

Exploração de dados - Banco Czech

Contents

Objetivo	3
Carregar os dados	3
Tratamento dos dados	3
Cliente	3
Distrito	3
Cartão	4
Transação	4

Objetivo

O objetivo do nosso trabalho é identificar quais fatores que podem impactar no atraso e não pagamento das dívidas.

Carregar os dados

Utilizamos a função `read.csv2` que nos permite carregar os dados disponíveis em texto, no formato CSV.

```
read.csv2("./dados/account.asc", stringsAsFactors = FALSE) -> account
read.csv2("./dados/client.asc", stringsAsFactors = FALSE) -> client
read.csv2("./dados/card.asc", stringsAsFactors = FALSE) -> card
read.csv2("./dados/disp.asc", stringsAsFactors = FALSE) -> disp
read.csv2("./dados/district.asc", stringsAsFactors = FALSE) -> district
read.csv2("./dados/trans.asc", stringsAsFactors = FALSE) -> trans
```

Tratamento dos dados

Cliente

O tratamento inicial da relação cliente envolve a separação do campo data de nascimento e gênero, que será tratado por M e F. E a transformação do campo `birth_number` em uma data válida para o R. Para isso transformamos o `birth_number` em um campo numérico, obtemos 4 dígitos da 3ª à 4ª posição, sendo este valor superior a 50 consideramos como feminino, pois o mês de nascimento das mulheres está com uma soma de 50 unidades. Subtraímos 5000 do mês das mulheres, pois como um único campo numérico, implica em diminuir 50 do campo mensal. Após isso concatenamos o número 19 ao começo do `birth_number`, no intuito de deixar melhor preparado para a formatação da data, que ocorre logo em seguida. Logo após, calculamos a idade e selecionamos apenas os campos que nos serão úteis para o nosso estudo. Vamos a data de referência como 01/01/1998, devido a referência dos dados, para calcular a idade dos clientes.

```
currentdate <- as.Date("1998/01/01", format="%Y/%m/%d")
client <- client %>%
  mutate(mesajustado = as.numeric(stringr::str_sub(birth_number,3,4))) %>%
  mutate(gender = ifelse(mesajustado > 50, "F", "M")) %>%
  mutate(birth_number = ifelse(gender=="F", birth_number - 5000, birth_number)) %>%
  mutate(birth_number = paste0("19", birth_number)) %>%
  mutate(birth_number = as.Date(birth_number, "%Y%m%d")) %>%
  mutate(age = year(currentdate) - year(birth_number)) %>%
  select(client_id, age, district_id, gender)
```

Distrito

O tratamento inicial da relação distrito começa na renomeação dos campos para melhor entendimento. Conversão dos campos de `unemp_95` e `unemp_96` para numéricos. Limpeza dos valores NA. Cálculo da taxa de desemprego entre os anos 95 e 96. E seleção dos valores que serão usados neste estudo.

```
#Renomear campos para melhor entendimento
colnames(district)[1] <- 'district_id'
colnames(district)[2] <- 'district_name'
colnames(district)[11] <- 'avg_sal'
colnames(district)[12] <- 'unemp_95'
colnames(district)[13] <- 'unemp_96'
```

```
#Converter campos para numérico
```

```

district$unemp_95 = as.numeric(district$unemp_95)

## Warning: NAs introduzidos por coerção
district$unemp_96 = as.numeric(district$unemp_96)

#Limpeza de NA
district[is.na(district$unemp_95),12] <- 1

#Cálculo da taxa de desemprego e seleção de valores
district %>%
  mutate(unemp_r = ifelse(unemp_95 == 0 | unemp_96 == 0, 1, unemp_96/unemp_95)) %>%
  select(district_id, district_name, avg_sal, unemp_95, unemp_96) -> district

```

Cartão

O tratamento inicial da relação cartão seleciona os campos que serão úteis para este estudo. Preparamos então o tipo do cartão como categórica.

```

card <- card %>%
  select(card_id, disp_id, type) %>%
  mutate(type = as.factor(type))

```

Transação

Na relação de transação, identificamos alguns valores como “VYBER” que não estão descritos na documentação e não pareceu ser pertinente aos nossos estudos, por isso foi filtrado. O tipo de transação foi traduzido para facilitar o entendimento. E foi feito um agrupamento pela conta para facilitar o relacionamento de 1 para 1 com os valores que nos interessam.

```

read.csv2("../dados/trans.asc", stringsAsFactors = FALSE) -> trans

unique(trans$type)

## [1] "PRIJEM" "VYDAJ" "VYBER"

trans <- trans %>%
  dplyr::filter(type != "VYBER") %>%
  mutate(type = ifelse(
    type == "PRIJEM", "credito",
    ifelse(type == "VYDAJ", "debito", type))) %>%
  mutate(amount = as.numeric(amount)) %>%
  select(account_id, type, amount) %>%
  group_by(account_id) %>%
  summarise(
    credito = sum(ifelse(type == "credito", amount, 0)),
    debito = sum(ifelse(type == "debito", amount, 0)),
    count = n()) %>%
  mutate(withdraw_rate = debito / credito) %>%
  select(account_id, withdraw_rate, count)

```