra de teste, analisando a matriz de confusão do modelo de validação (teste).

Ao final deste exercício, o banco terá uma noção exata do perfil (nas variáveis selecionadas) dos bons e maus pagadores. A depender da performance do modelo, o banco poderá traçar ações individualmente segmentadas para os potenciais tomadores de empréstimo, como, por exemplo, oferecer empréstimo ou recusar um pedido de empréstimo; ou, ainda, poderá traçar ações voltadas para a melhoria de seus bancos de dados, como, por exemplo, aumentar a quantidade de clientes e de variáveis coletadas deles.

Para a simplificação do exercício, as variáveis independentes do modelo foram escolhidas de forma arbitrária. São elas:

Variáveis contínuas:
1. Idade do cliente dono da conta, na data de concessão do empréstimo (em dias);
2. Balanço médio da conta até a data de concessão do empréstimo;
3. Balanço médio de todas as contas no distrito;
4. Proporção de habitantes urbanos dos distritos;
5. Salário médio dos distritos;

Variáveis categóricas:
6. Gênero do cliente dono da conta;
7. Tipos de cartões de crédito vinculados a conta.

Para construir uma base de dados capaz de dar suporte para esse exercício, realiza-se a análise exploratória e os procedimentos de Extract, Transform, Load (ETL) para variáveis de interesse em sete das oito bases disponíveis: account.asc, loan.asc, trans.asc, client.asc, disp.asc, card.asc e district.asc.

2 Análise exploratória e ETL

Inicia-se a sessão no RStudio selecionando a pasta de trabalho onde se encontra os dados e para onde os resultados serão salvos, através do comando `setwd()`.

É desativada a notação científica na saída de dados do console:

```
options(scipen=999)
```

O conjunto de pacotes centrais do “tidyverse”, como o “dplyr” e “tidyr”, são carregados conjuntamente na sessão:

```
library(tidyverse)
```

Sabe-se que:
As bases de dados possuem cabeçalhos (`header=TRUE`);
O simbolo separador de valores é o ponto e vírgula (“;”);
O separador decimal é o ponto (“.”);
E os valores faltantes (NAs) em strings não estão codificados, estão vazios (““).

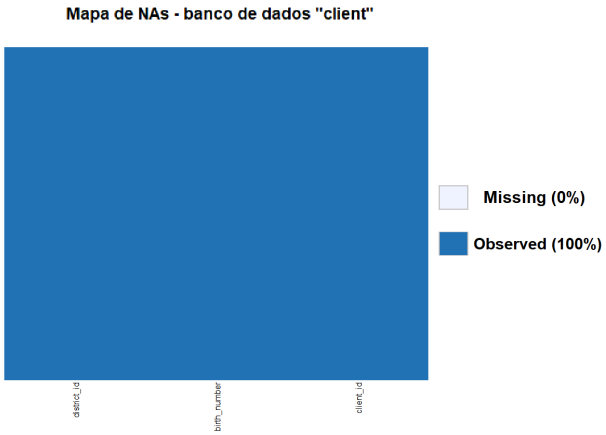
Com essas informações, escolhe-se o comando e os parâmetros corretos para a importação dos dados:

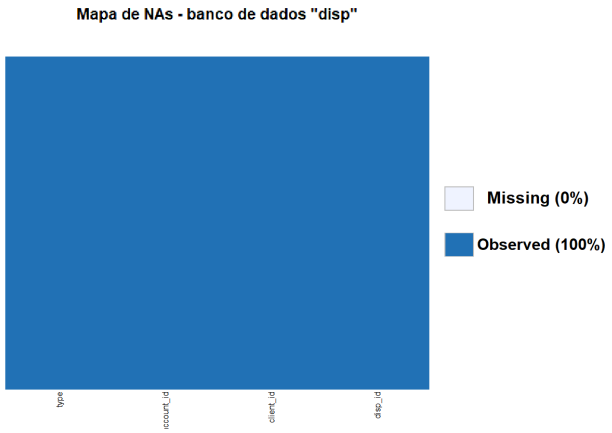
```
conta <- read.csv('account.asc', header = TRUE, sep = ";", na.strings = "")
emprest <- read.csv('loan.asc', header = TRUE, sep = ";", na.strings = "")
transacoes <- read.csv('trans.asc', header = TRUE, sep = ";", na.strings = "")
cliente <- read.csv('client.asc', header = TRUE, sep = ";", na.strings = "")
conector <- read.csv('disp.asc', header = TRUE, sep = ";", na.strings = "")
cartao <- read.csv('card.asc', header = TRUE, sep = ";", na.strings = "")
distrito <- read.csv('district.asc', header = TRUE, sep = ";", na.strings = "")
```

