# Data Science: Capstone HarvardX - PH125.9x - CYOP - Credit risk profile predictor

Javier Dominguez Lopez

25/11/2020

## Introduction

As part of the Professional Data Science Certification Program by HarvardX, students are required to complete one last course: Data Science: Capstone HarvardX - PH125.9x which is, in reality, a capstone project designed for students to be able to put in practice all the skills learned during the 8 previous courses which comprise the Program.

The capstone is divided in two projects:

The first project consists in the design, build, training and evaluation of a Machine Learning model able to make movie recommendations to users based in existing historical data of movies' ratings by users.

The second project is a Choose-Your-Own-Project where students can decide which kind of challenge they want to deal with and which methods and algorithms they're going to use to solve the problem.

In the case of this report, the CYO project consists in training an optimal algorithm which is able to assign a credit risk profile to bank customers given certain features such as the amount and duration of the credit or the education level of the client.

This document is structured as follows:

- Introduction
- Overview
- Data Ingestion
- Data pre-processing
- Dataset exploratory analysis
- Model build, training, testing and evaluation.
- Conclusion

## Overview

As explained above, the target of this project is to build an algorithm able to assign a credit risk profile "good" or "bad" to clients of a bank based on both personal and financial attributes.

For this purpose a dataset composed of 1000 observations will be used to train, test and validate the algorithm.

The original dataset used in this project has been downloaded from

- Statlog (German Credit Data) Data Set https://archive.ics.uci.edu/ml/machine-learning-databases/statlog/german/

The algorithm will be developed taking into account the different variables and using different approaches to determine which one provides the best performance. The resulting model should be able to predict a credit risk profile (“good” or “bad”) for any new bank customer.

Accuracy will be used as the metric to evaluate algorithms’ estimations although F1-scores will also be calculated for all models.

In this project 5 models will be trained and tested, resulting in an Accuracy value which will provide an idea of how good the algorithm is at estimating credit risk profiles.

The models which will be tested are:

- Logistic regresion
- Decision tree
- Random forest
- SVM
- kNN

The model with the best Accuracy will be chosen as the optimal model for this project.

# Methods and analysis

## Data Ingestion

This is the code which takes care of data ingestion:

```
### Data ingestion ###
if(!require(tidyverse)) install.packages("tidyverse", repos = "http://cran.us.r-project.org")
if(!require(caret)) install.packages("caret", repos = "http://cran.us.r-project.org")
if(!require(data.table)) install.packages("data.table", repos = "http://cran.us.r-project.org")
if(!require(lubridate)) install.packages("lubridate")
if(!require(randomForest)) install.packages("randomForest")

library(tidyverse)
library(caret)
library(data.table)
library(lubridate)
library(randomForest)

dl <- tempfile()
download.file("https://github.com/jdominguez-github/Capstone_CYOP_German_Credit_Risk/raw/master/german.c

credit <- fread(text = gsub(" ", ",", readLines(dl)),
                 col.names = c("Checking_acc_status","Duration","Credit_history","Purpose","Credit_amoun
                               "Current_empl_dur","Installment_rate","Personal_status_Sex","Other_debtor
                               "Residence_since","Property","Age","Other_installment_plans","Housing","N
                               "Job","N_dependant_people","Telephone","Foreign","Risk"))

# Split credit data set into training set (80%) and test set (20%)
set.seed(1, sample.kind="Rounding")
test_index <- createDataPartition(credit$Risk,1,0.2,list=FALSE)

credit_test <- credit[test_index, ]
credit_train <- credit[-test_index, ]
```

By executing this code chunk we'll end up with a full data set calle "credit" which is split into a training data set called "credit_train" and a test data set called "credit_test".

## Data pre-processing

The original data set (credit) uses a codification for its variable values which makes its interpretation very hard. For this reason, a process of translation will be applied so as to create an alternative data set ("credit_trans") with more descriptive column values which will be used for exploratory analysis.

This is the original codified data description as provided in the UCI site:

### Original dataset codification

- Attribute 1: (qualitative) - Status of existing checking account
  - A11 : ... < 0 DM
  - A12 : 0 <= ... < 200 DM

  - A13 : … >= 200 DM / salary assignments for at least 1 year
  - A14 : no checking account

- Attribute 2: (numerical) - Duration in month

- Attribute 3: (qualitative) - Credit history

  - A30 : no credits taken/all credits paid back duly
  - A31 : all credits at this bank paid back duly
  - A32 : existing credits paid back duly till now
  - A33 : delay in paying off in the past
  - A34 : critical account/other credits existing (not at this bank)

- Attribute 4: (qualitative) - Purpose

  - A40 : car (new)
  - A41 : car (used)
  - A42 : furniture/equipment
  - A43 : radio/television
  - A44 : domestic appliances
  - A45 : repairs
  - A46 : education
  - A47 : (vacation - does not exist?)
  - A48 : retraining
  - A49 : business
  - A410 : others

- Attribute 5: (numerical) - Credit amount

- Attibute 6: (qualitative) - Savings account/bonds

  - A61 : … < 100 DM
  - A62 : 100 <= … < 500 DM
  - A63 : 500 <= … < 1000 DM
  - A64 : .. >= 1000 DM
  - A65 : unknown/ no savings account

- Attribute 7: (qualitative) - Present employment since

  - A71 : unemployed
  - A72 : … < 1 year
  - A73 : 1 <= … < 4 years

  - A74 : 4 <= … < 7 years
  - A75 : .. >= 7 years

- Attribute 8: (numerical) - Installment rate in percentage of disposable income

- Attribute 9: (qualitative) - Personal status and sex

  - A91 : male : divorced/separated
  - A92 : female : divorced/separated/married
  - A93 : male : single
  - A94 : male : married/widowed
  - A95 : female : single

- Attribute 10: (qualitative) - Other debtors / guarantors

  - A101 : none
  - A102 : co-applicant
  - A103 : guarantor

- Attribute 11: (numerical) - Present residence since
- Attribute 12: (qualitative) - Property
  - A121 : real estate
  - A122 : if not A121 : building society savings agreement/life insurance
  - A123 : if not A121/A122 : car or other, not in attribute 6
  - A124 : unknown / no property
- Attribute 13: (numerical) - Age in years
- Attribute 14: (qualitative) - Other installment plans
  - A141 : bank
  - A142 : stores
  - A143 : none
- Attribute 15: (qualitative) - Housing
  - A151 : rent
  - A152 : own
  - A153 : for free
- Attribute 16: (numerical) - Number of existing credits at this bank
- Attribute 17: (qualitative) - Job
  - A171 : unemployed/ unskilled - non-resident
  - A172 : unskilled - resident
  - A173 : skilled employee / official
  - A174 : management/ self-employed/ highly qualified employee/ officer
- Attribute 18: (numerical) - Number of people being liable to provide maintenance for
- Attribute 19: (qualitative) - Telephone
  - A191 : none
  - A192 : yes, registered under the customers name
- Attribute 20: (qualitative) - foreign worker
  - A201 : yes
  - A202 : no

**Feature translation process**

This is the code that carries out column codes translation:

```r
### Data pre-processing

# Translate original data set codes into something more descriptive for exploratory analysis

credit_trans <- credit

# Checking_acc_status
credit_trans[credit$Checking_acc_status == "A11"]$Checking_acc_status <- "0"
credit_trans[credit$Checking_acc_status == "A12"]$Checking_acc_status <- "0-200"
credit_trans[credit$Checking_acc_status == "A13"]$Checking_acc_status <- ">200"
credit_trans[credit$Checking_acc_status == "A14"]$Checking_acc_status <- "No account"

# Credit_history
credit_trans[credit$Credit_history == "A30"]$Credit_history <- "No credit/Duly paid at other banks"
credit_trans[credit$Credit_history == "A31"]$Credit_history <- "Duly paid at this bank"
credit_trans[credit$Credit_history == "A32"]$Credit_history <- "Existing credit duly paid at this bank"
credit_trans[credit$Credit_history == "A33"]$Credit_history <- "Payment delay in the past"
credit_trans[credit$Credit_history == "A34"]$Credit_history <- "Critical account/Credit at other banks"

# Purpose
credit_trans[credit$Purpose == "A40"]$Purpose <- "car (new)"
credit_trans[credit$Purpose == "A41"]$Purpose <- "car (used)"
credit_trans[credit$Purpose == "A42"]$Purpose <- "furniture/equipment"
credit_trans[credit$Purpose == "A43"]$Purpose <- "radio/television"
credit_trans[credit$Purpose == "A44"]$Purpose <- "domestic appliances"
credit_trans[credit$Purpose == "A45"]$Purpose <- "repairs"
credit_trans[credit$Purpose == "A46"]$Purpose <- "education"
credit_trans[credit$Purpose == "A47"]$Purpose <- "vacation"
credit_trans[credit$Purpose == "A48"]$Purpose <- "retraining"
credit_trans[credit$Purpose == "A49"]$Purpose <- "business"
credit_trans[credit$Purpose == "A410"]$Purpose <- "others"

# Savings_account
credit_trans[credit$Savings_account == "A61"]$Savings_account <- "<100"
credit_trans[credit$Savings_account == "A62"]$Savings_account <- "100-500"
credit_trans[credit$Savings_account == "A63"]$Savings_account <- "501-1000"
credit_trans[credit$Savings_account == "A64"]$Savings_account <- ">1000"
credit_trans[credit$Savings_account == "A65"]$Savings_account <- "Unknown/No account"

# Current_empl_dur
credit_trans[credit$Current_empl_dur == "A71"]$Current_empl_dur <- "Unemployed"
credit_trans[credit$Current_empl_dur == "A72"]$Current_empl_dur <- "<1y"
credit_trans[credit$Current_empl_dur == "A73"]$Current_empl_dur <- "1y-4y"
credit_trans[credit$Current_empl_dur == "A74"]$Current_empl_dur <- "4y-7y"
credit_trans[credit$Current_empl_dur == "A75"]$Current_empl_dur <- ">7y"

# Personal_status_Sex
# New Personal_status feature
credit_trans <- credit_trans %>% mutate(Personal_status="")
credit_trans <- credit_trans %>% mutate(Sex="")

credit_trans[credit$Personal_status_Sex == "A91"]$Personal_status <- "divorced/separated"
credit_trans[credit$Personal_status_Sex == "A92"]$Personal_status <- "divorced/separated/married"
```

```r
credit_trans[credit$Personal_status_Sex == "A93"]$Personal_status <- "single"
credit_trans[credit$Personal_status_Sex == "A94"]$Personal_status <- "married/widowed"
credit_trans[credit$Personal_status_Sex == "A95"]$Personal_status <- "single"
# New Sex feature
credit_trans[credit$Personal_status_Sex == "A91"]$Sex <- "male"
credit_trans[credit$Personal_status_Sex == "A92"]$Sex <- "female"
credit_trans[credit$Personal_status_Sex == "A93"]$Sex <- "male"
credit_trans[credit$Personal_status_Sex == "A94"]$Sex <- "male"
credit_trans[credit$Personal_status_Sex == "A95"]$Sex <- "female"

credit_trans <- credit_trans %>% select(-Personal_status_Sex)

# Other_debtors_guarantors
credit_trans[credit$Other_debtors_guarantors == "A101"]$Other_debtors_guarantors <- "none"
credit_trans[credit$Other_debtors_guarantors == "A102"]$Other_debtors_guarantors <- "co-applicant"
credit_trans[credit$Other_debtors_guarantors == "A103"]$Other_debtors_guarantors <- "guarantor"

# Property
credit_trans[credit$Property == "A121"]$Property <- "real estate"
credit_trans[credit$Property == "A122"]$Property <- "building society savings agreement/life insurance"
credit_trans[credit$Property == "A123"]$Property <- "car or other"
credit_trans[credit$Property == "A124"]$Property <- "unknown / no property"

# Other_installment_plans
credit_trans[credit$Other_installment_plans == "A141"]$Other_installment_plans <- "bank"
credit_trans[credit$Other_installment_plans == "A142"]$Other_installment_plans <- "stores"
credit_trans[credit$Other_installment_plans == "A143"]$Other_installment_plans <- "none"

# Housing
credit_trans[credit$Housing == "A151"]$Housing <- "rent"
credit_trans[credit$Housing == "A152"]$Housing <- "own"
credit_trans[credit$Housing == "A153"]$Housing <- "for free"

# Job
credit_trans[credit$Job == "A171"]$Job <- "unemployed/unskilled - NR"
credit_trans[credit$Job == "A172"]$Job <- "unemployed/unskilled - R"
credit_trans[credit$Job == "A173"]$Job <- "skilled/official"
credit_trans[credit$Job == "A174"]$Job <- "management/self-employed/highly qualified/officer"

# Telephone
credit_trans[credit$Telephone == "A191"]$Telephone <- "no"
credit_trans[credit$Telephone == "A192"]$Telephone <- "yes"

# Foreign
credit_trans[credit$Foreign == "A201"]$Foreign <- "yes"
credit_trans[credit$Foreign == "A202"]$Foreign <- "no"

# Risk_profile
credit_trans <- credit_trans %>% mutate(Risk_profile="")

credit_trans[credit$Risk == "1"]$Risk_profile <- "good"
credit_trans[credit$Risk == "2"]$Risk_profile <- "bad"
```

```r
# Convert categorical features into factor
credit_trans$Checking_acc_status <- factor(credit_trans$Checking_acc_status)
credit_trans$Credit_history <- factor(credit_trans$Credit_history)
credit_trans$Purpose <- factor(credit_trans$Purpose)
credit_trans$Savings_account  <- factor(credit_trans$Savings_account )
credit_trans$Current_empl_dur <- factor(credit_trans$Current_empl_dur)
credit_trans$Other_debtors_guarantors <- factor(credit_trans$Other_debtors_guarantors)
credit_trans$Property <- factor(credit_trans$Property)
credit_trans$Other_installment_plans <- factor(credit_trans$Other_installment_plans)
credit_trans$Housing <- factor(credit_trans$Housing)
credit_trans$Job <- factor(credit_trans$Job)
credit_trans$Telephone <- factor(credit_trans$Telephone)
credit_trans$Foreign <- factor(credit_trans$Foreign)
credit_trans$Personal_status <- factor(credit_trans$Personal_status)
credit_trans$Sex <- factor(credit_trans$Sex)

# Convert outcome features into factor
credit_trans$Risk_profile = factor(credit_trans$Risk_profile)
credit$Risk = factor(credit$Risk)
credit_test$Risk = factor(credit_test$Risk)
credit_train$Risk = factor(credit_train$Risk)
```

The original codified data set "credit" will be used for model implementation and the translated data set "credit_trans" will be used for exploratory analysis.

## Data summary

With the initial loading and pre-processing of data complete, we can now take a look at the basic structure and stats of the data.

When looking at the general structure of the original data set we can see it consists of 1,000 observations and 21 variables of which, "Risk", is the outcome and the rest are features which can be used as potential predictors.

Risk is a factor variable with 2 levels: "good" and "bad" representing the two possible credit risk profiles that can be assigned to a client and that will be predicted by the algorithm.

We can see the summary statistics as well as a sample displaying the first 10 rows of both the original credit data set and the translated "credit_trans" data set which has the same amount of rows as it's only a translated version of the original one.

