Desafio Data Science

Henrique C Vieira

Junho 4, 2018

Analisando os dados bank

O dado **Bank Marketing** foi obtido do repositório de datasets https://archive.ics.uci.edu

Link para download: bank.zip

Carregando bibliotecas a serem usadas nesse projeto.

```
library(dplyr, warn.conflicts=FALSE, verbose=FALSE)
library(ggplot2, warn.conflicts=FALSE, verbose=FALSE)
library(tidyr, warn.conflicts=FALSE, verbose=FALSE)
library(readr, warn.conflicts=FALSE, verbose=FALSE)
library(FSelector)
```

Introdução

Este é um dado relacionado com as campanhas de marketing de uma instuição bancária de Portugal. São campanhas baseadas em ligações telefônicas para oferecer aos clientes o serviço de depósito a prazo fixo (bank term deposit), onde poderão retirar o valor após o prazo ter vencido.

O termo term deposit: A term deposit is a fixed-term deposit held at a financial institution. They are generally short-term deposits with maturities ranging anywhere from a month to a few years. When a term deposit is purchased, the client understands that the money can only be withdrawn after the term has ended or by giving a predetermined number of days notice. Investopedia

[Moro et al., 2011] S. Moro, R. Laureano and P. Cortez. Using Data Mining for Bank Direct Marketing: An Application of the CRISP-DM Methodology. In P. Novais et al. (Eds.), Proceedings of the European Simulation and Modelling Conference - ESM'2011, pp. 117-121, Guimarães, Portugal, October, 2011. EUROSIS.

Available at: [pdf] http://hdl.handle.net/1822/14838 [bib] http://www3.dsi.uminho.pt/pcortez/bib/2011-esm-1.txt

Carregando o dataset bank

Utilizaremos o pacote readr, que permite carregar dados em formato csv e tsv mais rápido que as funções básicas do R para carregar dados.

```
dataset <- read_delim('bank/bank.csv', delim = ';')
```

```
## Parsed with column specification:
## cols(
##   age = col_integer(),
##   job = col_character(),
##   marital = col_character(),
##   education = col_character(),
##   default = col_character(),
##   balance = col_integer(),
```

```
##   housing = col_character(),
##   loan = col_character(),
##   contact = col_character(),
##   day = col_integer(),
##   month = col_character(),
##   duration = col_integer(),
##   campaign = col_integer(),
##   pdays = col_integer(),
##   previous = col_integer(),
##   poutcome = col_character(),
##   y = col_character()
## )
```

```
glimpse(dataset)
```

```
## Observations: 4,521
## Variables: 17
## $ age       <int> 30, 33, 35, 30, 59, 35, 36, 39, 41, 43, 39, 43, 36, ...
## $ job       <chr> "unemployed", "services", "management", "management"...
## $ marital   <chr> "married", "married", "single", "married", "married"...
## $ education <chr> "primary", "secondary", "tertiary", "tertiary", "sec...
## $ default   <chr> "no", "no", "no", "no", "no", "no", "no", "no", "no"...
## $ balance   <int> 1787, 4789, 1350, 1476, 0, 747, 307, 147, 221, -88, ...
## $ housing   <chr> "no", "yes", "yes", "yes", "yes", "no", "yes", "yes"...
## $ loan      <chr> "no", "yes", "no", "yes", "no", "no", "no", "no", "n...
## $ contact   <chr> "cellular", "cellular", "cellular", "unknown", "unkn...
## $ day       <int> 19, 11, 16, 3, 5, 23, 14, 6, 14, 17, 20, 17, 13, 30,...
## $ month     <chr> "oct", "may", "apr", "jun", "may", "feb", "may", "ma...
## $ duration  <int> 79, 220, 185, 199, 226, 141, 341, 151, 57, 313, 273,...
## $ campaign  <int> 1, 1, 1, 4, 1, 2, 1, 2, 2, 1, 1, 2, 2, 1, 1, 2, 5, 1...
## $ pdays     <int> -1, 339, 330, -1, -1, 176, 330, -1, -1, 147, -1, -1,...
## $ previous  <int> 0, 4, 1, 0, 0, 3, 2, 0, 0, 2, 0, 0, 0, 0, 1, 0, 0, 2...
## $ poutcome  <chr> "unknown", "failure", "failure", "unknown", "unknown"...
## $ y         <chr> "no", "no", "no", "no", "no", "no", "no", "no", "no"...
```

Alterando o tipo da coluna para o tipo correto

Alterando as colunas do tipo texto para colunas do tipo categórico, em R são do tipo factor. As colunas com valores 'yes' e 'no' foram transfomadas em valores lógicos TRUE e FALSE.

```
dataset$job <- as.factor(dataset$job)
dataset$marital <- as.factor(dataset$marital)
dataset$education <- as.factor(dataset$education)
dataset$default <- ifelse(dataset$default == 'yes', TRUE, FALSE)
dataset$housing <- ifelse(dataset$housing == 'yes', TRUE, FALSE)
dataset$loan <- ifelse(dataset$loan == 'yes', TRUE, FALSE)
dataset$contact <- as.factor(dataset$contact)
dataset$day <- as.factor(dataset$day)
dataset$month <- as.factor(dataset$month)
dataset$campaign <- as.factor(dataset$campaign)
dataset$poutcome <- as.factor(dataset$poutcome)
dataset$y <- ifelse(dataset$y == 'yes', TRUE, FALSE)
dataset$term <- dataset$y
glimpse(dataset)
```

```
## Observations: 4,521
## Variables: 18
## $ age       <int> 30, 33, 35, 30, 59, 35, 36, 39, 41, 43, 39, 43, 36, ...
## $ job       <fctr> unemployed, services, management, management, blue-...
## $ marital   <fctr> married, married, single, married, married, single,...
## $ education <fctr> primary, secondary, tertiary, tertiary, secondary, ...
## $ default   <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FAL...
## $ balance   <int> 1787, 4789, 1350, 1476, 0, 747, 307, 147, 221, -88, ...
## $ housing   <lgl> FALSE, TRUE, TRUE, TRUE, TRUE, FALSE, TRUE, TRUE, TR...
## $ loan      <lgl> FALSE, TRUE, FALSE, TRUE, FALSE, FALSE, FALSE, FALSE...
## $ contact   <fctr> cellular, cellular, cellular, unknown, unknown, cel...
## $ day       <fctr> 19, 11, 16, 3, 5, 23, 14, 6, 14, 17, 20, 17, 13, 30...
## $ month     <fctr> oct, may, apr, jun, may, feb, may, may, may, apr, m...
## $ duration  <int> 79, 220, 185, 199, 226, 141, 341, 151, 57, 313, 273,...
## $ campaign  <fctr> 1, 1, 1, 4, 1, 2, 1, 2, 2, 1, 1, 2, 2, 1, 1, 2, 5, ...
## $ pdays     <int> -1, 339, 330, -1, -1, 176, 330, -1, -1, 147, -1, -1,...
## $ previous  <int> 0, 4, 1, 0, 0, 3, 2, 0, 0, 2, 0, 0, 0, 0, 1, 0, 0, 2...
## $ poutcome  <fctr> unknown, failure, failure, unknown, unknown, failur...
## $ y         <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FAL...
## $ term      <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FAL...
```

Sumarização dos dados

Podemos observar abaixo, os valores de minimo, máximo, 1º, 2º (mediana) e 3º quartil e a média para dados númericos e a contagem individual para cada valor para os dados categóricos.

Podemos perceber um grande desbalanceamento dos dados.

```
dataset %>%
  summary()
```

```
##       age                job          marital         education
##  Min.   :19.00   management :969   divorced: 528   primary  : 678
##  1st Qu.:33.00   blue-collar:946   married :2797   secondary:2306
##  Median :39.00   technician :768   single  :1196   tertiary :1350
##  Mean   :41.17   admin.     :478                   unknown  : 187
##  3rd Qu.:49.00   services   :417
##  Max.   :87.00   retired    :230
##                  (Other)    :713
##   default          balance       housing          loan
##  Mode :logical   Min.   :-3313   Mode :logical   Mode :logical
##  FALSE:4445      1st Qu.:   69   FALSE:1962      FALSE:3830
##  TRUE :76        Median :  444   TRUE :2559      TRUE :691
##                  Mean   : 1423
##                  3rd Qu.: 1480
##                  Max.   :71188
##
##       contact          day           month         duration
##  cellular :2896   20     : 257   may    :1398   Min.   :   4
##  telephone: 301   18     : 226   jul    : 706   1st Qu.: 104
##  unknown  :1324   19     : 201   aug    : 633   Median : 185
##                   21     : 198   jun    : 531   Mean   : 264
##                   14     : 195   nov    : 389   3rd Qu.: 329
##                   17     : 191   apr    : 293   Max.   :3025
```

```
##                   (Other):3253   (Other): 571
##     campaign        pdays           previous          poutcome
##  1      :1734   Min.   : -1.00   Min.   : 0.0000   failure: 490
##  2      :1264   1st Qu.: -1.00   1st Qu.: 0.0000   other  : 197
##  3      : 558   Median : -1.00   Median : 0.0000   success: 129
##  4      : 325   Mean   : 39.77   Mean   : 0.5426   unknown:3705
##  5      : 167   3rd Qu.: -1.00   3rd Qu.: 0.0000
##  6      : 155   Max.   :871.00   Max.   :25.0000
##  (Other): 318
##      y              term
##  Mode :logical   Mode :logical
##  FALSE:4000      FALSE:4000
##  TRUE :521       TRUE :521
##
##
##
##
```

Relações dos dados e gráficos

Número de individuos pela ocupação profissional e estado civil.

```
dataset %>%
  group_by(job) %>%
  count(marital, sort = TRUE)
```

```
## # A tibble: 35 x 3
## # Groups:   job [12]
##            job marital     n
##         <fctr>  <fctr> <int>
##  1 blue-collar married   693
##  2  management married   557
##  3  technician married   411
##  4  management  single   293
##  5  technician  single   268
##  6      admin. married   266
##  7    services married   236
##  8     retired married   176
##  9 blue-collar  single   174
## 10      admin.  single   143
## # ... with 25 more rows
```

Número total de sucessos por campanha

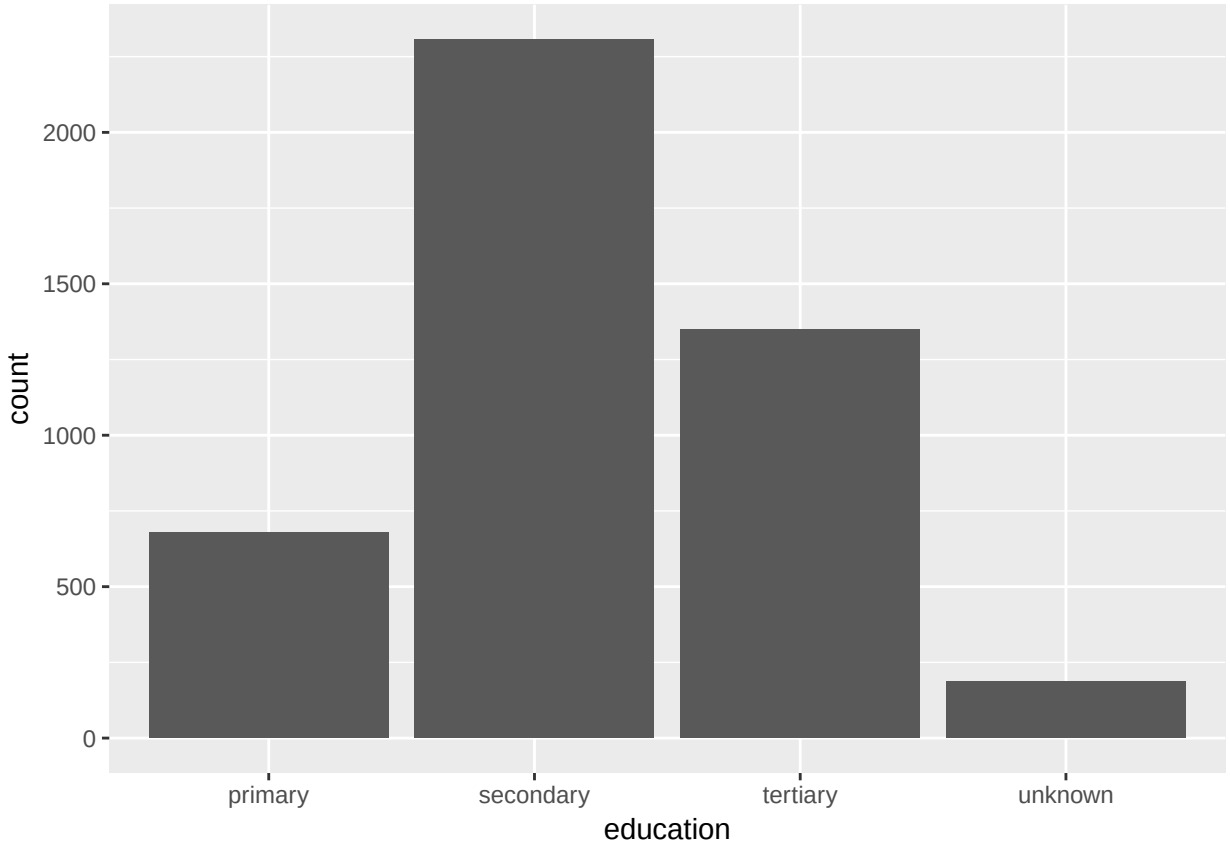
```
dataset %>%
  group_by(poutcome) %>%
  count(campaign, sort = TRUE)
```

```
## # A tibble: 59 x 3
## # Groups:   poutcome [4]
##    poutcome campaign     n
##      <fctr>   <fctr> <int>
##  1  unknown        1  1309
##  2  unknown        2  1057
```

```
##  3  unknown        3   474
##  4  unknown        4   288
##  5  failure        1   263
##  6  unknown        5   147
##  7  unknown        6   137
##  8  failure        2   131
##  9    other        1    88
## 10  success        1    74
## # ... with 49 more rows
```