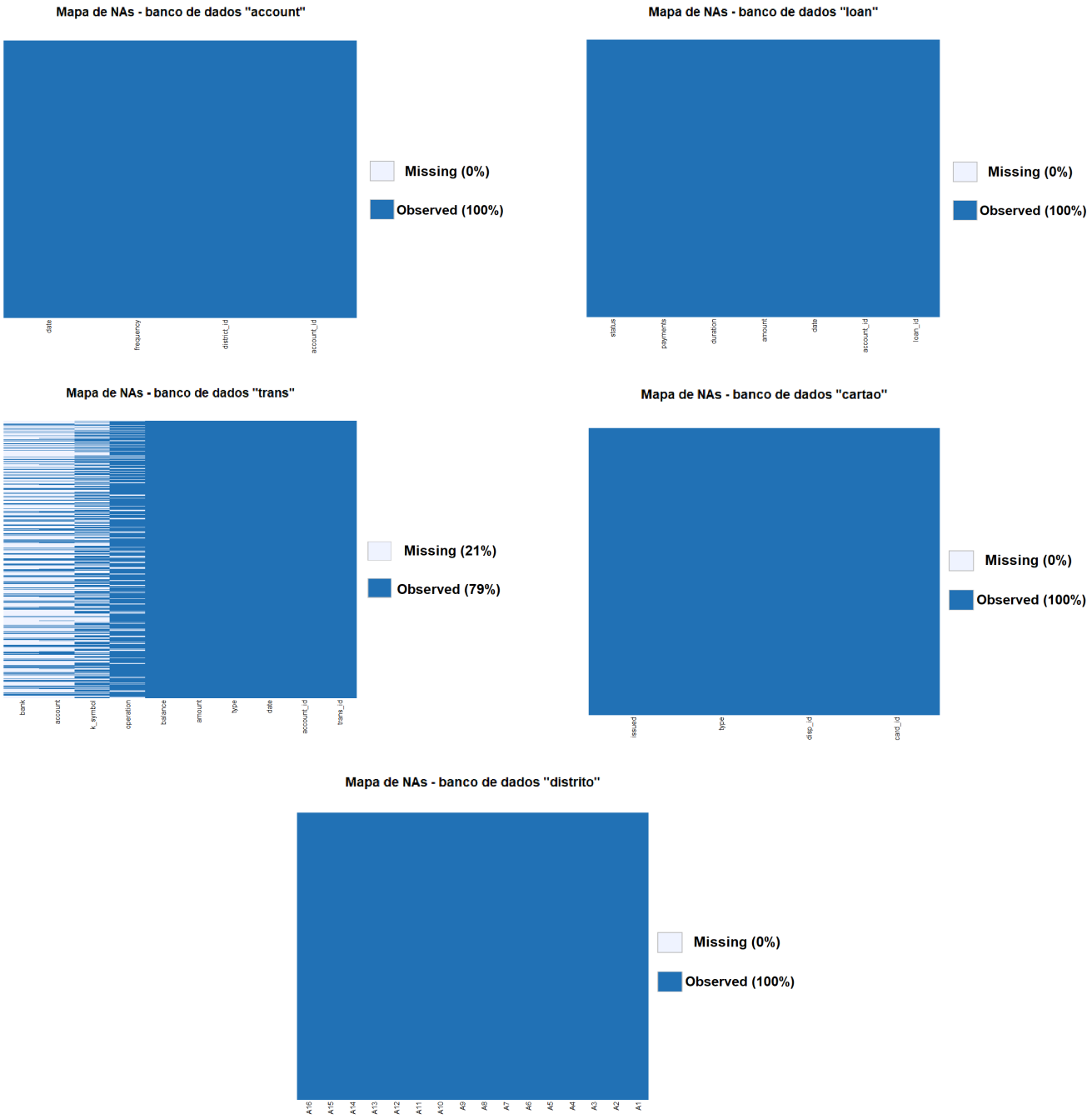
2.1 Mapeando NAs

Damos início a análise exploratória das variáveis. Primeiro, mapeamos os valores faltantes (NAs) nas bases, através da função `missmap()` do pacote `Amelia` :

```
install.packages("Amelia")
library(Amelia)
missmap(conta, main='Mapa de NAs - banco de dados "account"', x.cex = 1.0, y.labels= NULL, y.at
= NULL, margins = c(10, 10))
missmap(emprest, main='Mapa de NAs - banco de dados "loan"', x.cex = 1.0, y.labels= NULL, y.at =
NULL, margins = c(10, 10))
missmap(transacoes, main='Mapa de NAs - banco de dados "trans"', x.cex = 1.0, y.labels= NULL, y.
at = NULL, margins = c(10, 10))
missmap(cliente, main='Mapa de NAs - banco de dados "client"', x.cex = 1.0, y.labels= NULL, y.at
= NULL, margins = c(10, 10))
missmap(conector, main='Mapa de NAs - banco de dados "disp"', x.cex = 1.0, y.labels= NULL, y.at
= NULL, margins = c(10, 10))
missmap(cartao, main='Mapa de NAs - banco de dados "cartao"', x.cex = 1.0, y.labels= NULL, y.at
= NULL, margins = c(10, 10))
missmap(distrito, main='Mapa de NAs - banco de dados "distrito"', x.cex = 1.0, y.labels= NULL,
y.at = NULL, margins = c(10, 10))
```







Sabemos, então, que não existem NAs nas variáveis de interesse do exercício. Contudo, ao unir dados com observações (linhas) que representam registros de informações distintas, NAs podem surgir - mas esse tipo de NA carrega a informação de que aquela observação não possui atributo na variável integrada.

2.2 Análise do balanceamento

Nesse sentido, unimos as bases de contas e empréstimos para analisar o balanceamento das amostras:

```
# selecionando variáveis
conta2X <- conta %>%
  select(account_id, district_id)
emprest2X <- emprest %>%
  select(account_id, loan_id, status, emprest_date)
# unindo bancos
conta_loan <- conta2X %>%
  left_join(emprest2X, by = "account_id")
# limpando do ambiente de trabalho os objetos que não precisamos mais
rm(conta2X, emprest2X, emprest)
```

Contas com NAs nas variáveis (colunas) `loan_id` e `status` são contas sem empréstimo. Transformamos a variável do status do empréstimo em uma coluna binária onde:
1. Contratos de empréstimo finalizados ou em andamento, sem atraso nos pagamentos, são equalizados a 1;
2. Contratos de empréstimo finalizados ou em andamento, com atraso nos pagamentos, são equalizados a 0.

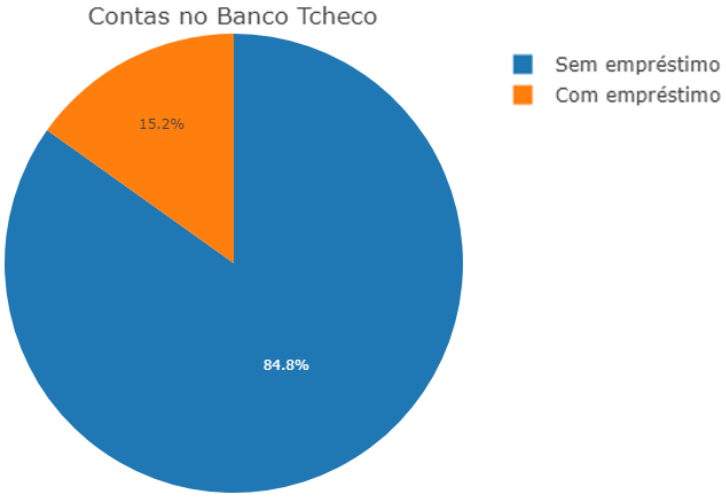
Pedimos um sumário da variável `status` com o comando `describe()` do pacote `Hmisc` para avaliar o balanço:

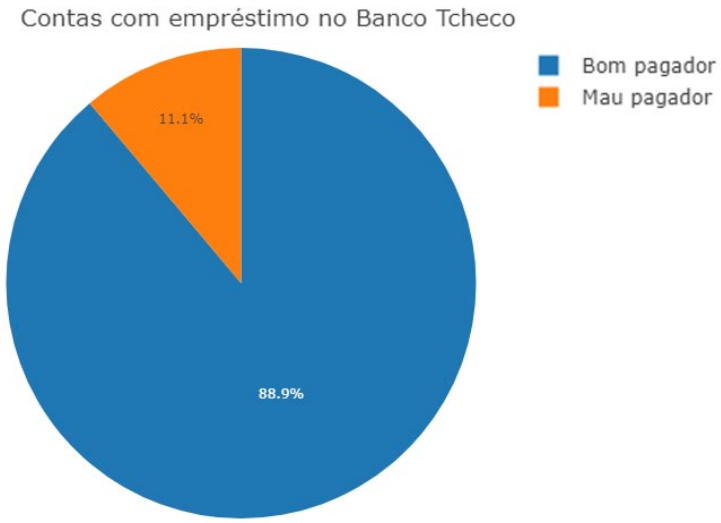
```
conta_loan$status <- ifelse(conta_loan$status=="A"|conta_loan$status=="C", 1, 0)
install.packages("Hmisc")
library(Hmisc)
describe(as.factor(conta_loan$status))
#as.factor(conta_loan$status)
#       n  missing distinct
#     682     3818        2
#
#Value          0     1
#Frequency     76   606
#Proportion 0.111 0.889
```

Apesar da ocorrência de maus pagadores ser um evento raro no nosso banco de dados, o balanceamento entre grupos ainda está dentro do aceitável para a aplicação de um modelo de regressão logística. Para uma melhor visualização do balanceamento, são adotados, ainda, os seguintes procedimentos:
Criamos uma coluna onde:
1. NAs em `loan_id` são transformados em 0;
2. Chaves de identificação dos empréstimos são equalizados a 1.

Um novo objeto é criado no ambiente de trabalho, onde contas sem contratos são suprimidas. Criamos também duas colunas com rótulos para as duas colunas binárias, e, por fim, renderiza-se dois gráficos de pizza com o pacote `plotly` para a análise do balanceamento:

```
conta_loan$emprestimo <- ifelse(is.na(conta_loan$loan_id), 0, 1)
conta_loan$emprestimoX <- ifelse(conta_loan$emprestimo==0, "Sem empréstimo", "Com empréstimo")
conta_loan$statusX <- ifelse(conta_loan$status==1, "Bom pagador", "Mau pagador")
conta_loanX <- subset(conta_loan, !is.na(status))
install.packages("plotly")
library(plotly)
fig <- plot_ly(conta_loan, labels = ~emprestimoX, values = ~length(emprestimo), type = 'pie')
fig <- fig %>% layout(title = 'Contas no Banco Tcheco',
                      xaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE),
                      yaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE))
fig2 <- plot_ly(conta_loanX, labels = ~statusX, values = ~length(status), type = 'pie')
fig2 <- fig2 %>% layout(title = 'Contas com empréstimo no Banco Tcheco',
                      xaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE),
                      yaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE))
fig;fig2
```





