Capstone CYO Project Report

Prediction of Credit Risk for German Bank

Endri Raco

Contents

Dedication

This project and all my work is dedicated to victims of Albanian earthquake 26 November 2019.

Acknowledgement

I would like to express my special thanks of gratitude to Prof. Rafael Irizarry for the wonderful material and thorough explanations he provided during all courses. Also I want to thank my friends of this course who share the same interests for Data Science.

Introduction

Investopedia webpage defines **Credit risk** as the possibility of a loss resulting from a borrower's failure to repay a loan or meet contractual obligations.(Investopedia 2019)

By this definition, we understand that subject who gives the loan (in our case the bank) takes a risk during this process, because it is not known a priori if client who applies and takes the loan will not default (fail to return the money).

Nowadays, banks and other financial institutions are taking advantange of machine learning methods to predict credit risks (refuse loans to risky customers).

The aim of this project is to use machine learning techniques such as logistic regression and decision trees for building a predictive model based on bank

client's features. This model should help on understanding which client can be considered as a risky one.

We will use dataset South German Credit. This dataset is a polished version of [German credit dataset] ('https://archive.ics.uci.edu/ml/machine-learning-databases/statlog/german/german.data') from **UCI Machine Learning Repository** portal, donated by the German professor Hans Hofmann via the European Statlog project.

German credit dataset from the UCI Machine Learning Repository, donated by the German professor Hans Hofmann via the European Statlog project, comes with an incorrect code table. Many variables are wrongly represented, which implies that the data cannot be adequately used for experimenting with methods for interpretable machine learning.

The **South German Credit** data are meaningful, credit scoring data from southern Germany In dataset, each entry represents a person who takes a credit by a bank.(Grömping 2019) Each person is classified as good or bad credit risks according to the set of attributes.

For all project calculations is used the following PC:

```
print("Operating System:")
```

```
## [1] "Operating System:"
```

```
version
```

```
##                _
## platform       x86_64-w64-mingw32
## arch           x86_64
## os             mingw32
## system         x86_64, mingw32
## status
## major          3
## minor          6.1
## year           2019
## month          07
## day            05
## svn rev        76782
## language       R
## version.string R version 3.6.1 (2019-07-05)
## nickname       Action of the Toes
```

Methods and Analysis

Importing data

For importing data in our project **data** folder we will use **readr** library, which is part of the **tidyverse** (Wickham 2019)package:

```r
# South German Credit Dataset:
# https://data.ub.uni-muenchen.de/23/2/kredit.asc Code for
# creating SouthGermanCredit.asc as described in
# @gromping2019
temp <- read.table("https://data.ub.uni-muenchen.de/23/2/kredit.asc",
    header = TRUE)
### recode pers and gastarb to the stated P2 coding
temp$pers <- 3 - temp$pers
temp$gastarb <- 3 - temp$gastarb
### put credit_risk is last
temp <- cbind(temp[, -1], kredit = temp$kredit)
write.table(temp, file = "./data/SouthGermanCredit.asc", row.names = FALSE,
    quote = FALSE)
## save dataset in our data folder
data_credit <- read.table("./data/SouthGermanCredit.asc", header = TRUE)
# remove temp file
remove(temp)
```

Let's have a look in the structure of downloaded dataset:

```r
# Check structure of downloaded dataset
str(data_credit)
```

```
## 'data.frame':    1000 obs. of  21 variables:
##  $ V1 : Factor w/ 4 levels "A11","A12","A13",..: 1 2 4 1 1 4 4 2 4 2 ...
##  $ V2 : int  6 48 12 42 24 36 24 36 12 30 ...
##  $ V3 : Factor w/ 5 levels "A30","A31","A32",..: 5 3 5 3 4 3 3 3 3 5 ...
##  $ V4 : Factor w/ 10 levels "A40","A41","A410",..: 5 5 8 4 1 8 4 2 5 1 ...
##  $ V5 : int  1169 5951 2096 7882 4870 9055 2835 6948 3059 5234 ...
##  $ V6 : Factor w/ 5 levels "A61","A62","A63",..: 5 1 1 1 1 5 3 1 4 1 ...
##  $ V7 : Factor w/ 5 levels "A71","A72","A73",..: 5 3 4 4 3 3 5 3 4 1 ...
##  $ V8 : int  4 2 2 2 3 2 3 2 2 4 ...
##  $ V9 : Factor w/ 4 levels "A91","A92","A93",..: 3 2 3 3 3 3 3 3 1 4 ...
##  $ V10: Factor w/ 3 levels "A101","A102",..: 1 1 1 3 1 1 1 1 1 1 ...
##  $ V11: int  4 2 3 4 4 4 4 2 4 2 ...
##  $ V12: Factor w/ 4 levels "A121","A122",..: 1 1 1 2 4 4 2 3 1 3 ...
##  $ V13: int  67 22 49 45 53 35 53 35 61 28 ...
##  $ V14: Factor w/ 3 levels "A141","A142",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ V15: Factor w/ 3 levels "A151","A152",..: 2 2 2 3 3 3 2 1 2 2 ...
##  $ V16: int  2 1 1 1 2 1 1 1 1 2 ...
##  $ V17: Factor w/ 4 levels "A171","A172",..: 3 3 2 3 3 2 3 4 2 4 ...
##  $ V18: int  1 1 2 2 2 2 1 1 1 1 ...
##  $ V19: Factor w/ 2 levels "A191","A192": 2 1 1 1 1 2 1 2 1 1 ...
##  $ V20: Factor w/ 2 levels "A201","A202": 1 1 1 1 1 1 1 1 1 1 ...
##  $ V21: int  1 2 1 1 2 1 1 1 1 2 ...
```

We are dealing with an object of class **dataframe** containing 21 columns(variables) and 1000 rows(observations).

Data Processing

Since variables are missing names, it is hard initially to understand information represented in dataframe. According to *Table 1: Distribution of categorical predictor variables for the South German Credit data, separately for good and bad credit risks* in gromping2019 we have all variables as numeric variables. Let's check this fact in the downloaded dataframe.

```
split(names(data_credit), sapply(data_credit, function(x) paste(class(x),
    collapse = " ")))
```

```
## $integer
##  [1] "laufkont" "laufzeit" "moral"    "verw"     "hoehe"    "sparkont"
##  [7] "beszeit"  "rate"     "famges"   "buerge"   "wohnzeit" "verm"
## [13] "alter"    "weitkred" "wohn"     "bishkred" "beruf"    "pers"
## [19] "telef"    "gastarb"  "kredit"
```

Everything seems OK. In the results we see numeric variable **credit**. This column contains bank evaluation for customer (1 = Good, 0 = Bad). We will rename first 21 columns with more understandable english names so we can work them easily in the future analysis.

```
# Rename columns
data_credit <- setNames(data_credit, c("account_status", "duration_month",
    "credit_history", "credit_purpose", "credit_amount", "savings_account",
    "employment_present", "installment_rate_pct", "status_sex",
    "other_debtors_guar", "residence_duration", "property", "age_years",
    "other_install_plans", "housing", "exist_credits_nr", "job",
    "dependents_nr", "telephone_nr", "foreign_worker", "customer_good_bad"))
```

Let's print again the structure of dataset:

```
# Check structure of downloaded dataset
str(data_credit)
```

```
## 'data.frame':    1000 obs. of  21 variables:
##  $ account_status      : Factor w/ 4 levels "A11","A12","A13",..: 1 2 4 1 1 4 4 2 4 2 .
##  $ duration_month      : int  6 48 12 42 24 36 24 36 12 30 ...
##  $ credit_history      : Factor w/ 5 levels "A30","A31","A32",..: 5 3 5 3 4 3 3 3 3 5 .
##  $ credit_purpose      : Factor w/ 10 levels "A40","A41","A410",..: 5 5 8 4 1 8 4 2 5 1
##  $ credit_amount       : int  1169 5951 2096 7882 4870 9055 2835 6948 3059 5234 ...
##  $ savings_account     : Factor w/ 5 levels "A61","A62","A63",..: 5 1 1 1 1 5 3 1 4 1 .
```

```
##  $ employment_present  : Factor w/ 5 levels "A71","A72","A73",..: 5 3 4 4 3 3 5 3 4 1 .
##  $ installment_rate_pct: int  4 2 2 2 3 2 3 2 2 4 ...
##  $ status_sex          : Factor w/ 4 levels "A91","A92","A93",..: 3 2 3 3 3 3 3 3 1 4 .
##  $ other_debtors_guar  : Factor w/ 3 levels "A101","A102",..: 1 1 1 3 1 1 1 1 1 1 ...
##  $ residence_duration  : int  4 2 3 4 4 4 4 2 4 2 ...
##  $ property            : Factor w/ 4 levels "A121","A122",..: 1 1 1 2 4 4 2 3 1 3 ...
##  $ age_years           : int  67 22 49 45 53 35 53 35 61 28 ...
##  $ other_install_plans : Factor w/ 3 levels "A141","A142",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ housing             : Factor w/ 3 levels "A151","A152",..: 2 2 2 3 3 3 2 1 2 2 ...
##  $ exist_credits_nr    : int  2 1 1 1 2 1 1 1 1 2 ...
##  $ job                 : Factor w/ 4 levels "A171","A172",..: 3 3 2 3 3 2 3 4 2 4 ...
##  $ dependents_nr       : int  1 1 2 2 2 2 1 1 1 1 ...
##  $ telephone_nr        : Factor w/ 2 levels "A191","A192": 2 1 1 1 1 2 1 2 1 1 ...
##  $ foreign_worker      : Factor w/ 2 levels "A201","A202": 1 1 1 1 1 1 1 1 1 1 ...
##  $ customer_good_bad   : int  1 2 1 1 2 1 1 1 1 2 ...
```

Now we show the first five lines for columns 1 to 3:

```
# First five lines using the function head
kable_styling(kable(head(data_credit[, 1:3], 5), digits = 3,
    row.names = FALSE, align = "c", caption = NULL, format = "latex"),
    latex_options = c("striped", "basic"), position = "center",
    full_width = FALSE)
```

account_status	duration_month	credit_history
1	18	4
1	9	4
2	12	2
1	12	4
1	12	4

Now we need to look if there are any missing values in our dataframe.

```
# Check data_credit dataframe for NA values
sapply(data_credit, function(x) sum(is.na(x)))
```

```
##       account_status       duration_month       credit_history
##                    0                   0                    0
##       credit_purpose        credit_amount      savings_account
##                    0                   0                    0
##   employment_present installment_rate_pct           status_sex
##                    0                   0                    0
##   other_debtors_guar   residence_duration             property
##                    0                   0                    0
##            age_years  other_install_plans              housing
```

```
##                    0                    0                    0
##     exist_credits_nr                  job        dependents_nr
##                    0                    0                    0
##         telephone_nr       foreign_worker    customer_good_bad
##                    0                    0                    0
```

Our dataset has no missing values. From attribute information provided in According to *Table 1: Distribution of categorical predictor variables for the South German Credit data, separately for good and bad credit risks* in gromping2019 we see that our variables **duration__month, credit__amount, age__years** are numeric and others are qualitative (categorical).

Let's transform our categorical variables to class **factor** for using them in our analysis.

```
# convert variables to class factor
variables <- c("account_status", "credit_history", "credit_purpose",
    "savings_account", "employment_present", "installment_rate_pct",
    "status_sex", "other_debtors_guar", "residence_duration",
    "property", "other_install_plans", "housing", "exist_credits_nr",
    "job", "dependents_nr", "telephone_nr", "foreign_worker",
    "customer_good_bad")
data_credit[, variables] <- lapply(data_credit[, variables],
    factor)
```

Result of transformations shown below:

```
# Factors vs numeric variables
split(names(data_credit), sapply(data_credit, function(x) paste(class(x),
    collapse = " ")))
```

```
## $factor
##  [1] "account_status"       "credit_history"       "credit_purpose"
##  [4] "savings_account"      "employment_present"   "installment_rate_pct"
##  [7] "status_sex"           "other_debtors_guar"   "residence_duration"
## [10] "property"             "other_install_plans"  "housing"
## [13] "exist_credits_nr"     "job"                  "dependents_nr"
## [16] "telephone_nr"         "foreign_worker"       "customer_good_bad"
##
## $integer
## [1] "duration_month" "credit_amount"  "age_years"
```

Let's view our dataset on which we will work further

```
# Transformed dataset
str(data_credit)
```

```
## 'data.frame':    1000 obs. of  21 variables:
##  $ account_status      : Factor w/ 4 levels "A11","A12","A13",..: 1 2 4 1 1 4 4 2 4 2 .
##  $ duration_month      : int  6 48 12 42 24 36 24 36 12 30 ...
##  $ credit_history      : Factor w/ 5 levels "A30","A31","A32",..: 5 3 5 3 4 3 3 3 3 5 .
##  $ credit_purpose      : Factor w/ 10 levels "A40","A41","A410",..: 5 5 8 4 1 8 4 2 5 1
##  $ credit_amount       : int  1169 5951 2096 7882 4870 9055 2835 6948 3059 5234 ...
##  $ savings_account     : Factor w/ 5 levels "A61","A62","A63",..: 5 1 1 1 1 5 3 1 4 1 .
##  $ employment_present  : Factor w/ 5 levels "A71","A72","A73",..: 5 3 4 4 3 3 5 3 4 1 .
##  $ installment_rate_pct: Factor w/ 4 levels "1","2","3","4": 4 2 2 2 3 2 3 2 2 4 ...
##  $ status_sex          : Factor w/ 4 levels "A91","A92","A93",..: 3 2 3 3 3 3 3 3 1 4 .
##  $ other_debtors_guar  : Factor w/ 3 levels "A101","A102",..: 1 1 1 3 1 1 1 1 1 1 ...
##  $ residence_duration  : Factor w/ 4 levels "1","2","3","4": 4 2 3 4 4 4 4 2 4 2 ...
##  $ property            : Factor w/ 4 levels "A121","A122",..: 1 1 1 2 4 4 2 3 1 3 ...
##  $ age_years           : int  67 22 49 45 53 35 53 35 61 28 ...
##  $ other_install_plans : Factor w/ 3 levels "A141","A142",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ housing             : Factor w/ 3 levels "A151","A152",..: 2 2 2 3 3 3 2 1 2 2 ...
##  $ exist_credits_nr    : Factor w/ 4 levels "1","2","3","4": 2 1 1 1 2 1 1 1 1 2 ...
##  $ job                 : Factor w/ 4 levels "A171","A172",..: 3 3 2 3 3 2 3 4 2 4 ...
##  $ dependents_nr       : Factor w/ 2 levels "1","2": 1 1 2 2 2 2 1 1 1 1 ...
##  $ telephone_nr        : Factor w/ 2 levels "A191","A192": 2 1 1 1 1 2 1 2 1 1 ...
##  $ foreign_worker      : Factor w/ 2 levels "A201","A202": 1 1 1 1 1 1 1 1 1 1 ...
##  $ customer_good_bad   : int  1 0 1 1 0 1 1 1 1 0 ...
```

Exploring Data

Data Summary

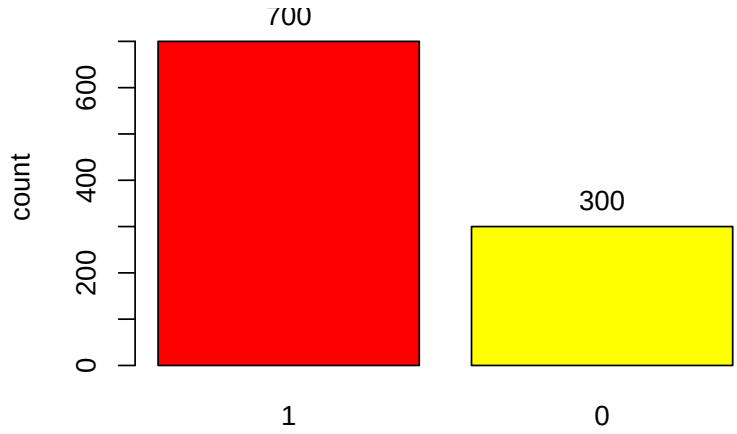
Our main goal is to build a model which should help on understanding which client can be considered as a risky one. As prof. Rafael Irizarry mentions(Irizarry 2019) : In Machine Learning, data comes in the form of:

1. the outcome we want to predict and
2. the features that we will use to predict the outcome

For us outcome we want to predict is variable **customer_good_bad**. This outcome depends on the features that we will use to predict so we need to take a look in all the features and identify the ones of interest for the model.