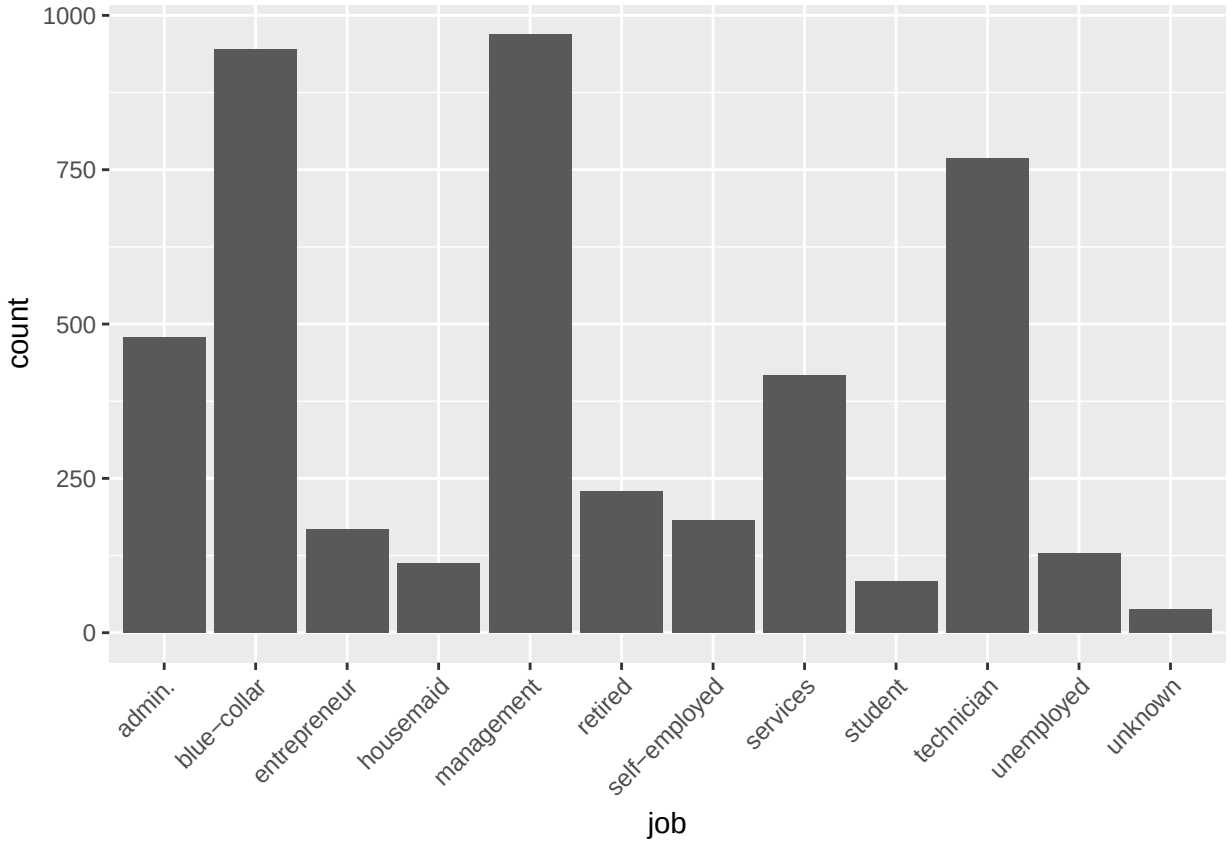
Número de individuos por nível escolar

```
ggplot(dataset, aes(x=education)) +
  geom_bar()
```



Número de individuos pela ocupação profissional

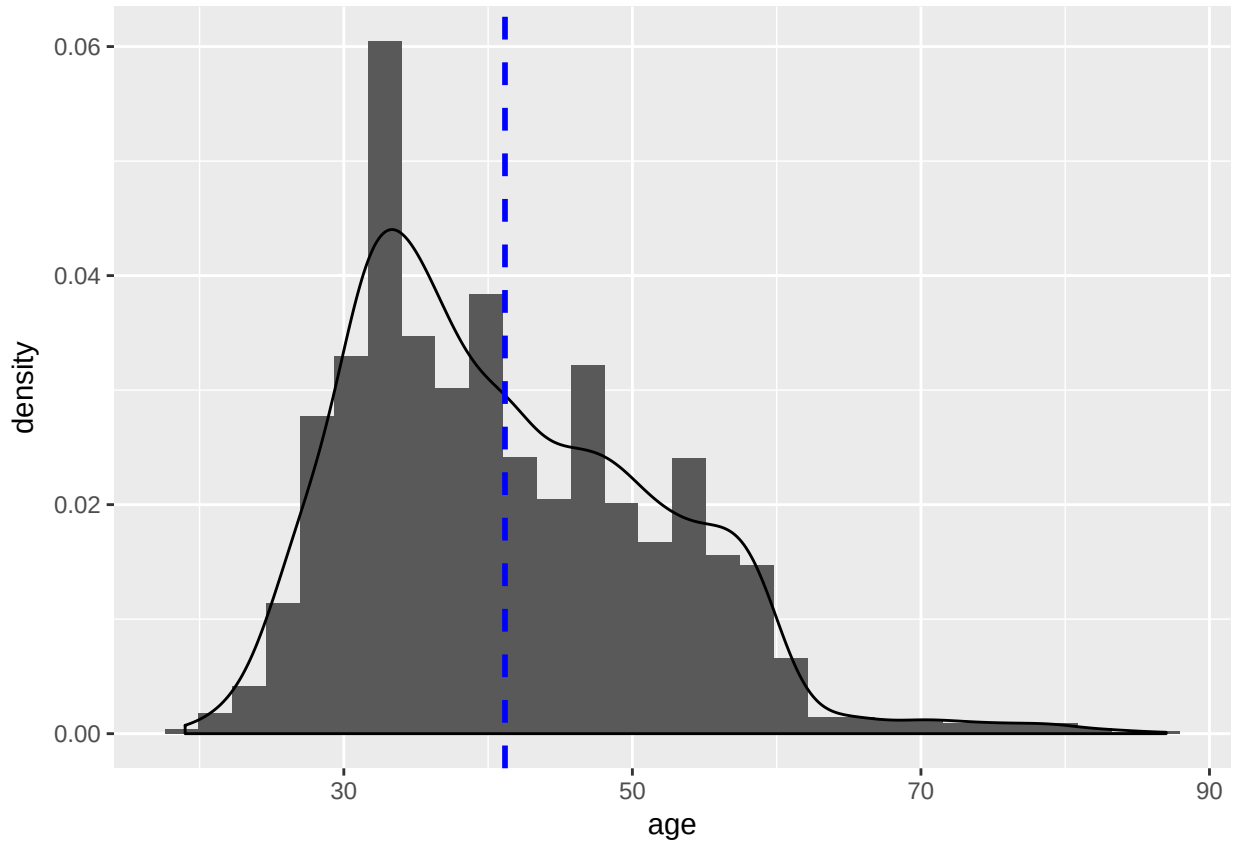
```
ggplot(dataset, aes(x=job)) +
  geom_bar() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Distribuição da idade

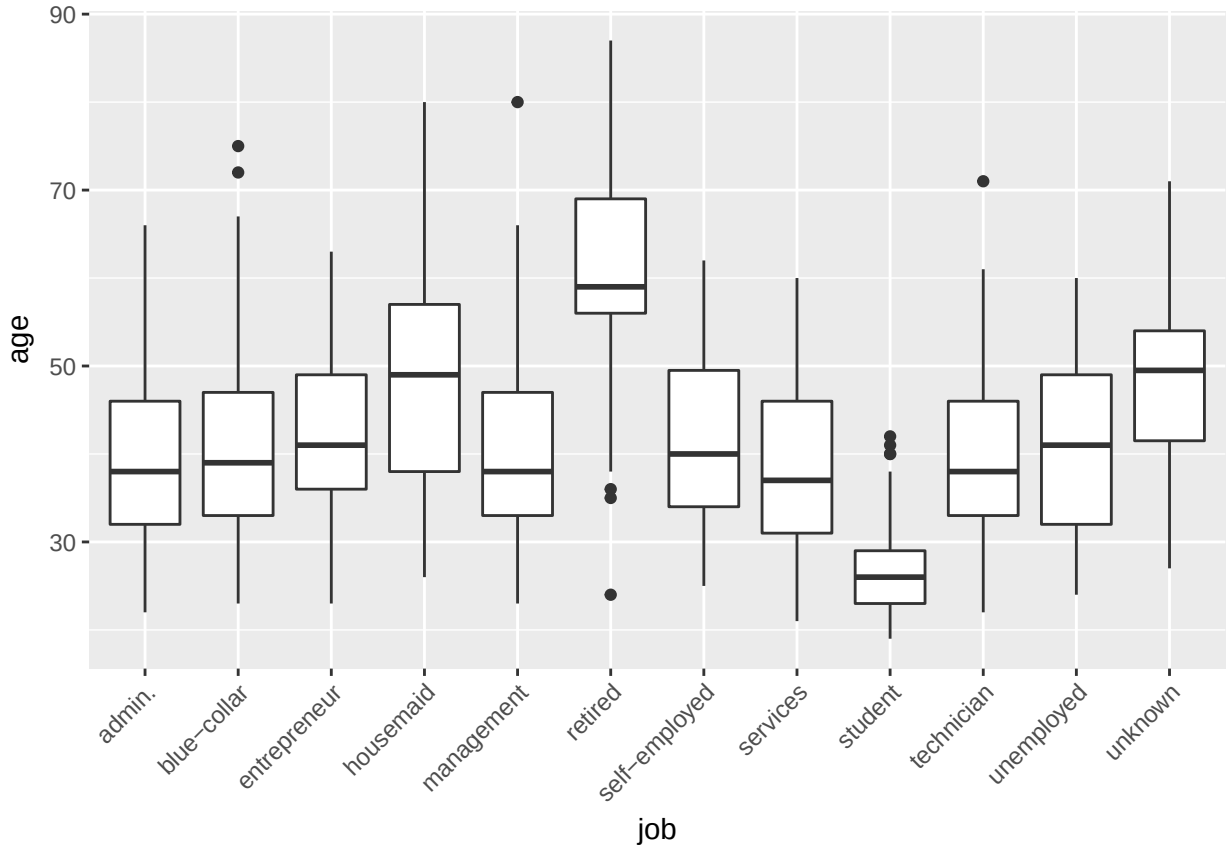
```
ggplot(dataset, aes(x=age, y=..density..)) +
  geom_histogram() +
  geom_density() +
  geom_vline(aes(xintercept=mean(age)),
             color="blue", linetype="dashed", size=1)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