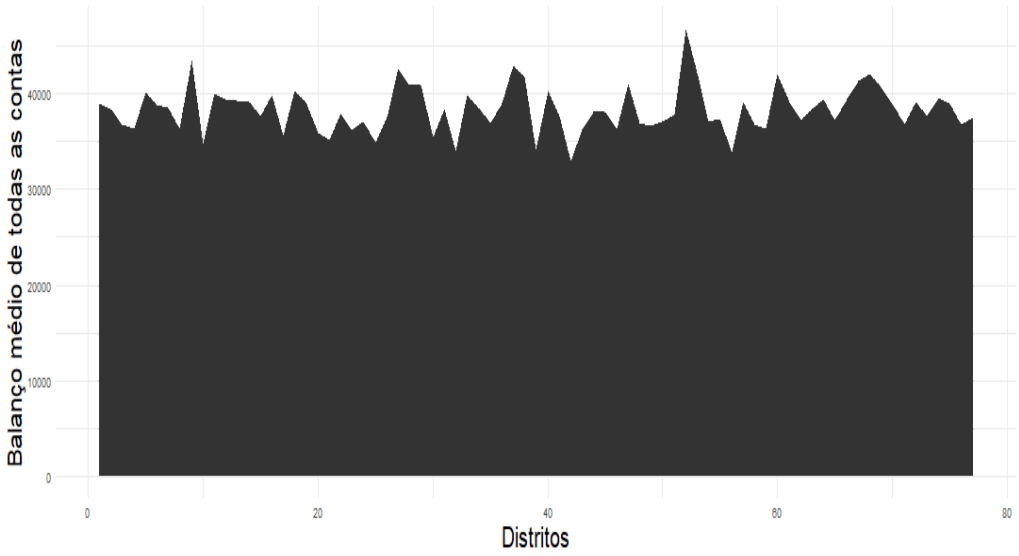
A visualização em gráfico do balanceamento confirma que tomar empréstimo é um evento raro neste banco tcheco - e mais raro ainda é a ocorrência de maus pagadores. Essa análise reforça a constatação de que o banco hipotético possui uma margem para aumentar seu volume de clientes tomadores de empréstimo.

2.3 Análise do balanço médio de todas as contas nos distritos

Em sequência, criamos a nossa primeira variável - balanço médio de todas as contas nos distritos:

```
transacoes1 <- transacoes %>%
  select(account_id, balance)
## unindo dados
trans_acc <- transacoes1 %>%
  left_join(conta, by = "account_id")
## sumarizando um balanço médio para cada distrito
trans_acc1 <- trans_acc %>%
  select(district_id, balance) %>%
  group_by(district_id) %>%
  summarise(balance_distr=mean(balance, na.rm=T)) %>%
  ungroup()
## limpando ambiente de trabalho
rm(transacoes1, trans_acc, conta)

ggplot(trans_acc1) +
  aes(x = district_id, y = balance_distr) +
  geom_area(size = 1.5) +
  labs(
    x = "Distritos",
    y = "Balanço médio de todas as contas"
  ) +
  theme_minimal() +
  theme(
    axis.title.y = element_text(size = 20L),
    axis.title.x = element_text(size = 20L)
  )
```



O gráfico indica pouca variação no balanço médio de todas as contas, com a maioria dos distritos com suas médias concentradas entre 35.000 e 40.000.

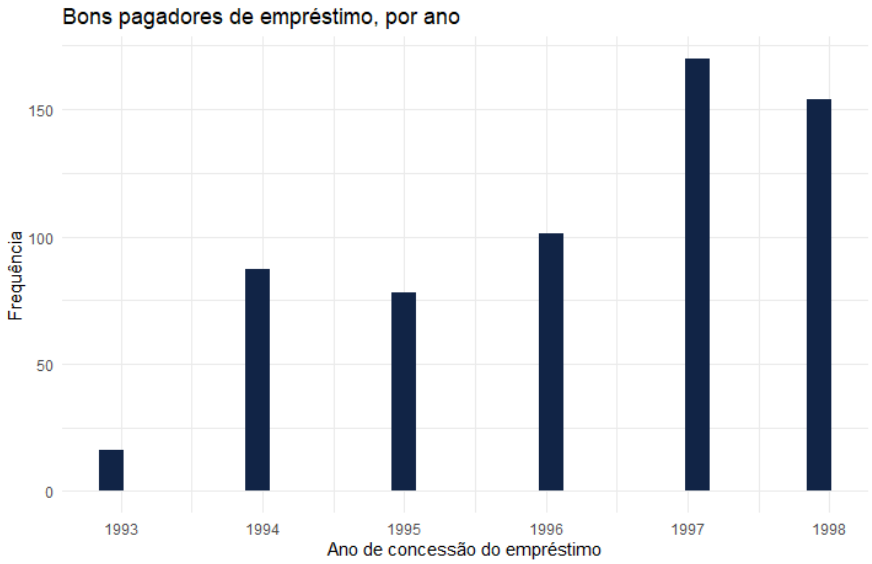
2.4 Histograma da data de concessão do empréstimo

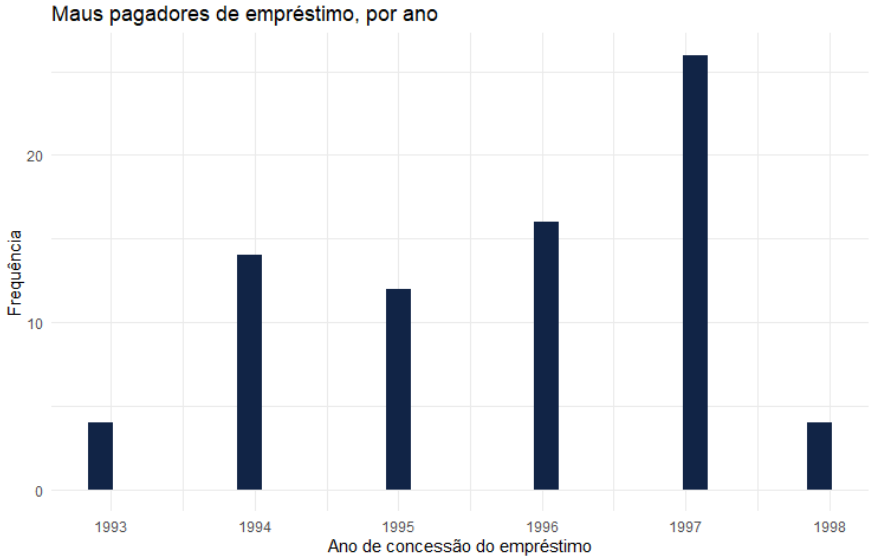
Prosseguindo com a análise exploratória, busca-se compreender o período de concessão dos empréstimos. As observações de contas sem empréstimo são removidas da base e é atribuído o formato de data para a nova coluna da data do empréstimo:

```
conta_loan <- subset(conta_loan, !is.na(loan_id))
conta_loan$emprest_date <- parse_date(as.character(conta_loan$date), format="%y%m%d")
```

Cria-se dois objetos: uma base para bons e outra para maus pagadores. Extraímos apenas o ano de concessão do empréstimo e renderizamos os gráficos com `ggplot2` :

```
conta_loanX <- subset(conta_loan, status==1)
conta_loanY <- subset(conta_loan, status==0)
conta_loanX <- conta_loanX %>%
  mutate(ano = substr(emprest_date, 1, 4),
         ano = as.integer(ano))
conta_loanY <- conta_loanY %>%
  mutate(ano = substr(emprest_date, 1, 4),
         ano = as.integer(ano))
ggplot(conta_loanX) +
  aes(x = ano) +
  geom_histogram(bins = 30L, fill = "#112446") +
  labs(
    x = "Ano de concessão do empréstimo",
    y = "Frequência",
    title = "Bons pagadores de empréstimo, por ano"
  ) +
  theme_minimal()
ggplot(conta_loanY) +
  aes(x = ano) +
  geom_histogram(bins = 30L, fill = "#112446") +
  labs(
    x = "Ano de concessão do empréstimo",
    y = "Frequência",
    title = "Maus pagadores de empréstimo, por ano"
  ) +
  theme_minimal()
```





Observa-se que o ano de maior concessão de empréstimos foi 1997, e que a proporção entre bons e maus pagadores se matem estável até 1998, ano em que a ocorrência de maus pagadores se reduz. Por se tratar da data mais recente, é razoável pensar que, se não houve mudanças estruturais, a proporção volte ao patamar dos anos anteriores com o passar dos anos - ou a tendência de alta pode se confirmar, caso o ano de 1998 marque uma mudança estrutural, como uma crise.

