# Credit Risk Analysis: Decriptive Analysis

*LvdT*

## Contents

**1 Credit Rating** **1**

**2 The Data** **1**

**3 Functions** **3**

**4 Descriptive Analysis** **5**

4.1 Credit Rating . . . . . . 6
4.2 Account Balance . . . . . . 6
4.3 Credit Duration . . . . . . 9
4.4 Previous Credit Payment Status . . . . . . 11
4.5 Credit Purpose . . . . . . 11
4.6 Credit Amount . . . . . . 13
4.7 Savings . . . . . . 16
4.8 Employment Duration & Installment Rate . . . . . . 17
4.9 Marital Status & Guarantor . . . . . . 19
4.10 Residence Duration & Current Assets . . . . . . 21
4.11 Remaining features . . . . . . 21

**5 Export** **24**

# 1 Credit Rating

This project will assess data about the credit risk of certain customers of a German bank. In this dataset, the customers are classified as being a credit risk or not based on previously collected data.

Credit risk can be defined as defaulting on a debt, due to the borrower being unable to make the stipulated debt payments in the agreed upon time frame. It is useful for a bank to conduct risk analysis, in order to ascertain whether or not a specific customer is likely to defaul on his debt.

This part of the risk analysis will conduct decriptive analysis on the data set in order to determine which features are important and whether or not they display a relationship with credit risk. The subsequent step of the project will build predictive models using various machine learning techniques in order to build valuable predictive models for the bank to predict customer credit risk based on specific customer traits.

# 2 The Data

The data is loaded in, dependencies are imported, column titles are adjusted and the data is investigated at a glance. Finally, the data is checked for NAs.

```r
# Dependencies
library(gridExtra)
library(pastecs)
```

```
## Loading required package: boot
```

```r
library(ggplot2)
library(gmodels)
library(car)
```

```
##
## Attaching package: 'car'

## The following object is masked from 'package:boot':
##
##     logit
```

```r
# Import dataset as dataframe and clean the column headers
credit.df <- read.csv("german_credit.csv", header = TRUE, sep = ",")

colnames(credit.df) <- c("credit.rating", "account.balance", "credit.duration.months", "previous.credit

# Dataset information
head(credit.df, n = 3)
```

```
##   credit.rating account.balance credit.duration.months
## 1             1               1                     18
## 2             1               1                      9
## 3             1               2                     12
##   previous.credit.payment.status credit.purpose credit.amount savings
## 1                              4              2          1049       1
## 2                              4              0          2799       1
## 3                              2              9           841       2
##   employment.duration installment.rate marital.status guarantor
## 1                   2                4              2         1
## 2                   3                2              3         1
## 3                   4                2              2         1
##   residence.duration current.assets age other.credits apartment.type
## 1                  4              2  21             3              1
## 2                  2              1  36             3              1
## 3                  4              1  23             3              1
##   bank.credits occupation dependents telephone foreign.worker
## 1            1          3          1         1              1
## 2            2          3          2         1              1
## 3            1          2          1         1              1
```

```r
tail(credit.df, n = 3)
```

```
##      credit.rating account.balance credit.duration.months
## 998              0               4                     21
## 999              0               2                     12
## 1000             0               1                     30
##      previous.credit.payment.status credit.purpose credit.amount savings
## 998                               4              0         12680       5
## 999                               2              3          6468       5
## 1000                              2              2          6350       5
##      employment.duration installment.rate marital.status guarantor
## 998                    5                4              3         1
## 999                    1                2              3         1
## 1000                   5                4              3         1
##      residence.duration current.assets age other.credits apartment.type
## 998                   4              4  30             3              3
## 999                   1              4  52             3              2
## 1000                  4              2  31             3              2
##      bank.credits occupation dependents telephone foreign.worker
## 998             1          4          1         2              1
## 999             1          4          1         2              1
## 1000            1          3          1         1              1
```

```r
str(credit.df)
```

```
## 'data.frame':    1000 obs. of  21 variables:
##  $ credit.rating                 : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ account.balance               : int  1 1 2 1 1 1 1 1 4 2 ...
##  $ credit.duration.months        : int  18 9 12 12 12 10 8 6 18 24 ...
##  $ previous.credit.payment.status: int  4 4 2 4 4 4 4 4 4 2 ...
##  $ credit.purpose                : int  2 0 9 0 0 0 0 0 3 3 ...
##  $ credit.amount                 : int  1049 2799 841 2122 2171 2241 3398 1361 1098 3758 ...
##  $ savings                       : int  1 1 2 1 1 1 1 1 1 3 ...
##  $ employment.duration           : int  2 3 4 3 3 2 4 2 1 1 ...
##  $ installment.rate              : int  4 2 2 3 4 1 1 2 4 1 ...
##  $ marital.status                : int  2 3 2 3 3 3 3 3 2 2 ...
##  $ guarantor                     : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ residence.duration            : int  4 2 4 2 4 3 4 4 4 4 ...
##  $ current.assets                : int  2 1 1 1 2 1 1 1 3 4 ...
##  $ age                           : int  21 36 23 39 38 48 39 40 65 23 ...
##  $ other.credits                 : int  3 3 3 3 1 3 3 3 3 3 ...
##  $ apartment.type                : int  1 1 1 1 2 1 2 2 2 1 ...
##  $ bank.credits                  : int  1 2 1 2 2 2 2 1 2 1 ...
##  $ occupation                    : int  3 3 2 2 2 2 2 2 1 1 ...
##  $ dependents                    : int  1 2 1 2 1 2 1 2 1 1 ...
##  $ telephone                     : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ foreign.worker                : int  1 1 1 2 2 2 2 2 1 1 ...
```

```r
# No NAs, complete data
sum(is.na(credit.df)) > 0
```

```
## [1] FALSE
```

```r
sum(complete.cases(credit.df))
```

```
## [1] 1000
```

```r
nrow(credit.df) == sum(complete.cases(credit.df))
```

```
## [1] TRUE
```