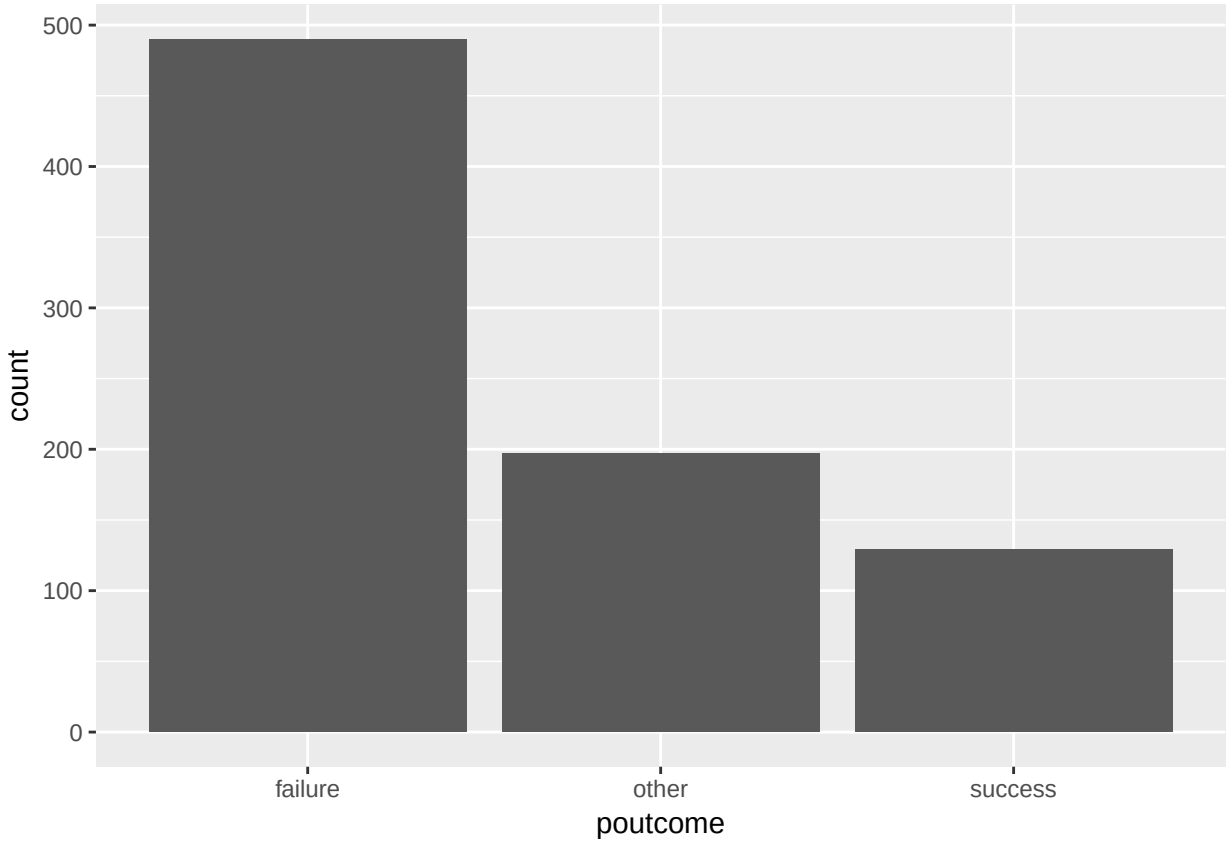
Relação entre ocupação profissional e a idade

```
ggplot(dataset, aes(x=job, y=age)) +
  geom_boxplot() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



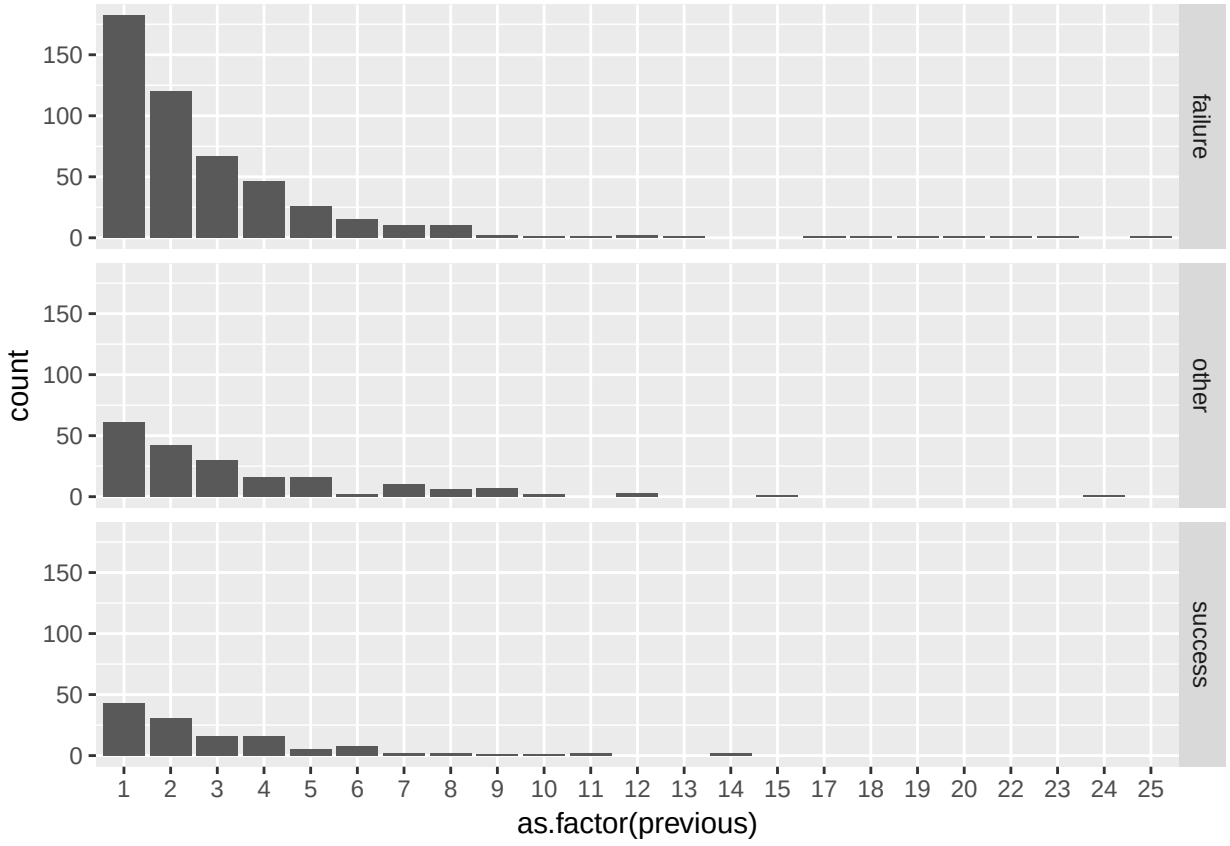
Número de contratos assinados na campanha anterior

```
dataset %>%
  filter(poutcome != 'unknown') %>%
  ggplot(aes(x=poutcome)) +
  geom_bar()
```



Número de contatos pela campanha anterior e estado (sucesso, falha e outros)

```
dataset %>%
  filter(poutcome != 'unknown') %>%
  ggplot(aes(x=as.factor(previous))) +
  geom_bar() +
  facet_grid(poutcome ~ .)
```

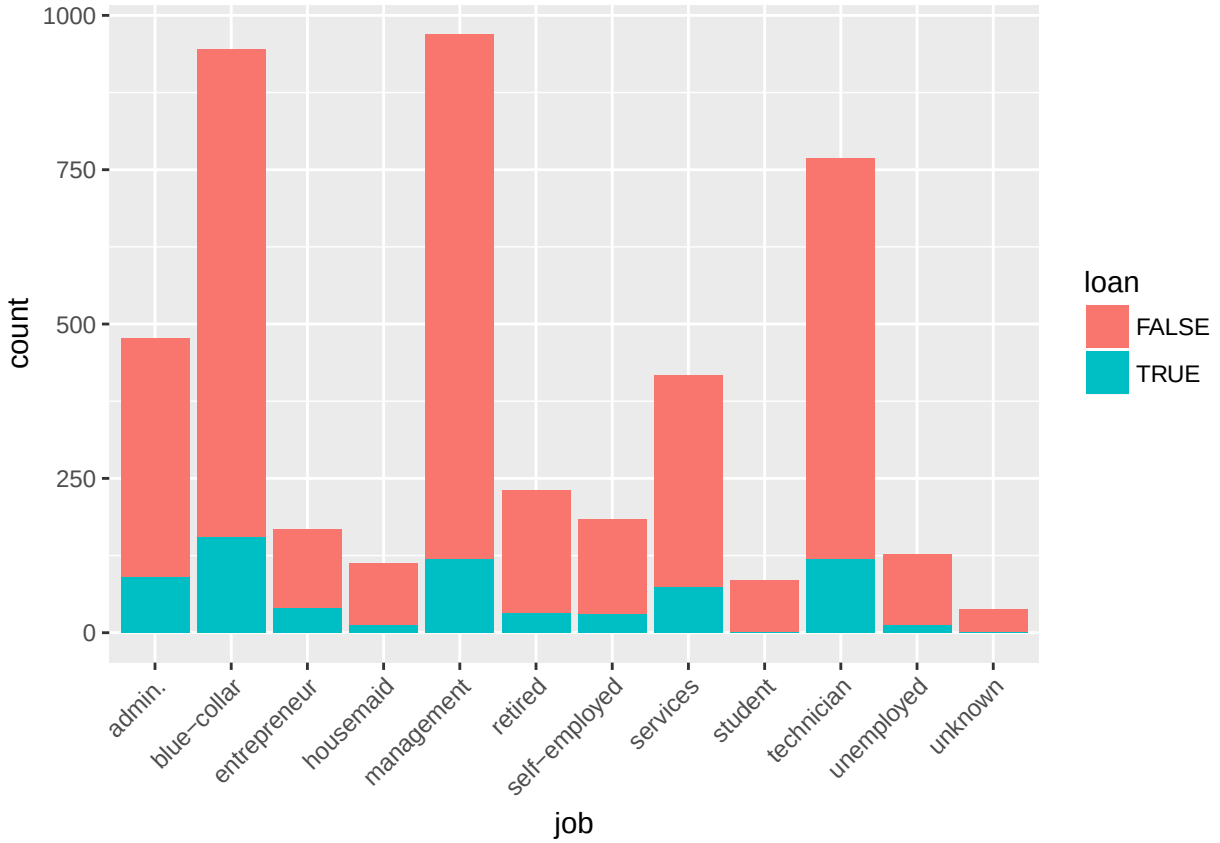


Respondendo as questões:

1. Qual profissão tem mais tendência a fazer um empréstimo? De qual tipo?

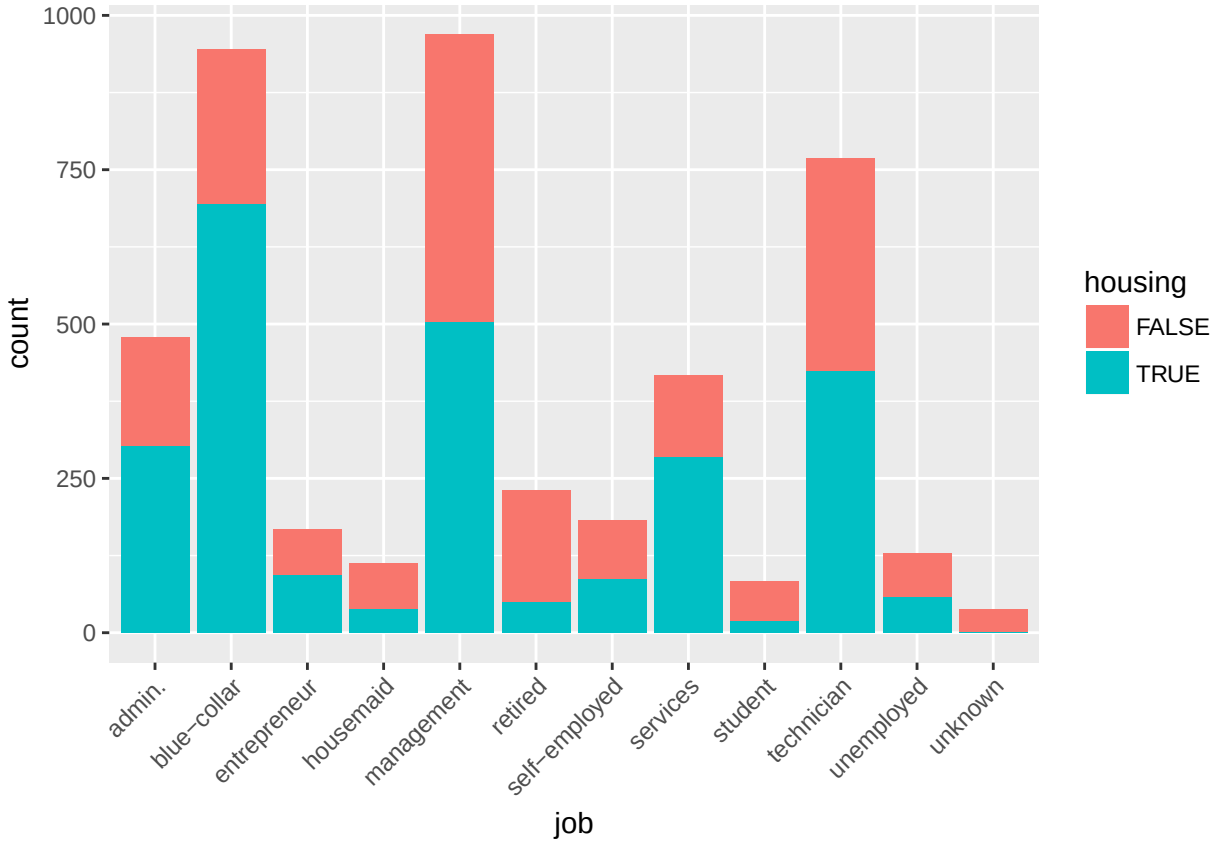
Visualizando as profissões por empréstimo

```
ggplot(dataset, aes(x=job, fill=loan)) +
  geom_bar() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Visualizando as profissões por empréstimo imobiliário

```
ggplot(dataset, aes(x=job, fill=housing)) +
  geom_bar() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Como podemos observar pelos dois gráficos, o blue-collar e management são as duas profissões que mais se destacam. Vamos filtrar por essas duas profissões e analisar qual apresentar o maior número de empréstimos e de qual tipo.