### Data overview: Original "credit" data set

```r
# Overview of original credit dataset
# Total number of rows
c("Number of users" = nrow(credit))
```

```
## Number of users
##            1000
```

```r
# General structure
str(credit)
```

```
## Classes 'data.table' and 'data.frame':   1000 obs. of  21 variables:
##  $ Checking_acc_status     : chr  "A11" "A12" "A14" "A11" ...
##  $ Duration                : int  6 48 12 42 24 36 24 36 12 30 ...
##  $ Credit_history          : chr  "A34" "A32" "A34" "A32" ...
##  $ Purpose                 : chr  "A43" "A43" "A46" "A42" ...
##  $ Credit_amount           : int  1169 5951 2096 7882 4870 9055 2835 6948 3059 5234 ...
##  $ Savings_account         : chr  "A65" "A61" "A61" "A61" ...
##  $ Current_empl_dur        : chr  "A75" "A73" "A74" "A74" ...
##  $ Installment_rate        : int  4 2 2 2 3 2 3 2 2 4 ...
##  $ Personal_status_Sex     : chr  "A93" "A92" "A93" "A93" ...
##  $ Other_debtors_guarantors: chr  "A101" "A101" "A101" "A103" ...
##  $ Residence_since         : int  4 2 3 4 4 4 4 2 4 2 ...
##  $ Property                : chr  "A121" "A121" "A121" "A122" ...
##  $ Age                     : int  67 22 49 45 53 35 53 35 61 28 ...
##  $ Other_installment_plans : chr  "A143" "A143" "A143" "A143" ...
##  $ Housing                 : chr  "A152" "A152" "A152" "A153" ...
##  $ N_credits               : int  2 1 1 1 2 1 1 1 1 2 ...
##  $ Job                     : chr  "A173" "A173" "A172" "A173" ...
##  $ N_dependant_people      : int  1 1 2 2 2 2 1 1 1 1 ...
##  $ Telephone               : chr  "A192" "A191" "A191" "A191" ...
##  $ Foreign                 : chr  "A201" "A201" "A201" "A201" ...
##  $ Risk                    : Factor w/ 2 levels "1","2": 1 2 1 1 2 1 1 1 1 2 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```r
# Summary stats
summary(credit)
```

```
##  Checking_acc_status    Duration     Credit_history       Purpose
##  Length:1000         Min.   : 4.0   Length:1000        Length:1000
##  Class :character    1st Qu.:12.0   Class :character   Class :character
##  Mode  :character    Median :18.0   Mode  :character   Mode  :character
##                      Mean   :20.9
##                      3rd Qu.:24.0
##                      Max.   :72.0
##  Credit_amount   Savings_account    Current_empl_dur   Installment_rate
##  Min.   :  250   Length:1000        Length:1000        Min.   :1.000
##  1st Qu.: 1366   Class :character   Class :character   1st Qu.:2.000
##  Median : 2320   Mode  :character   Mode  :character   Median :3.000
##  Mean   : 3271                                         Mean   :2.973
##  3rd Qu.: 3972                                         3rd Qu.:4.000
##  Max.   :18424                                         Max.   :4.000
##  Personal_status_Sex Other_debtors_guarantors Residence_since
##  Length:1000         Length:1000              Min.   :1.000
##  Class :character    Class :character         1st Qu.:2.000
##  Mode  :character    Mode  :character         Median :3.000
##                                               Mean   :2.845
##                                               3rd Qu.:4.000
##                                               Max.   :4.000
##    Property               Age        Other_installment_plans   Housing
##  Length:1000         Min.   :19.00   Length:1000             Length:1000
```

```
##  Class :character   1st Qu.:27.00   Class :character        Class :character
##  Mode  :character   Median :33.00   Mode  :character        Mode  :character
##                     Mean   :35.55
##                     3rd Qu.:42.00
##                     Max.   :75.00
##    N_credits         Job              N_dependant_people  Telephone
##  Min.   :1.000   Length:1000        Min.   :1.000      Length:1000
##  1st Qu.:1.000   Class :character   1st Qu.:1.000      Class :character
##  Median :1.000   Mode  :character   Median :1.000      Mode  :character
##  Mean   :1.407                      Mean   :1.155
##  3rd Qu.:2.000                      3rd Qu.:1.000
##  Max.   :4.000                      Max.   :2.000
##    Foreign            Risk
##  Length:1000        1:700
##  Class :character   2:300
##  Mode  :character
##
##
##
```

```
# Sample of first few rows
head(credit)
```

```
##    Checking_acc_status Duration Credit_history Purpose Credit_amount
## 1:                 A11        6            A34     A43          1169
## 2:                 A12       48            A32     A43          5951
## 3:                 A14       12            A34     A46          2096
## 4:                 A11       42            A32     A42          7882
## 5:                 A11       24            A33     A40          4870
## 6:                 A14       36            A32     A46          9055
##    Savings_account Current_empl_dur Installment_rate Personal_status_Sex
## 1:             A65              A75                4                 A93
## 2:             A61              A73                2                 A92
## 3:             A61              A74                2                 A93
## 4:             A61              A74                2                 A93
## 5:             A61              A73                3                 A93
## 6:             A65              A73                2                 A93
##    Other_debtors_guarantors Residence_since Property Age
## 1:                     A101               4     A121  67
## 2:                     A101               2     A121  22
## 3:                     A101               3     A121  49
## 4:                     A103               4     A122  45
## 5:                     A101               4     A124  53
## 6:                     A101               4     A124  35
##    Other_installment_plans Housing N_credits  Job N_dependant_people Telephone
## 1:                    A143    A152         2 A173                  1      A192
## 2:                    A143    A152         1 A173                  1      A191
## 3:                    A143    A152         1 A172                  2      A191
## 4:                    A143    A153         1 A173                  2      A191
## 5:                    A143    A153         2 A173                  2      A191
## 6:                    A143    A153         1 A172                  2      A192
##    Foreign Risk
## 1:    A201    1
## 2:    A201    2
```

```
## 3:    A201    1
## 4:    A201    1
## 5:    A201    2
## 6:    A201    1
```

**Data overview: Translated "credit_trans" data set**

```r
# Overview of the dataset with translated variables
# General structure
str(credit_trans)
```

```
## Classes 'data.table' and 'data.frame':   1000 obs. of  23 variables:
##  $ Checking_acc_status     : Factor w/ 4 levels ">200","0","0-200",..: 2 3 4 2 2 4 4 3 4 3 ...
##  $ Duration                : int  6 48 12 42 24 36 24 36 12 30 ...
##  $ Credit_history          : Factor w/ 5 levels "Critical account/Credit at other banks",..: 1 3 1 3
##  $ Purpose                 : Factor w/ 10 levels "business","car (new)",..: 8 8 5 6 2 5 6 3 8 2 ...
##  $ Credit_amount           : int  1169 5951 2096 7882 4870 9055 2835 6948 3059 5234 ...
##  $ Savings_account         : Factor w/ 5 levels "<100",">1000",..: 5 1 1 1 1 5 4 1 2 1 ...
##  $ Current_empl_dur        : Factor w/ 5 levels "<1y",">7y","1y-4y",..: 2 3 4 4 3 3 2 3 4 5 ...
##  $ Installment_rate        : int  4 2 2 2 3 2 3 2 2 4 ...
##  $ Other_debtors_guarantors: Factor w/ 3 levels "co-applicant",..: 3 3 3 2 3 3 3 3 3 3 ...
##  $ Residence_since         : int  4 2 3 4 4 4 4 2 4 2 ...
##  $ Property                : Factor w/ 4 levels "building society savings agreement/life insurance",
##  $ Age                     : int  67 22 49 45 53 35 53 35 61 28 ...
##  $ Other_installment_plans : Factor w/ 3 levels "bank","none",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ Housing                 : Factor w/ 3 levels "for free","own",..: 2 2 2 1 1 1 2 3 2 2 ...
##  $ N_credits               : int  2 1 1 1 2 1 1 1 1 2 ...
##  $ Job                     : Factor w/ 4 levels "management/self-employed/highly qualified/officer",
##  $ N_dependant_people      : int  1 1 2 2 2 2 1 1 1 1 ...
##  $ Telephone               : Factor w/ 2 levels "no","yes": 2 1 1 1 1 2 1 2 1 1 ...
##  $ Foreign                 : Factor w/ 2 levels "no","yes": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Risk                    : int  1 2 1 1 2 1 1 1 1 2 ...
##  $ Personal_status         : Factor w/ 4 levels "divorced/separated",..: 4 2 4 4 4 4 4 4 1 3 ...
##  $ Sex                     : Factor w/ 2 levels "female","male": 2 1 2 2 2 2 2 2 2 2 ...
##  $ Risk_profile            : Factor w/ 2 levels "bad","good": 2 1 2 2 1 2 2 2 2 1 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```r
# Summary stats
summary(credit_trans)
```

```
##  Checking_acc_status    Duration
##  >200       : 63      Min.   : 4.0
##  0          :274      1st Qu.:12.0
##  0-200      :269      Median :18.0
##  No account:394       Mean   :20.9
##                       3rd Qu.:24.0
##                       Max.   :72.0
##
##                               Credit_history                Purpose
##  Critical account/Credit at other banks:293    radio/television   :280
##  Duly paid at this bank                : 49    car (new)          :234
```

```
##  Existing credit duly paid at this bank:530    furniture/equipment:181
##  No credit/Duly paid at other banks    : 40    car (used)         :103
##  Payment delay in the past             : 88    business           : 97
##                                                education          : 50
##                                                (Other)            : 55
##  Credit_amount             Savings_account   Current_empl_dur Installment_rate
##  Min.   :  250   <100              :603    <1y       :172     Min.   :1.000
##  1st Qu.: 1366   >1000             : 48    >7y       :253     1st Qu.:2.000
##  Median : 2320   100-500           :103    1y-4y     :339     Median :3.000
##  Mean   : 3271   501-1000          : 63    4y-7y     :174     Mean   :2.973
##  3rd Qu.: 3972   Unknown/No account:183    Unemployed: 62     3rd Qu.:4.000
##  Max.   :18424                                                Max.   :4.000
##
##  Other_debtors_guarantors Residence_since
##  co-applicant: 41         Min.   :1.000
##  guarantor   : 52         1st Qu.:2.000
##  none        :907         Median :3.000
##                           Mean   :2.845
##                           3rd Qu.:4.000
##                           Max.   :4.000
##
##                                              Property        Age
##  building society savings agreement/life insurance:232   Min.   :19.00
##  car or other                                     :332   1st Qu.:27.00
##  real estate                                      :282   Median :33.00
##  unknown / no property                            :154   Mean   :35.55
##                                                          3rd Qu.:42.00
##                                                          Max.   :75.00
##
##  Other_installment_plans     Housing      N_credits
##  bank  :139              for free:108   Min.   :1.000
##  none  :814              own     :713   1st Qu.:1.000
##  stores: 47              rent    :179   Median :1.000
##                                         Mean   :1.407
##                                         3rd Qu.:2.000
##                                         Max.   :4.000
##
##                                                Job      N_dependant_people
##  management/self-employed/highly qualified/officer:148   Min.   :1.000
##  skilled/official                                 :630   1st Qu.:1.000
##  unemployed/unskilled - NR                        : 22   Median :1.000
##  unemployed/unskilled - R                         :200   Mean   :1.155
##                                                          3rd Qu.:1.000
##                                                          Max.   :2.000
##
##  Telephone Foreign        Risk                        Personal_status
##  no :596   no : 37   Min.   :1.0   divorced/separated        : 50
##  yes:404   yes:963   1st Qu.:1.0   divorced/separated/married:310
##                      Median :1.0   married/widowed           : 92
##                      Mean   :1.3   single                    :548
##                      3rd Qu.:2.0
##                      Max.   :2.0
##
##     Sex      Risk_profile
```

```
##  female:310   bad :300
##  male  :690   good:700
##
##
##
##
##
```

```r
# Sample of first few rows
head(credit_trans)
```

```
##    Checking_acc_status Duration                         Credit_history
## 1:                   0        6 Critical account/Credit at other banks
## 2:               0-200       48 Existing credit duly paid at this bank
## 3:          No account       12 Critical account/Credit at other banks
## 4:                   0       42 Existing credit duly paid at this bank
## 5:                   0       24                Payment delay in the past
## 6:          No account       36 Existing credit duly paid at this bank
##                Purpose Credit_amount    Savings_account Current_empl_dur
## 1:    radio/television          1169 Unknown/No account              >7y
## 2:    radio/television          5951               <100            1y-4y
## 3:           education          2096               <100            4y-7y
## 4: furniture/equipment          7882               <100            4y-7y
## 5:           car (new)          4870               <100            1y-4y
## 6:           education          9055 Unknown/No account            1y-4y
##    Installment_rate Other_debtors_guarantors Residence_since
## 1:                4                     none               4
## 2:                2                     none               2
## 3:                2                     none               3
## 4:                2                guarantor               4
## 5:                3                     none               4
## 6:                2                     none               4
##                                         Property Age
## 1:                                   real estate  67
## 2:                                   real estate  22
## 3:                                   real estate  49
## 4: building society savings agreement/life insurance  45
## 5:                            unknown / no property  53
## 6:                            unknown / no property  35
##    Other_installment_plans  Housing N_credits                         Job
## 1:                    none      own         2         skilled/official
## 2:                    none      own         1         skilled/official
## 3:                    none      own         1 unemployed/unskilled - R
## 4:                    none for free         1         skilled/official
## 5:                    none for free         2         skilled/official
## 6:                    none for free         1 unemployed/unskilled - R
##    N_dependant_people Telephone Foreign Risk            Personal_status    Sex
## 1:                  1       yes     yes    1                     single   male
## 2:                  1        no     yes    2 divorced/separated/married female
## 3:                  2        no     yes    1                     single   male
## 4:                  2        no     yes    1                     single   male
## 5:                  2        no     yes    2                     single   male
## 6:                  2       yes     yes    1                     single   male
##    Risk_profile
```

```
## 1:          good
## 2:           bad
## 3:          good
## 4:          good
## 5:           bad
## 6:          good
```

## Data exploratory analysis

### General information

Number of credit risk profiles categorized as "good" / "bad"

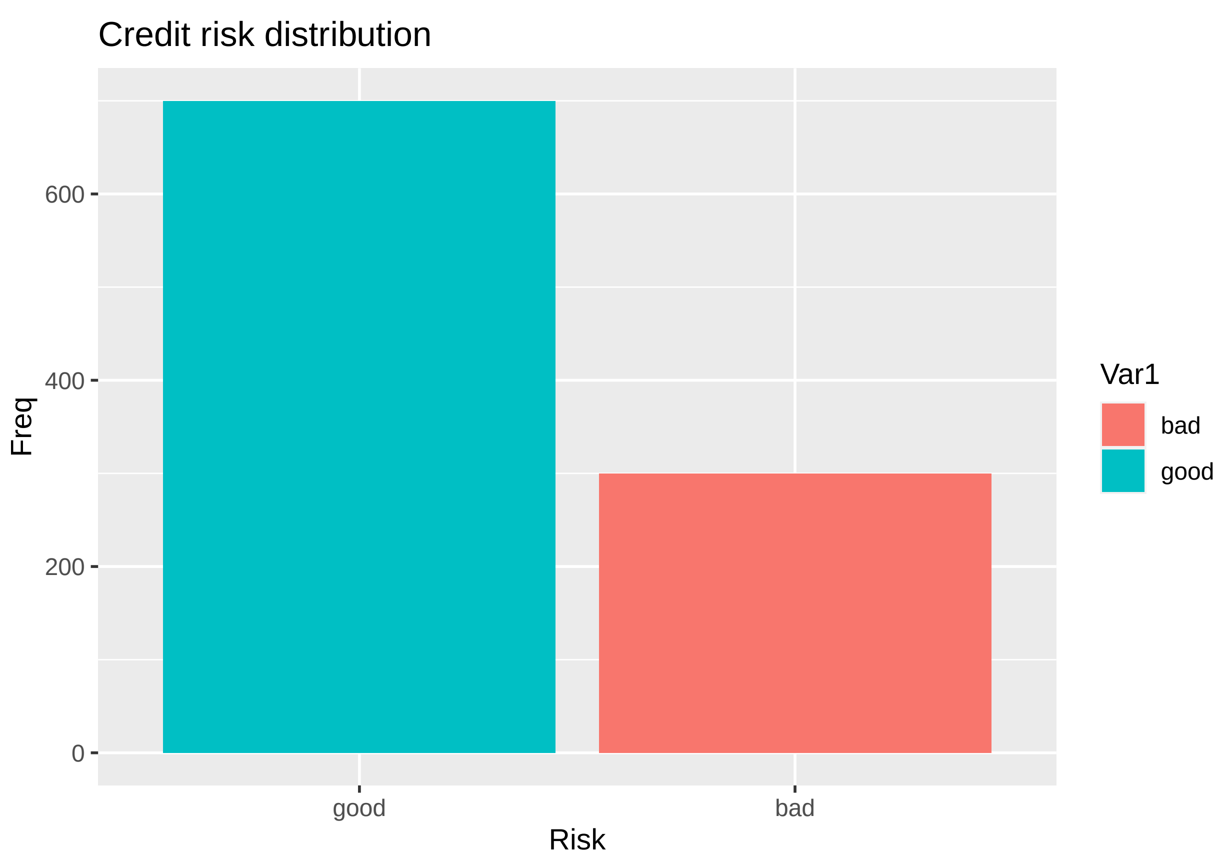
```
##
##  bad good
##  300  700
```

Proportion of credit risk profiles categorized as "good"

```
## Proportion of "good" credit risk profiles
##                                       0.7
```

70% of the profiles have been categorized as "good" and 30% have been categorized as "bad". To visualize this, here's a plot of the proportion of good vs bad profiles