Let's first have a thorough look at summary statistics for our outcome **customer_good_bad**. We will use function **tab1** from **epiDisplay** library (Chongsuvivatwong 2018).

```
# summary customer_good_bad
tab1(data_credit$customer_good_bad, sort.group = "decreasing",
    cum.percent = FALSE, bar.values = "frequency", cex = 1, cex.names = 1,
    main = "Distribution of data", xlab = "customer_good_bad",
    ylab = "count", col = c("red", "yellow", "blue"))
```

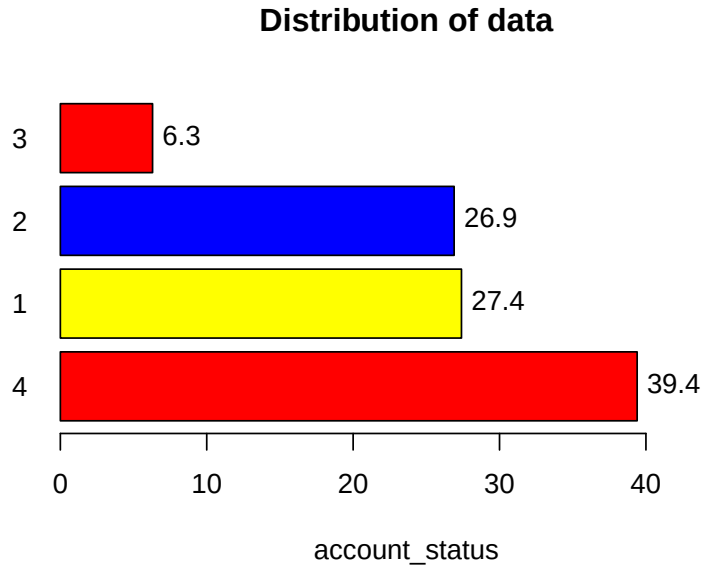



```
## data_credit$customer_good_bad :
##          Frequency Percent
## 1              700      70
## 0              300      30
##   Total       1000     100
```

We see that our outcome variable contains values of 0s and 1s. As mentioned in attribute information value 1 correnspondes to customer status **Good** (reliable) and value 2 correnspondes to status **Bad**(non reliable). As we can see from barchart customers with status **Good** are 700 much more than customers with status **Bad** 300.

Now we go on by taking some insights on other features and see which of them can be potential candidates on affecting our output. First, we start with **account_status**

```
# summary account_status
tab1(data_credit$account_status, sort.group = "decreasing", cum.percent = FALSE,
    bar.values = "percent", cex = 1, cex.names = 1, main = "Distribution of data",
    xlab = "account_status", ylab = "count", col = c("red", "yellow",
        "blue"), horiz = TRUE)
```



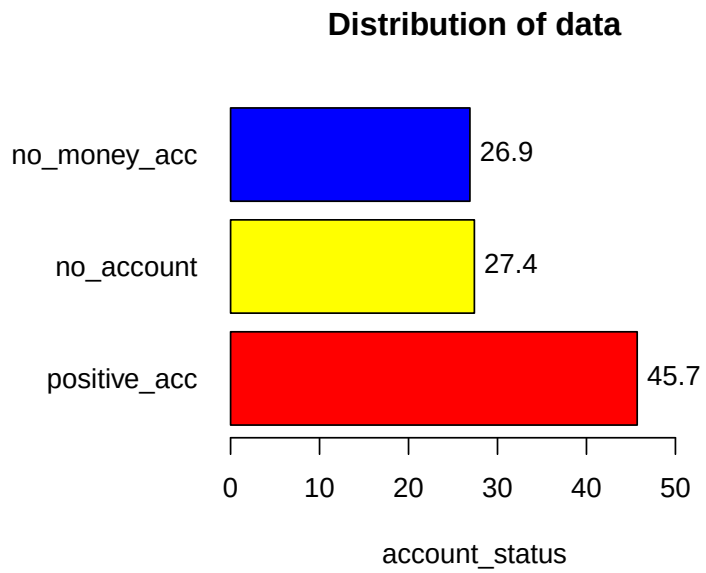

```
## data_credit$account_status :
##         Frequency Percent
## 4             394    39.4
## 1             274    27.4
## 2             269    26.9
## 3              63     6.3
##   Total      1000   100.0
```

We see that categorical variable **account_status** has 4 levels. According to *Table 1: Distribution of categorical predictor variables for the South German Credit data, separately for good and bad credit risks* in gromping2019 , We notice that largest percent of customers (39.4 %) have account status **>= 200 DM**. Beside that only 6.3 % have balance less than 200 DM. This level has very few observations comparing to other levels so maybe it is a good idea to recode variable **account_status** in only 3 levels and giving to these labels some more informative labels. Based on attribute meaning our 3 new levels for this variable will be: **no_account, no_money_acc, positive_acc**

```
# recode account_status
account_status_temp <- recode(data_credit$account_status, `3` = "positive_acc",
    `4` = "positive_acc", `1` = "no_account", `2` = "no_money_acc")
data_credit$account_status <- account_status_temp
```

The results of recoding:

```r
# recoding results
tab1(data_credit$account_status, sort.group = "decreasing", cum.percent = FALSE,
    bar.values = "percent", cex = 1, cex.names = 1, main = "Distribution of data",
    xlab = "account_status", ylab = "count", col = c("red", "yellow",
        "blue"), horiz = TRUE)
```



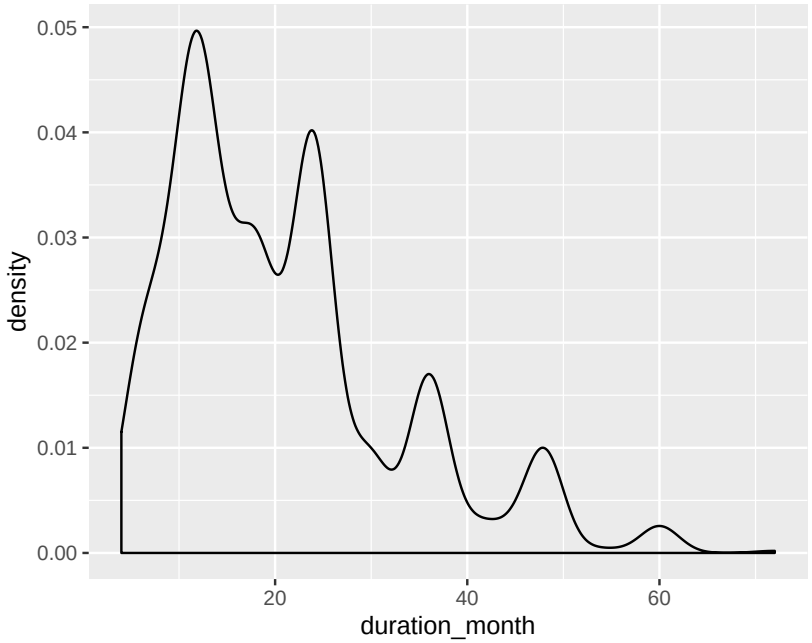

```
## data_credit$account_status :
##               Frequency Percent
## positive_acc        457    45.7
## no_account          274    27.4
## no_money_acc        269    26.9
##   Total            1000   100.0
```

Now let's continue with our numeric variable **duration_month** .

```r
# summary duration_month
summary(data_credit$duration_month)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     4.0    12.0    18.0    20.9    24.0    72.0
```

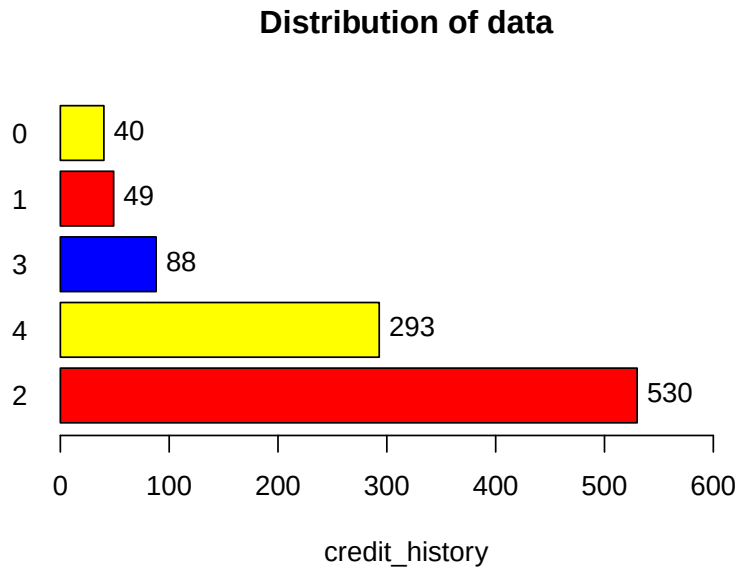
```
data_credit %>% ggplot(aes(duration_month)) + geom_density()
```



We notice that credits have a minimum duration of 4 months and a maximum duration of 72 months. The average duration of credits is 20.9 months.

Next feature is **credit_history**.

```
# summary credit_history
tab1(data_credit$credit_history, sort.group = "decreasing", cum.percent = FALSE,
    bar.values = "frequency", cex = 1, cex.names = 1, main = "Distribution of data",
    xlab = "credit_history", ylab = "count", col = c("red", "yellow",
        "blue"), horiz = TRUE)
```



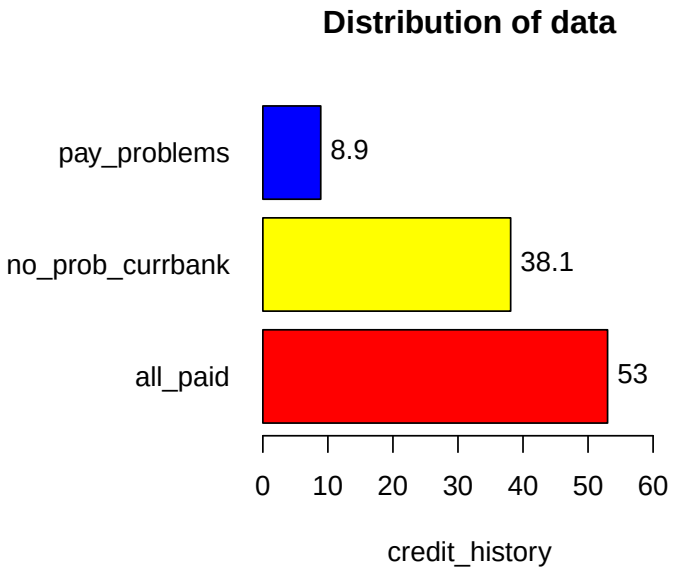

```
## data_credit$credit_history :
##         Frequency Percent
## 2             530    53.0
## 4             293    29.3
## 3              88     8.8
## 1              49     4.9
## 0              40     4.0
##   Total      1000   100.0
```

We see that **credit_history** variable has 5 levels. According to *Table 1: Distribution of categorical predictor variables for the South German Credit data, separately for good and bad credit risks* in gromping2019 , We see that most clients have no credits taken/all credits paid back duly (53 % of customers). Levels 0, 1, 3 (4% , 4.9%, 8.8%) has very few observations comparing to other levels so maybe it is a good idea to recode this variable in only 3 levels : **pay_problems, all_paid, no_prob_currbank**

```
# recode credit_history
credit_history_temp <- recode(data_credit$credit_history, `0` = "pay_problems",
    `1` = "pay_problems", `2` = "all_paid", `3` = "no_prob_currbank",
    `4` = "no_prob_currbank")
data_credit$credit_history <- credit_history_temp
```

The results of recoding:

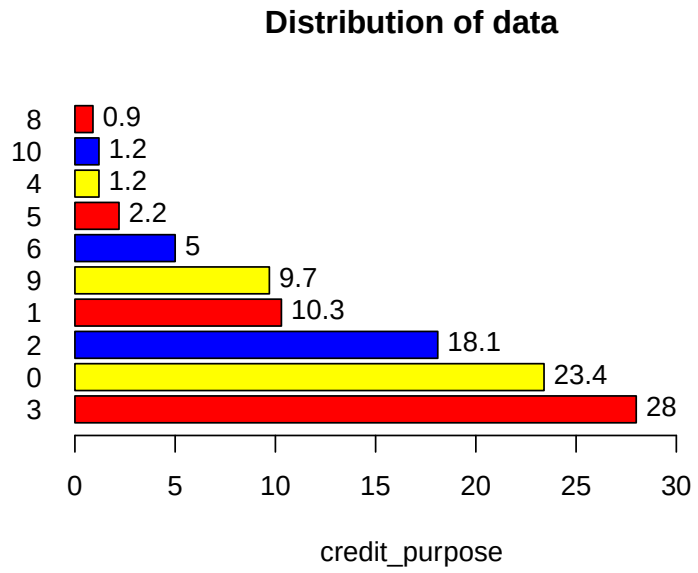
```r
# recoding results
tab1(data_credit$credit_history, sort.group = "decreasing", cum.percent = FALSE,
    bar.values = "percent", cex = 1, cex.names = 1, main = "Distribution of data",
    xlab = "credit_history", ylab = "count", col = c("red", "yellow",
        "blue"), horiz = TRUE)
```




```
## data_credit$credit_history :
##                   Frequency Percent
## all_paid                530    53.0
## no_prob_currbank        381    38.1
## pay_problems             89     8.9
##   Total                1000   100.0
```

Next we move to feature **credit_purpose**.

```r
# summary credit_purpose
tab1(data_credit$credit_purpose, sort.group = "decreasing", cum.percent = FALSE,
    bar.values = "percent", main = "Distribution of data", xlab = "credit_purpose",
    ylab = "count", col = c("red", "yellow", "blue"), horiz = TRUE)
```




```
## data_credit$credit_purpose :
##         Frequency Percent
## 3             280    28.0
## 0             234    23.4
## 2             181    18.1
## 1             103    10.3
## 9              97     9.7
## 6              50     5.0
## 5              22     2.2
## 4              12     1.2
## 10             12     1.2
## 8               9     0.9
##   Total      1000   100.0
```

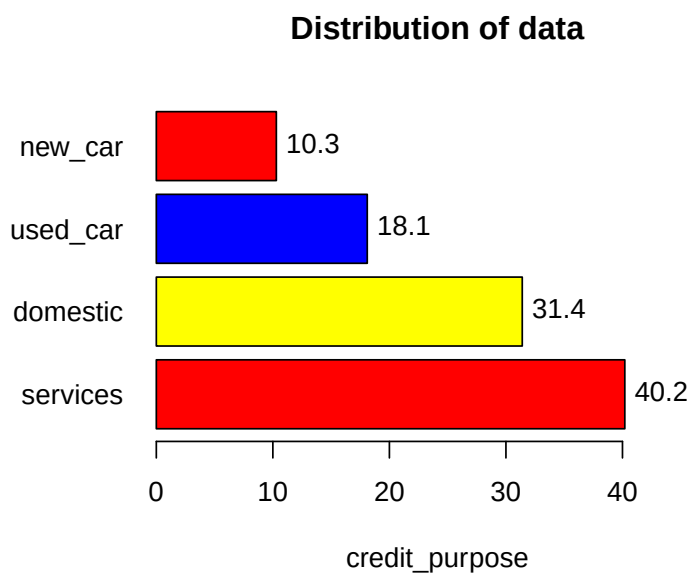
According to *Table 1: Distribution of categorical predictor variables for the South German Credit data, separately for good and bad credit risks* in gromping2019 „ this categorical variable should have 11 levels, but in our plot we see only 10. Level 7(education) has no observations. Levels 8, 10, 4, 5, 6 have very few observations comparing to other levels so maybe it is a good idea to recode this variable in 4 levels: **new_car, used_car, domestic, services**.