```
dataset %>%
  filter(loan == TRUE | housing == TRUE) %>%
  filter(job == 'blue-collar' | job == 'management') %>%
  count(job, loan, housing)
```

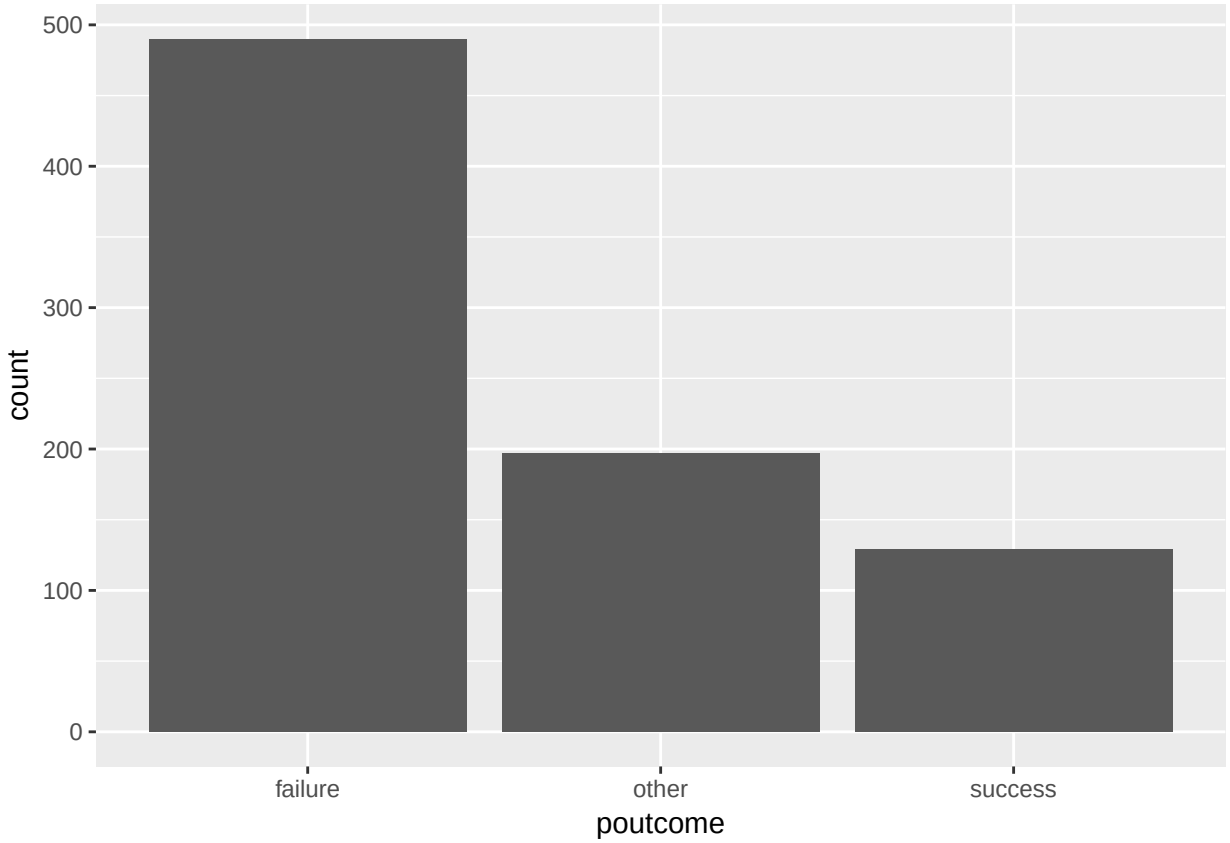
```
## # A tibble: 6 x 4
##           job  loan housing     n
##        <fctr> <lgl>   <lgl> <int>
## 1 blue-collar FALSE    TRUE   592
## 2 blue-collar  TRUE   FALSE    53
## 3 blue-collar  TRUE    TRUE   103
## 4  management FALSE    TRUE   442
## 5  management  TRUE   FALSE    59
## 6  management  TRUE    TRUE    61
```

blue-collar é a profissão que mais realiza empréstimos e do tipo housing.

2. Fazendo uma relação entre número de contatos e sucesso da campanha quais são os pontos relevantes a serem observados?

No gráfico abaixo vemos o resultado da campanha anterior.

```
dataset %>%
  filter(poutcome != 'unknown') %>%
  ggplot(aes(x=poutcome)) +
  geom_bar()
```




```
dataset %>%
  filter(poutcome == "success") %>%
  count(previous, sort = TRUE)
```

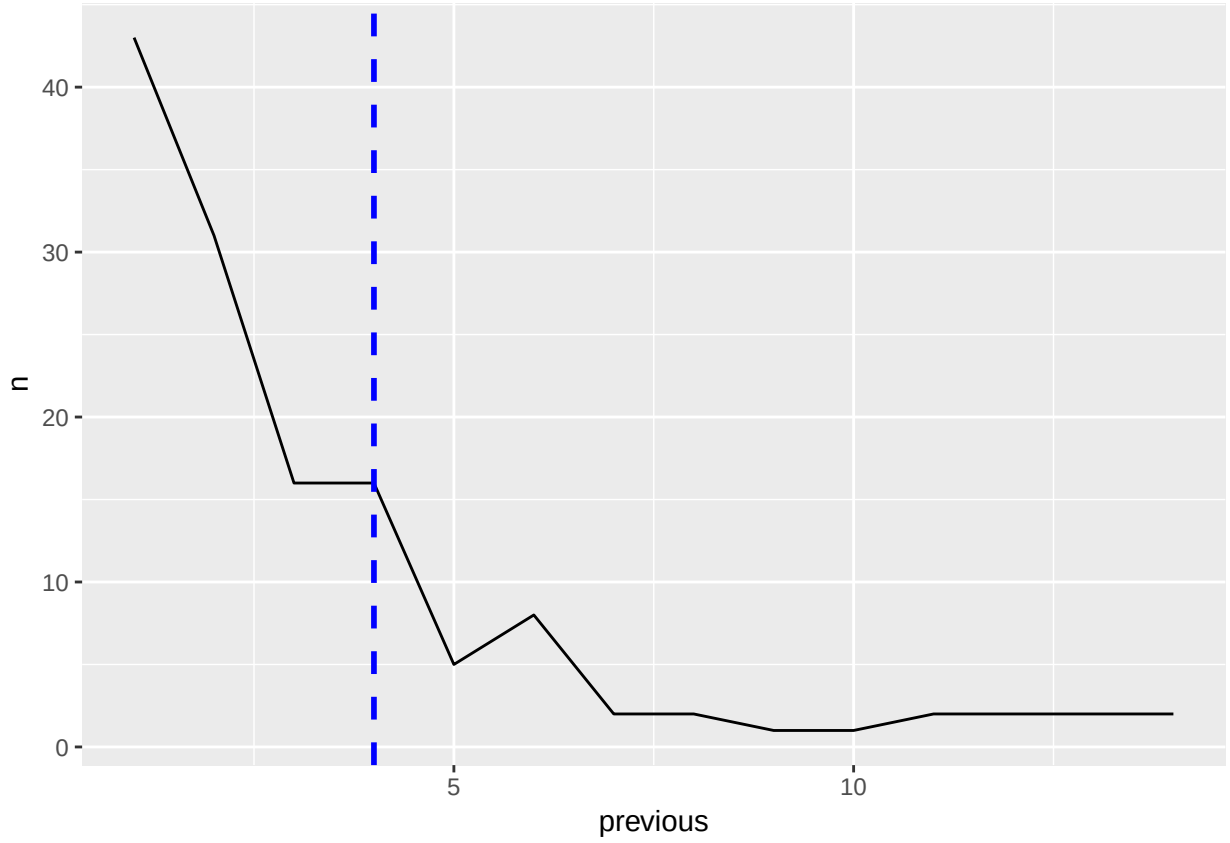
```
## # A tibble: 12 x 2
##    previous     n
##       <int> <int>
##  1        1    43
##  2        2    31
##  3        3    16
##  4        4    16
##  5        6     8
##  6        5     5
##  7        7     2
##  8        8     2
##  9       11     2
## 10       14     2
## 11        9     1
## 12       10     1
```

Temos um grande numero de falhas na campanha anterior. Os contatos onde obtiveram sucesso se destacaram realizando de 1, 2, 3 ou até 4 contatos com o cliente. A partir de 6 ligações há um número menor de clientes

que mudam de opinião e passam aderem.

3. Baseando-se nos resultados de adesão desta campanha qual o número médio e o máximo de ligações que você indica para otimizar a adesão?

```
dataset %>%
  filter(poutcome == "success") %>%
  count(previous, sort = TRUE) %>%
  ggplot(aes(x=previous, y=n)) +
  geom_line() +
  geom_vline(aes(xintercept=4),
             color="blue", linetype="dashed", size=1)
```



Observandoo gráfico acima, podemos perceber que há uma queda muito grande a partir de 5 ligações, sugiro em média 2 ligações e no máximo 4 para maximizar a adesão de novos clientes.

4. O resultado da campanha anterior tem relevância na campanha atual?

```
table(dataset$poutcome, dataset$term)
```

```
##
##           FALSE TRUE
##   failure   427   63
##   other     159   38
```

```
##   success    46   83
##   unknown  3368  337
```

Houve pouca relevância nessa campanha, o número de adesão é baixo em relação a atual com a anterior, inclusive havendo uma desistência dos que assinaram na campanha anterior.

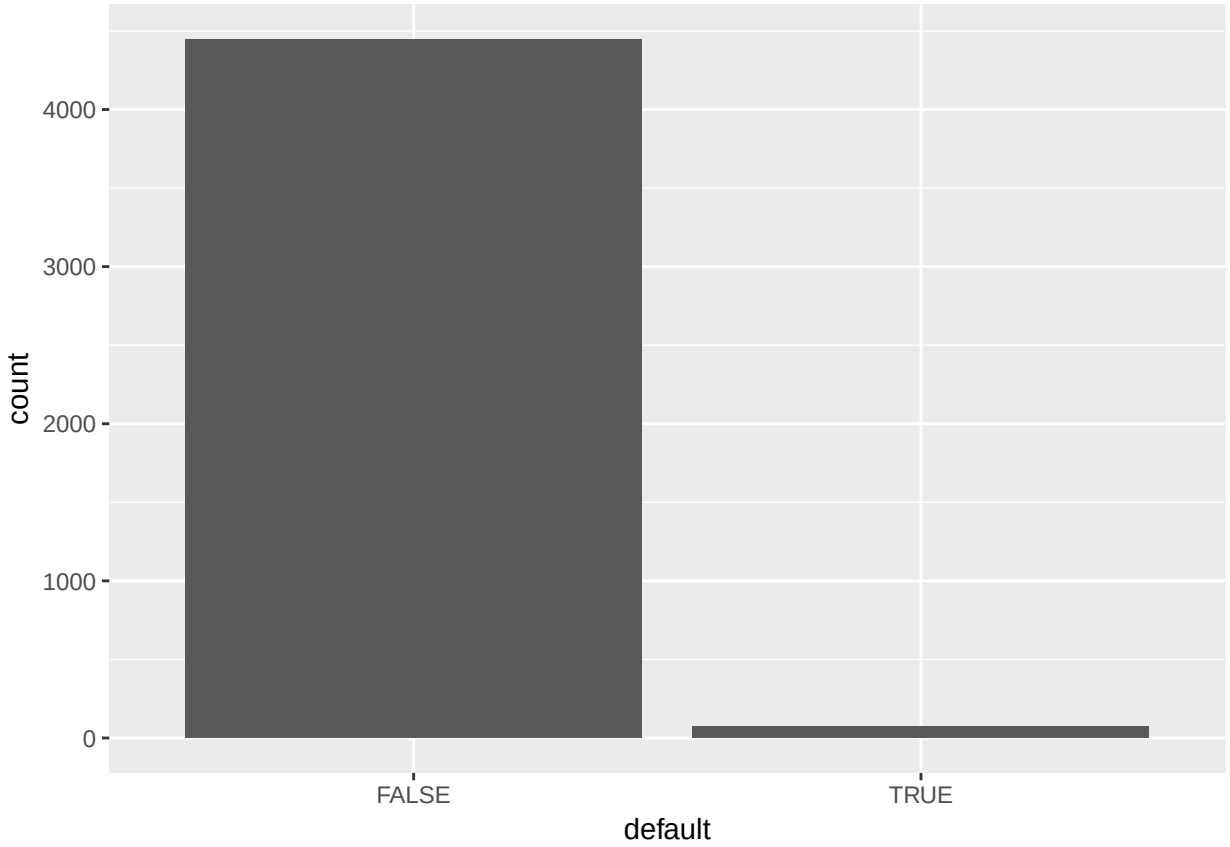
5. Qual o fator determinante para que o banco exija um seguro de crédito?

Segundo a pagina euler hermes:

"O seguro de crédito protege o seu negócio contra o não pagamento da dívida de transações comerciais."

Os clientes inadimplentes, identificado pela categoria 'default', tendem a criar novas dívidas, para estes clientes, deverá ser exigido o seguro de crédito. Neste dados temos poucos inadimplentes, como visto no gráfico abaixo.