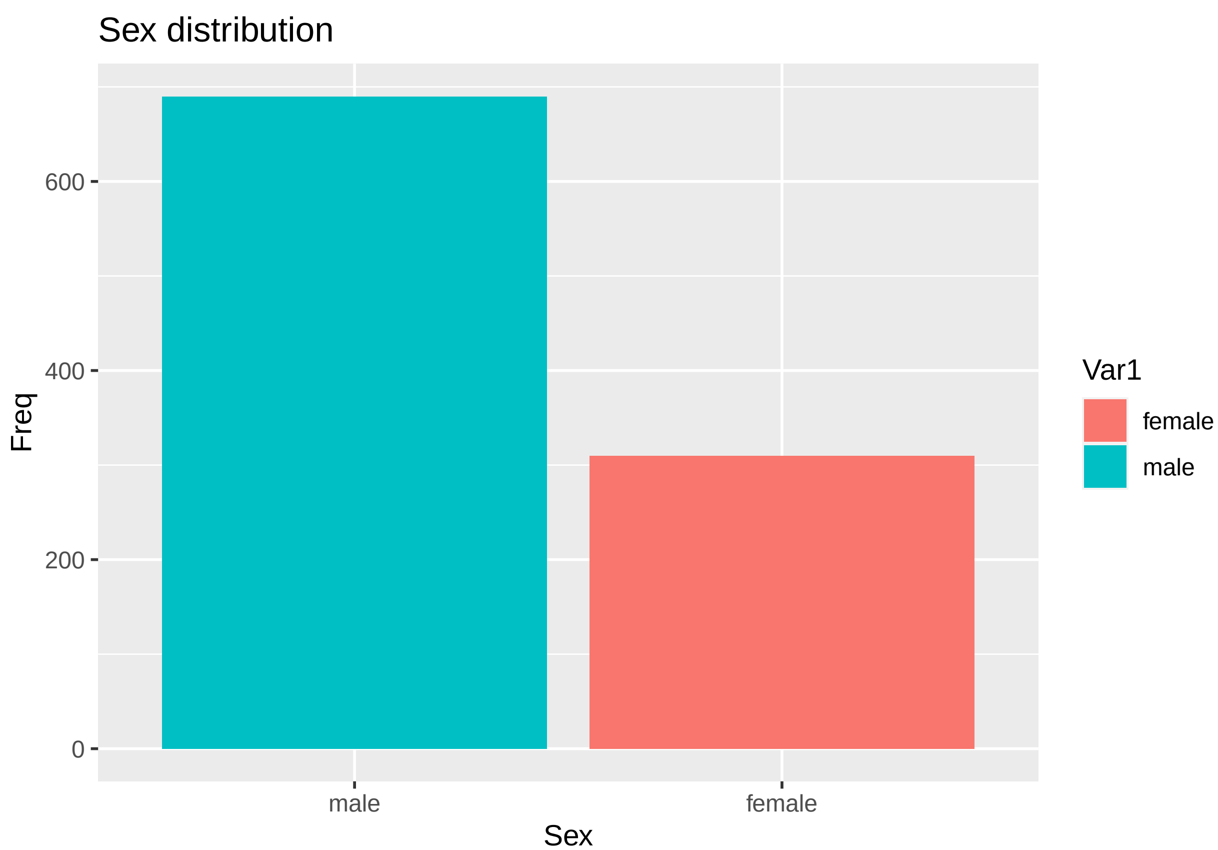
```
# Credit risk distribution
cont_succ <- table(credit_trans$Risk_profile)
data.frame(cont_succ) %>%
  ggplot(aes(x= reorder(Var1,-Freq),Freq,fill=Var1)) +
  geom_bar(stat ="identity") +
  xlab("Risk") +
  ggtitle("Credit risk distribution")
```



Number of male/female customers

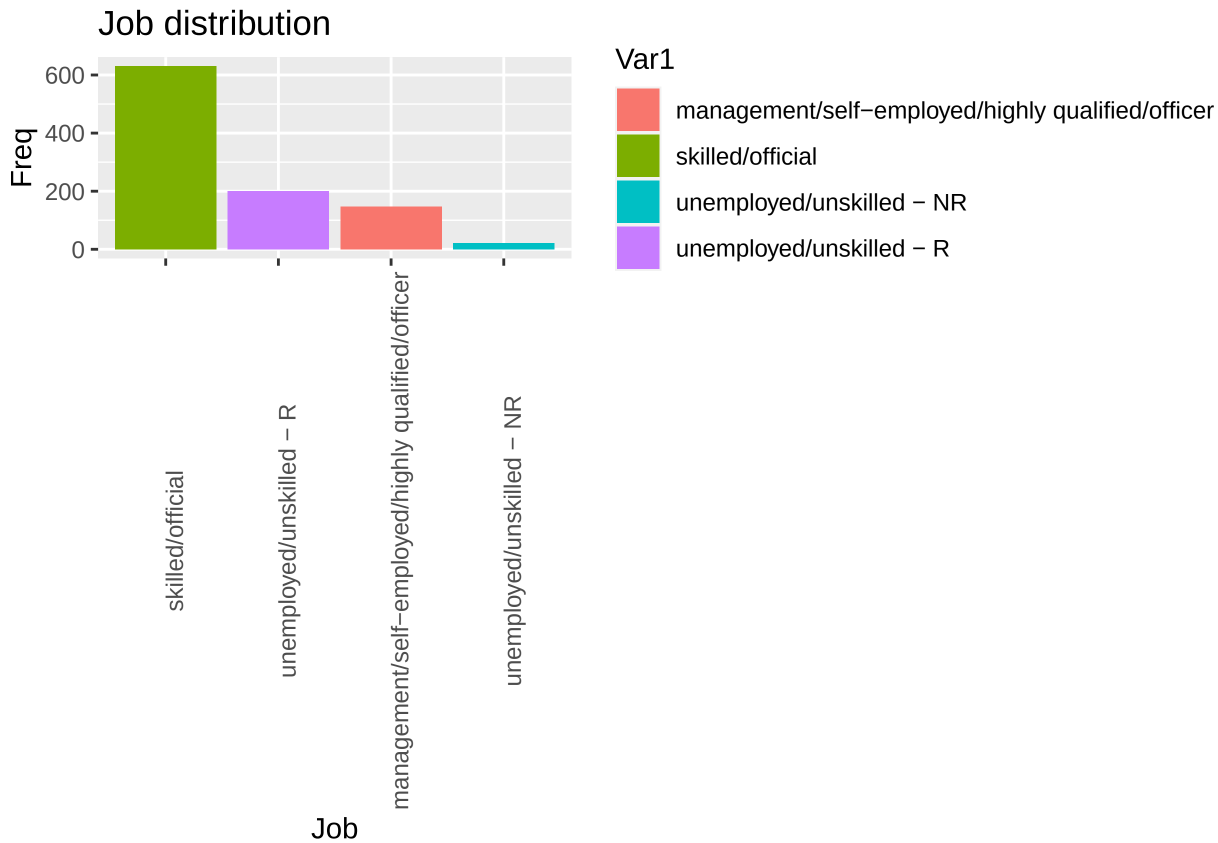
```
##
## female   male
##    310    690
```

```r
# Sex distribution
cont_succ <- table(credit_trans$Sex)
data.frame(cont_succ) %>%
  ggplot(aes(x= reorder(Var1,-Freq),Freq,fill=Var1)) +
  geom_bar(stat ="identity") +
  xlab("Sex") +
  ggtitle("Sex distribution")
```

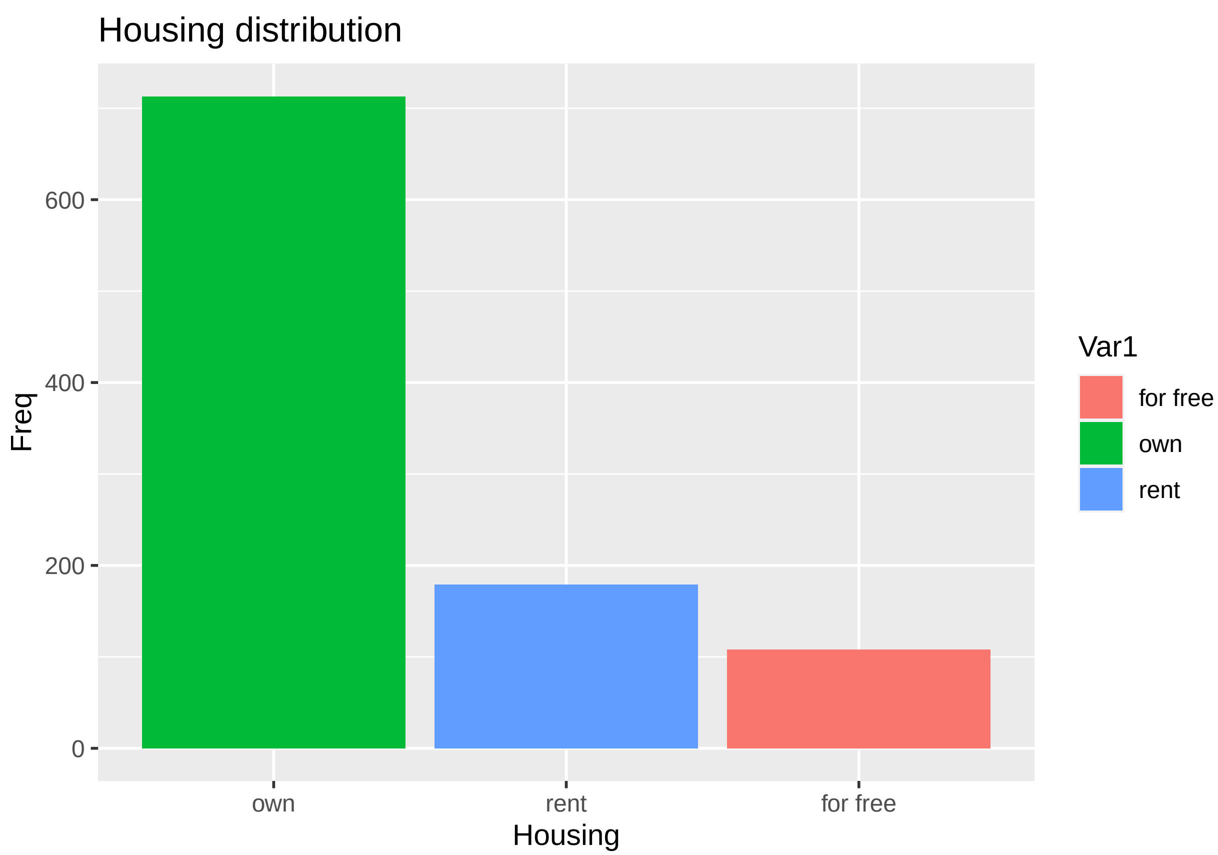



The number of male credit clients is approximately double than the number of female credit clients.

```r
# Job distribution
cont_succ <- table(credit_trans$Job)
data.frame(cont_succ) %>%
  ggplot(aes(x= reorder(Var1,-Freq),Freq,fill=Var1)) +
  geom_bar(stat ="identity") +
  xlab("Job") +
  ggtitle("Job distribution") +
  theme(axis.text.x = element_text(angle = 90))
```

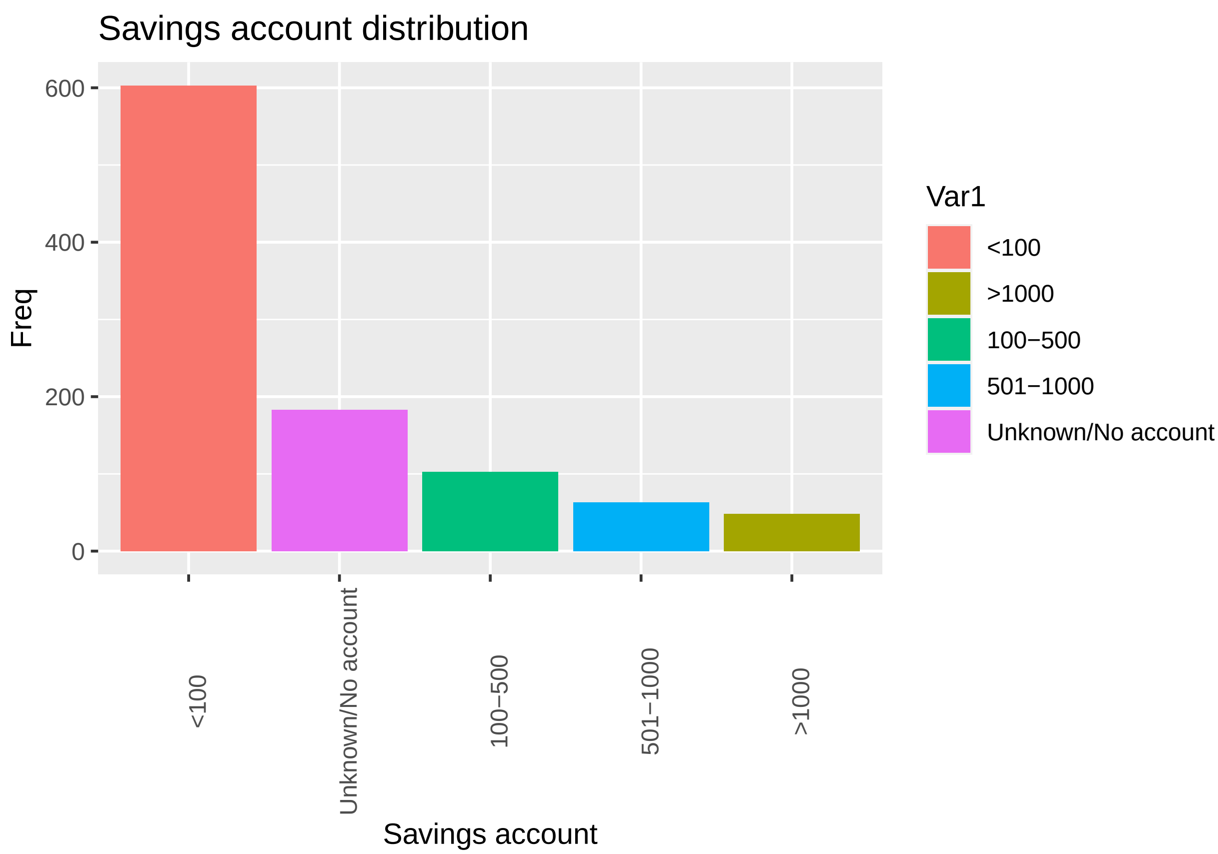


```r
# Housing distribution
cont_succ <- table(credit_trans$Housing)
data.frame(cont_succ) %>%
  ggplot(aes(x= reorder(Var1,-Freq),Freq,fill=Var1)) +
  geom_bar(stat ="identity") +
  xlab("Housing") +
  ggtitle("Housing distribution")
```



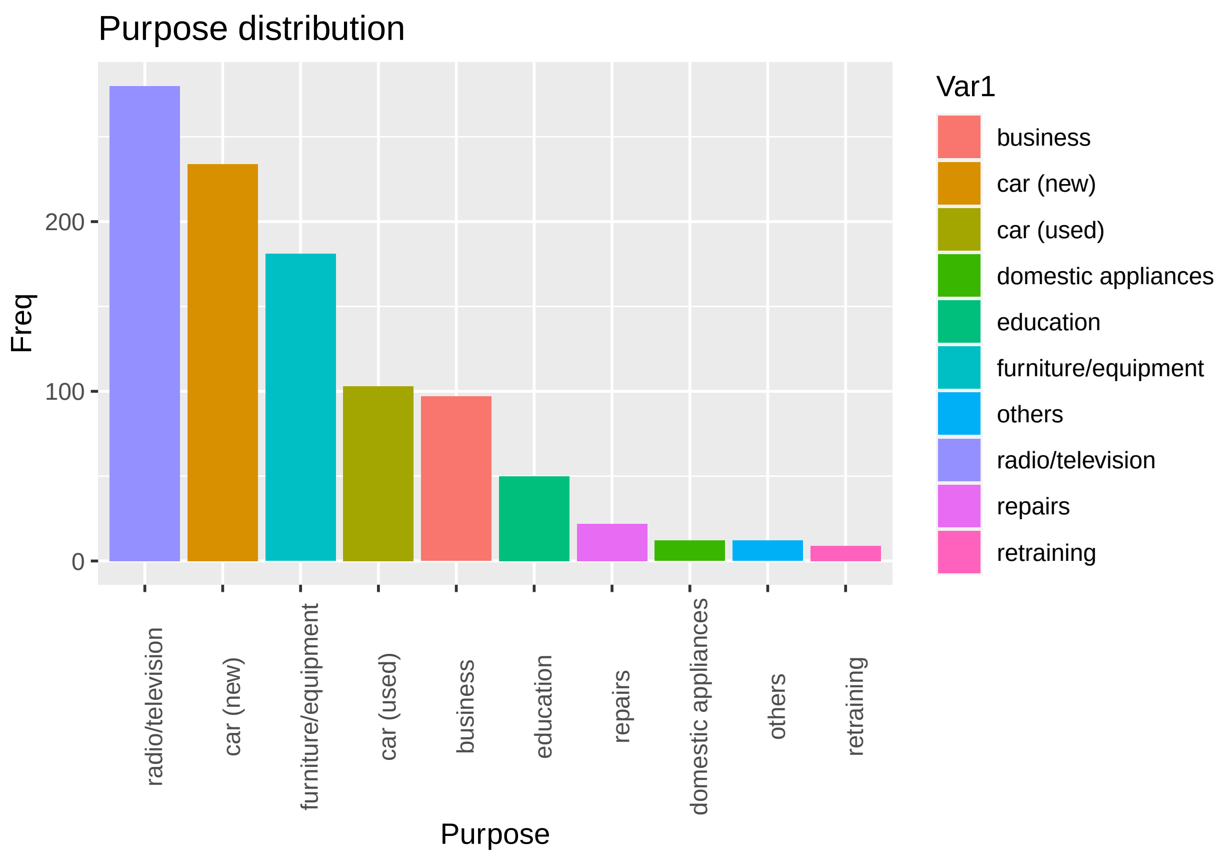

Most credit clients are skilled workers.

```r
# Savings account distribution
cont_succ <- table(credit_trans$Savings_account)
data.frame(cont_succ) %>%
  ggplot(aes(x= reorder(Var1,-Freq),Freq,fill=Var1)) +
  geom_bar(stat ="identity") +
  xlab("Savings account") +
  ggtitle("Savings account distribution") +
  theme(axis.text.x = element_text(angle = 90))
```