2.5 Boxplot do balanço médio dos bons e maus pagadores

Para criar a variável de balanço médio entre bons e maus pagadores é preciso calcular o balanço até a data do empréstimo, para que a variável não seja influenciada pelo empréstimo em si. Para isso, unimos a base de transações com a base `conta_loan`. Removemos as transações de contas sem empréstimo, e dividimos a base por dois critérios: ano de concessão do empréstimo e transações até a data de concessão:

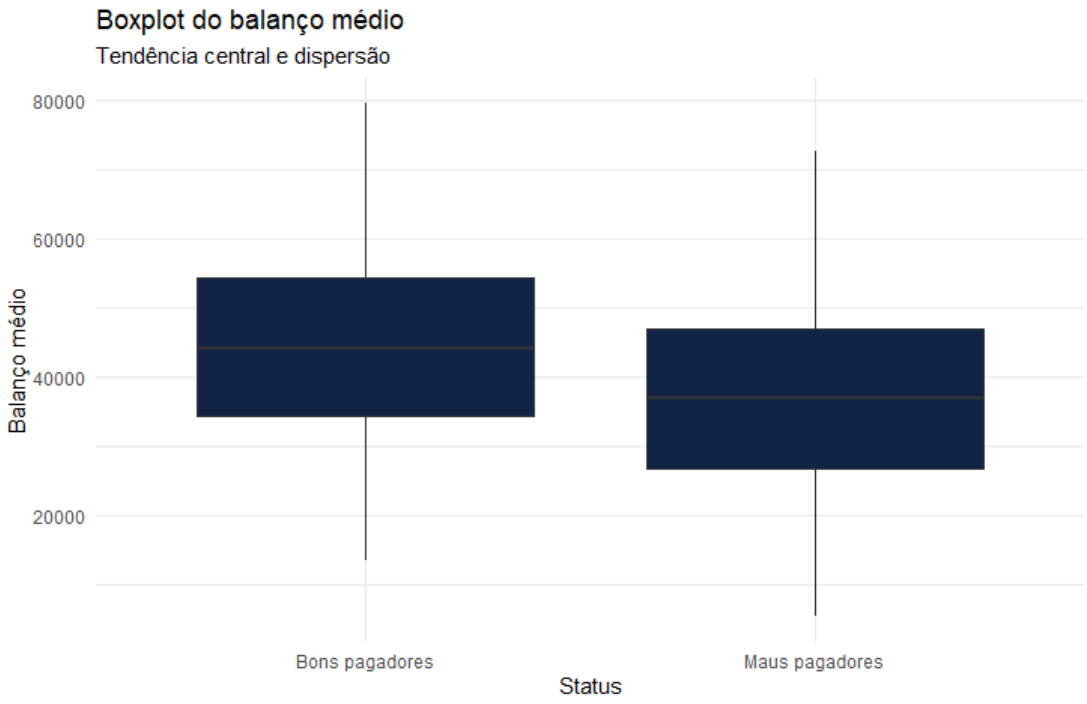
```
trans_loan <- transacoes %>%
  left_join(conta_loan, by = "account_id")
trans_loan <- subset(trans_loan, !is.na(loan_id))
trans_loan$trans_date <- parse_date(as.character(trans_loan$date), format="%y%m%d")
trans_loan93 <- subset(trans_loan, emprest_date<=as.Date("1993-12-31"))
trans_loan93 <- subset(trans_loan93, trans_date<emprest_date)
trans_loan94 <- subset(trans_loan, emprest_date >= as.Date("1994-01-01") & emprest_date <=  as.D
ate("1994-12-31"))
trans_loan94 <- subset(trans_loan94, trans_date<emprest_date)
trans_loan95 <- subset(trans_loan, emprest_date >= as.Date("1995-01-01") & emprest_date <=  as.D
ate("1995-12-31"))
trans_loan95 <- subset(trans_loan95, trans_date<emprest_date)
trans_loan96 <- subset(trans_loan, emprest_date >= as.Date("1996-01-01") & emprest_date <=  as.D
ate("1996-12-31"))
trans_loan96 <- subset(trans_loan96, trans_date<emprest_date)
trans_loan97 <- subset(trans_loan, emprest_date >= as.Date("1997-01-01") & emprest_date <=  as.D
ate("1997-12-31"))
trans_loan97 <- subset(trans_loan97, trans_date<emprest_date)
trans_loan98 <- subset(trans_loan, emprest_date >= as.Date("1998-01-01") & emprest_date <=  as.D
ate("1998-12-31"))
```

Reduzimos o comprimento (vertical) dos dados para termos somente uma observação para cada conta, através da sumarização do balanço médio até o ano do empréstimo, por conta. Depois, reintegramos os dados em um único objeto:

```
trans_loan93 <- trans_loan93 %>%
  select(account_id, balance, status, emprest_date) %>% # selecionando apenas as variáveis de in
teresse
  group_by(account_id, status, emprest_date) %>% # agrupando o cálculo por conta e status
  summarise(balance=mean(balance, na.rm=T)) %>% # sumarizando o balanço médio
  ungroup()
trans_loan94 <- trans_loan94 %>%
  select(account_id, balance, status, emprest_date) %>% # selecionando apenas as variáveis de in
teresse
  group_by(account_id, status, emprest_date) %>% # agrupando o cálculo por conta e status
  summarise(balance=mean(balance, na.rm=T)) %>% # sumarizando o balanço médio
  ungroup()
trans_loan95 <- trans_loan95 %>%
  select(account_id, balance, status, emprest_date) %>% # selecionando apenas as variáveis de in
teresse
  group_by(account_id, status, emprest_date) %>% # agrupando o cálculo por conta e status
  summarise(balance=mean(balance, na.rm=T)) %>% # sumarizando o balanço médio
  ungroup()
trans_loan96 <- trans_loan96 %>%
  select(account_id, balance, status, emprest_date) %>% # selecionando apenas as variáveis de in
teresse
  group_by(account_id, status, emprest_date) %>% # agrupando o cálculo por conta e status
  summarise(balance=mean(balance, na.rm=T)) %>% # sumarizando o balanço médio
  ungroup()
trans_loan97 <- trans_loan97 %>%
  select(account_id, balance, status, emprest_date) %>% # selecionando apenas as variáveis de in
teresse
  group_by(account_id, status, emprest_date) %>% # agrupando o cálculo por conta e status
  summarise(balance=mean(balance, na.rm=T)) %>% # sumarizando o balanço médio
  ungroup()
trans_loan98 <- trans_loan98 %>%
  select(account_id, balance, status, emprest_date) %>% # selecionando apenas as variáveis de in
teresse
  group_by(account_id, status, emprest_date) %>% # agrupando o cálculo por conta e status
  summarise(balance=mean(balance, na.rm=T)) %>% # sumarizando o balanço médio
  ungroup()
# unindo bases
transloan <- rbind(trans_loan93,trans_loan94,trans_loan95,trans_loan96,trans_loan97,trans_loan9
8)
# limpando ambiente de trabalho
rm(trans_loan,trans_loan93,trans_loan94,trans_loan95,trans_loan96,trans_loan97,trans_loan98)
```

Agora renderizamos um boxplot com `ggplot2` :

```
ggplot(transloan) +
  aes(x = status, y = balance) +
  geom_boxplot(fill = "#112446") +
  labs(
    x = "Status",
    y = "Balanço médio",
    title = "Boxplot do balanço médio",
    subtitle = "Tendência central e dispersão"
  ) +
  theme_minimal()
```