```
ggplot(dataset, aes(x=default)) +
  geom_bar()
```



6. Quais são as características mais proeminentes de um cliente que possua empréstimo imobiliário?

Utilizamos o V de Cramer para calcular a relação entre as caracteristicas. Quanto mais proximo de 1, indica uma relação mais forte entre as caracteristicas observadas.

```
housing_result <- chi.squared(housing ~ ., dataset)
housing_result
```

```
##           attr_importance
## age           0.236825273
## job           0.289918963
## marital       0.042755530
## education     0.124149204
## default       0.006880645
## balance       0.136269860
## loan          0.018450768
## contact       0.218924768
## day           0.171232332
## month         0.490185378
## duration      0.000000000
## campaign      0.086134127
## pdays         0.154704795
## previous      0.000000000
## poutcome      0.135471994
## y             0.104683400
## term          0.104683400
```

Usaremos aqui a função cutoff.k que irá selecionar as 6 caracteristicas com valor de V de Cramer mais alto.

```
cutoff.k(housing_result, 6)
```

```
## [1] "month"   "job"     "age"     "contact" "day"     "pdays"
```

Descartando as caracteristicas “month”, “contact”, “day” e “pdays”, pois não são relacionadas diretamente ao perfil cliente, temos então “job” (emprego) e “age” (idade) como fatores principais que caracterizam um cliente com empréstimo imobiliário.

Predição

Usando tecnicas de aprendizado de maquina, vamos substituir os valores ‘unknown’ por valores preditos.

```
library(rpart)
library(naivebayes)
```

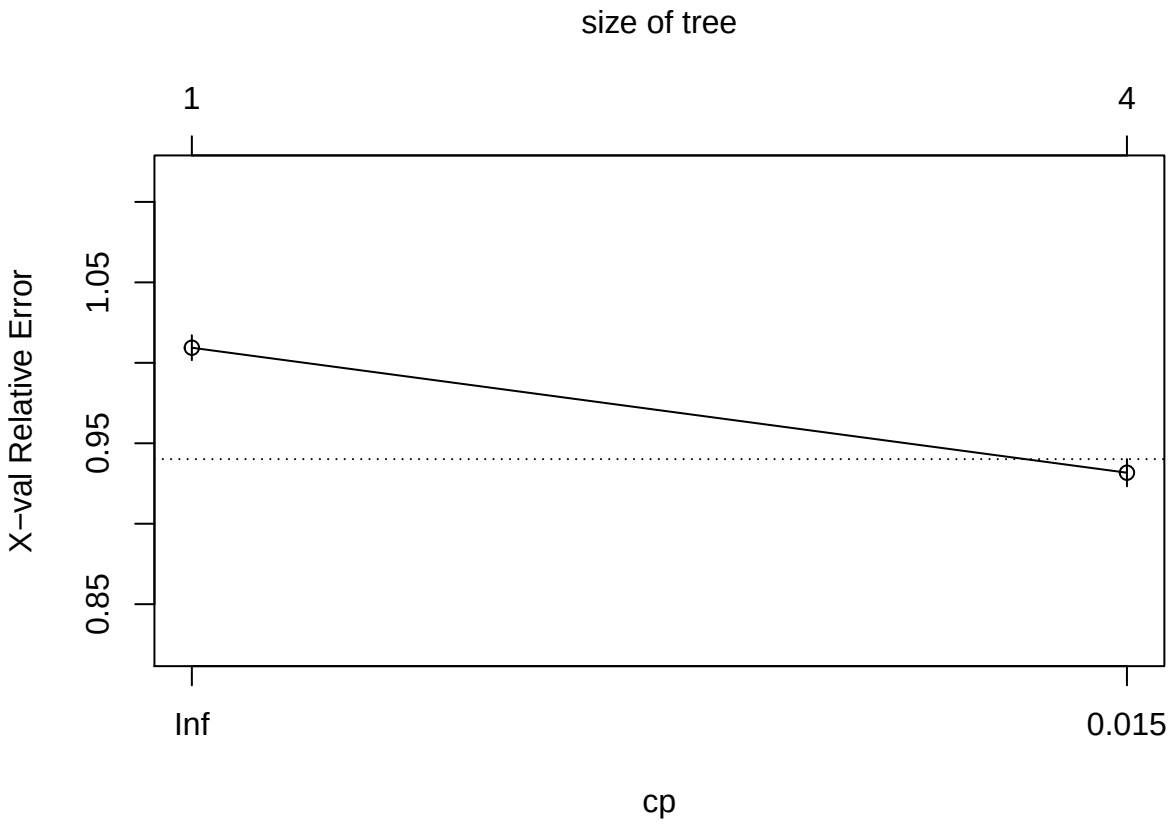
Job

```
job_unknown <- dataset$job == 'unknown'
dataset_job_real <- dataset[!job_unknown,]
dataset_job_unknown <- dataset[job_unknown,]
dataset_job_real$job <- droplevels(dataset_job_real$job)
dataset_job_unknown <- dataset_job_unknown[, colnames(dataset_job_unknown) != 'job']
```

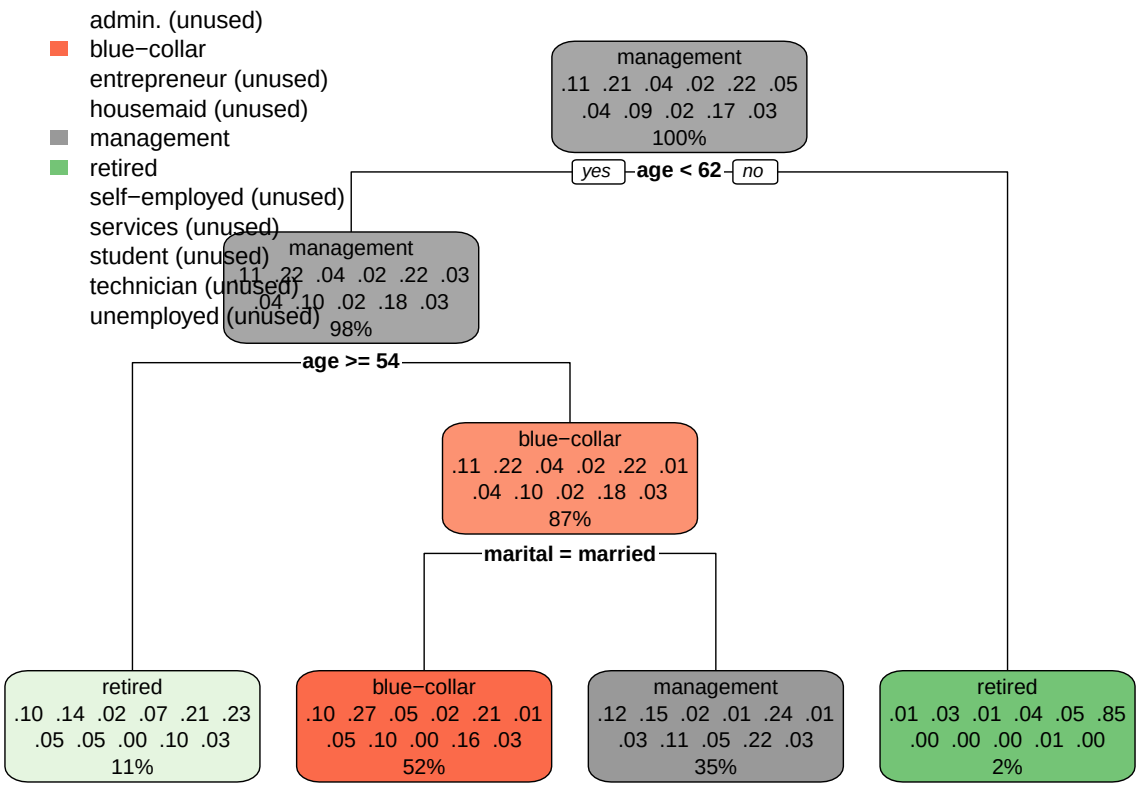
```
job_model <- rpart(job ~ age+marital, dataset_job_real, method = 'class')
```

```
job_pred <- predict(job_model, dataset_job_unknown, type='class')
```

```
plotcp(job_model)
```

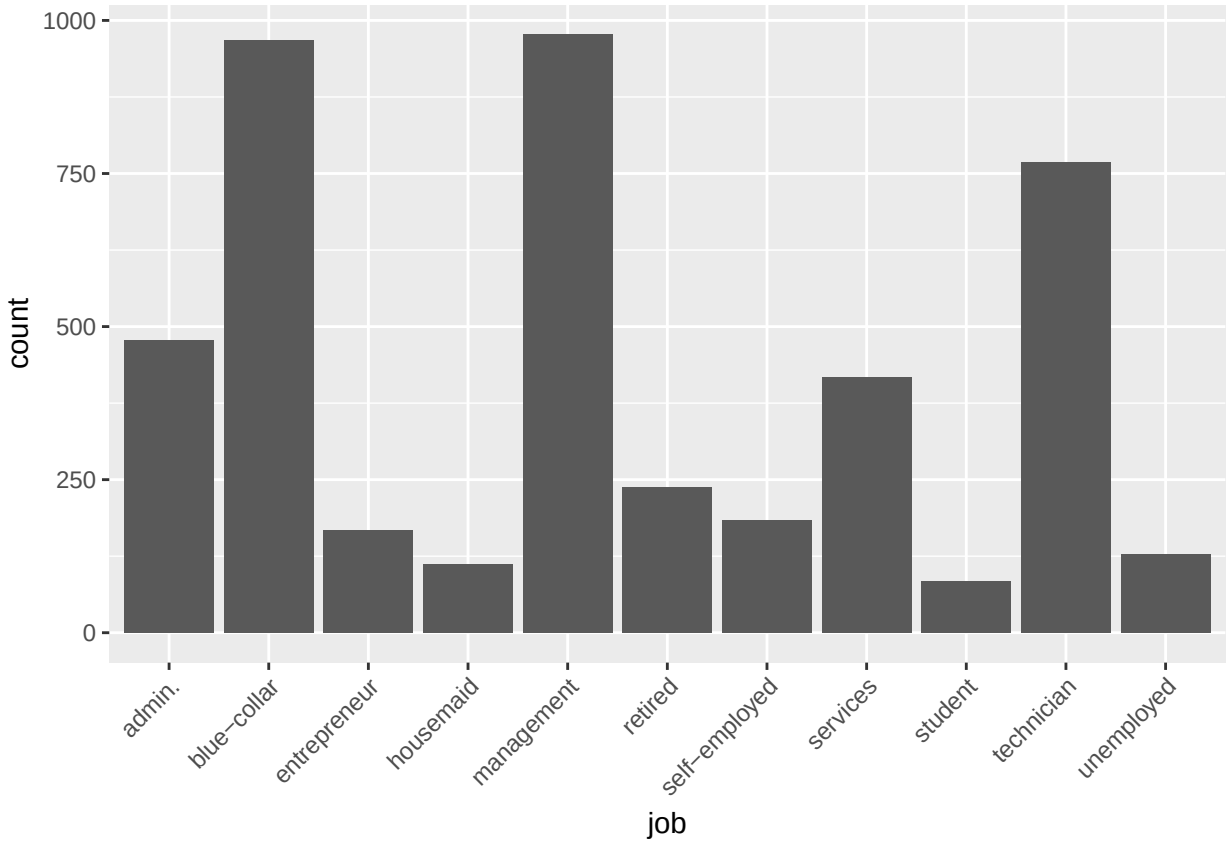



```
rpart.plot::rpart.plot(job_model)
```



Inserindo os valores preditos na caracteristica 'job'