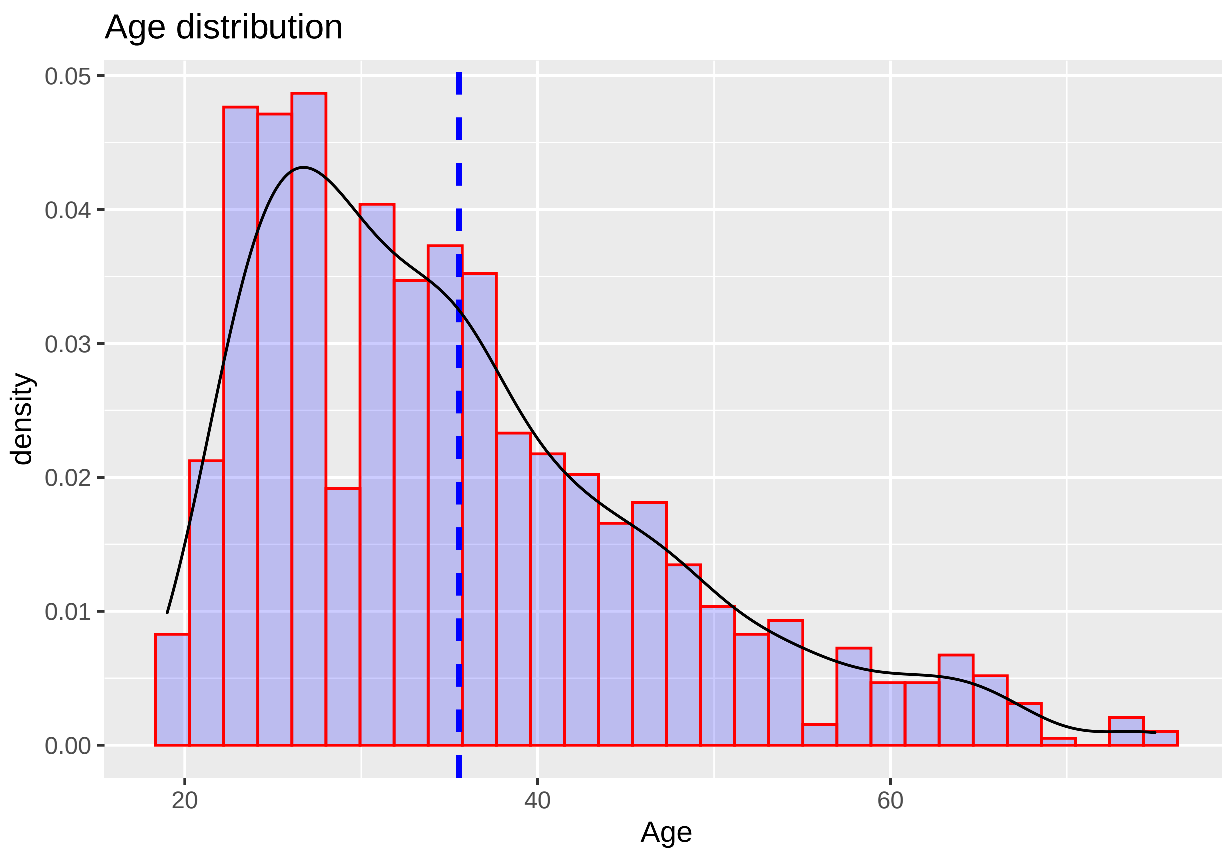



A majority of credit clients are owners of their house

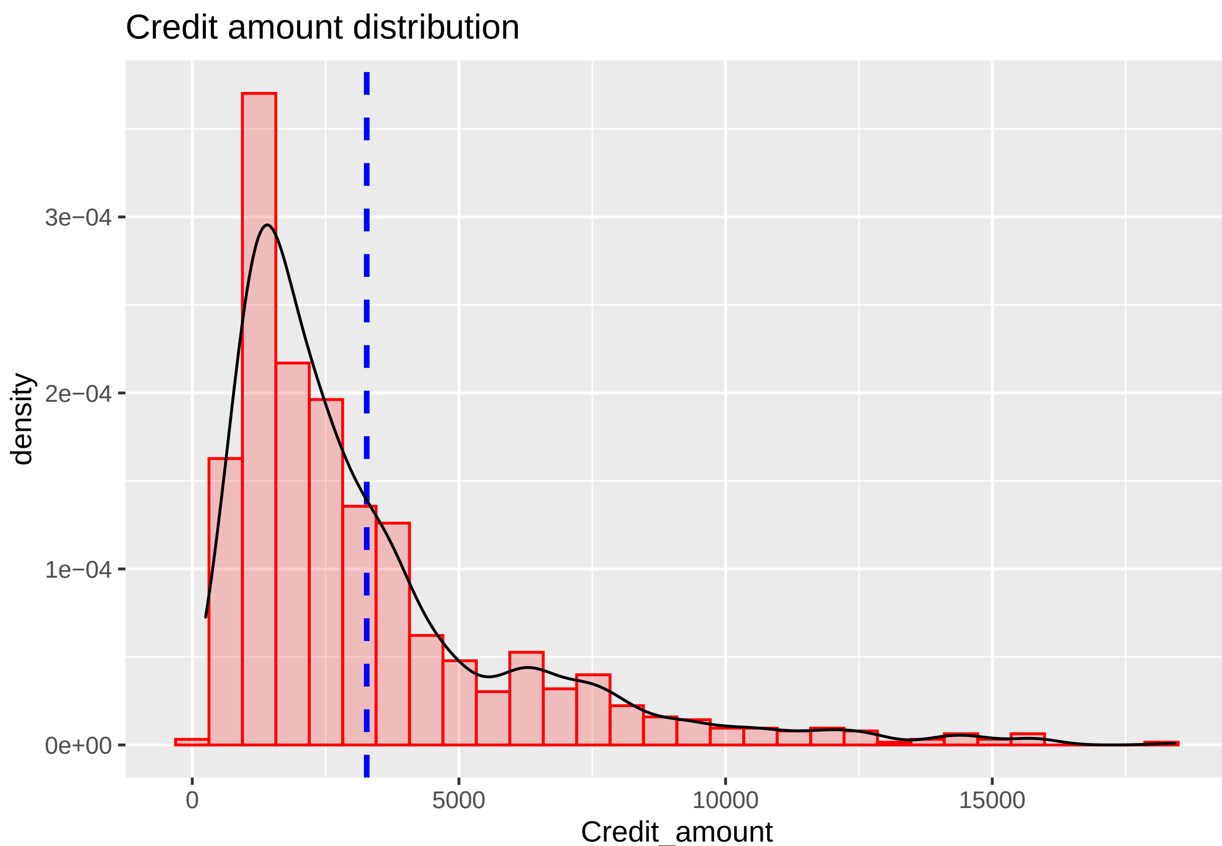
```r
# Purpose distribution
cont_succ <- table(credit_trans$Purpose)
data.frame(cont_succ) %>%
  ggplot(aes(x= reorder(Var1,-Freq),Freq,fill=Var1)) +
  geom_bar(stat ="identity") +
  xlab("Purpose") +
  theme(axis.text.x = element_text(angle = 90)) +
  ggtitle("Purpose distribution")
```



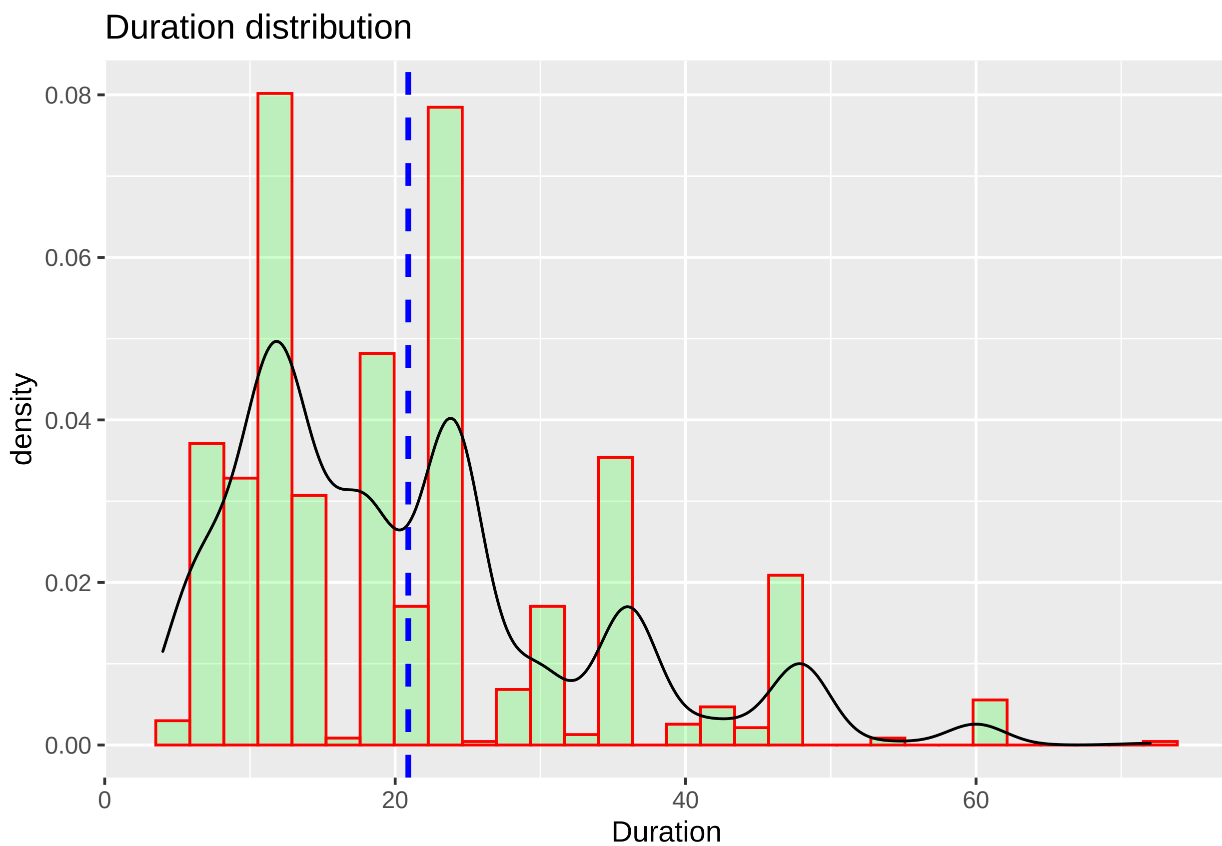
```r
# Age distribution
credit_trans %>% ggplot(aes(Age)) +
  geom_histogram(aes(y=..density..),col="red",fill="blue",alpha=.2) +
  ggtitle("Age distribution") +
  geom_vline(aes(xintercept=mean(Age)),color="blue", linetype="dashed", size=1)+
  geom_density(alpha=.2)
```



```r
# Credit amount distribution
credit_trans %>% ggplot(aes(Credit_amount)) +
  geom_histogram(aes(y=..density..),col="red",fill="red",alpha=.2) +
  ggtitle("Credit amount distribution") +
  geom_vline(aes(xintercept=mean(Credit_amount)),color="blue", linetype="dashed", size=1)+
  geom_density(alpha=.2)
```

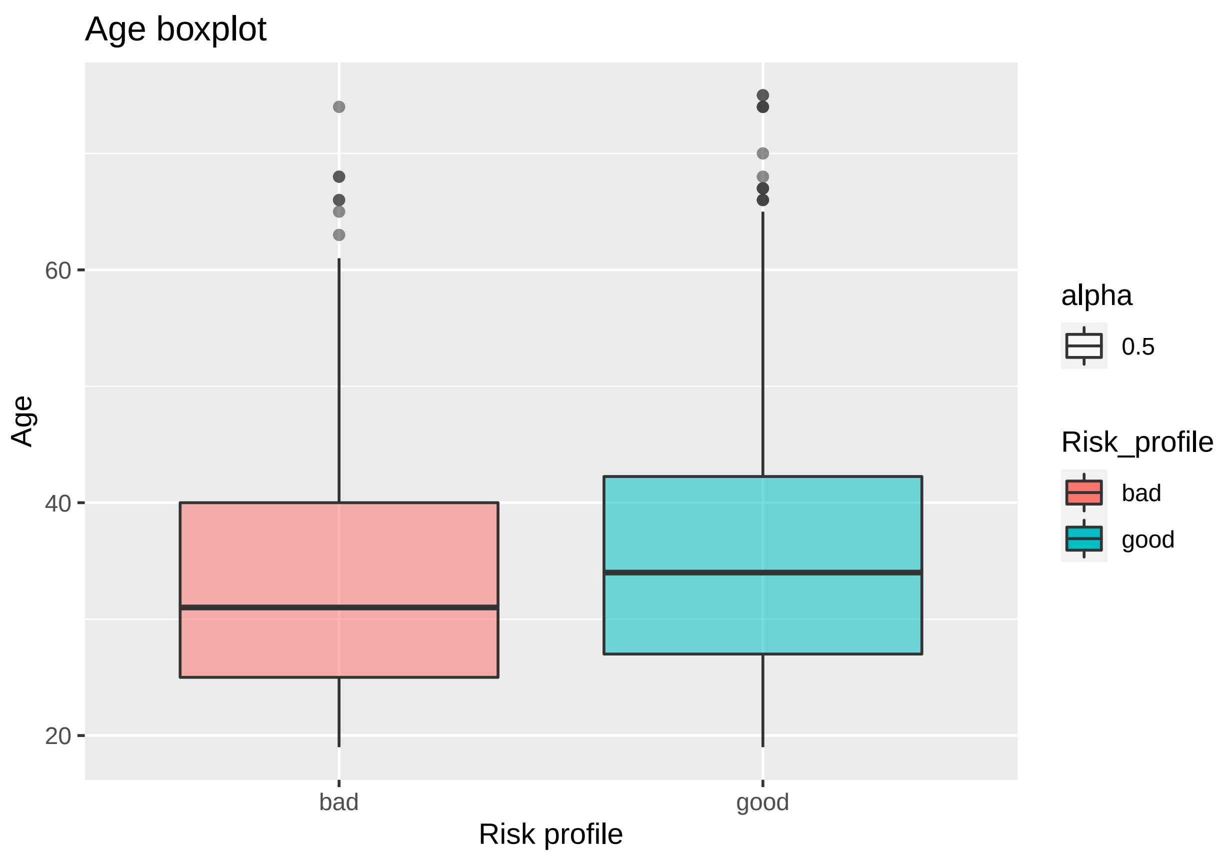


```r
# Duration distribution
credit_trans %>% ggplot(aes(Duration)) +
  geom_histogram(aes(y=..density..),col="red",fill="green",alpha=.2) +
  ggtitle("Duration distribution") +
  geom_vline(aes(xintercept=mean(Duration)),color="blue", linetype="dashed", size=1)+
  geom_density(alpha=.2)
```