nbsp;

```
# recode credit_purpose
credit_purpose_temp <- recode(data_credit$credit_purpose, `0` = "services",
    `1` = "new_car", `2` = "used_car", `3` = "domestic", `4` = "domestic",
    `5` = "domestic", `6` = "services", `7` = "services", `8` = "services",
    `9` = "services", `10` = "services")
data_credit$credit_purpose <- credit_purpose_temp
```

The results of recoding:

```
# recoding results
tab1(data_credit$credit_purpose, sort.group = "decreasing", cum.percent = FALSE,
    bar.values = "percent", cex = 1, cex.names = 1, main = "Distribution of data",
    xlab = "credit_purpose", ylab = "count", col = c("red", "yellow",
        "blue"), horiz = TRUE)
```



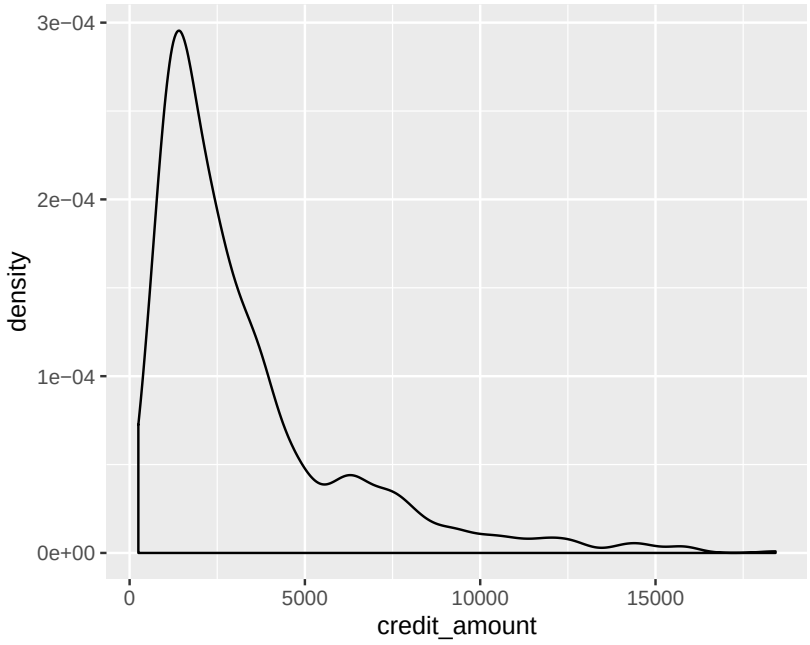

```
## data_credit$credit_purpose :
##          Frequency Percent
## services       402    40.2
## domestic       314    31.4
## used_car       181    18.1
## new_car        103    10.3
##   Total       1000   100.0
```

Now let's continue with our numeric variable **credit_amount** .

```
# summary of credit_amount
summary(data_credit$credit_amount)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     250    1366    2320    3271    3972   18424
```

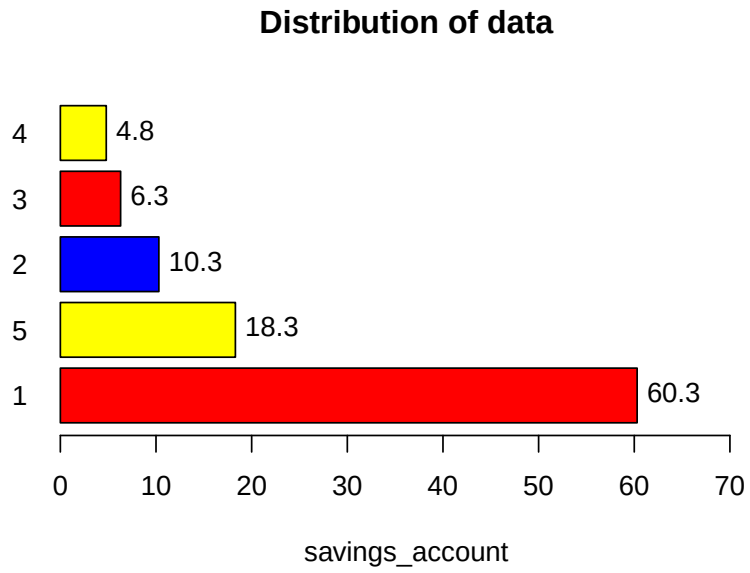
```
data_credit %>% ggplot(aes(credit_amount)) + geom_density()
```



We notice that credits required by bank clients have an average of 3271 DM.

Next we move to feature **savings_account**.

```
# summary savings_account
tab1(data_credit$savings_account, sort.group = "decreasing",
    cum.percent = FALSE, bar.values = "percent", main = "Distribution of data",
    xlab = "savings_account", ylab = "count", col = c("red",
        "yellow", "blue"), horiz = TRUE)
```




```
## data_credit$savings_account :
##          Frequency Percent
## 1              603    60.3
## 5              183    18.3
## 2              103    10.3
## 3               63     6.3
## 4               48     4.8
##   Total       1000   100.0
```

According to *Table 1: Distribution of categorical predictor variables for the South German Credit data, separately for good and bad credit risks* in gromping2019 „ this categorical variable has 5 levels. Levels 3 and 4 have very few observations comparing to other levels so maybe it is a good idea to recode this variable in 4 new levels **no_sav, less100, 100to1000, over1000**

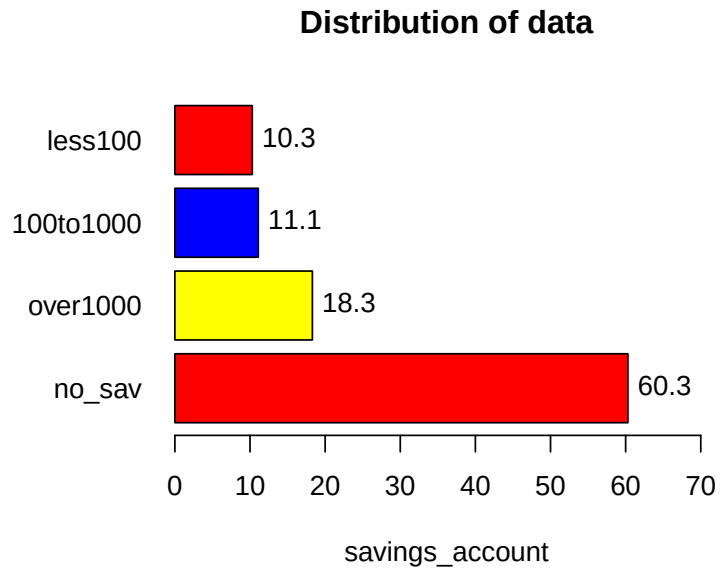
nbsp;

```
# recode savings_account
savings_account_temp <- recode(data_credit$savings_account, `1` = "no_sav",
    `2` = "less100", `3` = "100to1000", `4` = "100to1000", `5` = "over1000")
data_credit$savings_account <- savings_account_temp
```

The results of recoding:

```
# recoding results
tab1(data_credit$savings_account, sort.group = "decreasing",
    cum.percent = FALSE, bar.values = "percent", cex = 1, cex.names = 1,
```

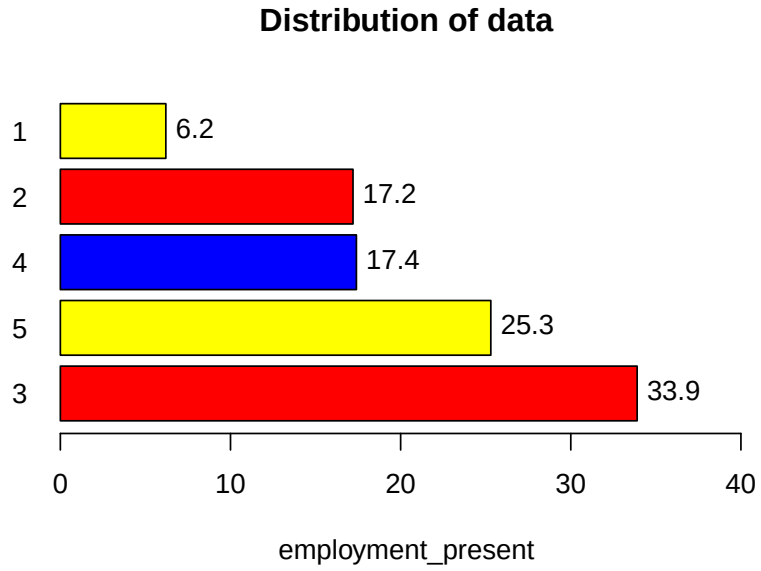
```
    main = "Distribution of data", xlab = "savings_account",
    ylab = "count", col = c("red", "yellow", "blue"), horiz = TRUE)
```




```
## data_credit$savings_account :
##            Frequency Percent
## no_sav           603    60.3
## over1000         183    18.3
## 100to1000        111    11.1
## less100          103    10.3
##   Total         1000   100.0
```

Now let's see feature **employment_present**.

```
# summary employment_present
tab1(data_credit$employment_present, sort.group = "decreasing",
    cum.percent = FALSE, bar.values = "percent", main = "Distribution of data",
    xlab = "employment_present", ylab = "count", col = c("red",
        "yellow", "blue"), horiz = TRUE)
```




```
## data_credit$employment_present :
##         Frequency Percent
## 3             339    33.9
## 5             253    25.3
## 4             174    17.4
## 2             172    17.2
## 1              62     6.2
##   Total      1000   100.0
```

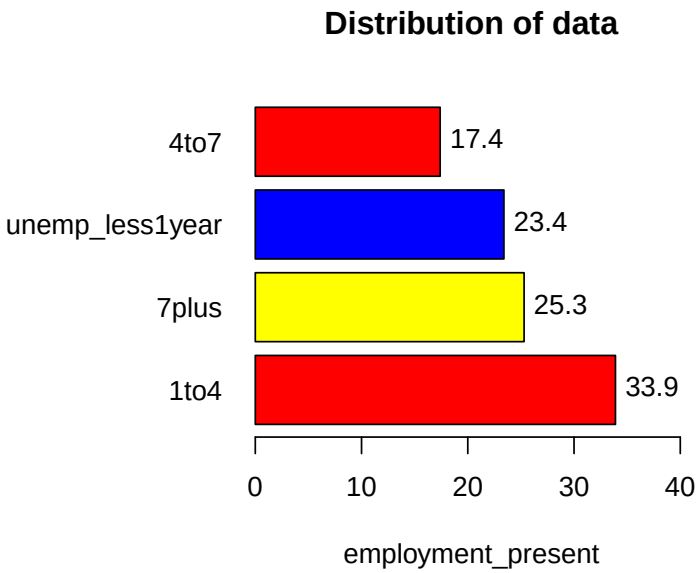
According to *Table 1: Distribution of categorical predictor variables for the South German Credit data, separately for good and bad credit risks* in gromping2019 „ this categorical variable has 5 levels. Levels 1 has very few observations comparing to other levels so maybe it is a good idea to recode this variable in 4 levels: **unemp_less1year, 1to4, 4to7, 7plus**

nbsp;

```
# recode employment_present
employment_present_temp <- recode(data_credit$employment_present,
    `1` = "unemp_less1year", `2` = "unemp_less1year", `3` = "1to4",
    `4` = "4to7", `5` = "7plus")
data_credit$employment_present <- employment_present_temp
```

The results of recoding:

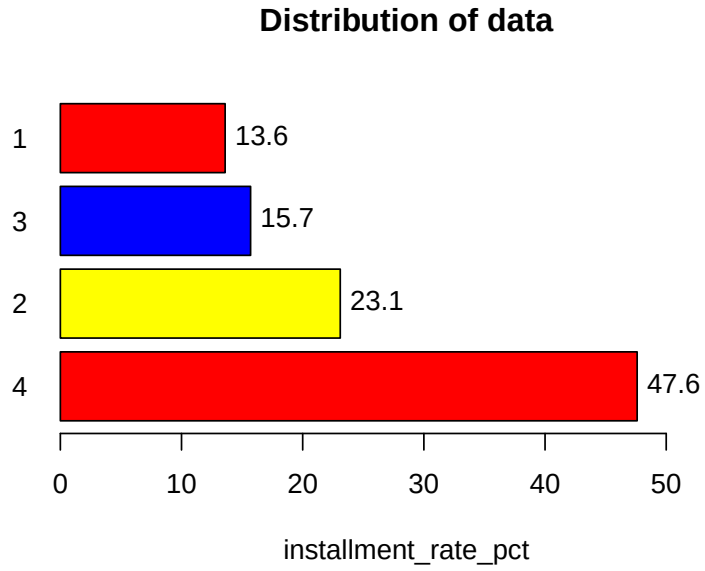
```r
# recoding results
tab1(data_credit$employment_present, sort.group = "decreasing",
    cum.percent = FALSE, bar.values = "percent", cex = 1, cex.names = 1,
    main = "Distribution of data", xlab = "employment_present",
    ylab = "count", col = c("red", "yellow", "blue"), horiz = TRUE)
```




```
## data_credit$employment_present :
##                  Frequency Percent
## 1to4                   339    33.9
## 7plus                  253    25.3
## unemp_less1year        234    23.4
## 4to7                   174    17.4
##   Total               1000   100.0
```

Now let's see feature **installment_rate_pct**.

```r
# summary installment_rate_pct
tab1(data_credit$installment_rate_pct, sort.group = "decreasing",
    cum.percent = FALSE, bar.values = "percent", main = "Distribution of data",
    xlab = "installment_rate_pct", ylab = "count", col = c("red",
        "yellow", "blue"), horiz = TRUE)
```

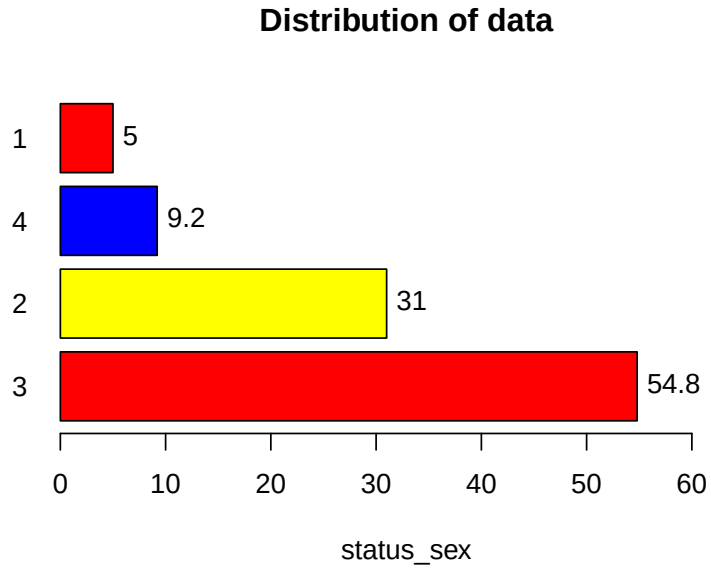



```
## data_credit$installment_rate_pct :
##          Frequency Percent
## 4              476    47.6
## 2              231    23.1
## 3              157    15.7
## 1              136    13.6
##   Total       1000   100.0
```

According to *Table 1: Distribution of categorical predictor variables for the South German Credit data, separately for good and bad credit risks* in gromping2019 , this categorical variable has 4 levels. This levels are shown also in plot and frequency table. We will not transform this variable.