```
dataset$job[job_unknown] <- job_pred
dataset$job <- droplevels(dataset$job)
ggplot(dataset, aes(x=job)) +
  geom_bar() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



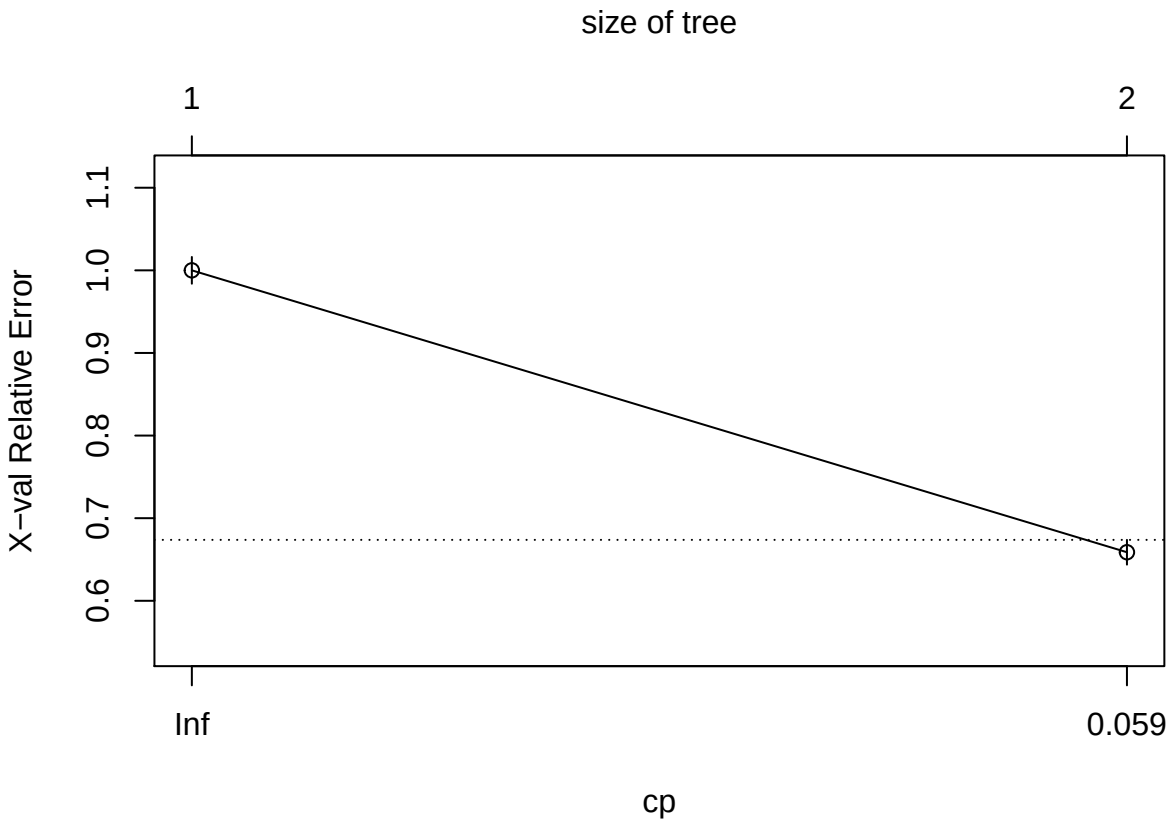
Education

```
education_unknown <- dataset$education == 'unknown'
dataset_education_real <- dataset[!education_unknown,]
dataset_education_unknown <- dataset[education_unknown,]
dataset_education_real$education <- droplevels(dataset_education_real$education)
dataset_education_unknown <- dataset_education_unknown[, colnames(dataset_education_unknown) != 'educat
```

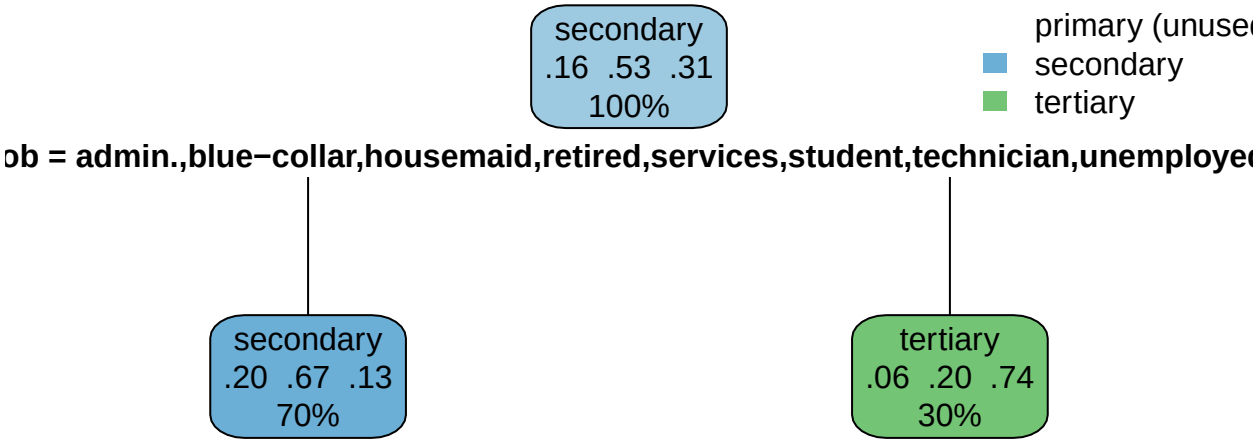
```
education_model <- rpart(education ~ age+marital+job, dataset_education_real, method = 'class')
```

```
education_pred <- predict(education_model, dataset_education_unknown, type='class')
```

```
plotcp(education_model)
```

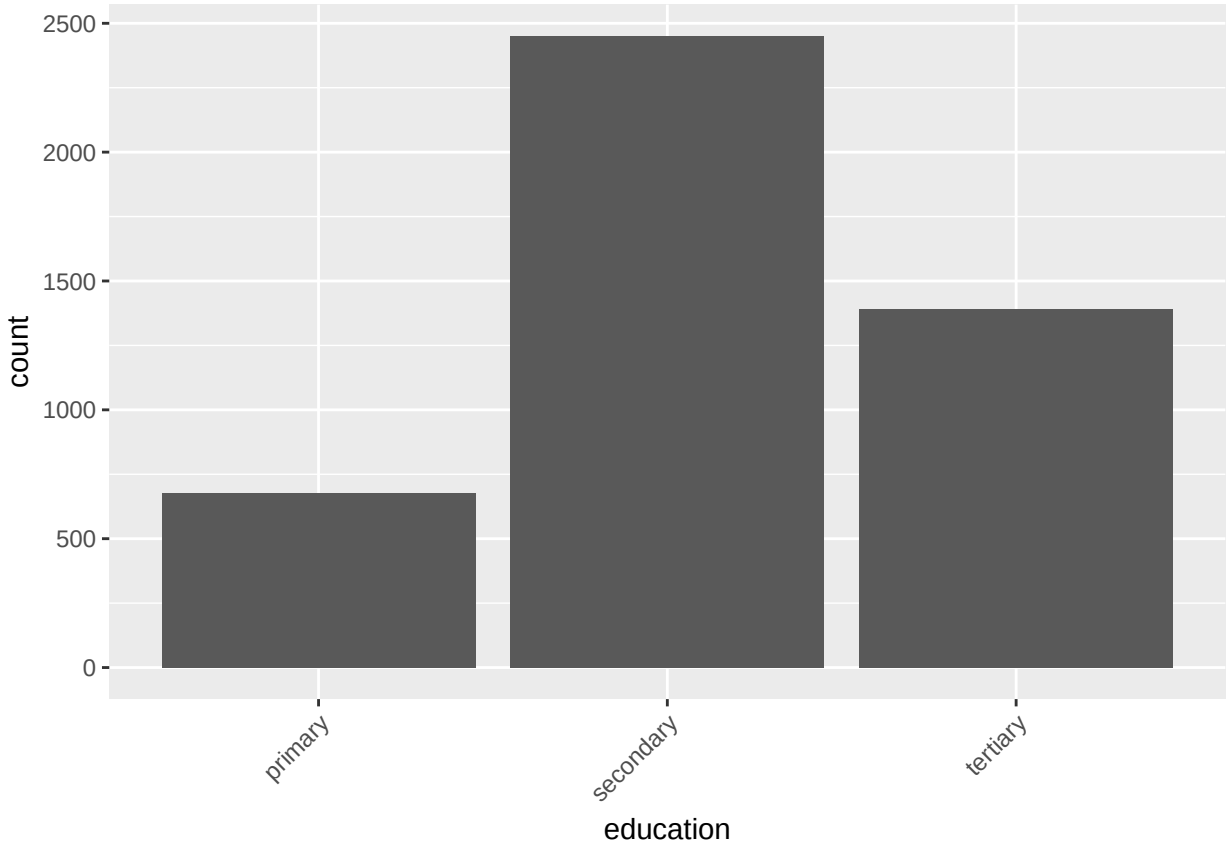



```
rpart.plot::rpart.plot(education_model)
```



Inserindo os valores preditos na caracteristica 'education'

```
dataset$education[education_unknown] <- education_pred
dataset$education <- droplevels(dataset$education)
ggplot(dataset, aes(x=education)) +
  geom_bar() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



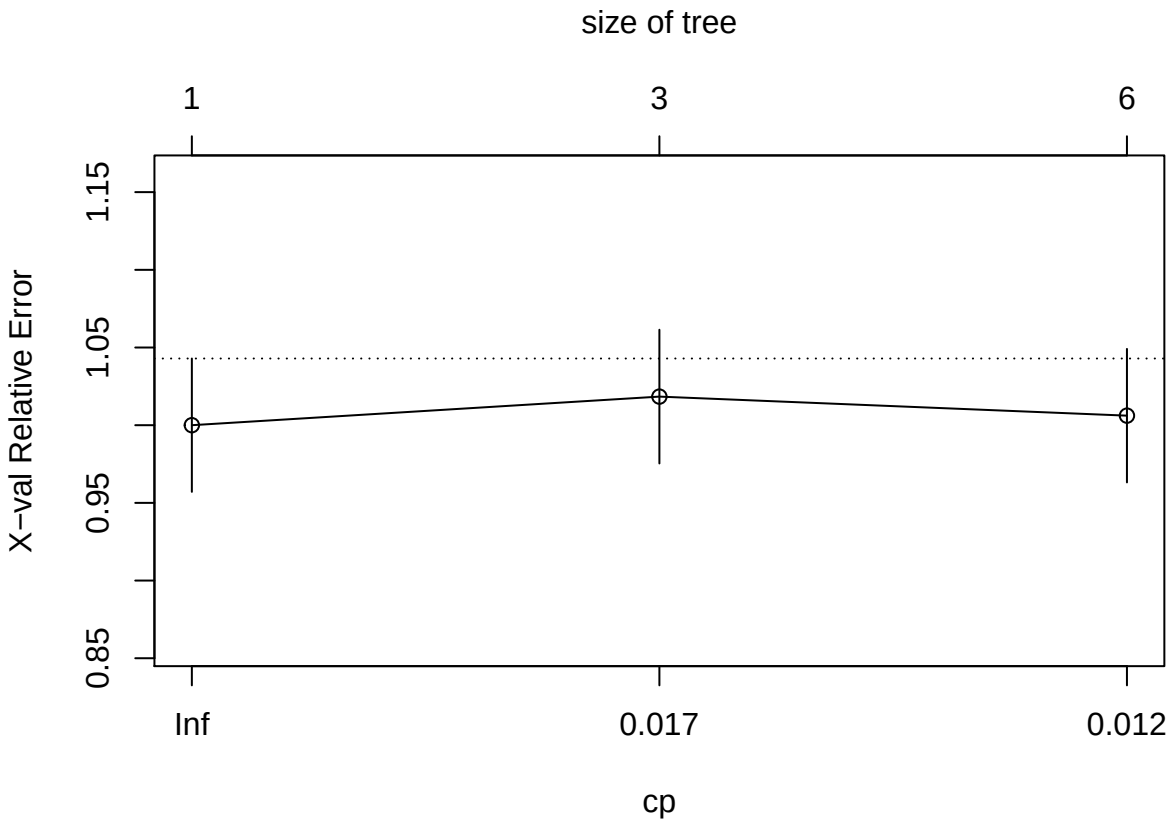
Poutcome