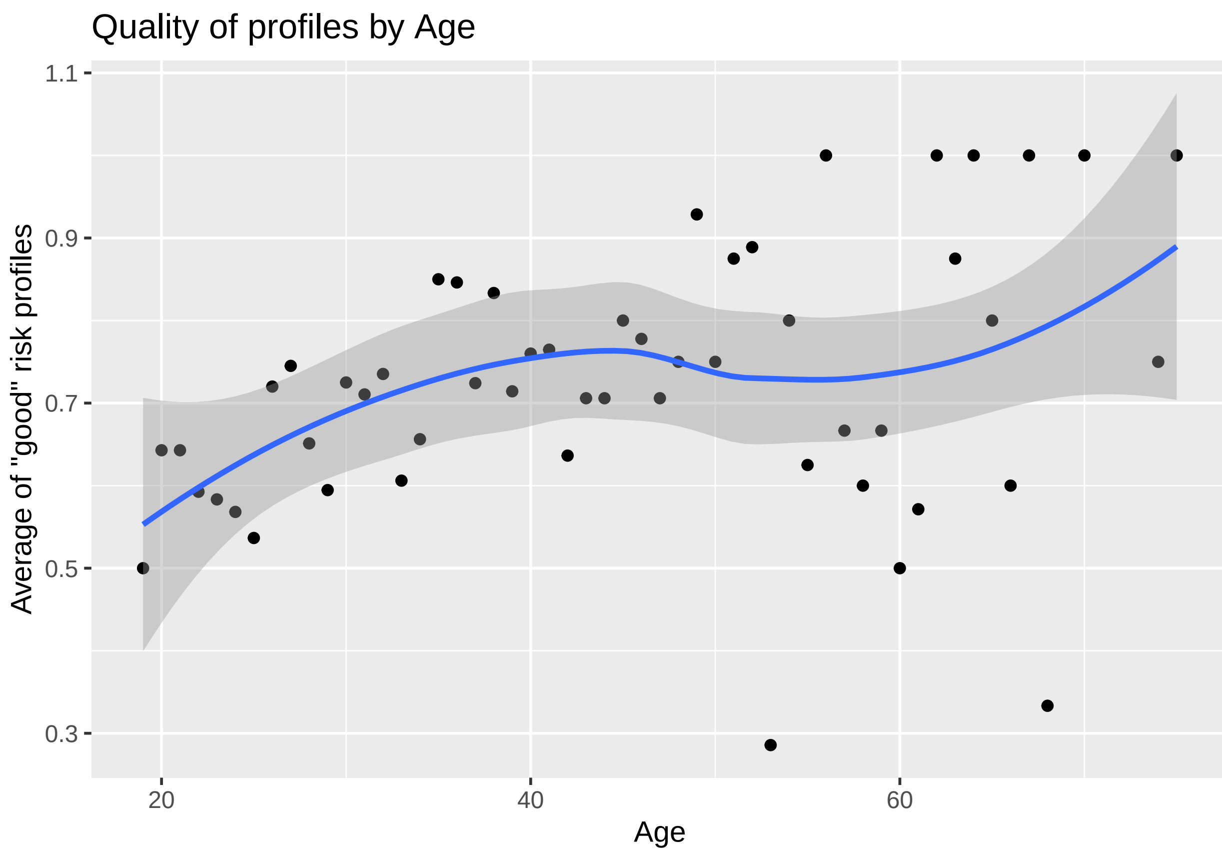


Features correlation check

```
# Age boxplot
credit_trans %>% ggplot(aes(Risk_profile,Age,fill=Risk_profile,alpha=.5)) +
  geom_boxplot() +
  ggtitle("Age boxplot") +
  xlab("Risk profile")
```

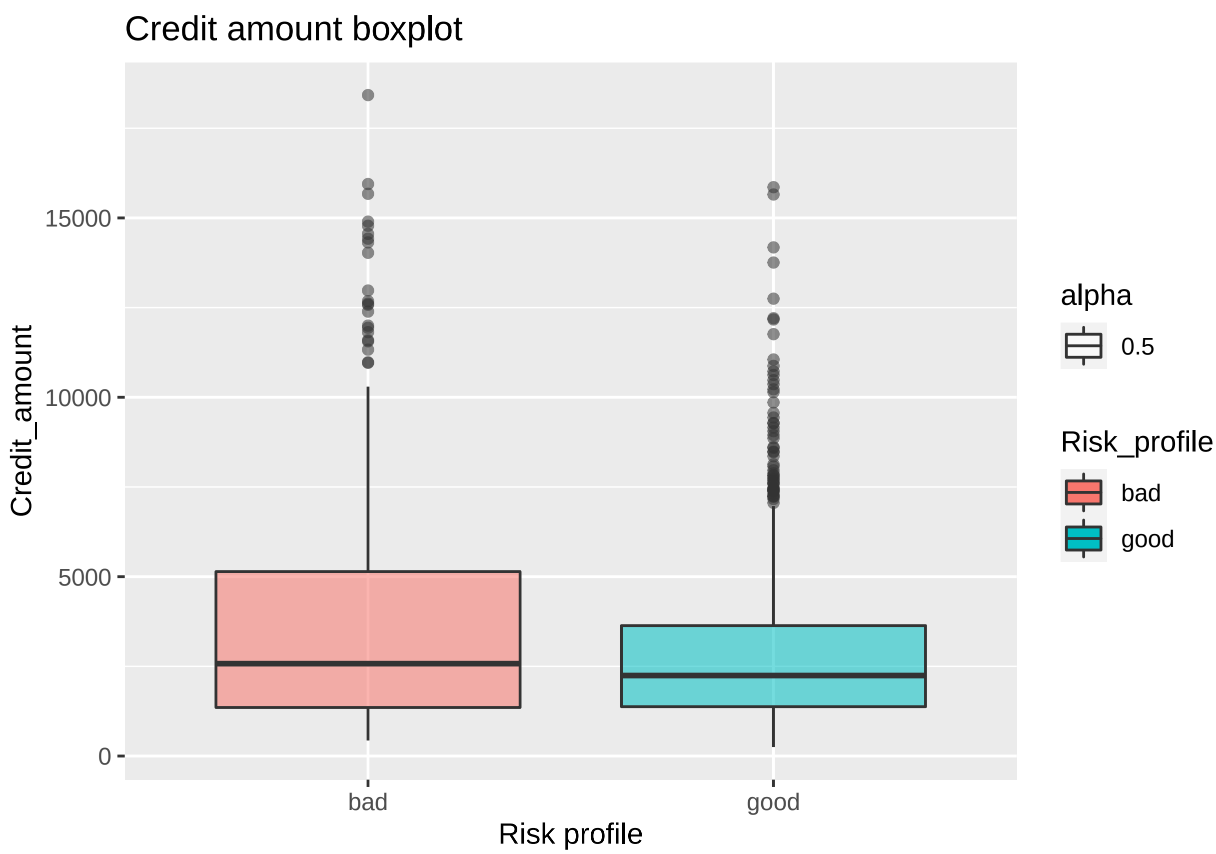


```r
# Quality of profiles by Age
credit_trans %>%
  group_by(Age) %>%
  summarize(avg = mean(Risk_profile=="good")) %>%
  ggplot(aes(Age,avg)) +
  geom_point() +
  geom_smooth() +
  ggtitle("Quality of profiles by Age") +
  ylab("Average of \"good\" risk profiles")
```

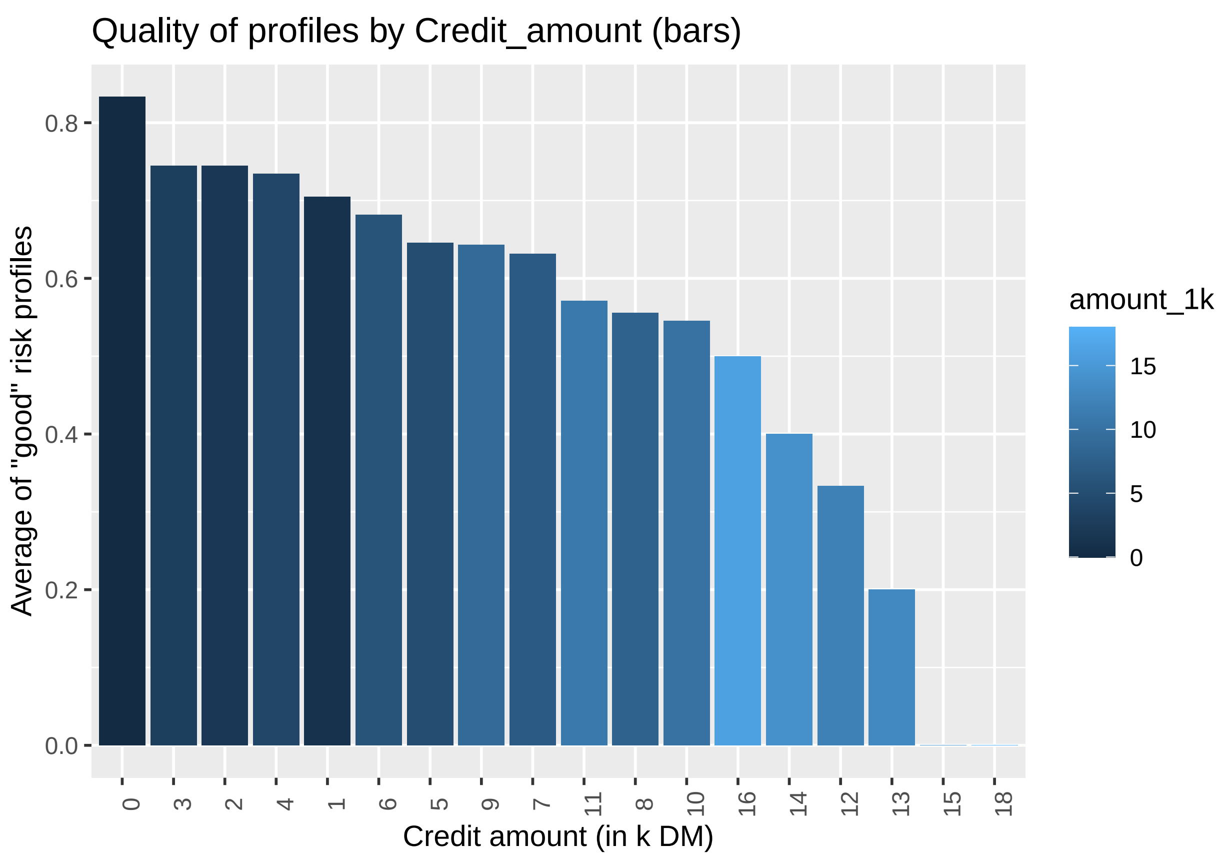



The two plots above show there's a correlation between age and risk profile quality. The older the client the higher the chance for them to have a good risk profile. One could expect this as we can make the assumption that older clients will have a more stable economic situation than younger ones.

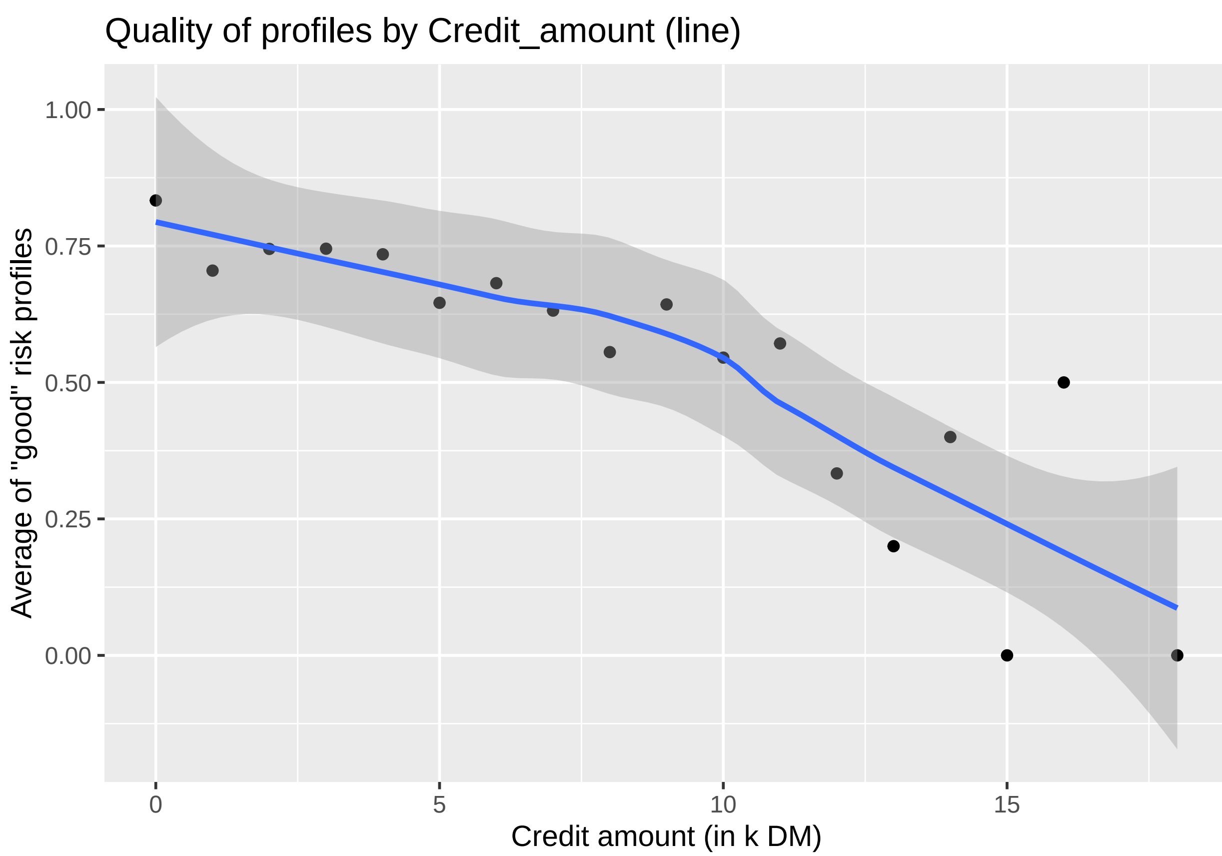
```r
# Credit amount boxplot
credit_trans %>% ggplot(aes(Risk_profile,Credit_amount,fill=Risk_profile,alpha=.5)) +
  geom_boxplot() +
  ggtitle("Credit amount boxplot") +
  xlab("Risk profile")
```



```
# Quality of profiles by Credit_amount (bars)
credit_trans %>%
  mutate(amount_1k = round(Credit_amount/1000)) %>%
  group_by(amount_1k) %>%
  summarize(avg_good = mean(Risk_profile=="good")) %>%
  ggplot(aes(reorder(amount_1k,-avg_good),avg_good,fill=amount_1k)) +
  geom_col() +
  theme(axis.text.x = element_text(angle = 90)) +
  xlab("Credit amount (in k DM)") +
  ggtitle("Quality of profiles by Credit_amount (bars)") +
  ylab("Average of \"good\" risk profiles")
```

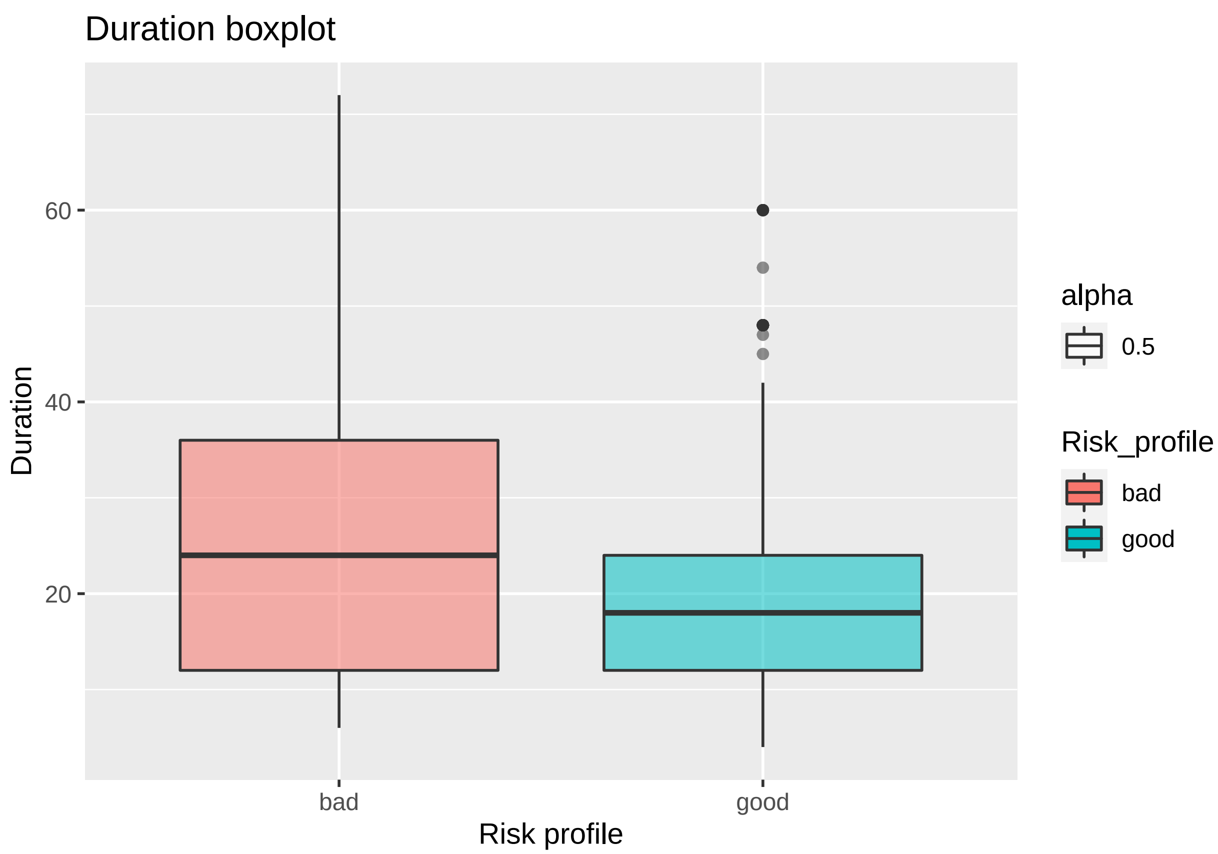


```
# Quality of profiles by Credit_amount (line)
credit_trans %>%
  mutate(amount_1k = round(Credit_amount/1000)) %>%
  group_by(amount_1k) %>%
  summarize(avg = mean(Risk_profile=="good")) %>%
  ggplot(aes(amount_1k,avg)) +
  geom_point() +
  geom_smooth() +
  xlab("Credit amount (in k DM)") +
  ggtitle("Quality of profiles by Credit_amount (line)") +
  ylab("Average of \"good\" risk profiles")
```