O boxplot indica uma tendência de bons pagadores possuírem balanços médios de transações em contas superiores aos maus pagadores. Realizamos um teste de médias para verificar se a tendência é estatisticamente relevante:

```
transloanX <- transloan %>%
  mutate(status=ifelse(status=="A"|status=="C", 1, 0),
         status=as.factor(status))
var.test(transloanX$balance ~ transloanX$status, alternative="two.sided")
#   F test to compare two variances
#
#data:  transloanX$balance by transloanX$status
#F = 1.0921, num df = 75, denom df = 605, p-value = 0.5762
#alternative hypothesis: true ratio of variances is not equal to 1
#95 percent confidence interval:
# 0.7941768 1.5720528
#sample estimates:
#ratio of variances
#          1.092053

t.test(transloan$balance ~ transloan$status, alternative = "two.sided", var.equal= T)
#   Welch Two Sample t-test
#
#data:  transloan$balance by transloan$status
#t = -4.4918, df = 680, p-value = 0.000008295
#alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
#95 percent confidence interval:
# -10484.822  -4106.653
#sample estimates:
#mean in group 0 mean in group 1
#       37326.56        44622.30
rm(transloanX)
```

Apesar da variância igual, o p-valor confirma a diferença entre médias.

2.6 Histograma da idade e proporção entre gêneros

Para a inclusão das variáveis da idade dos clientes na data de concessão do empréstimo e do gênero, seleciona-se apenas características dos donos das contas na base `conector`, e prosseguimos com o tratamento da variável de data de nascimento integrada a codificação do gênero:

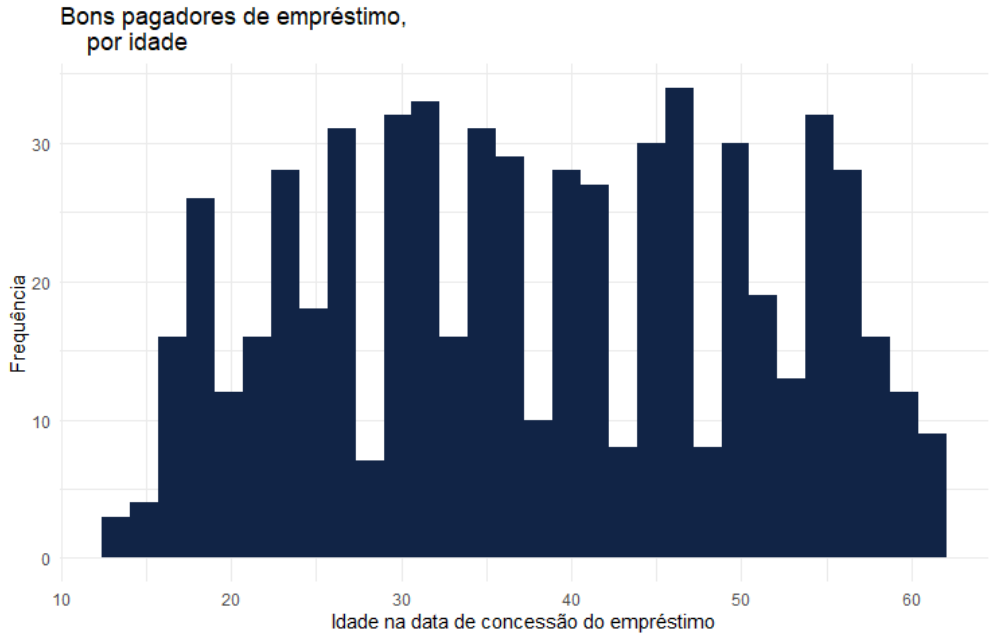
```
conector <- subset(conector, type=="OWNER")
cliente <- cliente %>%
  select(client_id, birth_number)
cliente$ano <- substr(cliente$birth_number, 1, 2)
cliente$ano <- paste(19, cliente$ano, sep = "")
cliente$ano <- as.numeric(cliente$ano)
cliente$mes <- as.numeric(substr(cliente$birth_number, 3, 4))
cliente$gnr <- ifelse(cliente$mes>50, 1, 0)  # 1 = Feminino // 0 = Masculino
cliente$mes <- ifelse(cliente$mes>50, cliente$mes - 50, cliente$mes)
cliente$dia <- as.numeric(substr(cliente$birth_number, 5, 6))
cliente$data_nasc <- as.Date(paste0(cliente$ano, "-", cliente$mes, "-", cliente$dia), format =
"%Y-%m-%d")
cliente$gnr <- as.factor(cliente$gnr)
cliente <- cliente %>%
  select(client_id, data_nasc, gnr)
conector <- conector %>%
  select(disp_id, account_id, client_id)
conect_acc <- conector %>%
  left_join(cliente, by = "client_id")
rm(cliente, conector)
dados_fn <- transloan %>%
  left_join(conect_acc, by = "account_id")
dados_fn <- dados_fn  %>%
  mutate(idade=as.numeric(emprest_date - data_nasc))
```

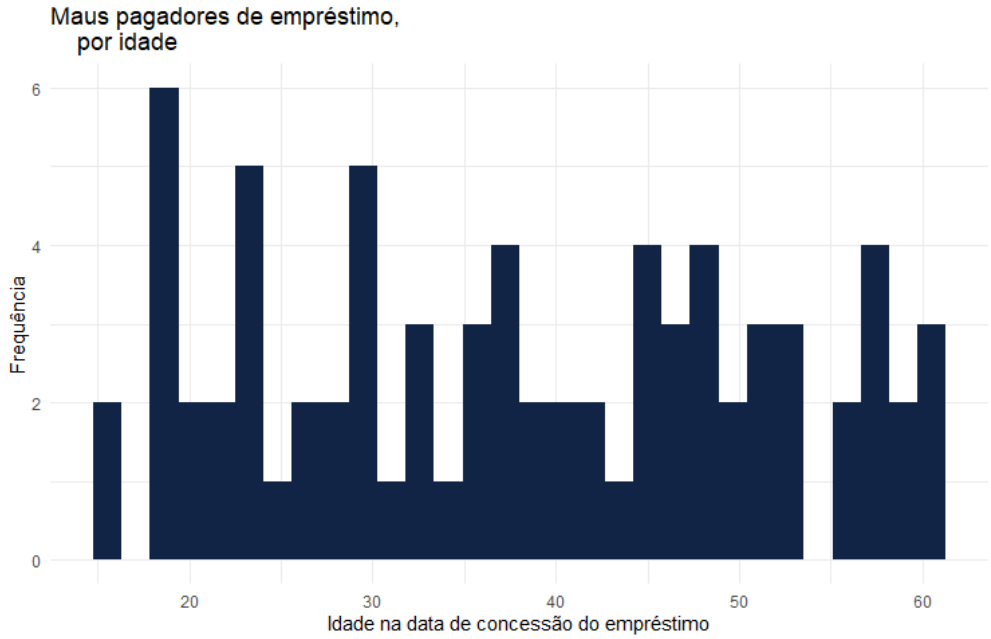
Após o tratamento, renderiza-se um histograma da distribuição da idade dos clientes bons e maus pagadores:

```
dados_fnX <- subset(dados_fn, status==1)
dados_fnY <- subset(dados_fn, status==0)

ggplot(dados_fnX) +
  aes(x = idade) +
  geom_histogram(bins = 30L, fill = "#112446") +
  labs(
    x = "Idade na data de concessão do empréstimo",
    y = "Frequência",
    title = "Bons pagadores de empréstimo,
    por idade"
  ) +
  theme_minimal()

ggplot(dados_fnY) +
  aes(x = idade) +
  geom_histogram(bins = 30L, fill = "#112446") +
  labs(
    x = "Idade na data de concessão do empréstimo",
    y = "Frequência",
    title = "Maus pagadores de empréstimo,
    por idade"
  ) +
  theme_minimal()
```