```
poutcome_unknown <- dataset$poutcome == 'unknown'
dataset_poutcome_real <- dataset[!poutcome_unknown,]
dataset_poutcome_unknown <- dataset[poutcome_unknown,]
dataset_poutcome_real$poutcome <- droplevels(dataset_poutcome_real$poutcome)
dataset_poutcome_unknown <- dataset_poutcome_unknown[, colnames(dataset_poutcome_unknown) != 'poutcome']
```

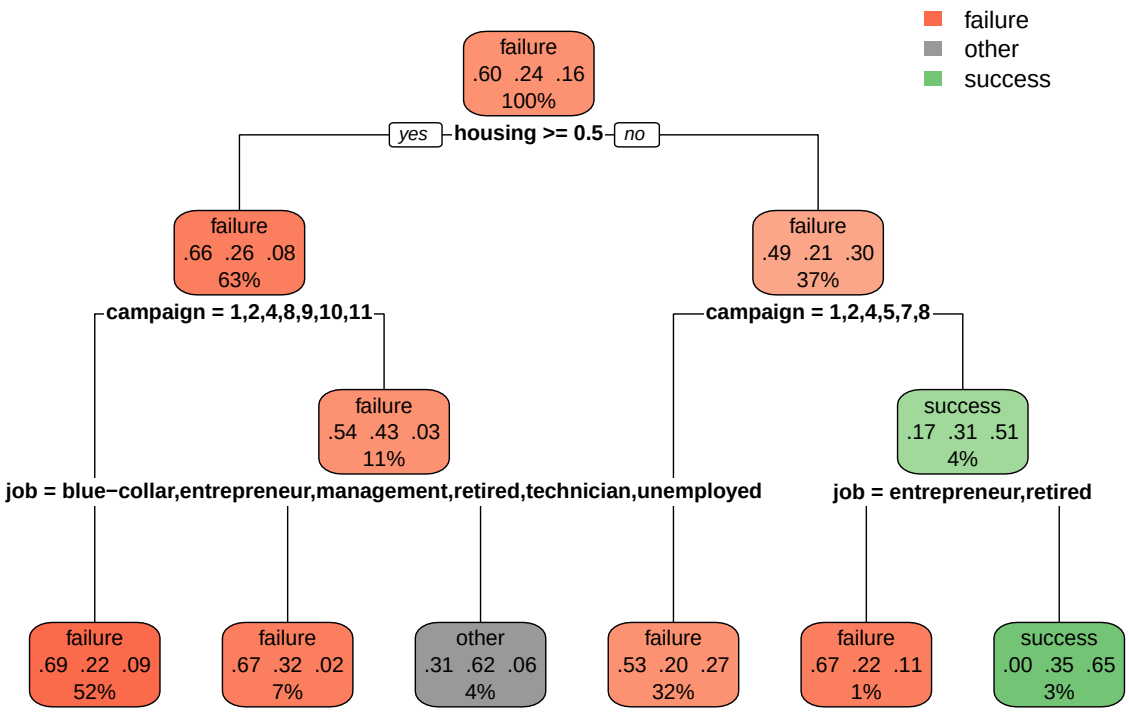
```
poutcome_model <- rpart(poutcome ~ age+marital+job+default+housing+loan+campaign, dataset_poutcome_real
```

```
poutcome_pred <- predict(poutcome_model, dataset_poutcome_unknown, type='class')
```

```
plotcp(poutcome_model)
```

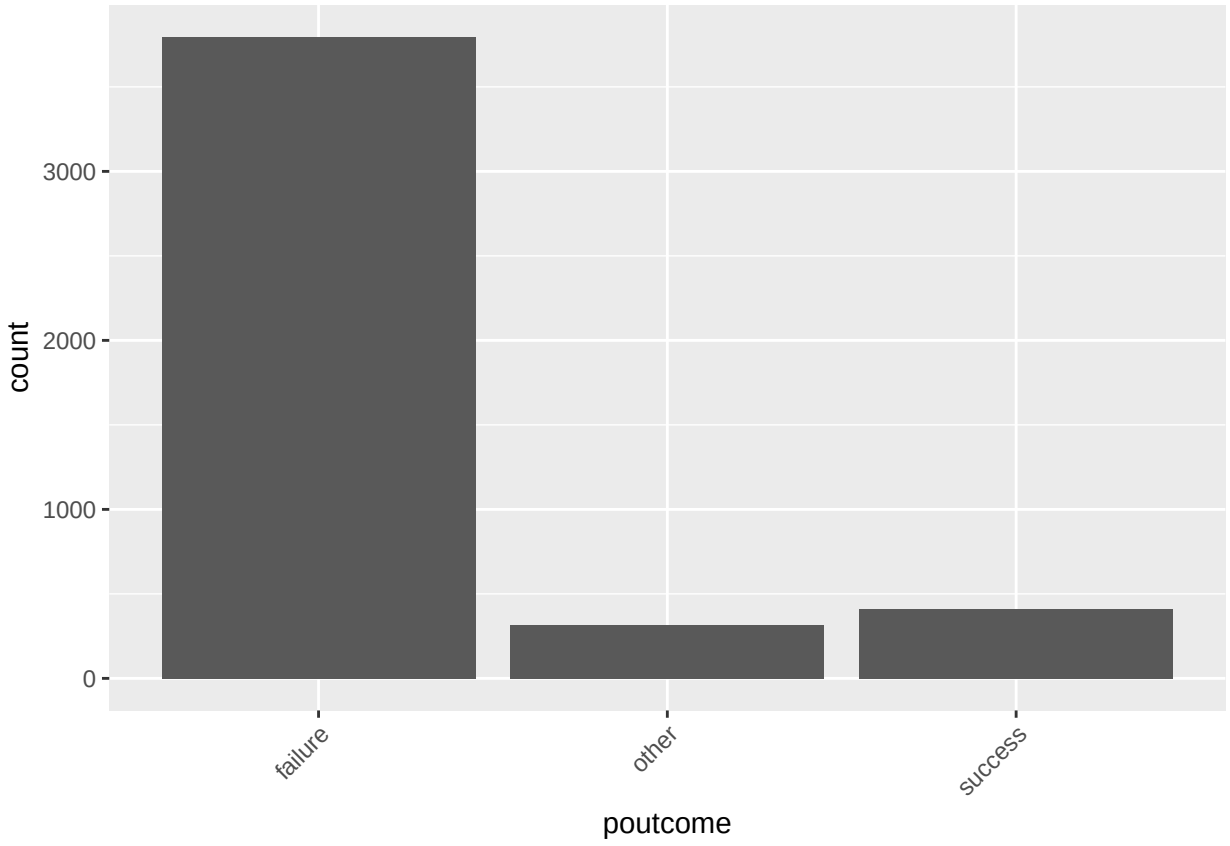



```
rpart.plot::rpart.plot(poutcome_model)
```



Inserindo os valores preditos na caracteristica 'poutcome'

```
dataset$poutcome[poutcome_unknown] <- poutcome_pred
dataset$poutcome <- droplevels(dataset$poutcome)
ggplot(dataset, aes(x=poutcome)) +
  geom_bar() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Predição de Term

Divisão em dois grupos: train and control

```
set.seed(100)
pos <- sample(1:nrow(dataset), round(nrow(dataset)*0.1, 0))
dataset_train <- dataset[-pos,]
dataset_test <- dataset[pos,]
class <- dataset_test$term
dataset_test <- dataset_test[,!colnames(dataset_test)=='term']
```

Para esta etapa utilizaremos os classificadores naive bayes e decision tree, ambos classificadores supervisionados que podem ser usados para a predição de valores categoricos.

Naive bayes

Usando todas as caracteristicas categoricas

```
# job
# marital
# education
# default
# housing
```

```r
# loan
# contact
# day
# month
# campaign
# poutcome

m <- naive_bayes(term ~ duration +
                        job +
                        marital +
                        education +
                        default +
                        housing +
                        loan +
                        contact +
                        day +
                        month +
                        campaign +
                        poutcome, dataset_train)
```

```r
pred <- predict(m, dataset_test)
```

```r
confusion_matrix <- table(class, pred)
confusion_matrix
```

```
##        pred
## class   FALSE TRUE
##   FALSE   391   15
##   TRUE     29   17
```

```r
accuracy <- (confusion_matrix[2,2] + confusion_matrix[1,1]) / sum(confusion_matrix)
recall <- confusion_matrix[2,2] / (confusion_matrix[2,2] + confusion_matrix[2,1])
precision <- confusion_matrix[2,2] / (confusion_matrix[2,2] + confusion_matrix[1,2])

accuracy
```

```
## [1] 0.9026549
```

```r
recall
```

```
## [1] 0.3695652
```

```r
precision
```

```
## [1] 0.53125
```

Decision tree

Selecionando as caracteristicas para o modelo.

```r
# job
# marital
# education
# default
# housing
# loan
# contact
```

```r
# day
# month
# campaign
# poutcome
d <- dataset[, colnames(dataset) != 'y']
weights <- chi.squared(term ~ ., d)
print(weights)
```

```
##           attr_importance
## age           0.139874536
## job           0.118974056
## marital       0.064878796
## education     0.058893492
## default       0.001302653
## balance       0.085708896
## housing       0.104683400
## loan          0.070517035
## contact       0.139412818
## day           0.141898141
## month         0.235389272
## duration      0.410052810
## campaign      0.088226286
## pdays         0.272667383
## previous      0.162037684
## poutcome      0.148414055
```

```r
cutoff.k(weights, 5)
```

```
## [1] "duration" "pdays"    "month"    "previous" "poutcome"
```

```r
m2 <- rpart(term ~ duration +
                  # job +
                  # marital +
                  # education +
                  # default +
                  # housing +
                  # loan +
                  # contact +
                  # day +
                  month +
                  pdays +
                  previous +
                  # campaign +
                  poutcome,
                  dataset_train, method = 'class')
```

```r
pred2   <- predict(m2,  dataset_test,   type    =   "class")
```

```r
confusion_matrix <- table(class, pred2)
confusion_matrix
```

```
##        pred2
## class   FALSE TRUE
##   FALSE   385   21
##   TRUE     27   19
```

```
accuracy <- (confusion_matrix[2,2] + confusion_matrix[1,1]) / sum(confusion_matrix)
recall <- confusion_matrix[2,2] / (confusion_matrix[2,2] + confusion_matrix[2,1])
precision <- confusion_matrix[2,2] / (confusion_matrix[2,2] + confusion_matrix[1,2])

accuracy
```

```
## [1] 0.8938053
```

```
recall
```

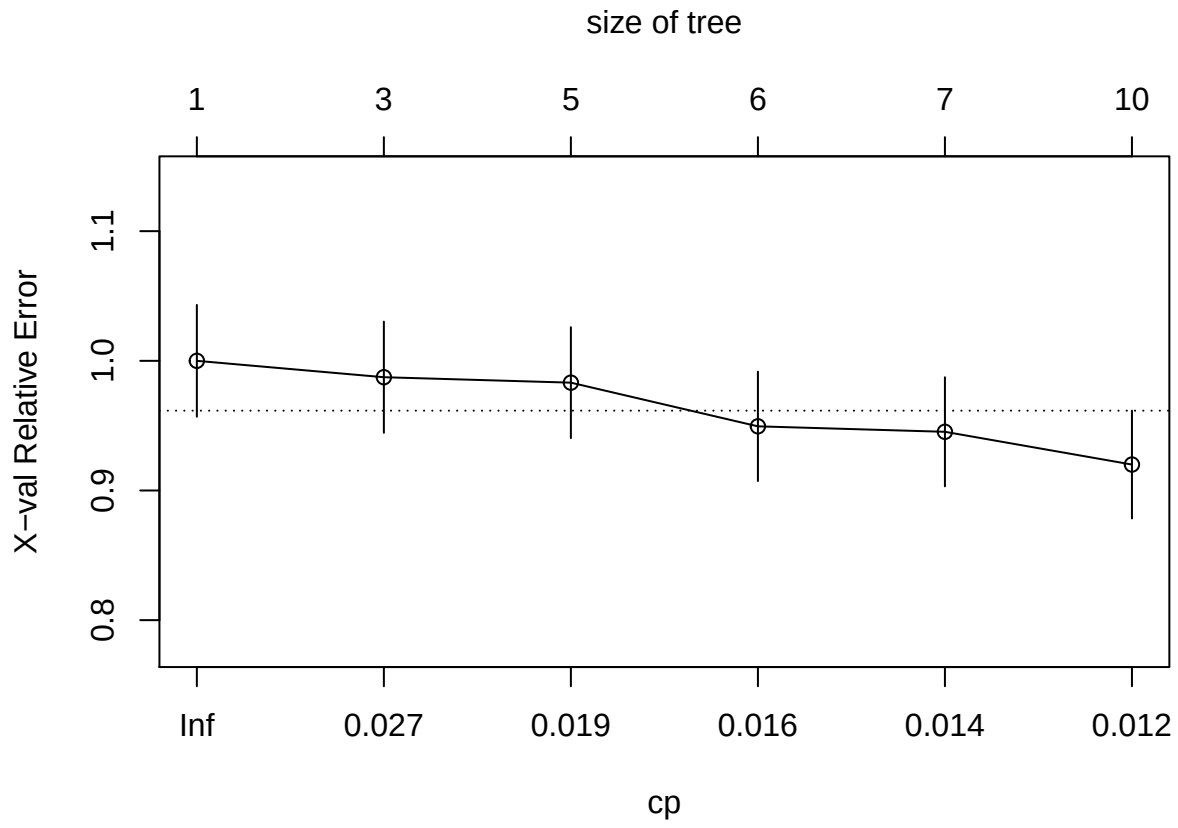
```
## [1] 0.4130435
```

```
precision
```