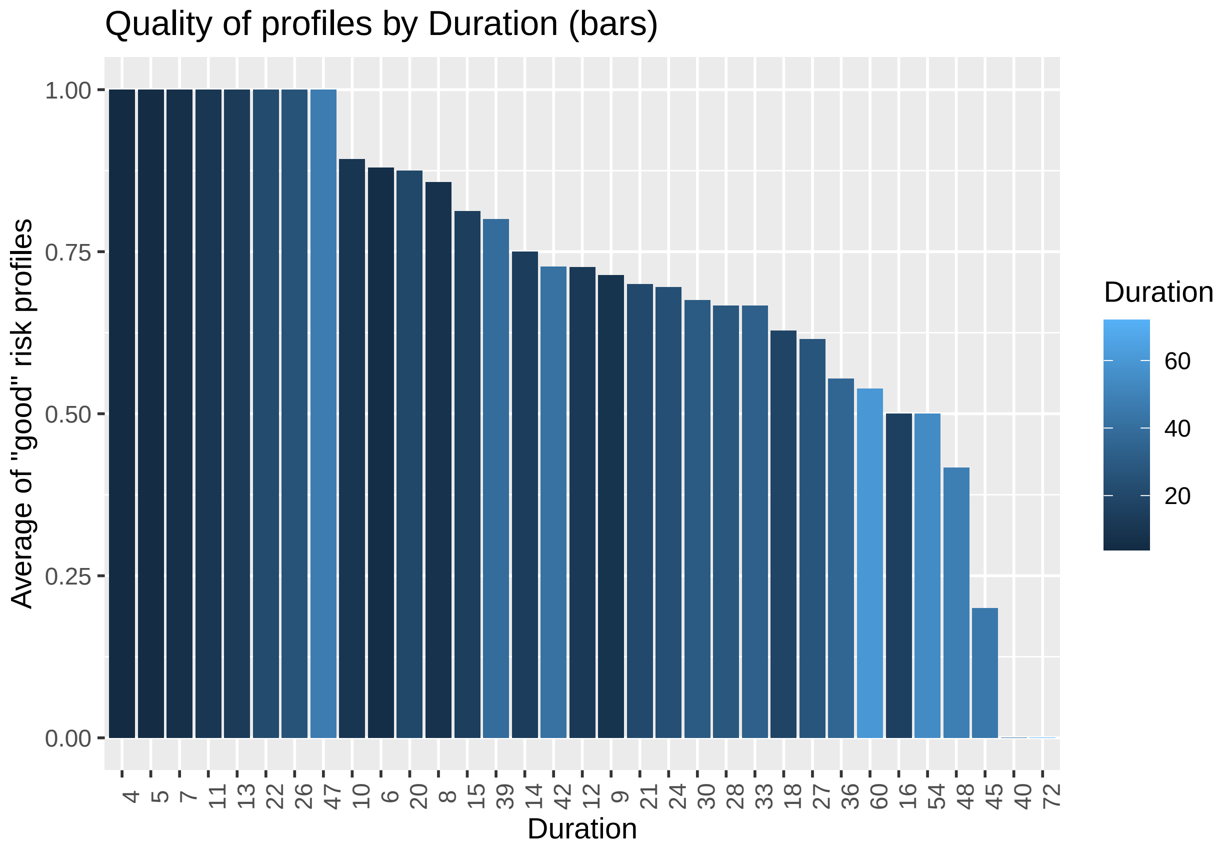



As shown in the three previous plots, there seems to be a trend by which holders of higher amount credits have a worse risk profile than lower ones. This makes sense as high amount credits are riskier than lower amount ones.

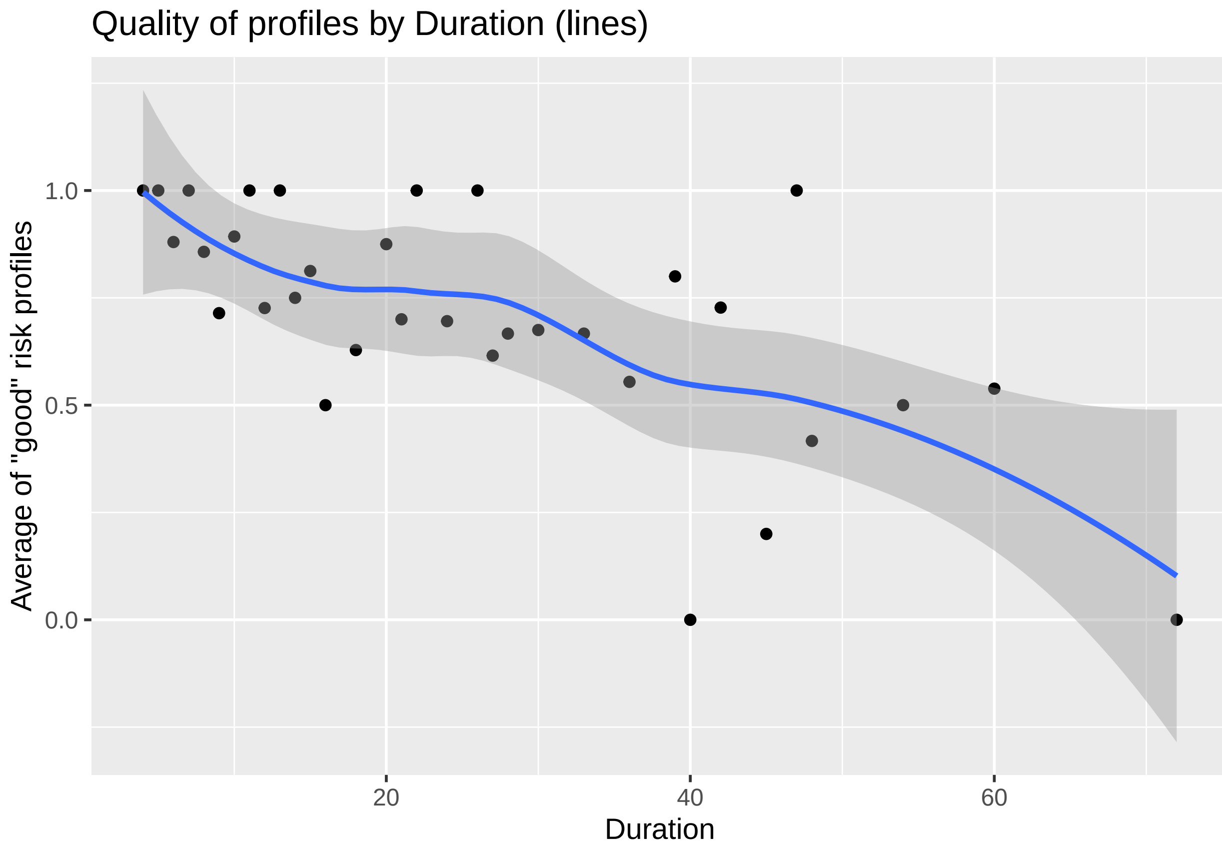
```r
# Duration boxplot
credit_trans %>% ggplot(aes(Risk_profile,Duration,fill=Risk_profile,alpha=.5)) +
  geom_boxplot() +
  ggtitle("Duration boxplot") +
  xlab("Risk profile")
```




```r
# Quality of profiles by Duration (bars)
credit_trans %>%
  group_by(Duration) %>%
  summarize(avg_good = mean(Risk_profile=="good")) %>%
  ggplot(aes(reorder(Duration,-avg_good),avg_good,fill=Duration)) +
  geom_col() +
  theme(axis.text.x = element_text(angle = 90)) +
  xlab("Duration") +
  ggtitle("Quality of profiles by Duration (bars)") +
  ylab("Average of \"good\" risk profiles")
```

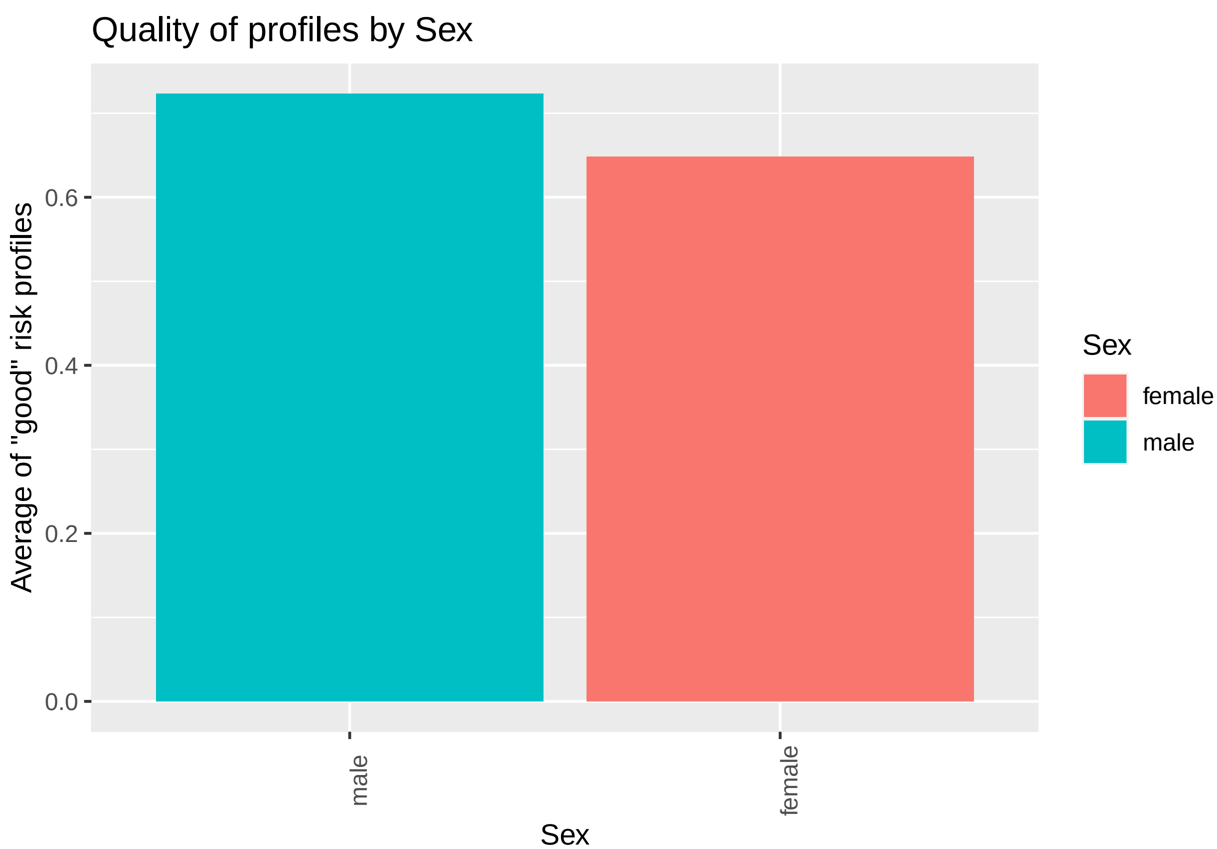



```
# Quality of profiles by Duration (lines)
credit_trans %>%
  group_by(Duration) %>%
  summarize(avg = mean(Risk_profile=="good")) %>%
  ggplot(aes(Duration,avg)) +
  geom_point() +
  geom_smooth() +
  ggtitle("Quality of profiles by Duration (lines)") +
  ylab("Average of \"good\" risk profiles")
```

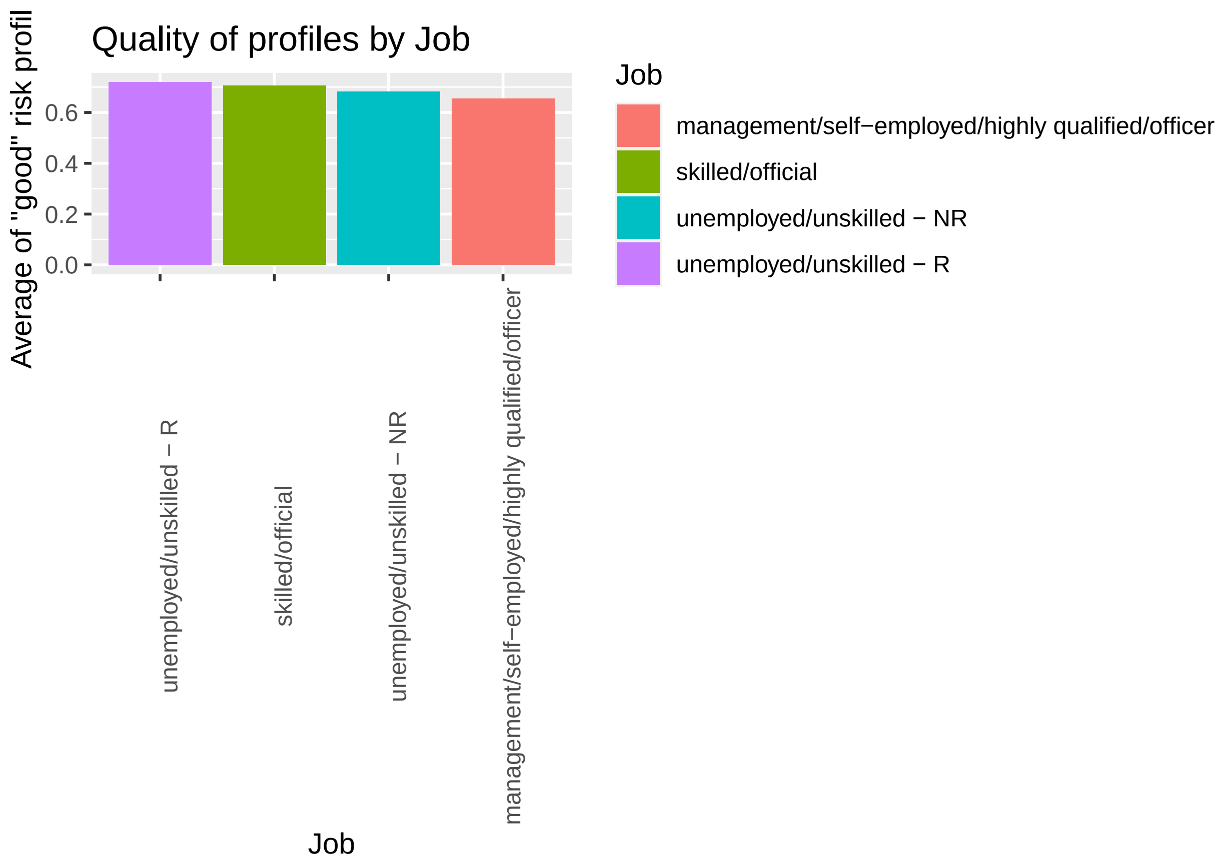



The three plots above show that clients with longer duration credits have a worse risk profile than shorter ones. This makes sense as the risk is higher when the bank has to face the possibility of default for a longer time.