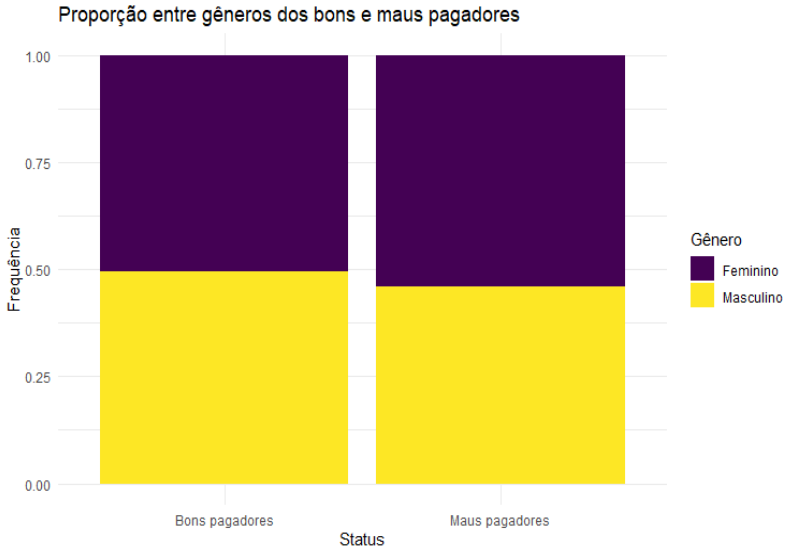




Observa-se que, apesar de bem distribuídos, os maiores picos de frequência entre bons pagadores ocorrem acima da faixa dos 30 anos, enquanto que os maiores picos de maus pagadores ocorre abaixo da mesma faixa etária.

Seguimos a análise renderizando um gráfico com a proporção entre gêneros dos bons e maus pagadores:

```
ggplot(dados_fn) +
  aes(x = status1, fill = genero) +
  geom_bar(position = "fill") +
  scale_fill_viridis_d(option = "viridis", direction = 1) +
  labs(
    x = "Status",
    y = "Frequência",
    title = "Proporção entre gêneros dos bons e maus pagadores",
    fill = "Gênero"
  ) +
  theme_minimal()
rm(dados_fnX, dados_fnY, dados_fn)
```



O gráfico indica uma maior proporção de homens no grupo de bons pagadores, e de mulheres no grupo de maus pagadores - o que levanta outras questões, como o salário médio entre homens e mulheres no país e como isso influencia a relação observada no gráfico.

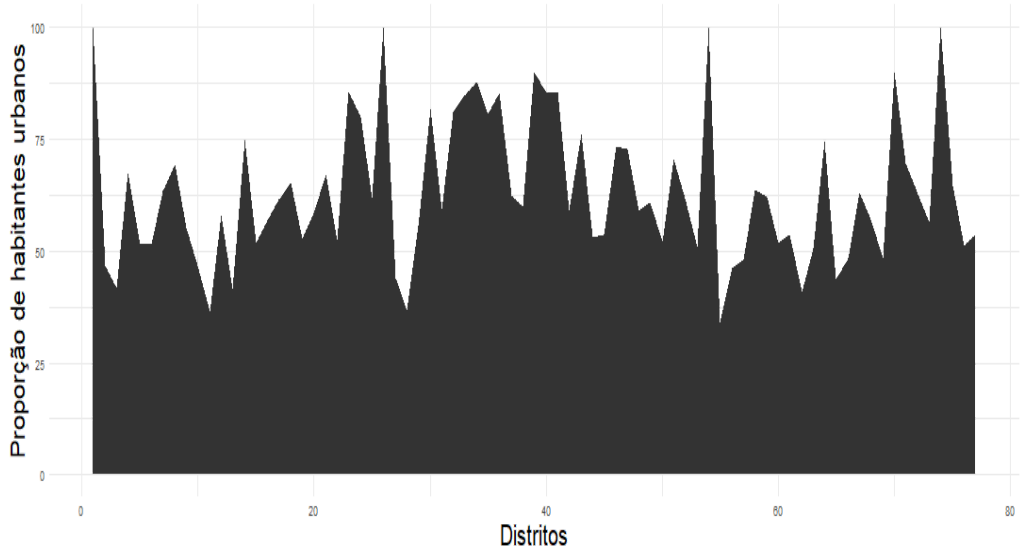
2.7 Variabilidade das características dos distritos

Seleciona-se as variáveis de interesse nos dados sobre distritos e prosseguimos com a análise dos gráficos:

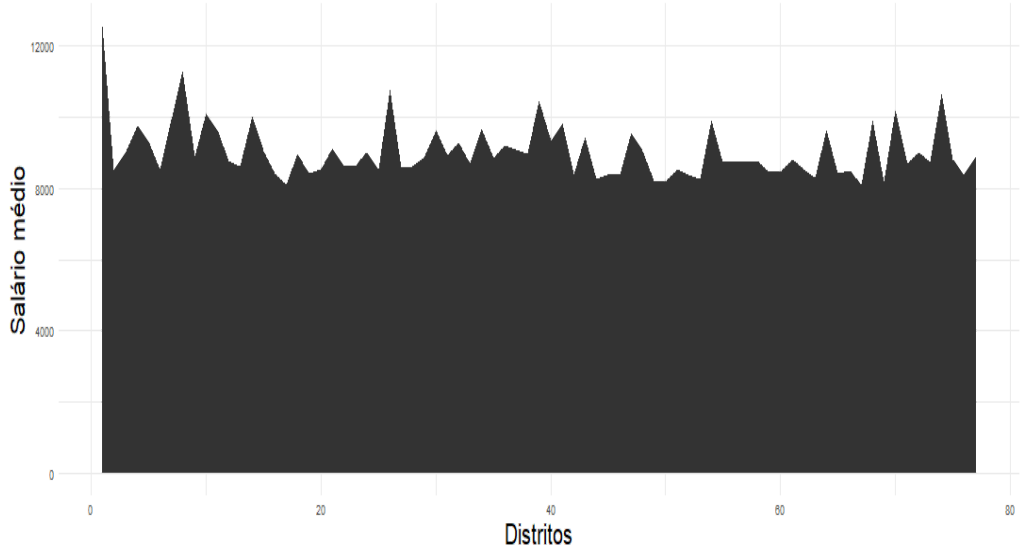
```
distrito <- distrito %>%
  select(A1, A10, A11) %>%
  rename(district_id='A1')

ggplot(distrito) +
  aes(x = district_id, y = A10) +
  geom_area(size = 0.5) +
  labs(x = "Distritos", y = "Proporção de habitantes urbanos") +
  theme_minimal()

ggplot(distrito) +
  aes(x = district_id, y = A11) +
  geom_area(size = 1.5) +
  labs(x = "Distritos", y = "Salário médio") +
  theme_minimal()
```



Observa-se uma variação da proporção de habitantes urbanos nos distritos entre 25 e 100%, com a maioria dos distritos concentrados na faixa entre 50 e 75%.



Já o salário médio possui uma variabilidade menor, com a maioria dos distritos concentrados um pouco acima de $8.000.