Now let's see feature **status_sex**.

```
# summary status_sex
tab1(data_credit$status_sex, sort.group = "decreasing", cum.percent = FALSE,
    bar.values = "percent", main = "Distribution of data", xlab = "status_sex",
    ylab = "count", col = c("red", "yellow", "blue"), horiz = TRUE)
```




```
## data_credit$status_sex :
##          Frequency Percent
## 3              548    54.8
## 2              310    31.0
## 4               92     9.2
## 1               50     5.0
##   Total       1000   100.0
```

According to *Table 1: Distribution of categorical predictor variables for the South German Credit data, separately for good and bad credit risks* in gromping2019 „ this categorical variable has 4 levels. We see that largest part of bank customers fall in group 3 (Male married or widowed). Levels 1 and 4 have very few observations comparing to other levels so maybe it is a good idea to recode this variable in 3 levels **m_single_divorced, m_married_wid, female**

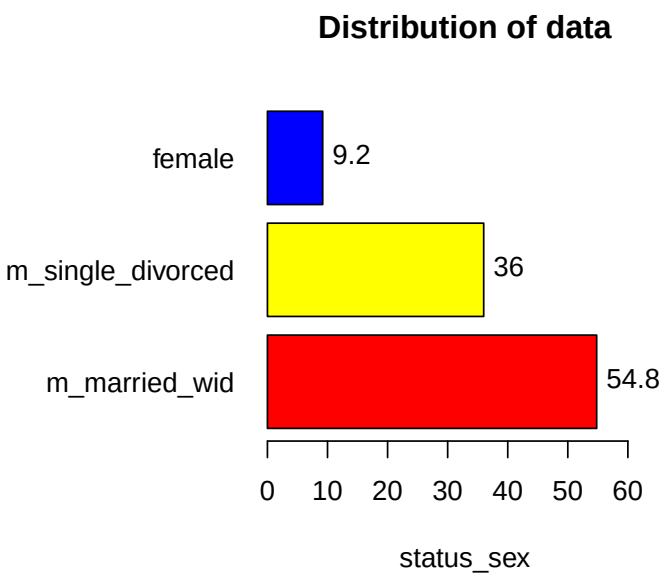
nbsp;

```
# recode status_sex
status_sex_temp <- recode(data_credit$status_sex, `1` = "m_single_divorced",
    `2` = "m_single_divorced", `3` = "m_married_wid", `4` = "female")
data_credit$status_sex <- status_sex_temp
```

The results of recoding:

```
# recoding results
tab1(data_credit$status_sex, sort.group = "decreasing", cum.percent = FALSE,
```

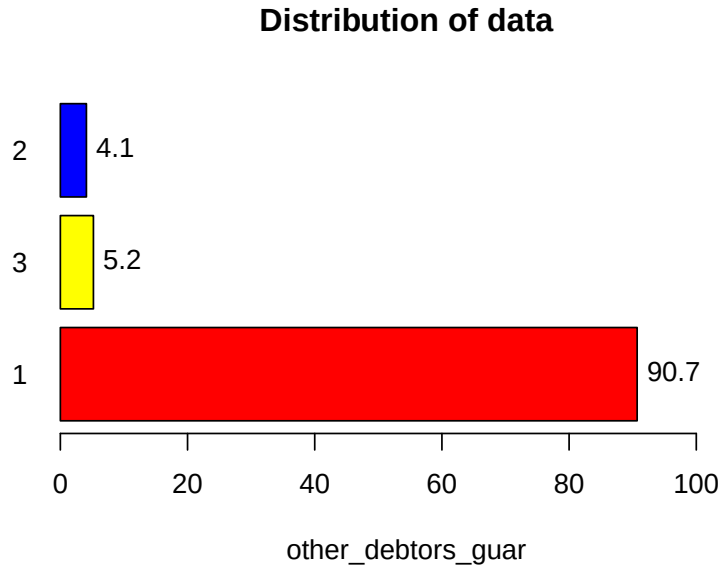
```
    bar.values = "percent", cex = 1, cex.names = 1, main = "Distribution of data",
    xlab = "status_sex", ylab = "count", col = c("red", "yellow",
        "blue"), horiz = TRUE)
```




```
## data_credit$status_sex :
##                    Frequency Percent
## m_married_wid            548    54.8
## m_single_divorced        360    36.0
## female                    92     9.2
##   Total                 1000   100.0
```

Now let's see feature **other_debtors_guar**.

```
# summary other_debtors_guar
tab1(data_credit$other_debtors_guar, sort.group = "decreasing",
    cum.percent = FALSE, bar.values = "percent", main = "Distribution of data",
    xlab = "other_debtors_guar", ylab = "count", col = c("red",
        "yellow", "blue"), horiz = TRUE)
```




```
## data_credit$other_debtors_guar :
##          Frequency Percent
## 1              907    90.7
## 3               52     5.2
## 2               41     4.1
##   Total       1000   100.0
```

According to *Table 1: Distribution of categorical predictor variables for the South German Credit data, separately for good and bad credit risks* in gromping2019 , this categorical variable has 3 levels. This variable has data about known debtors / guarantors so we can recode this variable to 2 levels **yes, no** because levels 2 and 3 have very few observations comparing to other levels.

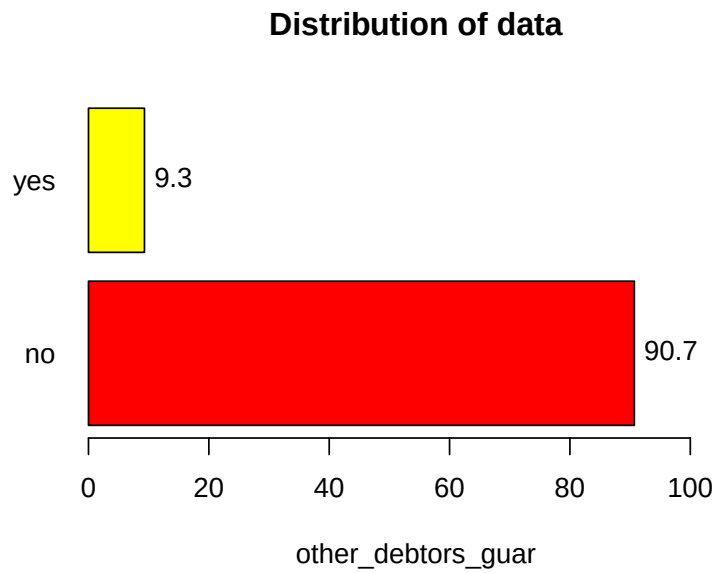
nbsp;

```
# recode other_debtors_guar
other_debtors_guar_temp <- recode(data_credit$other_debtors_guar,
    `1` = "no", `2` = "yes", `3` = "yes")
data_credit$other_debtors_guar <- other_debtors_guar_temp
```

The results of recoding:

```
# recoding results
tab1(data_credit$other_debtors_guar, sort.group = "decreasing",
    cum.percent = FALSE, bar.values = "percent", cex = 1, cex.names = 1,
    main = "Distribution of data", xlab = "other_debtors_guar",
```

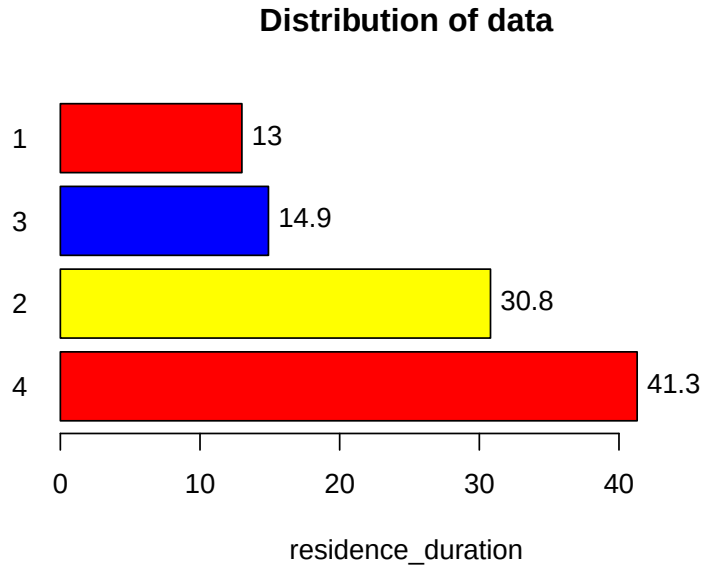
```
    ylab = "count", col = c("red", "yellow", "blue"), horiz = TRUE)
```




```
## data_credit$other_debtors_guar :
##          Frequency Percent
## no             907    90.7
## yes             93     9.3
##   Total       1000   100.0
```

Now let's see feature **residence_duration**.

```
# summary residence_duration
tab1(data_credit$residence_duration, sort.group = "decreasing",
    cum.percent = FALSE, bar.values = "percent", main = "Distribution of data",
    xlab = "residence_duration", ylab = "count", col = c("red",
        "yellow", "blue"), horiz = TRUE)
```

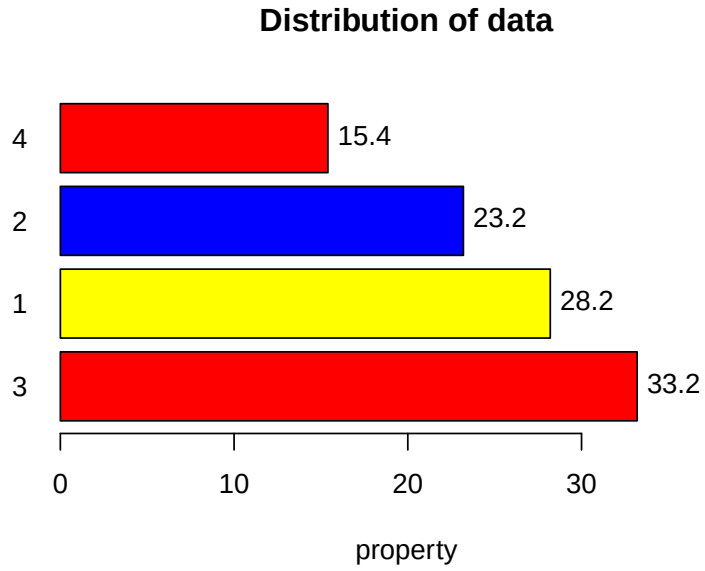



```
## data_credit$residence_duration :
##         Frequency Percent
## 4             413    41.3
## 2             308    30.8
## 3             149    14.9
## 1             130    13.0
##   Total      1000   100.0
```

According to *Table 1: Distribution of categorical predictor variables for the South German Credit data, separately for good and bad credit risks* in gromping2019 „ this categorical variable has 4 levels. This variable will stay as it is.

Now let's see feature **property**.

```
# summary property
tab1(data_credit$property, sort.group = "decreasing", cum.percent = FALSE,
    bar.values = "percent", main = "Distribution of data", xlab = "property",
    ylab = "count", col = c("red", "yellow", "blue"), horiz = TRUE)
```




```
## data_credit$property :
##         Frequency Percent
## 3             332    33.2
## 1             282    28.2
## 2             232    23.2
## 4             154    15.4
##   Total      1000   100.0
```

According to *Table 1: Distribution of categorical predictor variables for the South German Credit data, separately for good and bad credit risks* in gromping2019 „ this categorical variable has 4 levels. We will not change this variable.

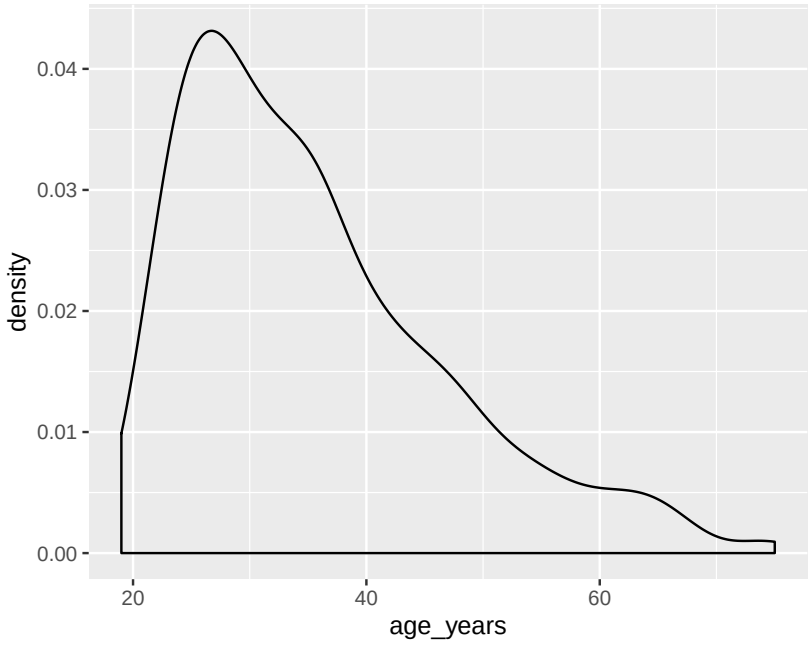
nbsp;

We will continue with numeric feature **age__years**.

```
# summary age_years
summary(data_credit$age_years)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   19.00   27.00   33.00   35.55   42.00   75.00
```

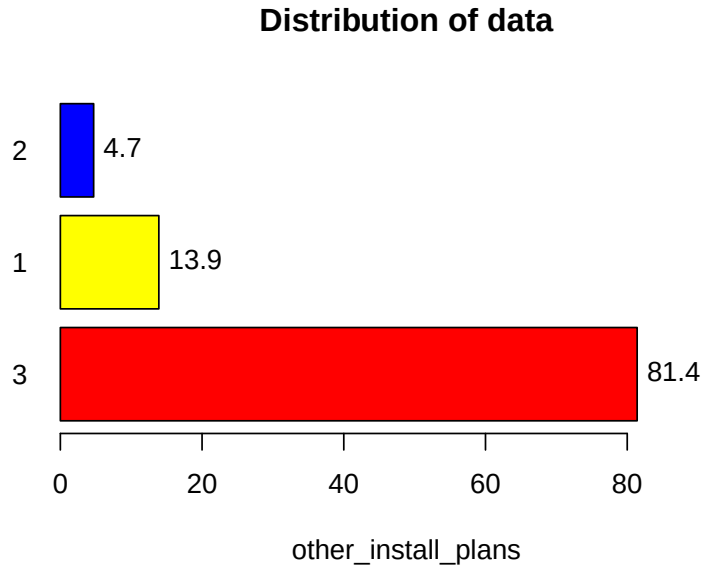
```
data_credit %>% ggplot(aes(age_years)) + geom_density()
```



We notice that youngest customer is of age 19 and in the average bank customers are between 35-36 years old.

Now let's see feature **other_install_plans**.

```
# summary other_install_plans
tab1(data_credit$other_install_plans, sort.group = "decreasing",
    cum.percent = FALSE, bar.values = "percent", main = "Distribution of data",
    xlab = "other_install_plans", ylab = "count", col = c("red",
        "yellow", "blue"), horiz = TRUE)
```




```
## data_credit$other_install_plans :
##         Frequency Percent
## 3             814    81.4
## 1             139    13.9
## 2              47     4.7
##   Total      1000   100.0
```

According to *Table 1: Distribution of categorical predictor variables for the South German Credit data, separately for good and bad credit risks* in gromping2019 „ this categorical variable has 3 levels and has data about other credits that customer has in other banks or stores . We will recode this variable to 2 levels **yes, no** because level 2 has very few observations comparing to other levels.