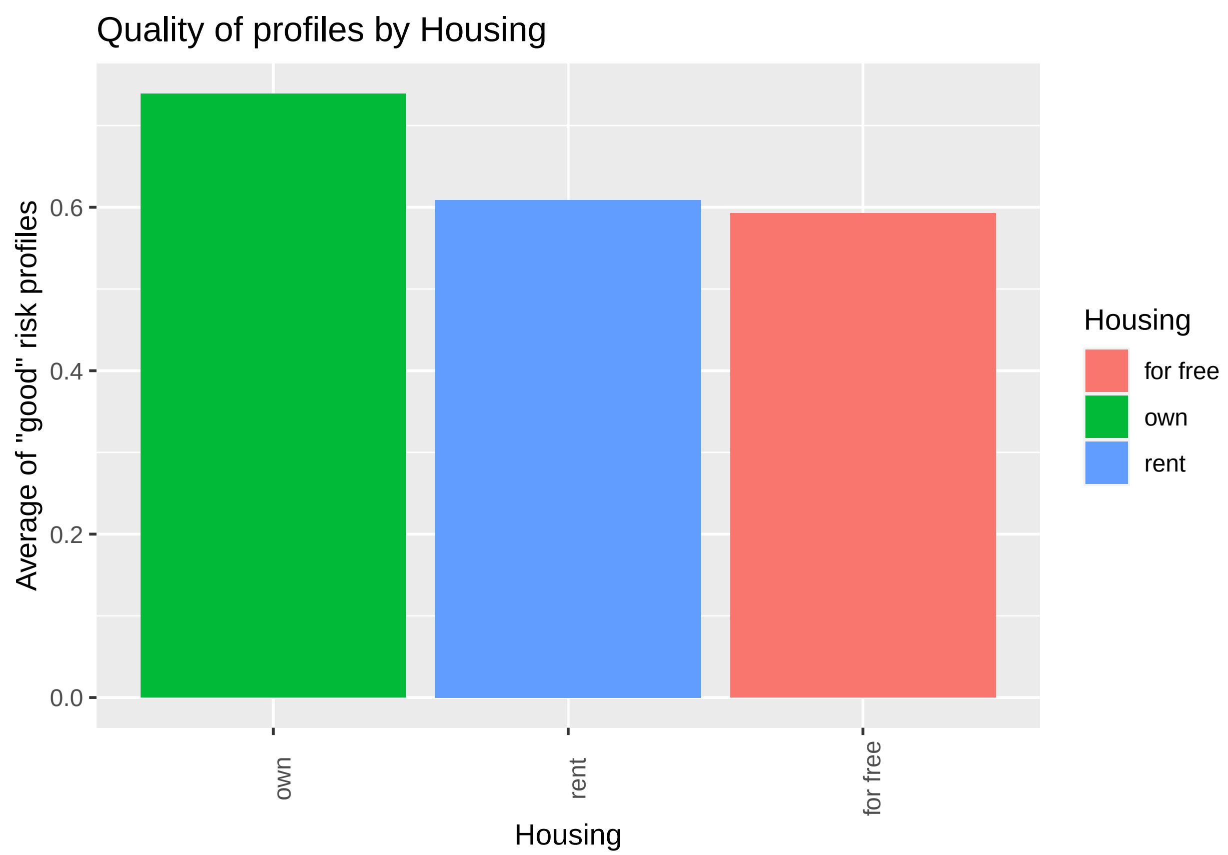
```r
# Quality of profiles by Sex
credit_trans %>%
  group_by(Sex) %>%
  summarize(avg_good = mean(Risk_profile=="good")) %>%
  ggplot(aes(reorder(Sex,-avg_good),avg_good,fill=Sex)) +
  geom_col() +
  theme(axis.text.x = element_text(angle = 90)) +
  xlab("Sex") +
  ggtitle("Quality of profiles by Sex") +
  ylab("Average of \"good\" risk profiles")
```



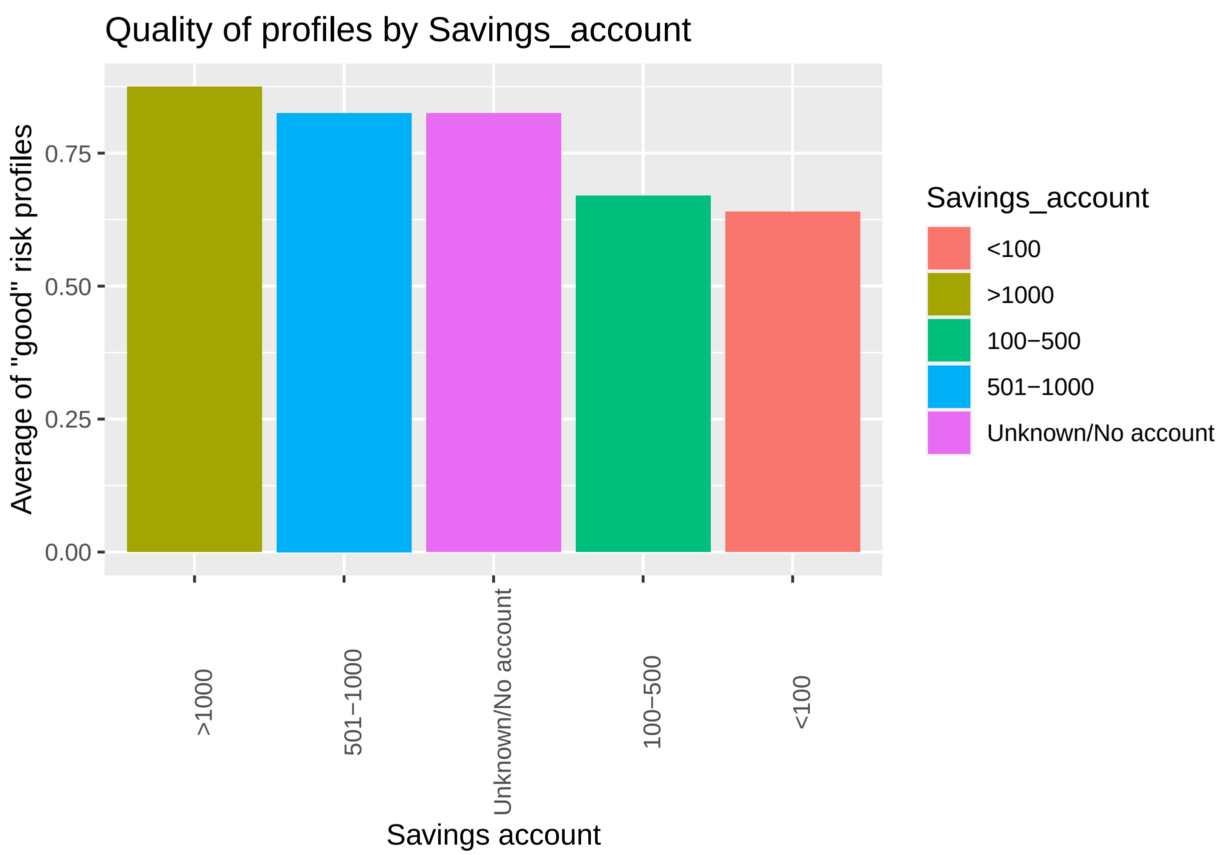
```
# Quality of profiles by Job
credit_trans %>%
  group_by(Job) %>%
  summarize(avg_good = mean(Risk_profile=="good")) %>%
  ggplot(aes(reorder(Job,-avg_good),avg_good,fill=Job)) +
  geom_col() +
  theme(axis.text.x = element_text(angle = 90)) +
  xlab("Job") +
  ggtitle("Quality of profiles by Job") +
  ylab("Average of \"good\" risk profiles")
```



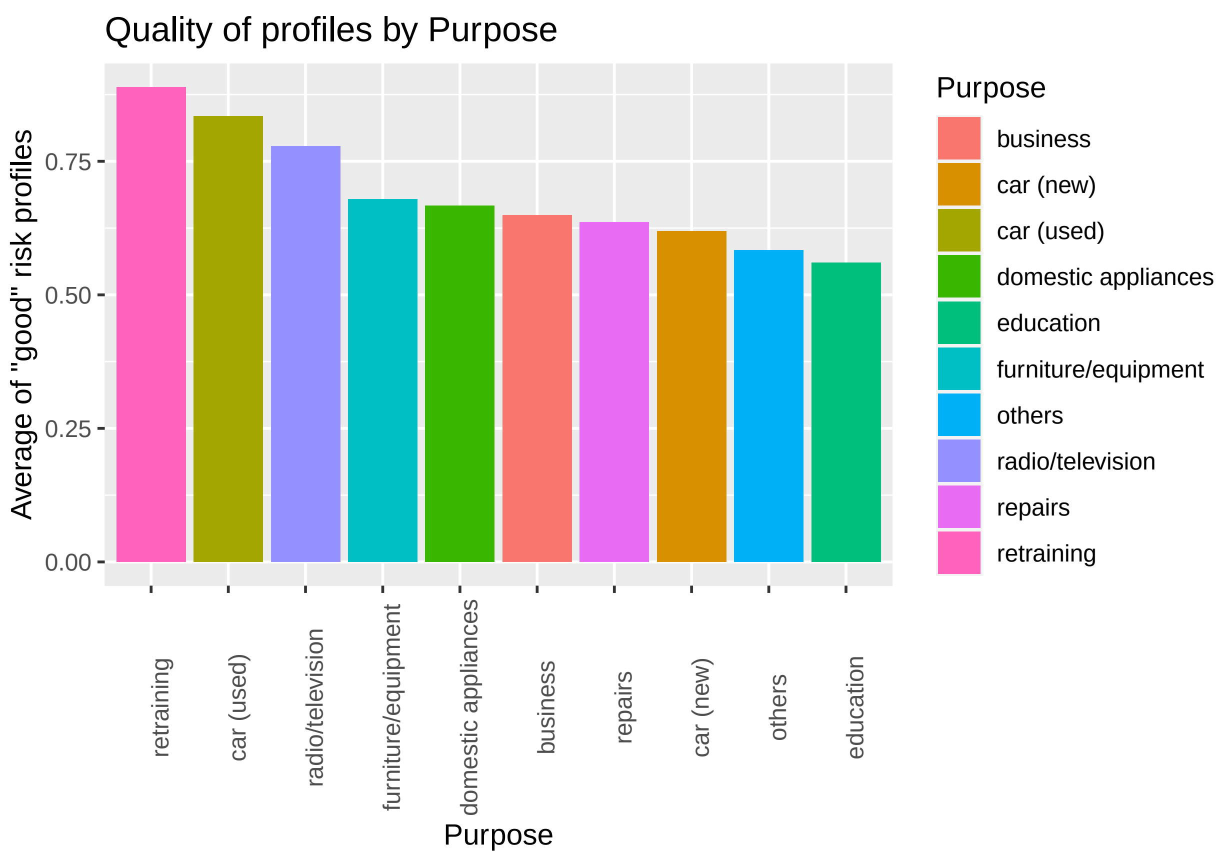
```r
# Quality of profiles by Housing
credit_trans %>%
  group_by(Housing) %>%
  summarize(avg_good = mean(Risk_profile=="good")) %>%
  ggplot(aes(reorder(Housing,-avg_good),avg_good,fill=Housing)) +
  geom_col() +
  theme(axis.text.x = element_text(angle = 90)) +
  xlab("Housing") +
  ggtitle("Quality of profiles by Housing") +
  ylab("Average of \"good\" risk profiles")
```



```r
# Quality of profiles by Savings_account
credit_trans %>%
  group_by(Savings_account) %>%
  summarize(avg_good = mean(Risk_profile=="good")) %>%
  ggplot(aes(reorder(Savings_account,-avg_good),avg_good,fill=Savings_account)) +
  geom_col() +
  theme(axis.text.x = element_text(angle = 90)) +
  xlab("Savings account") +
  ggtitle("Quality of profiles by Savings_account") +
  ylab("Average of \"good\" risk profiles")
```



```r
# Quality of profiles by Purpose
credit_trans %>%
  group_by(Purpose) %>%
  summarize(avg_good = mean(Risk_profile=="good")) %>%
  ggplot(aes(reorder(Purpose,-avg_good),avg_good,fill=Purpose)) +
  geom_col() +
  theme(axis.text.x = element_text(angle = 90)) +
  xlab("Purpose") +
  ggtitle("Quality of profiles by Purpose") +
  ylab("Average of \"good\" risk profiles")
```




Sex, Job, Housing, Savings account and Purpose seem to have a less significant correlation with the risk profile.

## Modeling approach

The data frame perf_results is created to keep track of the different models performance so as they can be easily compared.

```
perf_results <- data_frame()
```

Also, a function for F1-scores calculation is created as it will be applied to all models.

The F1-score metric is calculated as:

$$F_1 - Score = 2 * \frac{precision * recall}{precision + recall}$$

```
# F1-score calculation
f1 <- function(y_hat,y){
  precision <- posPredValue(y_hat, y, positive="1")
  recall <- sensitivity(y_hat, y, positive="1")
  F1 <- (2 * precision * recall) / (precision + recall)
  F1
}
```

All models are trained and parameters fitted by using the functions included in the caret package.

### Model 1: Logistic regression

Logistic regression is an algorithm used very commonly for binary classification problems although it can be applied in many other cases.

In this case, considering the apparent high correlation between some predictors and the outcome, the first approach will be to apply a logistic regression model to the data.

All variables will be considered as predictors for the model fitting process.

```
#------------------------
### Logistic regresion
#------------------------

# trainctrl <- trainControl(verboseIter = TRUE)

# Train model
fit_glm <- train(Risk ~ ., method="glm", data = credit_train)

# Calculate predictions using fitted model
y_hat_glm <- predict(fit_glm, credit_test, type = "raw")

# Display results
cm_glm <- confusionMatrix(y_hat_glm,credit_test$Risk)
Acc_glm <- cm_glm$overall[["Accuracy"]]
F1_glm <- f1(y_hat_glm,credit_test$Risk)

# Save first metric result in perf_results
perf_results <- data_frame(method = "Logistic regresion", Accuracy = Acc_glm, F1_score = F1_glm)
perf_results %>% knitr::kable()
```

| method | Accuracy | F1_score |
|---|---|---|
| Logistic regresion | 0.775 | 0.8432056 |

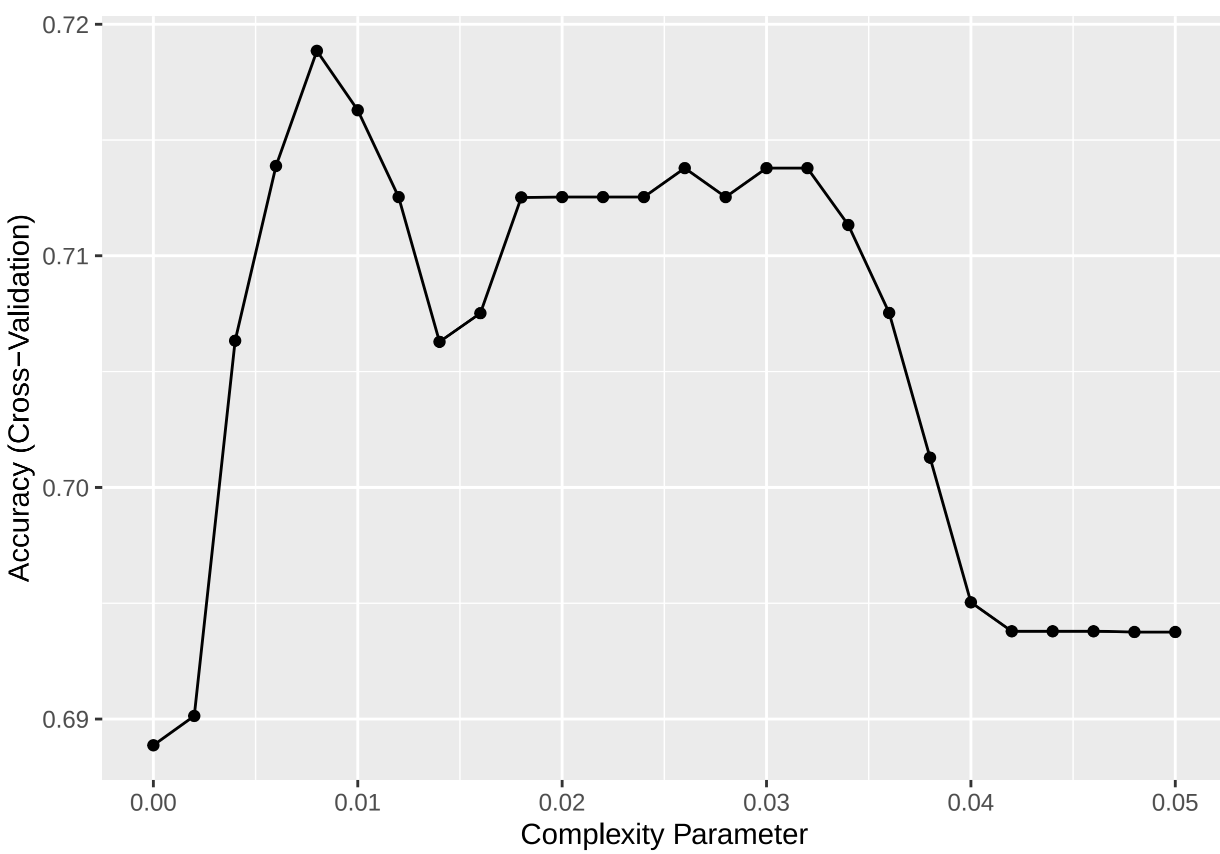
#### Model 2: Decision tree

Parameter *cp* will be optimized with cross-validation.

```
#-------------------------
### Decision tree
#-------------------------

# Train model
fit_dt <- train(Risk ~ ., data = credit_train,method="rpart",
                   tuneGrid = data.frame(cp = seq(0, 0.05, 0.002)),
                trControl = trainControl(method = "cv"))

# Calculate predictions using fitted model and check results
y_hat_dt <- predict(fit_dt, credit_test, type = "raw")
cm_dt <- confusionMatrix(y_hat_dt, credit_test$Risk)
Acc_dt <- cm_dt$overall[["Accuracy"]]
F1_dt <- f1(y_hat_dt,credit_test$Risk)
```