2.8 Análise dos cartões e finalização do ETL

Por fim, sumarizamos os dados sobre cartões e criamos variáveis binárias indicando a quais contas bancárias eles estão vinculados:

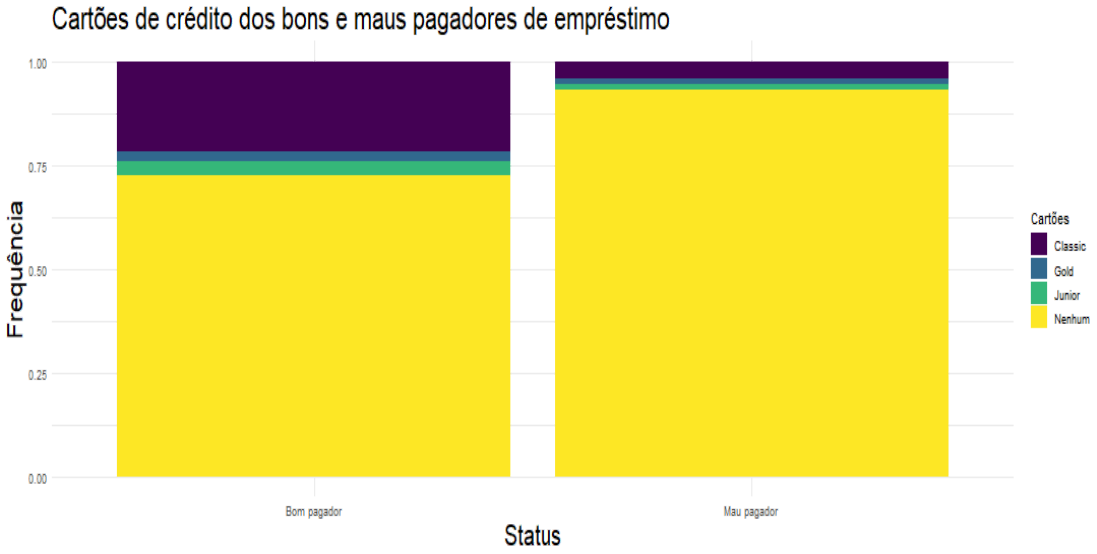
```
cartao <- cartao %>%
  mutate(junior=ifelse(type=="junior", 1, 0),
         classic=ifelse(type=="classic", 1, 0),
         gold=ifelse(type=="gold", 1, 0)) %>%
  select(disp_id,junior,classic,gold) %>%
  group_by(disp_id) %>%
  summarise(junior=sum(junior), classic=sum(classic), gold=sum(gold)) %>%
  ungroup()
```

Finalizamos o tratamento dos dados unindo as bases, transformando as variáveis categóricas e indicando a classe das colunas:

```
# unindo balanço médio do distrito
dados_final <- transloan %>%
  left_join(trans_acc1, by = "district_id")
# unindo proporção de habitantes urbanos e salário médio dos distritos
dados_final <- dados_final %>%
  left_join(distrito, by = "district_id")
# unindo conector
dados_final <- dados_final %>%
  left_join(conect_acc, by = "account_id")
# unindo cartão
dados_final <- dados_final %>%
  left_join(cartao, by = "disp_id")
dados_final <- dados_final  %>%
  mutate(idade=as.numeric(emprest_date - data_nasc))
rm(transloan, conect_acc, trans_acc1, distrito, cartao)
dados_final <- dados_final  %>%
  mutate(status=ifelse(status=="A"|status=="C", 1, 0), # 1= bom pagador // 0= mau pagador
         status=as.factor(status),
         junior=ifelse(is.na(junior), 0, junior),
         classic=ifelse(is.na(classic), 0, classic),
         gold=ifelse(is.na(gold), 0, gold),
         junior=as.factor(junior),
         classic=as.factor(classic),
         gold=as.factor(gold))
dados_final <- dados_final %>%
  select(account_id, status, balance, idade, gnr, junior, classic, gold, balance_distr, A10, A1
1)
```

Renderizamos gráficos para a análise da distribuição de cartões entre bons e maus pagadores:

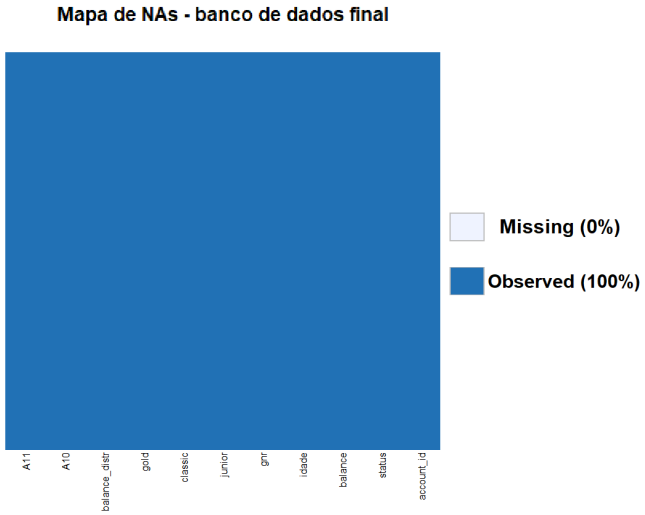
```
dados_finalX <- dados_final  %>%
  mutate(card=ifelse(junior==1, "Junior",
                     ifelse(classic==1, "Classic",
                            ifelse(gold==1, "Gold", "Nenhum"))),
         card=as.factor(card),
         status=ifelse(status==1, "Bom pagador", "Mau pagador"),
         status=as.factor(status))
ggplot(dados_finalX) +
  aes(x = status, fill = card) +
  geom_bar(position = "fill") +
  scale_fill_viridis_d(option = "viridis", direction = 1) +
  labs(
    x = "Status",
    y = "Frequência",
    title = "Cartões de crédito dos bons e maus pagadores de empréstimo"
  ) +
  theme_minimal()
rm(dados_finalX)
```



O gráfico indica que, em todos os tipos de cartão de crédito, existem mais bons do que maus tomadores de empréstimo, com cartões do tipo ‘Classic’ proporcionalmente mais propensos a serem bons pagadores. Contudo, a maioria dos clientes com contratos de empréstimo encerrados ou ativos não possuem cartões, sinalizando uma área com margem de crescimento para o banco.

Agora, analisamos se todos os casos dos dados finais estão completos:

```
missmap(dados_final, main='Mapa de NAs - banco de dados final', x.cex = 1.0, y.labels= NULL, y.at = NULL, margins = c(10, 10))
```



Salvamos a base final, adequada para a análise preditiva:

```
saveRDS(dados_final, 'dados_final')
```

3 Análise preditiva - Modelo de Regressão Logística

Carregando bibliotecas:

```
library(caret)
library(glmnet)
library(MASS)
library(ROCR)
library(leaps)
library(caret)
library(doParallel)
library(InformationValue)
library(tidyverse)
```

3.1 Construindo funções e indicando os parâmetros

Cálculo do limiar que maximiza a especificidade e a sensibilidade simultaneamente (mais informações: Hosmer e Lemeshow, 2000):

```
SelecionaCutOff <- function(fitted,y){
  ROCRpred <- prediction(fitted, y)
  sens <- data.frame(x=unlist(performance(ROCRpred, "sens")@x.values),
                     y=unlist(performance(ROCRpred, "sens")@y.values))
  spec <- data.frame(x=unlist(performance(ROCRpred, "spec")@x.values),
                     y=unlist(performance(ROCRpred, "spec")@y.values))
  sens %>% ggplot(aes(x,y)) +
    geom_line() +
    geom_line(data=spec, aes(x,y,col="red")) +
    scale_y_continuous(sec.axis = sec_axis(~., name = "Specificity")) +
    labs(x='Cutoff', y="Sensitivity") +
    theme(axis.title.y.right = element_text(colour = "red"), legend.position="none")
  sens <- cbind(unlist(performance(ROCRpred, "sens")@x.values), unlist(performance(ROCRpred, "se
ns")@y.values))
  spec <- cbind(unlist(performance(ROCRpred, "spec")@x.values), unlist(performance(ROCRpred, "sp
ec")@y.values))
  limiar <- sens[which.min(apply(sens, 1, function(x) min(colSums(abs(t(spec) - x))))), 1]

  return(limiar)

}
```

Cálculo das métricas para avaliação das previsões a partir da Matriz de Confusão (Hosmer e Lemeshow, 2000):

```
MetricasPrevisao <- function(real,previsto){
  mat <- table(real, previsto)
  if (dim(mat)[2] != 2){
    mat <- matrix(0,2,2)
  }

  TP <- mat[2,2]
  TN <- mat[1,1]
  FP <- mat[1,2]
  FN <- mat[2,1]

  tab_metricas <- matrix(NA,nrow=1,ncol=5)
  colnames(tab_metricas) <- c("accur","sens","spec","prec","f1")

  tab_metricas[1,1] <- (TP + TN) / (TP + FP + FN + TN)
  tab_metricas[1,2] <- TP/(TP + FN)
  tab_metricas[1,3] <- TN/(TN + FP)
  tab_metricas[1,4] <- TP/(TP+FP)
  tab_metricas[1,5] <- 2*(tab_metricas[1,2]*tab_metricas[1,4])/(tab_metricas[1,2]+tab_metricas
[1,4])


  return(tab_metricas)
}
```