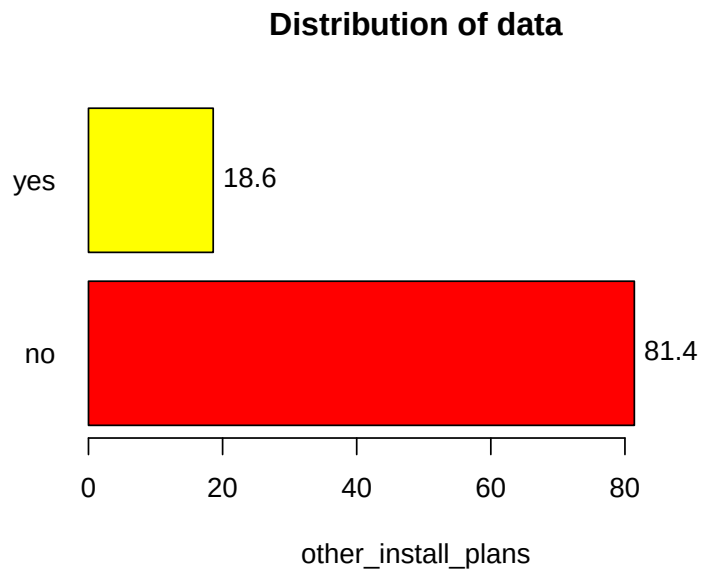
nbsp;

```
# recode other_install_plans
other_install_plans_temp <- recode(data_credit$other_install_plans,
    `1` = "yes", `2` = "yes", `3` = "no")
data_credit$other_install_plans <- other_install_plans_temp
```

The results of recoding:

```
# recoding results
tab1(data_credit$other_install_plans, sort.group = "decreasing",
    cum.percent = FALSE, bar.values = "percent", cex = 1, cex.names = 1,
    main = "Distribution of data", xlab = "other_install_plans",
```

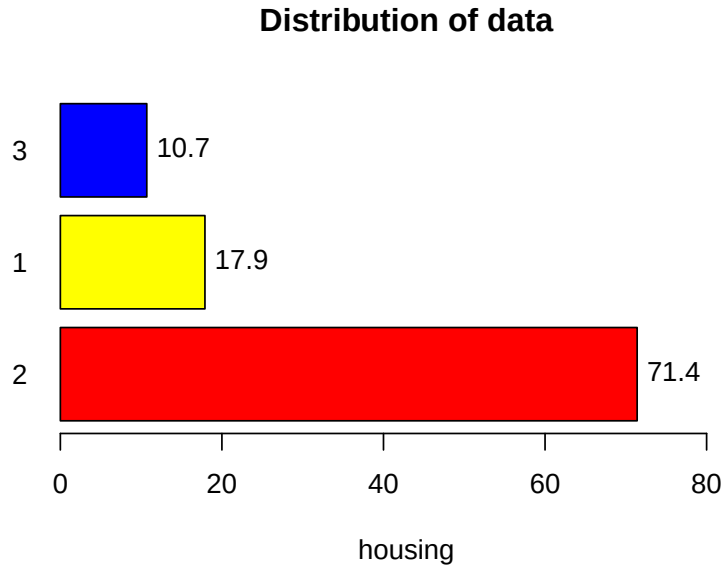
```
    ylab = "count", col = c("red", "yellow", "blue"), horiz = TRUE)
```




```
## data_credit$other_install_plans :
##          Frequency Percent
## no             814    81.4
## yes            186    18.6
##   Total       1000   100.0
```

Next feature is **housing**.

```
# summary housing
tab1(data_credit$housing, sort.group = "decreasing", cum.percent = FALSE,
    bar.values = "percent", main = "Distribution of data", xlab = "housing",
    ylab = "count", col = c("red", "yellow", "blue"), horiz = TRUE)
```



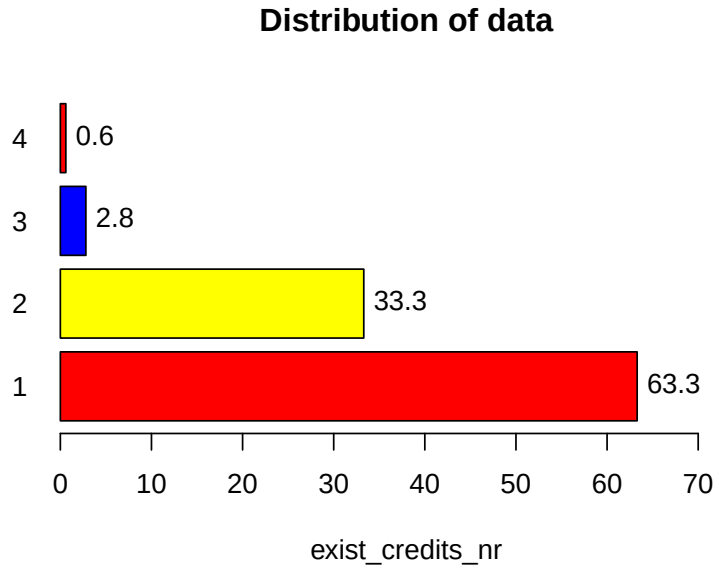

```
## data_credit$housing :
##          Frequency Percent
## 2              714    71.4
## 1              179    17.9
## 3              107    10.7
##   Total       1000   100.0
```

According to *Table 1: Distribution of categorical predictor variables for the South German Credit data, separately for good and bad credit risks* in gromping2019 , this categorical variable has 3 levels. This variable will not be subject of transformation.

nbsp;

Now let's see feature **exist_credits_nr**.

```
# summary exist_credits_nr
tab1(data_credit$exist_credits_nr, sort.group = "decreasing",
    cum.percent = FALSE, bar.values = "percent", main = "Distribution of data",
    xlab = "exist_credits_nr", ylab = "count", col = c("red",
        "yellow", "blue"), horiz = TRUE)
```




```
## data_credit$exist_credits_nr :
##          Frequency Percent
## 1              633    63.3
## 2              333    33.3
## 3               28     2.8
## 4                6     0.6
##   Total       1000   100.0
```

According to *Table 1: Distribution of categorical predictor variables for the South German Credit data, separately for good and bad credit risks* in gromping2019 , this categorical variable has 4 levels. This variable has data about other known credits inside bank where customer is applying for a new credit. We will recode this variable to 2 levels ** one_credit, morethan1**.

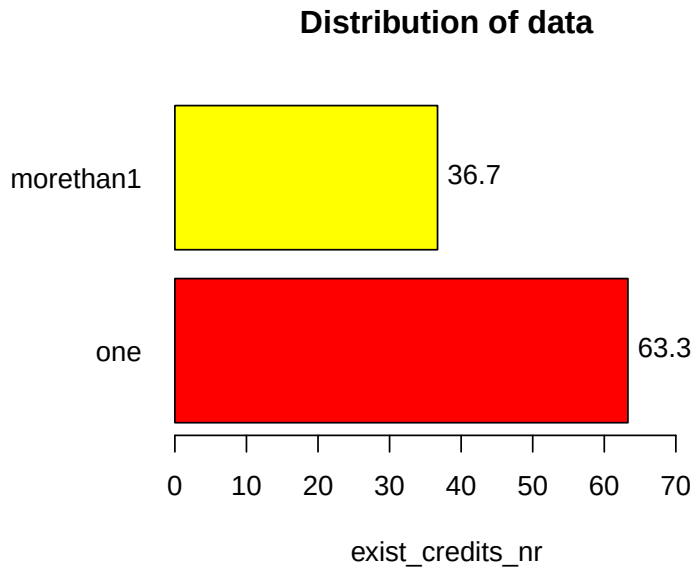
nbsp;

```
# recode exist_credits_nr
exist_credits_nr_temp <- recode(data_credit$exist_credits_nr,
    `1` = "one", `2` = "morethan1", `3` = "morethan1", `4` = "morethan1")
data_credit$exist_credits_nr <- exist_credits_nr_temp
```

The results of recoding:

```
# recoding results
tab1(data_credit$exist_credits_nr, sort.group = "decreasing",
    cum.percent = FALSE, bar.values = "percent", cex = 1, cex.names = 1,
```

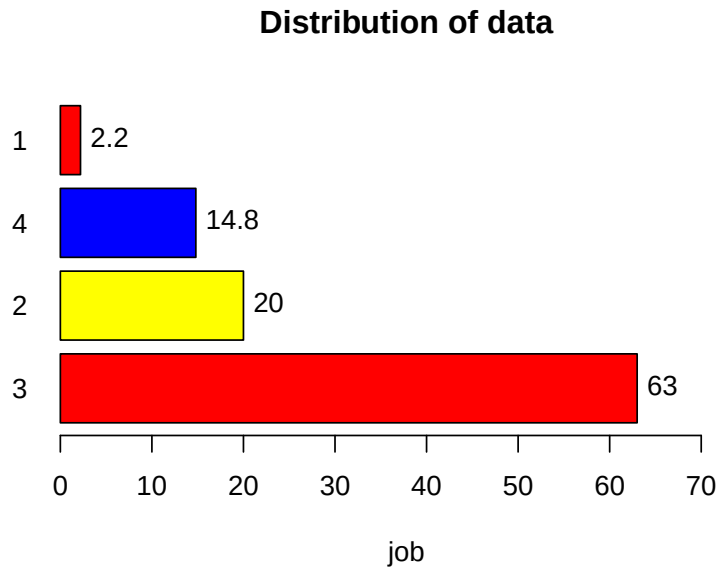
```
    main = "Distribution of data", xlab = "exist_credits_nr",
    ylab = "count", col = c("red", "yellow", "blue"), horiz = TRUE)
```




```
## data_credit$exist_credits_nr :
##           Frequency Percent
## one             633    63.3
## morethan1       367    36.7
##   Total        1000   100.0
```

Now let's see feature **job**.

```
# summary job
tab1(data_credit$job, sort.group = "decreasing", cum.percent = FALSE,
    bar.values = "percent", main = "Distribution of data", xlab = "job",
    ylab = "count", col = c("red", "yellow", "blue"), horiz = TRUE)
```



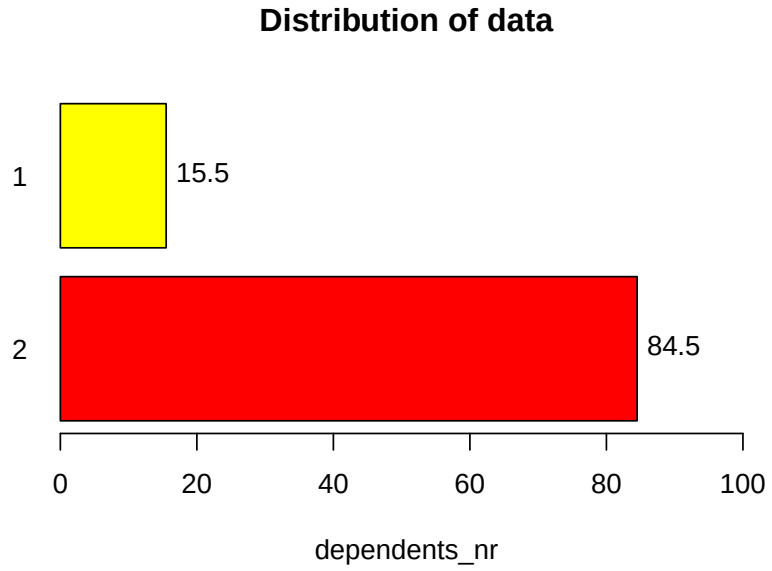

```
## data_credit$job :
##          Frequency Percent
## 3              630    63.0
## 2              200    20.0
## 4              148    14.8
## 1               22     2.2
##   Total       1000   100.0
```

According to *Table 1: Distribution of categorical predictor variables for the South German Credit data, separately for good and bad credit risks* in gromping2019 , this categorical variable has 4 levels. This variable has data about job skills level of bank customer who is applying for a new credit. We will not recode this variable.

nbsp;

Next feature is **dependents_nr**.

```
# summary dependents_nr
tab1(data_credit$dependents_nr, sort.group = "decreasing", cum.percent = FALSE,
    bar.values = "percent", main = "Distribution of data", xlab = "dependents_nr",
    ylab = "count", col = c("red", "yellow", "blue"), horiz = TRUE)
```

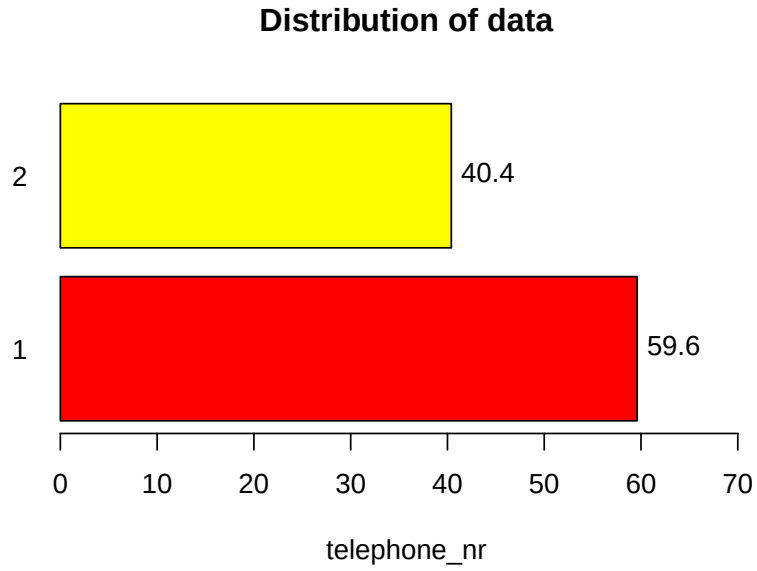



```
## data_credit$dependents_nr :
##          Frequency Percent
## 2              845    84.5
## 1              155    15.5
##   Total       1000   100.0
```

This categorical variable is binary and we will not transform it.

Now let's see feature **telephone_nr**.

```
# summary telephone_nr
tab1(data_credit$telephone_nr, sort.group = "decreasing", cum.percent = FALSE,
    bar.values = "percent", main = "Distribution of data", xlab = "telephone_nr",
    ylab = "count", col = c("red", "yellow", "blue"), horiz = TRUE)
```

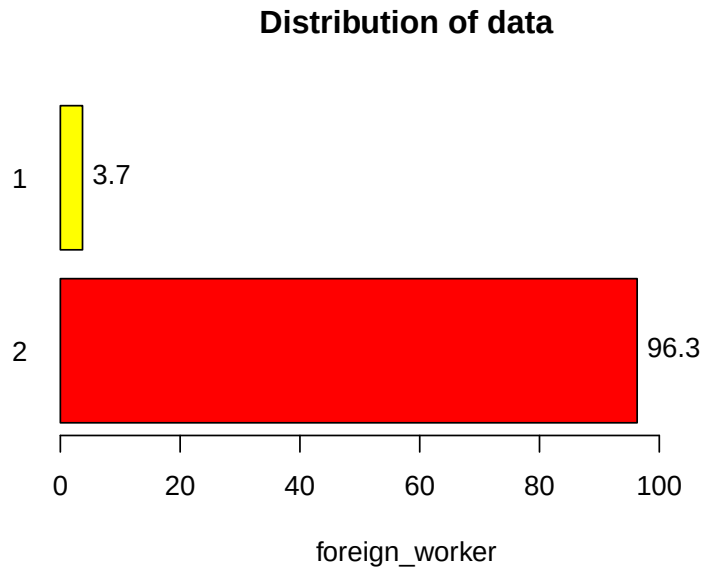



```
## data_credit$telephone_nr :
##         Frequency Percent
## 1             596    59.6
## 2             404    40.4
##   Total      1000   100.0
```

This categorical variable is binary and we will not transform it.

Last feature is **foreign_worker**.

```
# summary foreign_worker
tab1(data_credit$foreign_worker, sort.group = "decreasing", cum.percent = FALSE,
    bar.values = "percent", main = "Distribution of data", xlab = "foreign_worker",
    ylab = "count", col = c("red", "yellow", "blue"), horiz = TRUE)
```




```
## data_credit$foreign_worker :
##          Frequency Percent
## 2              963    96.3
## 1               37     3.7
##   Total       1000   100.0
```

This categorical variable is binary and we will not transform it.

Relationship between variables

Now we will study relationships between our outcome **customer_good_bad** and other features.