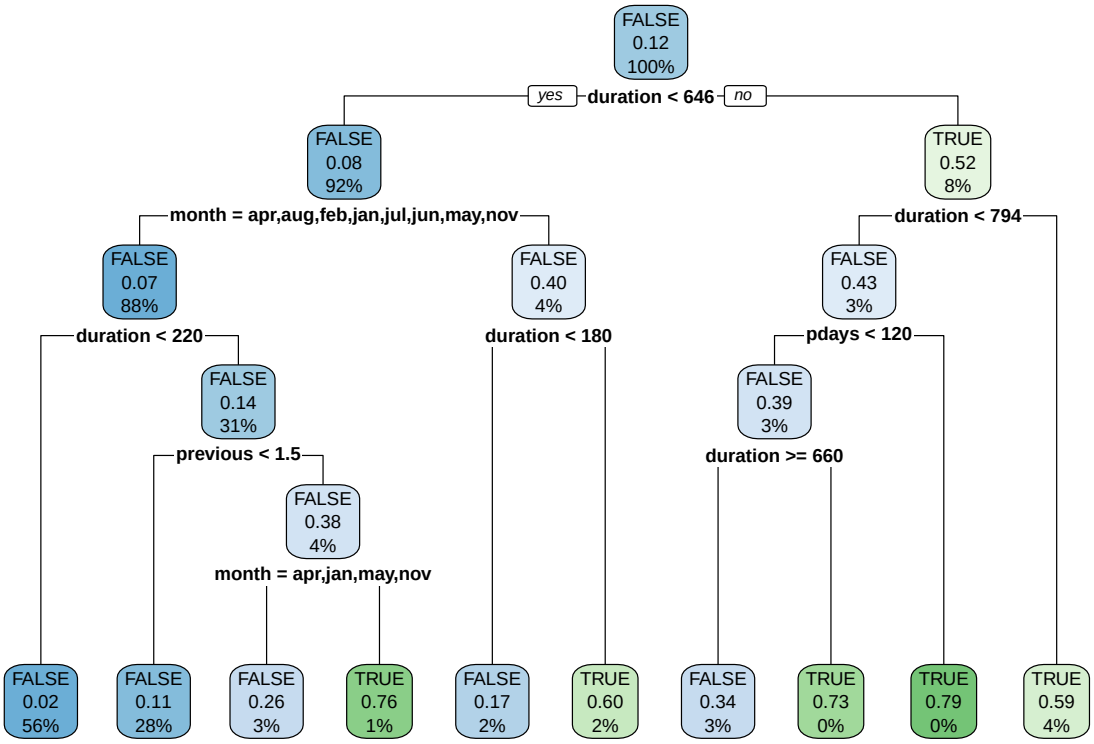
```
## [1] 0.475
```

Complexidade da arvore.

```
plotcp(m2)
```




```
rpart.plot::rpart.plot(m2)
```



Dado este resultado obtido com naive bayes e decision tree, podemos observar que o desbalanceamento da váriavel *term* (inicialmente chamada de y) causa um grande problema ao classificador. Temos uma alta acurácia, mas uma baixa cobertura com os Verdadeiro Positivos. Nos proximos passos iremos aplicar uma forma de balanceamento da classe.

Balanceamento

Abaixo iremos balancear este dado e vamos observar se temos uma melhora na predição. Podemos ver abaixo que o numero de valores FALSE é oito vezes maior que o numero de TRUE.

```
table(dataset$term)
```

```
##
## FALSE  TRUE
##  4000   521
```

Selecionaremos 500 amostras aleatórias da classe 'term' cujo resultado tenha sido FALSE.

```
set.seed(110)
term_false <- which(dataset$term == FALSE)
term_true <- which(dataset$term != FALSE)
# sortear 500 valores
select_samples <- term_false[sample(1:length(term_false), 500)]
dataset_balanced <- dataset[c(term_true, select_samples),]
```

```
table(dataset_balanced$term)
```

```
##
## FALSE  TRUE
##   500   521
```

Predição com o dado balanceado

Divisão em dois grupos: train and control

```
set.seed(110)
pos <- sample(1:nrow(dataset_balanced), round(nrow(dataset_balanced)*0.1, 0))
dataset_train <- dataset_balanced[-pos,]
dataset_test <- dataset_balanced[pos,]
class <- dataset_test$term
dataset_test <- dataset_test[,!colnames(dataset_test)=='term']
```

Usaremos somente o decision tree e o mesmo modelo anterior para esta etapa.

Decision tree

```
m3 <- rpart(term ~ duration +
                   # job +
                   # marital +
                   # education +
                   # default +
                   # housing +
                   # loan +
                   # contact +
                   # day +
                   month +
                   pdays +
                   previous +
                   # campaign +
                   poutcome,
                   dataset_train, method = 'class')
```

```
pred3   <- predict(m3,  dataset_test,   type    =   "class")
```

```
confusion_matrix <- table(class, pred3)
confusion_matrix
```

```
##        pred3
## class   FALSE TRUE
##   FALSE    35   21
##   TRUE      6   40
```

```
accuracy <- (confusion_matrix[1,1] + confusion_matrix[2,2]) / sum(confusion_matrix)
recall <- confusion_matrix[2,2] / (confusion_matrix[2,2] + confusion_matrix[2,1])
precision <- confusion_matrix[2,2] / (confusion_matrix[2,2] + confusion_matrix[1,2])

accuracy
```

```
## [1] 0.7352941
```

```
recall
```

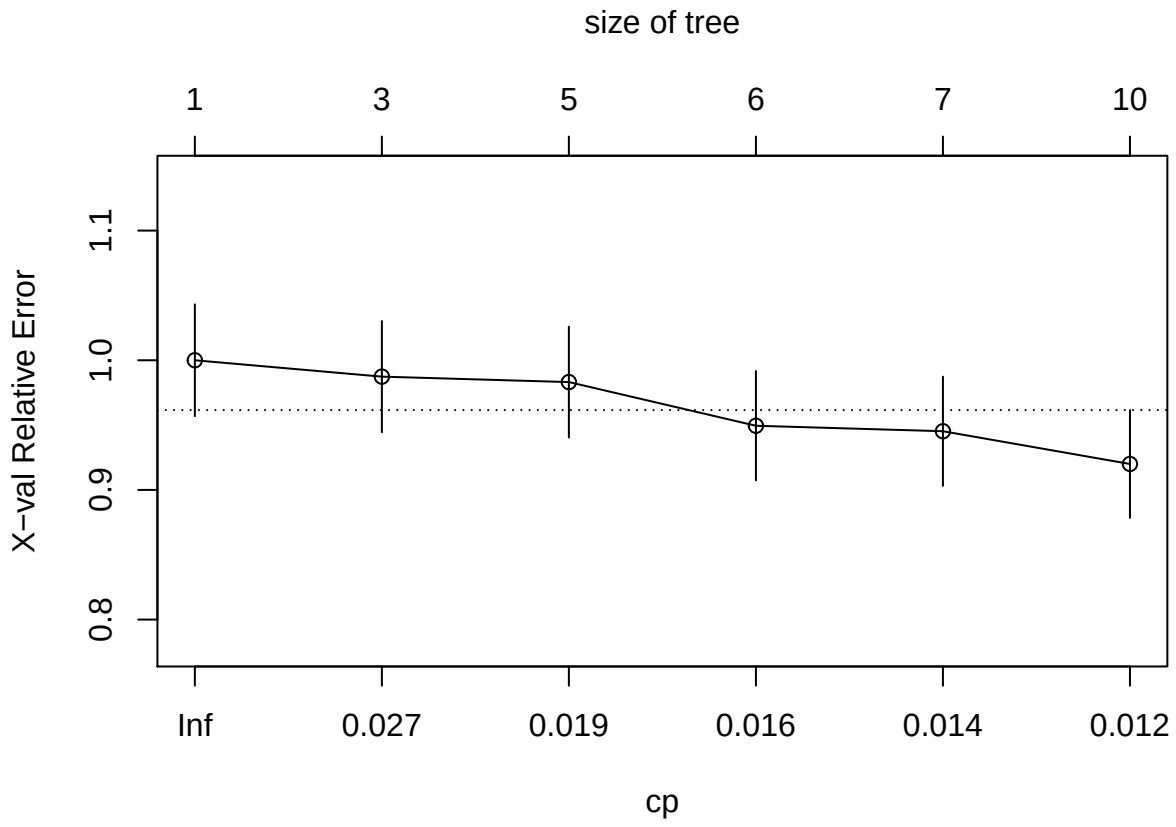
```
## [1] 0.8695652
```

```
precision
```

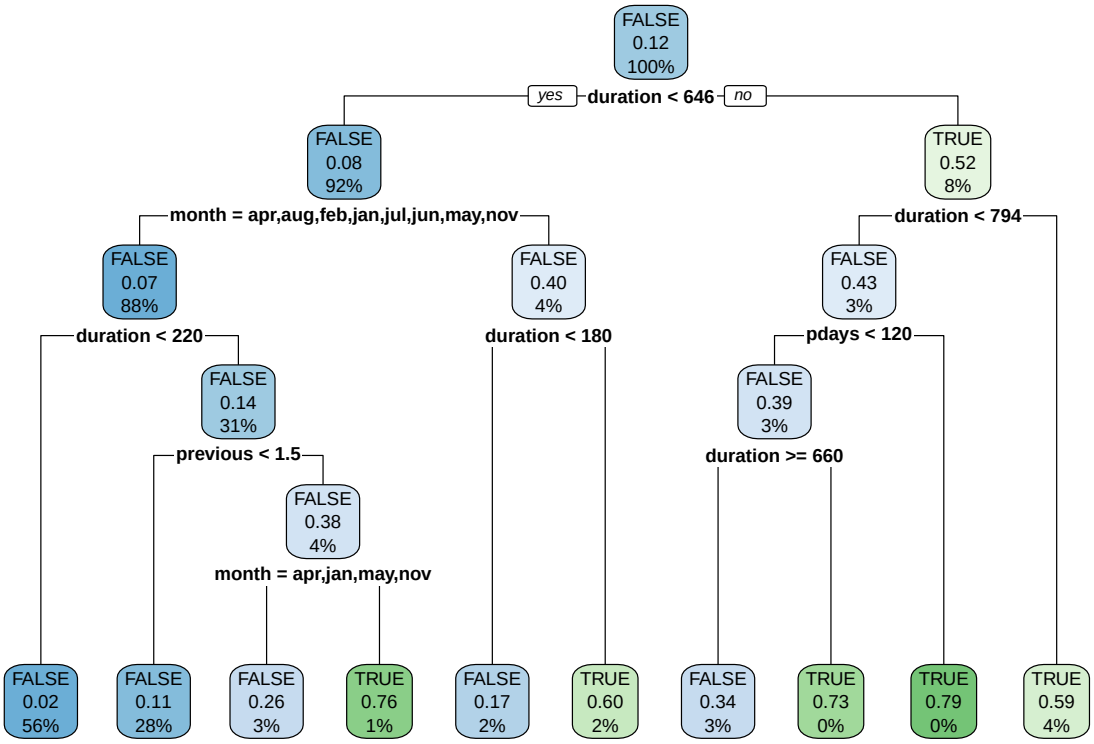
```
## [1] 0.6557377
```

Complexidade da arvore.

```
plotcp(m2)
```




```
rpart.plot::rpart.plot(m2)
```



Com o balanceamento conseguimos aumentar muito a cobertura dos Verdadeiros Positivos (0.8695652), tivemos uma queda na acurácia e um aumento na precisão. Este resultado pode ser mais refinado, melhorando a seleção de caracteristicas, melhorando o modelo e podando a arvore de decisão.