Definindo o número de clusters para o k-fold cross-validation:

```
num_kfold  <-  10
```

`Set.seed()` para reprodutibilidade do exercício e dividindo dados entre duas amostras, uma de treino com 70% das observações, e uma de teste, com 30%:

```
set.seed(1)
sample <- sample(c(TRUE, FALSE), nrow(dados_final), replace=TRUE, prob=c(0.7,0.3))
dados_treino <- dados_final[sample, ]
dados_teste <- dados_final[!sample, ]
```

Constantes de regularização α (alfa) e λ (lamba) mantidas em 1 e 0, respectivamente:

```
tuneGrid<-expand.grid(alpha = 1, lambda=0)
```

Método de clusterização - coeficiente de variação:

```
trControl <- trainControl(method = 'cv', number = num_kfold, savePredictions =  TRUE,
                          classProbs = FALSE,
                          allowParallel = TRUE)
```

3.2 Aplicando modelo Logit e analisando desempenho

```
model.fit <- train(status ~ idade+balance+gnr+junior+classic+gold+balance_distr+A10+A11, data =
dados_treino,
                   method = "glmnet", trControl = trControl,
                   tuneGrid=tuneGrid, verbose=FALSE,
                   family="binomial", savePredictions = TRUE)
```

Verificando os coeficientes das variáveis explicativas, os correspondentes sinais e a significância a um nível de significância de 5%:

```
coef(model.fit$finalModel,model.fit$finalModel$lambdaOpt)
#(Intercept)   -2.43177779989
#idade          0.00001302629
#balance        0.00002594232
#gnr1          -0.50617475989
#junior1        0.30013297928
#classic1       1.04999511436
#gold1         -0.02530097904
#balance_distr  0.00007543199
#A10           -0.00759807449
#A11            0.00012097503
```

Pessoas com cartão tipo “Classic” têm uma log-chance 1.04999511436 maior de serem boas pagadoras, enquanto pessoas do gênero feminino ou com cartão do tipo “Gold” possuem uma log-chance maior de serem más pagadoras. As demais variáveis do nosso modelo não são significativas.

Definindo as métricas de desempenho e o limiar γ através da K-fold cross-validation:

```
# Limiar e Cutoff no CV
mat.cutoff <- matrix(NA,ncol=1,nrow=num_kfold,
                     dimnames = list(paste("fold",c(1:num_kfold),sep="")))

metricas.in <- matrix(NA, nrow= num_kfold, ncol = 5,
                      dimnames = list(paste("fold",c(1:num_kfold),sep=""),
                                      c("Acc","Sens","Spec","Prec","F1")))
for (k in 1:num_kfold){
  if(num_kfold >= 10){

    if(k < 10){
      index_fold_aux <- filter(model.fit$pred, Resample == paste("Fold0",k,sep="") )
    }
    if(k >= 10){
      index_fold_aux <- filter(model.fit$pred, Resample == paste("Fold",k,sep="") )
    }

  }
  if(num_kfold < 10){
    index_fold_aux <- filter(model.fit$pred, Resample == paste("Fold",k,sep="") )
  }

  prev_k_fold <- predict.train(model.fit, type = "prob", newdata = dados_treino[index_fold_aux$r
owIndex , ] )
  mat.cutoff[k, 1 ] <- SelecionaCutOff(fitted = prev_k_fold$`1` , y = dados_treino$status[index_
fold_aux$rowIndex])

  real <-  dados_treino$status[index_fold_aux$rowIndex]
  previsto <- if_else(prev_k_fold$`1` <  mat.cutoff[ k,1] , 0  ,1)
  metricas.in[k, ] <-  MetricasPrevisao(real = real , previsto = previsto)

}
```

Analisando métricas de performance dos modelos de treino e teste:

```
Metricas.Totais.In <- colMeans(metricas.in)
cut.off.selected <- mean(mat.cutoff)
prob_in <-  predict.train(model.fit, type = "prob", newdata = dados_treino )
previsto.in <- if_else(prob_in$`1` <  cut.off.selected , 0  ,1)
Conf_in <- InformationValue::confusionMatrix(dados_treino$status, previsto.in)
Conf_in
   0   1
0 32 166
1 16 265
Conf_in <- caret::confusionMatrix(as.factor(previsto.in),as.factor(dados_treino$status))
Conf_in

Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0  32 166
         1  16 265

               Accuracy : 0.62
                 95% CI : (0.5749, 0.6637)
    No Information Rate : 0.8998
    P-Value [Acc > NIR] : 1

                  Kappa : 0.1179

 Mcnemar's Test P-Value : <0.0000000000000002

            Sensitivity : 0.66667
            Specificity : 0.61485
         Pos Pred Value : 0.16162
         Neg Pred Value : 0.94306
             Prevalence : 0.10021
         Detection Rate : 0.06681
   Detection Prevalence : 0.41336
      Balanced Accuracy : 0.64076

       'Positive' Class : 0

prob_out <- predict.train(model.fit, type = "prob", newdata = dados_teste )
previsto.out <- if_else(prob_out$`1` <  cut.off.selected , 0  ,1)
Conf_out <- InformationValue::confusionMatrix(dados_teste$status, previsto.out)
Conf_out
   0   1
0 18  62
1 10 113
Conf_out <- caret::confusionMatrix(as.factor(previsto.out),as.factor(dados_teste$status))
Conf_out

Confusion Matrix and Statistics

          Reference
```

```
Prediction   0   1
         0  18  62
         1  10 113

               Accuracy : 0.6453
                 95% CI : (0.5753, 0.711)
    No Information Rate : 0.8621
    P-Value [Acc > NIR] : 1

                  Kappa : 0.1621

 Mcnemar's Test P-Value : 0.000000001851

            Sensitivity : 0.64286
            Specificity : 0.64571
         Pos Pred Value : 0.22500
         Neg Pred Value : 0.91870
             Prevalence : 0.13793
         Detection Rate : 0.08867
   Detection Prevalence : 0.39409
      Balanced Accuracy : 0.64429

       'Positive' Class : 0
```

Baixa capacidade preditiva (acurácia entre 62-64%), com muitos falsos-negativos [bons pagadores (1) previstos como maus (0)].

4 Conclusões

O banco, agora, conhece melhor o perfil dos seus clientes, podendo buscar uma fidelização em nicho ou maximizar sua presença em quadrantes demográficos menos explorados, como abordamos ao longo do exercício.

Variáveis monetárias possuem pouca variação e foram insuficientes para criar diferenciações claras entre clientes. Em um cenário como este, uma instituição financeira, com um grande volume de dados de natureza monetária, precisa traçar estratégias de captação de outros tipos de variáveis, seja através de conexões com dados de outras instituições ou desenvolvendo um plano de ação em que a coleta dessas informações ocorra de forma natural.

Existem vários tipos de dados não-monetários que podem auxiliar na previsão de um comportamento econômico: a pontuação escolar em matemática, que indica se um indivíduo é capaz manter uma contabilidade pessoal; exames clínicos, indicando a longevidade e estabilidade física e emocional do cliente; dados de deslocamento, que indicam a probabilidade de um indivíduo estar próximo ou distante de uma agência bancária (e, por tanto, a probabilidade de atrasar pagamentos), entre outros.

Pensa-se que o melhor caminho para nosso banco hipotético é traçar uma estratégia híbrida entre expandir sua presença e coletar mais informações, viabilizando testes estatísticos diversificados.

5 Referências

David W. Hosmer e Stanley Lemeshow. Applied logistic regression. John Wiley & Sons, 2000.