For this we will build **crosstables**, also perform **chi-square** test for each pair (outcome,feature). We will use function **CrossTable** from **gmodels** library (Warnes et al. 2018)

```
# Crosstables of outcome vs account_status
CrossTable(data_credit$customer_good_bad, data_credit$account_status,
    digits = 1, prop.r = F, prop.t = F, prop.chisq = F, chisq = T,
    dnn = c("customer_good_bad", "account_status"))
# Crosstables of outcome vs credit_history
CrossTable(data_credit$customer_good_bad, data_credit$credit_history,
    digits = 1, prop.r = F, prop.t = F, prop.chisq = F, chisq = T,
    dnn = c("customer_good_bad", "credit_history"))
# Crosstables of outcome vs credit_purpose
CrossTable(data_credit$customer_good_bad, data_credit$credit_purpose,
    digits = 1, prop.r = F, prop.t = F, prop.chisq = F, chisq = T,
```

```
    dnn = c("customer_good_bad", "credit_purpose"))
# Crosstables of outcome vs savings_account
CrossTable(data_credit$customer_good_bad, data_credit$savings_account,
    digits = 1, prop.r = F, prop.t = F, prop.chisq = F, chisq = T,
    dnn = c("customer_good_bad", "savings_account"))
# Crosstables of outcome vs employment_present
CrossTable(data_credit$customer_good_bad, data_credit$employment_present,
    digits = 1, prop.r = F, prop.t = F, prop.chisq = F, chisq = T,
    dnn = c("customer_good_bad", "employment_present"))
# Crosstables of outcome vs installment_rate_pct
CrossTable(data_credit$customer_good_bad, data_credit$installment_rate_pct,
    digits = 1, prop.r = F, prop.t = F, prop.chisq = F, chisq = T,
    dnn = c("customer_good_bad", "installment_rate_pct"))
# Crosstables of outcome vs status_sex
CrossTable(data_credit$customer_good_bad, data_credit$status_sex,
    digits = 1, prop.r = F, prop.t = F, prop.chisq = F, chisq = T,
    dnn = c("customer_good_bad", "status_sex"))
# Crosstables of outcome vs other_debtors_guar
CrossTable(data_credit$customer_good_bad, data_credit$other_debtors_guar,
    digits = 1, prop.r = F, prop.t = F, prop.chisq = F, chisq = T,
    dnn = c("customer_good_bad", "other_debtors_guar"))
# Crosstables of outcome vs residence_duration
CrossTable(data_credit$customer_good_bad, data_credit$residence_duration,
    digits = 1, prop.r = F, prop.t = F, prop.chisq = F, chisq = T,
    dnn = c("customer_good_bad", "residence_duration"))
# Crosstables of outcome vs property
CrossTable(data_credit$customer_good_bad, data_credit$property,
    digits = 1, prop.r = F, prop.t = F, prop.chisq = F, chisq = T,
    dnn = c("customer_good_bad", "property"))
# Crosstables of outcome vs other_install_plans
CrossTable(data_credit$customer_good_bad, data_credit$other_install_plans,
    digits = 1, prop.r = F, prop.t = F, prop.chisq = F, chisq = T,
    dnn = c("customer_good_bad", "other_install_plans"))
# Crosstables of outcome vs housing
CrossTable(data_credit$customer_good_bad, data_credit$housing,
    digits = 1, prop.r = F, prop.t = F, prop.chisq = F, chisq = T,
    dnn = c("customer_good_bad", "housing"))
# Crosstables of outcome vs exist_credits_nr
CrossTable(data_credit$customer_good_bad, data_credit$exist_credits_nr,
    digits = 1, prop.r = F, prop.t = F, prop.chisq = F, chisq = T,
    dnn = c("customer_good_bad", "exist_credits_nr"))
# Crosstables of outcome vs job
CrossTable(data_credit$customer_good_bad, data_credit$job, digits = 1,
    prop.r = F, prop.t = F, prop.chisq = F, chisq = T, dnn = c("customer_good_bad",
        "job"))
# Crosstables of outcome vs dependents_nr
CrossTable(data_credit$customer_good_bad, data_credit$dependents_nr,
    digits = 1, prop.r = F, prop.t = F, prop.chisq = F, chisq = T,
    dnn = c("customer_good_bad", "dependents_nr"))
```

```r
# Crosstables of outcome vs telephone_nr
CrossTable(data_credit$customer_good_bad, data_credit$telephone_nr,
    digits = 1, prop.r = F, prop.t = F, prop.chisq = F, chisq = T,
    dnn = c("customer_good_bad", "telephone_nr"))
# Crosstables of outcome vs foreign_worker
CrossTable(data_credit$customer_good_bad, data_credit$foreign_worker,
    digits = 1, prop.r = F, prop.t = F, prop.chisq = F, chisq = T,
    dnn = c("customer_good_bad", "foreign_worker"))
```

The results of code are shown in Appendix A. The **null hypothesis** for statistical test is that *there is no relationship between outcome and chosen feature.*

In the case of **customer_good_bad** and **account_status** test shows a p-value smaller than 0.05 so we have reject null hypothesis and keep alternative hypothesis *there is some relationship between outcome and chosen feature.* Also we can check from the table that 90% of people who have a possitive account balance are listed as good creditors, which fits to intutive.

In the case of **customer_good_bad** and **credit_history** test shows a p-value smaller than 0.05 so we have reject null hypothesis and keep alternative hypothesis *there is some relationship between outcome and chosen feature.* From the table we notice that percentage of customers who have payed their debts are listed as good creditors, which fits to intutive.

In the case of **customer_good_bad** and **savings_account** test shows a p-value smaller than 0.05 so~ we have reject null hypothesis and keep alternative hypothesis *there is some relationship between outcome and chosen feature.* From the table we notice that percentage of customers who have no money in their accounts are listed as bad creditors, which fits to intutive.

In the case of **customer_good_bad** and **employment_present** test shows a p-value smaller than 0.05 so we have reject null hypothesis and keep alternative hypothesis *there is some relationship between outcome and chosen feature.* From the table we notice that percentage of customers who are unemployed or have less than 1 year of work and bad creditors are 93 out of 300 (31 %) and higher than 141 out of 700 (20 %) who are who are unemployed or have less than 1 year of work and good creditors.

In the case of **customer_good_bad** and **credit_purpose** test shows a p-value smaller than 0.05 so we have reject null hypothesis and keep alternative hypothesis *there is some relationship between outcome and chosen feature.* From the table we notice that 75 % of bad creditors (225 out of 300) have taken credit for **domestic** or **other** purposes.

In the case of **customer_good_bad** and **installment_rate_pct** test shows a p-value bigger than 0.05 so we keep null hypothesis *there is no relationship between outcome and chosen feature.*

In the case of **customer_good_bad** and **status_sex** test shows a p-value smaller than 0.05 so we have reject null hypothesis and keep alternative hypothesis *there is some relationship between outcome and chosen feature.* We also see that married men are twice more than single when it comes to good creditors.

In the case of **customer_good_bad** and **other_debtors_guar** test shows a p-value bigger than 0.05 so we keep null hypothesis *there is no relationship between outcome and chosen feature.*

In the case of **customer_good_bad** and **residence_duration** test shows a p-value bigger than 0.05 so we keep null hypothesis *there is no relationship between outcome and chosen feature.*

In the case of **customer_good_bad** and **property** test shows a p-value smaller than 0.05 so we have reject null hypothesis and keep alternative hypothesis *there is some relationship between outcome and chosen feature.*

In the case of **customer_good_bad** and **other_install_plans** test shows a p-value smaller than 0.05 so we have reject null hypothesis and keep alternative hypothesis *there is some relationship between outcome and chosen feature.*

In the case of **customer_good_bad** and **exist_credits_nr** test shows a p-value bigger than 0.05 so we keep null hypothesis *there is no relationship between outcome and chosen feature.*

In the case of **customer_good_bad** and **housing** test shows a p-value smaller than 0.05 so we have reject null hypothesis and keep alternative hypothesis *there is some relationship between outcome and chosen feature.*

In the case of **customer_good_bad** and **job** test shows a p-value bigger than 0.05 so we keep null hypothesis *there is no relationship between outcome and chosen feature.*

In the case of **customer_good_bad** and **telephone_nr** test shows a p-value bigger than 0.05 so we keep null hypothesis *there is no relationship between outcome and chosen feature.*

In the case of **customer_good_bad** and **dependents_nr** test shows a p-value bigger than 0.05 so we keep null hypothesis *there is no relationship between outcome and chosen feature.*

In the case of **customer_good_bad** and **foreign_worker** test shows a p-value smaller than 0.05 so we have reject null hypothesis and keep alternative hypothesis *there is some relationship between outcome and chosen feature.*

Building models

Scaling numerical variables

We have in our **data_credit** 3 numerical variables **duration_month, credit_amount, age_years**. Before starting modeling process we need to apply on these variables z-score normalization (from each feature value we substract mean and result divide by standart deviation). We will use function **scale**.

```
# normalization of numeric features
data_credit %>% mutate_if(is.numeric, scale)
```

account_status	duration_month	credit_history	credit_purpose	credit_amount	savings_ac
no_account	-0.240736767	no_prob_currbank	used_car	-0.787262993	no_sav
no_account	-0.987078792	no_prob_currbank	services	-0.167300578	no_sav
no_money_acc	-0.738298117	all_paid	services	-0.860949955	less100
no_account	-0.738298117	no_prob_currbank	services	-0.407137466	no_sav
no_account	-0.738298117	no_prob_currbank	services	-0.389778519	no_sav
no_account	-0.904151900	no_prob_currbank	services	-0.364980022	no_sav
no_account	-1.070005683	no_prob_currbank	services	0.044903701	no_sav
no_account	-1.235859467	no_prob_currbank	services	-0.676732551	no_sav
positive_acc	-0.240736767	no_prob_currbank	domestic	-0.769904046	no_sav
no_money_acc	0.256824584	all_paid	domestic	0.172438826	100to1000
no_account	-0.821225008	no_prob_currbank	services	0.224515669	no_sav
no_account	0.754385934	no_prob_currbank	new_car	1.032946659	less100
no_account	-1.235859467	no_prob_currbank	domestic	-0.465591066	no_sav
no_money_acc	2.247069984	no_prob_currbank	services	1.527145271	less100
no_account	-0.240736767	all_paid	domestic	-0.473030615	over1000
no_account	-1.235859467	all_paid	domestic	-0.221148742	100to1000
no_account	-0.821225008	no_prob_currbank	services	0.236560653	no_sav
no_money_acc	-0.240736767	all_paid	domestic	-0.020635183	100to1000
no_money_acc	1.251947284	no_prob_currbank	domestic	-0.330970655	no_sav
positive_acc	-0.821225008	no_prob_currbank	services	1.401735730	no_sav
no_account	-1.235859467	no_prob_currbank	services	0.143389159	no_sav
no_money_acc	-0.738298117	no_prob_currbank	services	-0.052164700	no_sav
no_money_acc	1.251947284	all_paid	domestic	-0.314320236	no_sav
no_money_acc	-0.738298117	no_prob_currbank	domestic	-0.654413904	no_sav
no_account	-1.235859467	no_prob_currbank	services	0.511823966	over1000
no_money_acc	-0.821225008	no_prob_currbank	domestic	0.531308499	no_sav
no_account	-0.738298117	all_paid	used_car	-0.927905896	no_sav
no_money_acc	-0.987078792	no_prob_currbank	domestic	-0.750065248	no_sav
positive_acc	-0.489517442	all_paid	services	0.100877450	over1000
positive_acc	1.749508634	no_prob_currbank	new_car	0.540165105	no_sav
positive_acc	0.754385934	no_prob_currbank	domestic	-0.090070974	no_sav
positive_acc	1.251947284	no_prob_currbank	services	0.093437901	no_sav
positive_acc	1.251947284	no_prob_currbank	services	1.184217489	no_sav
positive_acc	0.256824584	all_paid	domestic	-0.671418588	100to1000
no_account	-0.489517442	all_paid	services	-0.549197426	no_sav
no_account	-1.235859467	no_prob_currbank	services	-0.854218934	no_sav
positive_acc	-0.738298117	no_prob_currbank	services	-0.629261143	no_sav
positive_acc	-0.738298117	no_prob_currbank	domestic	-0.473739143	no_sav
positive_acc	-0.240736767	all_paid	new_car	0.037818416	over1000
positive_acc	0.256824584	no_prob_currbank	new_car	0.211407892	no_sav
positive_acc	-0.738298117	no_prob_currbank	domestic	-0.806038998	over1000
no_account	0.256824584	all_paid	services	-0.537152442	no_sav
positive_acc	-0.240736767	no_prob_currbank	services	-0.794702542	no_sav
no_money_acc	0.256824584	no_prob_currbank	services	-0.158089707	over1000
no_money_acc	-0.240736767	all_paid	services	-0.719952788	over1000
no_account	-0.240736767	all_paid	services	-0.728100866	no_sav
positive_acc	0.256824584	all_paid	services	-0.713221768	no_sav
positive_acc	-0.240736767	no_prob_currbank	services	-0.498537640	less100
positive_acc	0.256824584	all_paid	services	-0.636700692	less100
no_account	0.256824584	no_prob_currbank	services	-0.669293002	less100
positive_acc	-0.738298117	all_paid	services	-0.932157067	no_sav
positive_acc	1.251947284	all_paid	domestic	0.229475368	no_sav
positive_acc	-0.987078792	no_prob_currbank	services	-0.725266752	no_sav
positive_acc	-0.738298117	no_prob_currbank	domestic	-0.333096241	over1000
positive_acc	0.256824584	all_paid	new_car	1.077583953	over1000
no_account	-0.738298117	no_prob_currbank	domestic	-1.022494447	no_sav
positive_acc	-0.738298117	no_prob_currbank	domestic	-0.572578865	no_sav

Data splitting

The **caret** package (Jed Wing et al. 2019) includes the function **createDataPartition** that helps us generates indexes for randomly splitting the data into training and test sets: We will split our data in half:

```
# Validation set will be 50% of South German Credit data Set
# seed as a starting point
set.seed(1, sample.kind = "Rounding")
# Store row numbers for train set: test_index
test_index <- createDataPartition(y = data_credit$customer_good_bad,
    times = 1, p = 0.5, list = FALSE)
```

We use the result of the **createDataPartition** function call to define the training and test sets like this:

```
# Create the train set
train <- data_credit[-test_index, ]
# Create the validation set
validation <- data_credit[test_index, ]
```

Now, we can save the result of steps above (**train** and **validation** dataframes) as R objects, so we can reload the final version of the data into the session for further analysis without repeating the process.

```
# Save our data as R objects
save(train, file = "./data/train.RData")
save(validation, file = "./data/validation.RData")
```

We see that **train** data frame has 500 rows and 21 variables, while **validation** data frame has 500 rows and 21.