Optimal *cp* parameter for Decision tree

```
# Optimal cp parameter
ggplot(fit_dt)
```

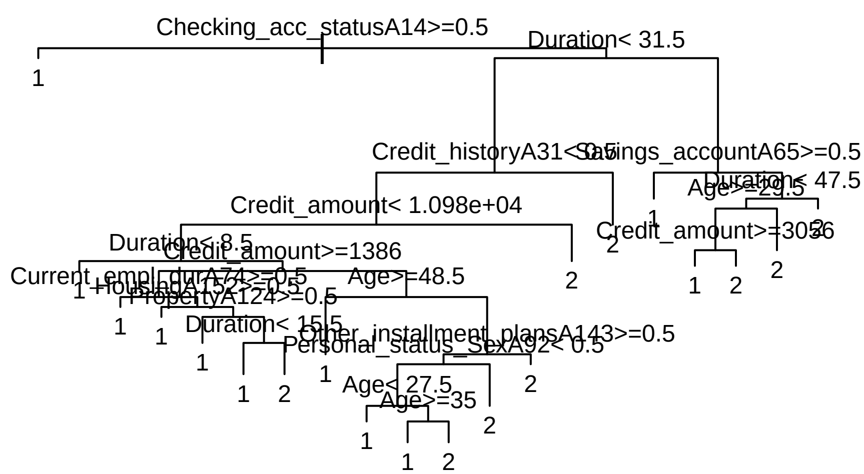


```r
fit_dt$bestTune
```

```
##      cp
## 5 0.008
```

Final model for decision tree

```r
# Tree visualization
plot(fit_dt$finalModel, margin = 0.1)
text(fit_dt$finalModel, cex = 0.75)
```




Decision tree results

```r
# Save metric in perf_results
perf_results <- bind_rows(perf_results, data_frame(method="Decision tree", Accuracy = Acc_dt, F1_score =
perf_results %>% knitr::kable()
```

| method | Accuracy | F1_score |
|---|---|---|
| Logistic regresion | 0.775 | 0.8432056 |
| Decision tree | 0.745 | 0.8305648 |

### Model 3: Random forest

Let's see if bagging multiple decision trees by using Random forest can improve the previous result.

Parameter *mtry* will be optimized by using cross-validation with the train() function. Parameter *ntree* will be set at a fixed value of 1000.

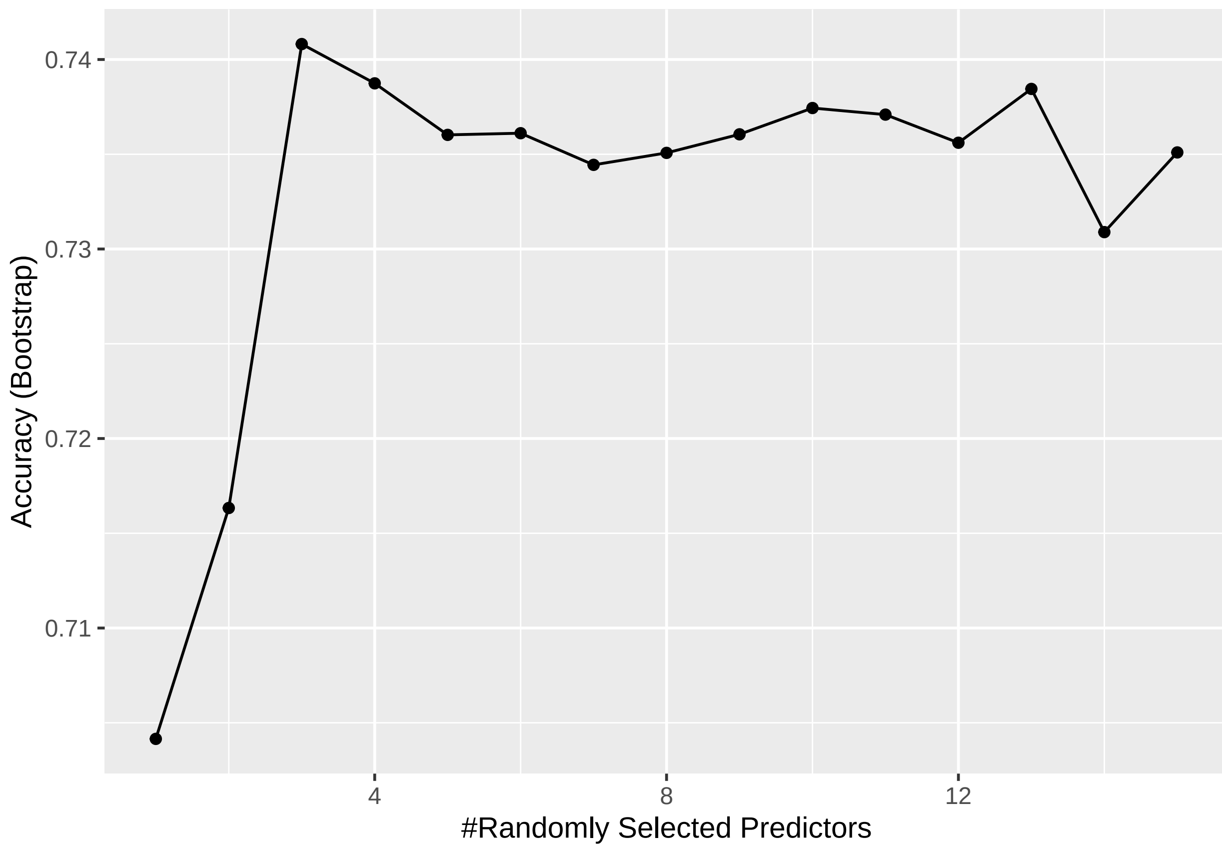
```
# Warning: Please note that this code takes a substantial amount of time to execute

trainctrl <- trainControl(number = 10)

# Parameters (mtry) fit. ntree parameter is set at a fixed value of 1000
fit_rf <- train(Risk ~ .
                , data = credit_train,method="rf",
                tuneGrid = data.frame(mtry = seq(1,15)),ntree=1000, trControl = trainctrl)
```

Optimal *mtry* parameter for Random forest

```
# Optimal mtry parameter
print(fit_rf)
```

```
## Random Forest
##
## 800 samples
##  20 predictor
##   2 classes: '1', '2'
##
## No pre-processing
## Resampling: Bootstrapped (10 reps)
## Summary of sample sizes: 800, 800, 800, 800, 800, 800, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##    1    0.7041512  0.00000000
##    2    0.7163345  0.08167287
##    3    0.7408150  0.21846999
##    4    0.7387431  0.24270451
##    5    0.7360230  0.24930107
##    6    0.7361112  0.25999503
##    7    0.7344407  0.26150814
##    8    0.7350727  0.26990214
##    9    0.7360526  0.27639550
##   10    0.7374391  0.28403231
##   11    0.7370917  0.28832570
##   12    0.7356070  0.28665289
##   13    0.7384478  0.30051468
##   14    0.7308924  0.28020114
##   15    0.7350966  0.29367862
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 3.
```

```r
ggplot(fit_rf)
```




```r
fit_rf$bestTune
```

```
##   mtry
## 3    3
```

Feature importance analysis

```r
# Feature importance analysis
imp <- varImp(fit_rf)
imp
```

```
## rf variable importance
##
##   only 20 most important variables shown (out of 48)
##
##                                Overall
## Credit_amount                   100.00
## Age                              85.74
## Duration                         80.37
## Checking_acc_statusA14           69.24
## Installment_rate                 37.32
## Residence_since                  34.73
```

```
## Credit_historyA34             22.64
## Other_installment_plansA143   21.74
## HousingA152                   21.54
## N_credits                     20.98
## Checking_acc_statusA12        20.79
## TelephoneA192                 18.02
## Personal_status_SexA92        17.67
## Savings_accountA65            17.60
## PropertyA123                  16.96
## Personal_status_SexA93        16.87
## JobA173                       16.39
## Current_empl_durA75           16.35
## Credit_historyA32             16.30
## PurposeA43                    16.03
```

Random forest results

```r
# Calculate predictions using fitted model and check results
y_hat_rf <- predict(fit_rf, credit_test, type = "raw")
cm_rf <- confusionMatrix(y_hat_rf,credit_test$Risk)
Acc_rf <- cm_rf$overall[["Accuracy"]]
F1_rf <- f1(y_hat_rf,credit_test$Risk)

# Save metric in perf_results
perf_results <- bind_rows(perf_results, data_frame(method="Random forest", Accuracy = Acc_rf, F1_score =
perf_results %>% knitr::kable()
```

| method | Accuracy | F1_score |
|---|---|---|
| Logistic regresion | 0.775 | 0.8432056 |
| Decision tree | 0.745 | 0.8305648 |
| Random forest | 0.755 | 0.8463950 |

**Model 4: SVM**

```
#------------------------
### SVM with Linear Kernel
#------------------------

# Set up Repeated k-fold Cross Validation
train_control <- trainControl(method="repeatedcv", number=25, repeats=3)

# Fit the model
svm <- train(Risk ~ ., data = credit_train,
              method = "svmLinear", trControl = train_control)
```

View of the SVM model

```
#View the model
svm
```

```
## Support Vector Machines with Linear Kernel
##
## 800 samples
##  20 predictor
##   2 classes: '1', '2'
##
## No pre-processing
## Resampling: Cross-Validated (25 fold, repeated 3 times)
## Summary of sample sizes: 768, 768, 769, 768, 767, 768, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.7510606  0.3715374
##
## Tuning parameter 'C' was held constant at a value of 1
```

SVM results

```
# Calculate predictions using fitted model and check results
y_hat_svm <- predict(svm, credit_test, type = "raw")
cm_svm <- confusionMatrix(y_hat_svm, credit_test$Risk)
Acc_svm <- cm_svm$overall[["Accuracy"]]
F1_svm <- f1(y_hat_svm,credit_test$Risk)

# Save metric in perf_results
perf_results <- bind_rows(perf_results, data_frame(method="SVM with Linear Kernel", Accuracy = Acc_svm,
perf_results %>% knitr::kable()
```

| method | Accuracy | F1_score |
|---|---|---|
| Logistic regresion | 0.775 | 0.8432056 |
| Decision tree | 0.745 | 0.8305648 |
| Random forest | 0.755 | 0.8463950 |
| SVM with Linear Kernel | 0.770 | 0.8413793 |

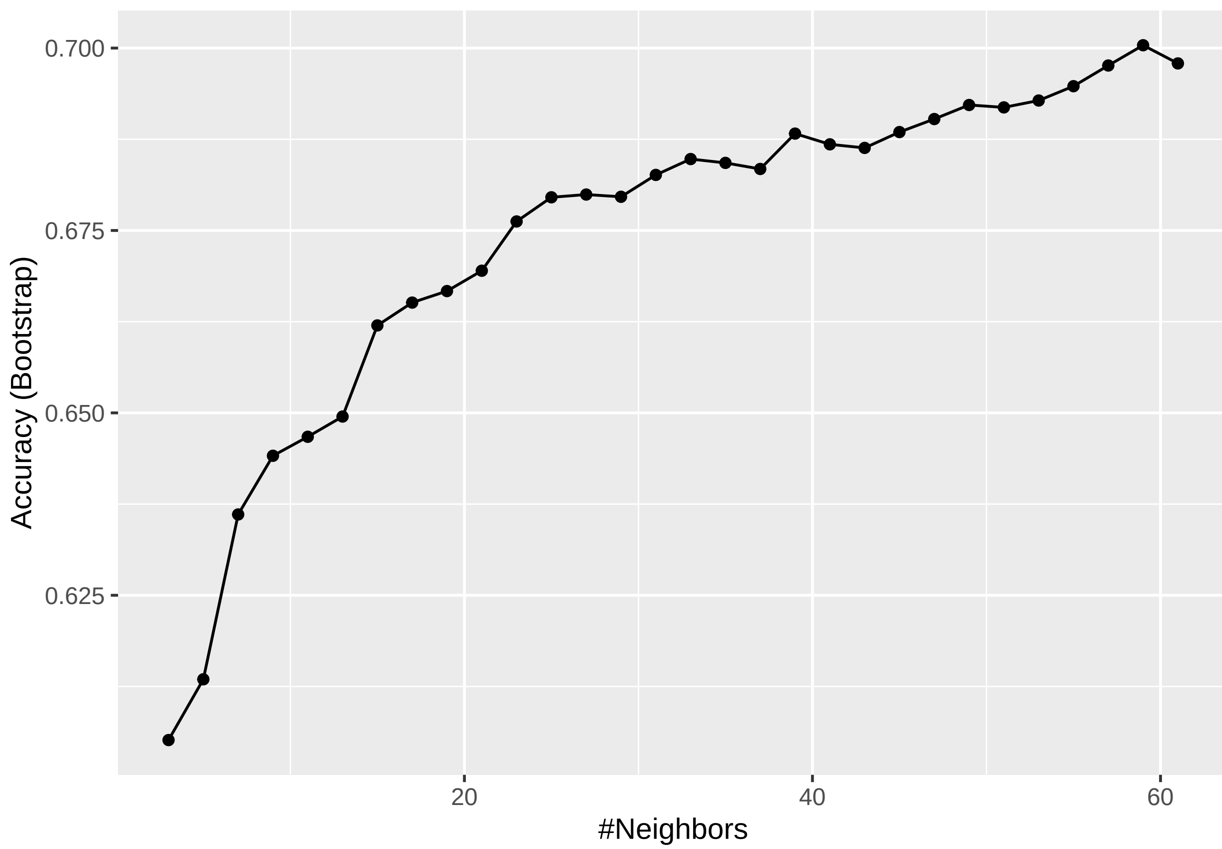
**Model 5: kNN**

Considering the relatively high number of variables, kNN is not expected to outperform the other methods.

```
control <- trainControl(method = "cv", number = 10, p = .9)

# Fit the model
fit_knn <- train(Risk ~ ., method = "knn",
                 data = credit_train,
                 tuneGrid = data.frame(k = seq(3, 61, 2)))
```

Optimal K parameter

```
# Optimal K parameter
ggplot(fit_knn)
```




```
fit_knn$bestTune
```

```
##     k
## 29 59
```

**kNN results**

```r
# Calculate predictions using fitted model and check results
y_hat_knn <- predict(fit_knn, credit_test, type = "raw")
cm_knn <- confusionMatrix(y_hat_knn, credit_test$Risk)
Acc_knn <- cm_knn$overall[["Accuracy"]]
F1_knn <- f1(y_hat_knn,credit_test$Risk)

# Save metric in perf_results
perf_results <- bind_rows(perf_results, data_frame(method="kNN", Accuracy = Acc_knn, F1_score = F1_knn )
perf_results %>% knitr::kable()
```

| method | Accuracy | F1_score |
|---|---|---|
| Logistic regresion | 0.775 | 0.8432056 |
| Decision tree | 0.745 | 0.8305648 |
| Random forest | 0.755 | 0.8463950 |
| SVM with Linear Kernel | 0.770 | 0.8413793 |
| kNN | 0.710 | 0.8263473 |

# Results

This is the final result

```
perf_results %>% knitr::kable()
```

| method | Accuracy | F1_score |
|---|---|---|
| Logistic regresion | 0.775 | 0.8432056 |
| Decision tree | 0.745 | 0.8305648 |
| Random forest | 0.755 | 0.8463950 |
| SVM with Linear Kernel | 0.770 | 0.8413793 |
| kNN | 0.710 | 0.8263473 |

The highest value for Accuracy is

```
## [1] 0.775
```

provided by the **logistic regression** model.

# Conclusion

A model to predict credit risk profile qualities has been built by testing different approaches and choosing the one with best results.

The optimal model accounts for the variability due to the different features available in the dataset.

The final accuracy obtained with the optimal model, which turned out to be **logistic regression**, is

```
## [1] 0.775
```