Now let's print features of both data frames **train** and **validation** together to reassure ourselves that both contain the same features. We will use function **rCompare** from **dataCompareR** library(Noble-Eddy et al. 2018)

```
# compare train and validation
library(dataCompareR)
comp_train_val <- rCompare(train, validation)
comp_summ <- summary(comp_train_val)
comp_summ[c("datasetSummary", "ncolInAOnly", "ncolInBOnly", "ncolCommon",
    "rowsInAOnly", "rowsInBOnly", "nrowCommon")]
```

```
## $datasetSummary
##   Dataset Name Number of Rows Number of Columns
```

```
## 1        train            500                21
## 2   validation            500                21
##
## $ncolInAOnly
## [1] 0
##
## $ncolInBOnly
## [1] 0
##
## $ncolCommon
## [1] 21
##
## $rowsInAOnly
## [1] indices_removed
## <0 rows> (or 0-length row.names)
##
## $rowsInBOnly
## [1] indices_removed
## <0 rows> (or 0-length row.names)
##
## $nrowCommon
## [1] 500
```

Logistic Regression model

We will start with a Logistic Regression model. Logistic regression is a specific case of a set of generalized linear models. For logistic regression, outcome is a categorical variable which fits very well to our case since we have outcome **customer_good_bad** as categorical variable with 2 levels. In R, we can fit the logistic regression model with the function glm: generalized linear models. This function is more general than logistic regression so we need to specify the model we want through the family parameter (Irizarry 2019):

```
# Fitting initial model
glm_model <- glm(customer_good_bad ~ ., family = "binomial",
    data = train)
# Obtain significance levels using summary()
summary(glm_model)
```

```
##
## Call:
## glm(formula = customer_good_bad ~ ., family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.9946  -0.5501   0.3511   0.6586   1.7217
##
## Coefficients:
```

```
##                                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)                     2.060e-01  1.318e+00   0.156  0.87575
## account_statusno_money_acc      4.715e-01  3.221e-01   1.464  0.14320
## account_statuspositive_acc      1.921e+00  3.303e-01   5.817 5.98e-09 ***
## duration_month                 -7.031e-03  1.386e-02  -0.507  0.61207
## credit_historyall_paid          2.336e-01  4.549e-01   0.513  0.60763
## credit_historyno_prob_currbank  1.211e+00  4.740e-01   2.555  0.01061 *
## credit_purposenew_car           1.512e+00  5.788e-01   2.613  0.00898 **
## credit_purposeused_car          9.620e-01  3.711e-01   2.593  0.00952 **
## credit_purposedomestic          6.204e-01  3.115e-01   1.992  0.04641 *
## credit_amount                  -2.192e-04  7.019e-05  -3.123  0.00179 **
## savings_accountless100          5.327e-01  4.048e-01   1.316  0.18819
## savings_account100to1000        1.075e+00  5.189e-01   2.071  0.03839 *
## savings_accountover1000         1.183e+00  4.010e-01   2.951  0.00317 **
## employment_present1to4          5.107e-01  3.435e-01   1.487  0.13708
## employment_present4to7          6.486e-01  4.262e-01   1.522  0.12804
## employment_present7plus         4.108e-01  4.050e-01   1.014  0.31040
## installment_rate_pct2          -6.524e-02  4.928e-01  -0.132  0.89468
## installment_rate_pct3          -7.409e-01  5.453e-01  -1.359  0.17423
## installment_rate_pct4          -1.332e+00  4.836e-01  -2.754  0.00588 **
## status_sexm_married_wid         8.414e-01  2.934e-01   2.868  0.00414 **
## status_sexfemale                4.499e-01  4.812e-01   0.935  0.34982
## other_debtors_guaryes           9.218e-02  4.417e-01   0.209  0.83469
## residence_duration2            -1.093e+00  4.422e-01  -2.473  0.01340 *
## residence_duration3            -1.372e+00  4.915e-01  -2.791  0.00525 **
## residence_duration4            -8.004e-01  4.332e-01  -1.848  0.06465 .
## property2                      -4.496e-01  3.498e-01  -1.285  0.19868
## property3                       1.024e-01  3.504e-01   0.292  0.76997
## property4                      -7.352e-01  6.028e-01  -1.220  0.22263
## age_years                       2.246e-02  1.362e-02   1.650  0.09903 .
## other_install_plansno           4.198e-01  3.413e-01   1.230  0.21870
## housing2                       -3.809e-02  3.542e-01  -0.108  0.91436
## housing3                       -5.652e-02  6.905e-01  -0.082  0.93476
## exist_credits_nrmorethan1      -3.824e-01  3.450e-01  -1.108  0.26769
## job2                           -3.649e-01  7.593e-01  -0.481  0.63079
## job3                           -1.729e-01  7.285e-01  -0.237  0.81244
## job4                           -1.172e-01  7.845e-01  -0.149  0.88129
## dependents_nr2                  1.156e-01  3.817e-01   0.303  0.76207
## telephone_nr2                   4.938e-01  2.993e-01   1.650  0.09897 .
## foreign_worker2                -6.551e-01  8.111e-01  -0.808  0.41926
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 610.86  on 499  degrees of freedom
## Residual deviance: 425.59  on 461  degrees of freedom
## AIC: 503.59
##
## Number of Fisher Scoring iterations: 5
```

We see that some of variables have 3 stars beside them. This variables are statistically significant(have p-values < 0.05). Le' s filter them:

```
# Filter significant values
sig <- summary(glm_model)$coeff[-1, 4] < 0.05
names(sig)[sig == T]
```

```
##  [1] "account_statuspositive_acc"     "credit_historyno_prob_currbank"
##  [3] "credit_purposenew_car"          "credit_purposeused_car"
##  [5] "credit_purposedomestic"         "credit_amount"
##  [7] "savings_account100to1000"       "savings_accountover1000"
##  [9] "installment_rate_pct4"          "status_sexm_married_wid"
## [11] "residence_duration2"            "residence_duration3"
```

Next let's obtain prediction using the **predict** function:

```
# Predictions
pred_logit <- predict(glm_model, newdata = validation, type = "response")
```

To form a prediction, we define a decision rule: predict *good_creditor* if pred_logit > 0.5. Let's evaluate the accuracy of our model using function **confusionMatrix** from **caret** library(Jed Wing et al. 2019):

```
# convert pred_logit to a vector of binary values : put as
# cut pred_logit > 0.5
y_hat_glm <- factor(ifelse(pred_logit > 0.5, 1, 0))
# print confusion matrix
confusionMatrix(y_hat_glm, reference = validation$customer_good_bad,
    positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0  66  48
##          1  84 302
##
##                Accuracy : 0.736
##                  95% CI : (0.695, 0.7741)
##     No Information Rate : 0.7
##     P-Value [Acc > NIR] : 0.042593
##
##                   Kappa : 0.3252
##
##  Mcnemar's Test P-Value : 0.002316
##
##             Sensitivity : 0.8629
##             Specificity : 0.4400
```

```
##          Pos Pred Value : 0.7824
##          Neg Pred Value : 0.5789
##              Prevalence : 0.7000
##          Detection Rate : 0.6040
##    Detection Prevalence : 0.7720
##       Balanced Accuracy : 0.6514
##
##        'Positive' Class : 1
##
```

Since the output contains too much information, let's print only value of **accuracy** :

```
# Print only accuracy
confusionMatrix(y_hat_glm, reference = validation$customer_good_bad,
    positive = "1")$overall["Accuracy"]
```

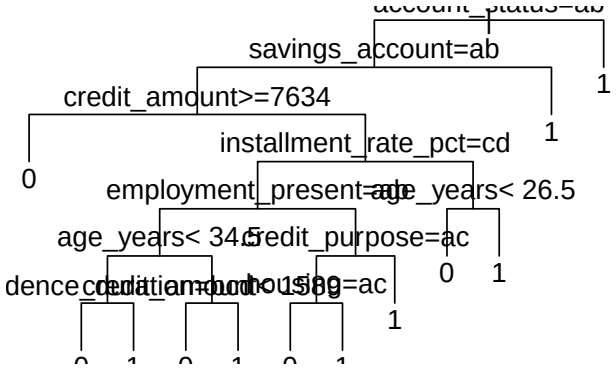
```
## Accuracy
##    0.736
```

We obtained an overall accuracy of 73.6 %. The prediction for bad creditors is good judged by value of **Specificity** 44%. Sensitivity is 86%, which is quite good. Also we notice that NPV is nearly 58%, and PPV is almost 78%.

Classification Tree model

Let' build our model will all features included. For building the decision tree we will use function **rpart** from **rpart** library(T. Therneau and Atkinson 2019)

```
# Decision Tree model
credit_tree <- rpart(customer_good_bad ~ ., method = "class",
    data = train)
# Plot the decision tree
plot(credit_tree, uniform = TRUE)
# Add labels
text(credit_tree)
```



Let's make our prediction as we did with previous model

```
# Predictions
pred_tree <- predict(credit_tree, newdata = validation, type = "class")
```

Let's evaluate the accuracy of our model:

```
# print confusion matrix
confusionMatrix(pred_tree, reference = validation$customer_good_bad,
    positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0  54  53
##          1  96 297
##
##                Accuracy : 0.702
##                  95% CI : (0.6598, 0.7418)
##     No Information Rate : 0.7
##     P-Value [Acc > NIR] : 0.4831349
##
##                   Kappa : 0.2272
##
##  Mcnemar's Test P-Value : 0.0005801
##
##             Sensitivity : 0.8486
##             Specificity : 0.3600
##          Pos Pred Value : 0.7557
```

```
##          Neg Pred Value : 0.5047
##              Prevalence : 0.7000
##          Detection Rate : 0.5940
##    Detection Prevalence : 0.7860
##       Balanced Accuracy : 0.6043
##
##        'Positive' Class : 1
##
```

Since the output contains too much information, let's print only value of **accuracy** :

```
# Print only accuracy
confusionMatrix(pred_tree, reference = validation$customer_good_bad,
    positive = "1")$overall["Accuracy"]
```

```
## Accuracy
##    0.702
```

We obtained an overall accuracy of 70.2 %. Sensitivity is 84.86%, a very good one, and specificity is 36%.

Results

- We build a logistic regression model with accuracy of 73.6 %. Specificity of this model was 44% and sensitivity 86 %.
- We build a classification tree model with accuracy of 70.2 %. Sensitivity is 84.86 and specificity is 36%.

Conclusions

In project with real-life credit data, we tried to model credit risk by using logistic regression and decision trees in R.

Credit risk models are very important for financial institutions such as banks. The risk of creditor not turning the loan is a parameter which can be modeled and measured apriori in order to minimize loses. Machine learning algorithms and techniques come in help in such problems.

Appendix A

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Col Total |
## |-------------------------|
##
##
```

```
## Total Observations in Table:  1000
##
##
##                    | account_status
## customer_good_bad |   no_account | no_money_acc | positive_acc |    Row Total |
## ------------------|--------------|--------------|--------------|--------------|
##                 0 |          135 |          105 |           60 |          300 |
##                   |          0.5 |          0.4 |          0.1 |              |
## ------------------|--------------|--------------|--------------|--------------|
##                 1 |          139 |          164 |          397 |          700 |
##                   |          0.5 |          0.6 |          0.9 |              |
## ------------------|--------------|--------------|--------------|--------------|
##      Column Total |          274 |          269 |          457 |         1000 |
##                   |          0.3 |          0.3 |          0.5 |              |
## ------------------|--------------|--------------|--------------|--------------|
##
##
## Statistics for All Table Factors
##
##
## Pearson's Chi-squared test
## ------------------------------------------------------------
## Chi^2 =  120.8438     d.f. =  2     p =  5.742621e-27
##
##
##

##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Col Total |
## |-------------------------|
##
##
## Total Observations in Table:  1000
##
##
##                    | credit_history
## customer_good_bad |     pay_problems |         all_paid | no_prob_currbank |        Row
## ------------------|------------------|------------------|------------------|-----------
##                 0 |               53 |              169 |               78 |
##                   |              0.6 |              0.3 |              0.2 |
## ------------------|------------------|------------------|------------------|-----------
##                 1 |               36 |              361 |              303 |
##                   |              0.4 |              0.7 |              0.8 |
## ------------------|------------------|------------------|------------------|-----------
##      Column Total |               89 |              530 |              381 |
##                   |              0.1 |              0.5 |              0.4 |
```

```
## ------------------|-----------------|-----------------|-----------------|-----------
##
##
## Statistics for All Table Factors
##
##
## Pearson's Chi-squared test
## ------------------------------------------------------------
## Chi^2 =  54.3761     d.f. =  2     p =  1.557328e-12
##
##
##

##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Col Total |
## |-------------------------|
##
##
## Total Observations in Table:  1000
##
##
##                   | credit_purpose
## customer_good_bad |  services |   new_car |  used_car |  domestic | Row Total |
## ------------------|-----------|-----------|-----------|-----------|-----------|
##                 0 |       151 |        17 |        58 |        74 |       300 |
##                   |       0.4 |       0.2 |       0.3 |       0.2 |           |
## ------------------|-----------|-----------|-----------|-----------|-----------|
##                 1 |       251 |        86 |       123 |       240 |       700 |
##                   |       0.6 |       0.8 |       0.7 |       0.8 |           |
## ------------------|-----------|-----------|-----------|-----------|-----------|
##      Column Total |       402 |       103 |       181 |       314 |      1000 |
##                   |       0.4 |       0.1 |       0.2 |       0.3 |           |
## ------------------|-----------|-----------|-----------|-----------|-----------|
##
##
## Statistics for All Table Factors
##
##
## Pearson's Chi-squared test
## ------------------------------------------------------------
## Chi^2 =  26.42789     d.f. =  3     p =  7.759191e-06
##
##
##

##
##
```

```
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Col Total |
## |-------------------------|
##
##
## Total Observations in Table:  1000
##
##
##                   | savings_account
## customer_good_bad |    no_sav |   less100 | 100to1000 |  over1000 | Row Total |
## ------------------|-----------|-----------|-----------|-----------|-----------|
##                 0 |       217 |        34 |        17 |        32 |       300 |
##                   |       0.4 |       0.3 |       0.2 |       0.2 |           |
## ------------------|-----------|-----------|-----------|-----------|-----------|
##                 1 |       386 |        69 |        94 |       151 |       700 |
##                   |       0.6 |       0.7 |       0.8 |       0.8 |           |
## ------------------|-----------|-----------|-----------|-----------|-----------|
##      Column Total |       603 |       103 |       111 |       183 |      1000 |
##                   |       0.6 |       0.1 |       0.1 |       0.2 |           |
## ------------------|-----------|-----------|-----------|-----------|-----------|
##
##
## Statistics for All Table Factors
##
##
## Pearson's Chi-squared test
## ------------------------------------------------------------
## Chi^2 =  35.77973     d.f. =  3     p =  8.335937e-08
##
##
##

##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Col Total |
## |-------------------------|
##
##
## Total Observations in Table:  1000
##
##
##                   | employment_present
## customer_good_bad | unemp_less1year |            1to4 |            4to7 |           7pl
## ------------------|-----------------|-----------------|-----------------|--------------
##                 0 |              93 |             104 |              39 |
```

```
##                   |             0.4 |             0.3 |             0.2 |             0
## ------------------|-----------------|-----------------|-----------------|--------------
##                 1 |             141 |             235 |             135 |             1
##                   |             0.6 |             0.7 |             0.8 |             0
## ------------------|-----------------|-----------------|-----------------|--------------
##      Column Total |             234 |             339 |             174 |             2
##                   |             0.2 |             0.3 |             0.2 |             0
## ------------------|-----------------|-----------------|-----------------|--------------
##
##
## Statistics for All Table Factors
##
##
## Pearson's Chi-squared test
## ------------------------------------------------------------
## Chi^2 =  18.08688     d.f. =  3     p =  0.0004220685
##
##
##

##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Col Total |
## |-------------------------|
##
##
## Total Observations in Table:  1000
##
##
##                   | installment_rate_pct
## customer_good_bad |         1 |         2 |         3 |         4 | Row Total |
## ------------------|-----------|-----------|-----------|-----------|-----------|
##                 0 |        34 |        62 |        45 |       159 |       300 |
##                   |       0.2 |       0.3 |       0.3 |       0.3 |           |
## ------------------|-----------|-----------|-----------|-----------|-----------|
##                 1 |       102 |       169 |       112 |       317 |       700 |
##                   |       0.8 |       0.7 |       0.7 |       0.7 |           |
## ------------------|-----------|-----------|-----------|-----------|-----------|
##      Column Total |       136 |       231 |       157 |       476 |      1000 |
##                   |       0.1 |       0.2 |       0.2 |       0.5 |           |
## ------------------|-----------|-----------|-----------|-----------|-----------|
##
##
## Statistics for All Table Factors
##
##
## Pearson's Chi-squared test
```

```
## ---------------------------------------------------------
## Chi^2 =  5.476792     d.f. =  3     p =  0.1400333
##
##
##
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Col Total |
## |-------------------------|
##
##
## Total Observations in Table:  1000
##
##
##                   | status_sex
## customer_good_bad | m_single_divorced |     m_married_wid |            female |
## ------------------|-------------------|-------------------|-------------------|--------
##                 0 |               129 |               146 |                25 |
##                   |               0.4 |               0.3 |               0.3 |
## ------------------|-------------------|-------------------|-------------------|--------
##                 1 |               231 |               402 |                67 |
##                   |               0.6 |               0.7 |               0.7 |
## ------------------|-------------------|-------------------|-------------------|--------
##      Column Total |               360 |               548 |                92 |
##                   |               0.4 |               0.5 |               0.1 |
## ------------------|-------------------|-------------------|-------------------|--------
##
##
## Statistics for All Table Factors
##
##
## Pearson's Chi-squared test
## ---------------------------------------------------------
## Chi^2 =  9.125183     d.f. =  2     p =  0.01043498
##
##
##
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Col Total |
## |-------------------------|
##
##
```

```
## Total Observations in Table:  1000
##
##
##                   | other_debtors_guar
## customer_good_bad |        no |       yes | Row Total |
## ------------------|-----------|-----------|-----------|
##                 0 |       272 |        28 |       300 |
##                   |       0.3 |       0.3 |           |
## ------------------|-----------|-----------|-----------|
##                 1 |       635 |        65 |       700 |
##                   |       0.7 |       0.7 |           |
## ------------------|-----------|-----------|-----------|
##      Column Total |       907 |        93 |      1000 |
##                   |       0.9 |       0.1 |           |
## ------------------|-----------|-----------|-----------|
##
##
## Statistics for All Table Factors
##
##
## Pearson's Chi-squared test
## ------------------------------------------------------------
## Chi^2 =  0.0005645345     d.f. =  1     p =  0.9810441
##
## Pearson's Chi-squared test with Yates' continuity correction
## ------------------------------------------------------------
## Chi^2 =  6.050627e-30     d.f. =  1     p =  1
##
##

##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Col Total |
## |-------------------------|
##
##
## Total Observations in Table:  1000
##
##
##                   | residence_duration
## customer_good_bad |         1 |         2 |         3 |         4 | Row Total |
## ------------------|-----------|-----------|-----------|-----------|-----------|
##                 0 |        36 |        97 |        43 |       124 |       300 |
##                   |       0.3 |       0.3 |       0.3 |       0.3 |           |
## ------------------|-----------|-----------|-----------|-----------|-----------|
##                 1 |        94 |       211 |       106 |       289 |       700 |
##                   |       0.7 |       0.7 |       0.7 |       0.7 |           |
```

```
## ------------------|-----------|-----------|-----------|-----------|-----------|
##      Column Total |       130 |       308 |       149 |       413 |      1000 |
##                   |       0.1 |       0.3 |       0.1 |       0.4 |           |
## ------------------|-----------|-----------|-----------|-----------|-----------|
##
##
## Statistics for All Table Factors
##
##
## Pearson's Chi-squared test
## ------------------------------------------------------------
## Chi^2 =  0.7492964     d.f. =  3     p =  0.8615521
##
##
##

##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Col Total |
## |-------------------------|
##
##
## Total Observations in Table:  1000
##
##
##                   | property
## customer_good_bad |         1 |         2 |         3 |         4 | Row Total |
## ------------------|-----------|-----------|-----------|-----------|-----------|
##                 0 |        60 |        71 |       102 |        67 |       300 |
##                   |       0.2 |       0.3 |       0.3 |       0.4 |           |
## ------------------|-----------|-----------|-----------|-----------|-----------|
##                 1 |       222 |       161 |       230 |        87 |       700 |
##                   |       0.8 |       0.7 |       0.7 |       0.6 |           |
## ------------------|-----------|-----------|-----------|-----------|-----------|
##      Column Total |       282 |       232 |       332 |       154 |      1000 |
##                   |       0.3 |       0.2 |       0.3 |       0.2 |           |
## ------------------|-----------|-----------|-----------|-----------|-----------|
##
##
## Statistics for All Table Factors
##
##
## Pearson's Chi-squared test
## ------------------------------------------------------------
## Chi^2 =  23.71955     d.f. =  3     p =  2.858442e-05
##
##
```

```
##
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Col Total |
## |-------------------------|
##
##
## Total Observations in Table:  1000
##
##
##                   | other_install_plans
## customer_good_bad |       yes |        no | Row Total |
## ------------------|-----------|-----------|-----------|
##                 0 |        76 |       224 |       300 |
##                   |       0.4 |       0.3 |           |
## ------------------|-----------|-----------|-----------|
##                 1 |       110 |       590 |       700 |
##                   |       0.6 |       0.7 |           |
## ------------------|-----------|-----------|-----------|
##      Column Total |       186 |       814 |      1000 |
##                   |       0.2 |       0.8 |           |
## ------------------|-----------|-----------|-----------|
##
##
## Statistics for All Table Factors
##
##
## Pearson's Chi-squared test
## ------------------------------------------------------------
## Chi^2 =  12.83353     d.f. =  1     p =  0.000340463
##
## Pearson's Chi-squared test with Yates' continuity correction
## ------------------------------------------------------------
## Chi^2 =  12.20607     d.f. =  1     p =  0.0004763431
##
##
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Col Total |
## |-------------------------|
##
##
## Total Observations in Table:  1000
```

```
##
##
##                    | housing
## customer_good_bad |         1 |         2 |         3 | Row Total |
## ------------------|-----------|-----------|-----------|-----------|
##                 0 |        70 |       186 |        44 |       300 |
##                   |       0.4 |       0.3 |       0.4 |           |
## ------------------|-----------|-----------|-----------|-----------|
##                 1 |       109 |       528 |        63 |       700 |
##                   |       0.6 |       0.7 |       0.6 |           |
## ------------------|-----------|-----------|-----------|-----------|
##      Column Total |       179 |       714 |       107 |      1000 |
##                   |       0.2 |       0.7 |       0.1 |           |
## ------------------|-----------|-----------|-----------|-----------|
##
##
## Statistics for All Table Factors
##
##
## Pearson's Chi-squared test
## ------------------------------------------------------------
## Chi^2 =  18.67401     d.f. =  2     p =  8.810311e-05
##
##
##
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Col Total |
## |-------------------------|
##
##
## Total Observations in Table:  1000
##
##
##                    | exist_credits_nr
## customer_good_bad |       one | morethan1 | Row Total |
## ------------------|-----------|-----------|-----------|
##                 0 |       200 |       100 |       300 |
##                   |       0.3 |       0.3 |           |
## ------------------|-----------|-----------|-----------|
##                 1 |       433 |       267 |       700 |
##                   |       0.7 |       0.7 |           |
## ------------------|-----------|-----------|-----------|
##      Column Total |       633 |       367 |      1000 |
##                   |       0.6 |       0.4 |           |
## ------------------|-----------|-----------|-----------|
```

```
##
##
## Statistics for All Table Factors
##
##
## Pearson's Chi-squared test
## ------------------------------------------------------------
## Chi^2 =  2.090998     d.f. =  1     p =  0.1481692
##
## Pearson's Chi-squared test with Yates' continuity correction
## ------------------------------------------------------------
## Chi^2 =  1.889093     d.f. =  1     p =  0.1693042
##
##

##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Col Total |
## |-------------------------|
##
##
## Total Observations in Table:  1000
##
##
##                   | job
## customer_good_bad |         1 |         2 |         3 |         4 | Row Total |
## ------------------|-----------|-----------|-----------|-----------|-----------|
##                 0 |         7 |        56 |       186 |        51 |       300 |
##                   |       0.3 |       0.3 |       0.3 |       0.3 |           |
## ------------------|-----------|-----------|-----------|-----------|-----------|
##                 1 |        15 |       144 |       444 |        97 |       700 |
##                   |       0.7 |       0.7 |       0.7 |       0.7 |           |
## ------------------|-----------|-----------|-----------|-----------|-----------|
##      Column Total |        22 |       200 |       630 |       148 |      1000 |
##                   |       0.0 |       0.2 |       0.6 |       0.1 |           |
## ------------------|-----------|-----------|-----------|-----------|-----------|
##
##
## Statistics for All Table Factors
##
##
## Pearson's Chi-squared test
## ------------------------------------------------------------
## Chi^2 =  1.885156     d.f. =  3     p =  0.5965816
##
##
##
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Col Total |
## |-------------------------|
##
##
## Total Observations in Table:  1000
##
##
##                    | dependents_nr
## customer_good_bad |         1 |         2 | Row Total |
## ------------------|-----------|-----------|-----------|
##                 0 |        46 |       254 |       300 |
##                   |       0.3 |       0.3 |           |
## ------------------|-----------|-----------|-----------|
##                 1 |       109 |       591 |       700 |
##                   |       0.7 |       0.7 |           |
## ------------------|-----------|-----------|-----------|
##      Column Total |       155 |       845 |      1000 |
##                   |       0.2 |       0.8 |           |
## ------------------|-----------|-----------|-----------|
##
##
## Statistics for All Table Factors
##
##
## Pearson's Chi-squared test
## ------------------------------------------------------------
## Chi^2 =  0.009089339     d.f. =  1     p =  0.9240463
##
## Pearson's Chi-squared test with Yates' continuity correction
## ------------------------------------------------------------
## Chi^2 =  0     d.f. =  1     p =  1
##
##
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Col Total |
## |-------------------------|
##
##
## Total Observations in Table:  1000
##
```

```
##
##                    | telephone_nr
## customer_good_bad |         1 |         2 | Row Total |
## ------------------|-----------|-----------|-----------|
##                 0 |       187 |       113 |       300 |
##                   |       0.3 |       0.3 |           |
## ------------------|-----------|-----------|-----------|
##                 1 |       409 |       291 |       700 |
##                   |       0.7 |       0.7 |           |
## ------------------|-----------|-----------|-----------|
##      Column Total |       596 |       404 |      1000 |
##                   |       0.6 |       0.4 |           |
## ------------------|-----------|-----------|-----------|
##
##
## Statistics for All Table Factors
##
##
## Pearson's Chi-squared test
## ------------------------------------------------------------
## Chi^2 =  1.329783     d.f. =  1     p =  0.2488438
##
## Pearson's Chi-squared test with Yates' continuity correction
## ------------------------------------------------------------
## Chi^2 =  1.172559     d.f. =  1     p =  0.2788762
##
##

##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Col Total |
## |-------------------------|
##
##
## Total Observations in Table:  1000
##
##
##                    | foreign_worker
## customer_good_bad |         1 |         2 | Row Total |
## ------------------|-----------|-----------|-----------|
##                 0 |         4 |       296 |       300 |
##                   |       0.1 |       0.3 |           |
## ------------------|-----------|-----------|-----------|
##                 1 |        33 |       667 |       700 |
##                   |       0.9 |       0.7 |           |
## ------------------|-----------|-----------|-----------|
##      Column Total |        37 |       963 |      1000 |
```

```
##                   |       0.0 |       1.0 |           |
## ------------------|-----------|-----------|-----------|
##
##
## Statistics for All Table Factors
##
##
## Pearson's Chi-squared test
## ------------------------------------------------------------
## Chi^2 =  6.737044     d.f. =  1     p =  0.009443096
##
## Pearson's Chi-squared test with Yates' continuity correction
## ------------------------------------------------------------
## Chi^2 =  5.821576     d.f. =  1     p =  0.01583075
##
##
```

References

Allaire, JJ, Yihui Xie, Jonathan McPherson, Javier Luraschi, Kevin Ushey, Aron Atkins, Hadley Wickham, Joe Cheng, Winston Chang, and Richard Iannone. 2019. *Rmarkdown: Dynamic Documents for R*. https://github.com/rstudio/rmarkdown.

Chongsuvivatwong, Virasakdi. 2018. *EpiDisplay: Epidemiological Data Display Package*. https://CRAN.R-project.org/package=epiDisplay.

Dowle, Matt, and Arun Srinivasan. 2019. *Data.table: Extension of 'Data.frame'*. https://CRAN.R-project.org/package=data.table.

Grömping, Ulrike. 2019. *South German Credit Data Correcting a Widely Used Data Set*. Reports in Mathematics, Physics and Chemistry. Herausgeber: Fachbereich II. http://www1.beuth-hochschule.de/FB_II/reports/Report-2019-004.pdf.

Henry, Lionel, and Hadley Wickham. 2019. *Purrr: Functional Programming Tools*. https://CRAN.R-project.org/package=purrr.

Investopedia. 2019. https://www.investopedia.com/terms/c/creditrisk.asp.

Irizarry, Rafael A. 2019. *Introduction to Data Science: Data Analysis and Prediction Algorithms with R*. Data Science Series. Chapman & Hall/CRC.

Jed Wing, Max Kuhn. Contributions from, Steve Weston, Andre Williams, Chris Keefer, Allan Engelhardt, Tony Cooper, Zachary Mayer, et al. 2019. *Caret: Classification and Regression Training*. https://CRAN.R-project.org/package=caret.

Müller, Kirill, and Hadley Wickham. 2019. *Tibble: Simple Data Frames*. https://CRAN.R-project.org/package=tibble.

Noble-Eddy, Rob, Sarah Johnston, Sarah Pollicott, and Merlijn van Horssen. 2018. *DataCompareR: Compare Two Data Frames and Summarise the Difference*.

https://CRAN.R-project.org/package=dataCompareR.

R Core Team. 2018. *Foreign: Read Data Stored by 'Minitab', 'S', 'Sas', 'Spss', 'Stata', 'Systat', 'Weka', 'dBase', ...* https://CRAN.R-project.org/package=foreign.

Ripley, Brian. 2016. *Nnet: Feed-Forward Neural Networks and Multinomial Log-Linear Models.* https://CRAN.R-project.org/package=nnet.

Sarkar, Deepayan. 2018. *Lattice: Trellis Graphics for R.* https://CRAN.R-project.org/package=lattice.

Sing, T., O. Sander, N. Beerenwinkel, and T. Lengauer. 2005. "ROCR: Visualizing Classifier Performance in R." *Bioinformatics* 21 (20): 7881. http://rocr.bioinf.mpi-sb.mpg.de.

Therneau, Terry, and Beth Atkinson. 2019. *Rpart: Recursive Partitioning and Regression Trees.* https://CRAN.R-project.org/package=rpart.

Therneau, Terry M. 2019. *Survival: Survival Analysis.* https://CRAN.R-project.org/package=survival.

Warnes, Gregory R., Ben Bolker, Thomas Lumley, Randall C Johnson. Contributions from Randall C. Johnson are Copyright SAIC-Frederick, Inc. Funded by the Intramural Research Program, of the NIH, National Cancer Institute, and Center for Cancer Research under NCI Contract NO1-CO-12400. 2018. *Gmodels: Various R Programming Tools for Model Fitting.* https://CRAN.R-project.org/package=gmodels.

Wickham, Hadley. 2019. *Tidyverse: Easily Install and Load the 'Tidyverse'.* https://CRAN.R-project.org/package=tidyverse.

Wickham, Hadley, Winston Chang, Lionel Henry, Thomas Lin Pedersen, Kohske Takahashi, Claus Wilke, Kara Woo, and Hiroaki Yutani. 2019. *Ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics.* https://CRAN.R-project.org/package=ggplot2.

Wickham, Hadley, Romain François, Lionel Henry, and Kirill Müller. 2019. *Dplyr: A Grammar of Data Manipulation.* https://CRAN.R-project.org/package=dplyr.

Wickham, Hadley, and Lionel Henry. 2019. *Tidyr: Tidy Messy Data.* https://CRAN.R-project.org/package=tidyr.

Xie, Yihui. 2019. *Knitr: A General-Purpose Package for Dynamic Report Generation in R.* https://CRAN.R-project.org/package=knitr.

Zhu, Hao. 2019. *KableExtra: Construct Complex Table with 'Kable' and Pipe Syntax.* https://CRAN.R-project.org/package=kableExtra.