# 3 Functions

Functions are written to transform categorical variables to factors, generate summary statistics and generate visualisation. All dependent on the type of data used.

```r
# Transform data
to.factors <- function(df, variables) {
  for (variable in variables) {
    df[[variable]] <- as.factor(df[[variable]])
  }
  return(df)
```

```
}

factor.vars <- c("credit.rating", "account.balance", "previous.credit.payment.status", "credit.purpose"

credit.df <- to.factors(df = credit.df, variables = factor.vars)

# Summary statistics: numerical
get.numeric.variable.stats <- function(indep.var, detailed = FALSE) {

  options(scipen = 100)
  options(digits = 2)

  if (detailed) {
    var.stats <- stat.desc(indep.var)
  } else {
    var.stats <- summary(indep.var)
  }

  df <- data.frame(round(as.numeric(var.stats), 2))
  colnames(df) <- deparse(substitute(indep.var))
  rownames(df) <- names(var.stats)

  if (names(dev.cur()) != "null device") {
    dev.off()
  }

  grid.table(t(df))
}

# Summary statistics: categorical
get.categorical.variable.stats <- function(indep.var) {

  feature.name <- deparse(substitute(indep.var))
  df1 <- data.frame(table(indep.var))
  colnames(df1) <- c (feature.name, "Frequency")

  df2 <- data.frame(prop.table(table(indep.var)))
  colnames(df2) <- c(feature.name, "Proportion")

  df <- merge(df1, df2, by = feature.name)

  ndf <- df[order(-df$Frequency), ]

  if (names(dev.cur()) != "null device") {
    dev.off()
  }

  grid.table(ndf)
}

# Generate contingency table
get.contingency.table <- function(dep.var, indep.var, stat.tests = F) {
  if (stat.tests == F) {
    CrossTable(dep.var, indep.var, digits = 1, prop.r = F, prop.t = F, prop.chisq = F)
  } else {
    CrossTable(dep.var, indep.var, digits = 1, prop.r = F, prop.t = F, prop.chisq = F, chisq = T, fishe
  }
}

# Visualisation
## Histograms and density plots
visualize.distribution <- function(indep.var) {

  pl1 <- qplot(indep.var, geom = "histogram", fill = I("gray"), binwidth = 5, col = I("black")) + theme_

  pl2 <- qplot(indep.var, geom = "density", fill = I("gray"), binwidth = 5, col = I("black")) + theme_b

  grid.arrange(pl1, pl2, ncol = 2)
}

## Box plots
visualize.boxplot <- function(indep.var, dep.var) {

  pl1 <- qplot(factor(0), indep.var, geom = "boxplot", xlab = deparse(substitute(indep.var)), ylab = "V

  pl2 <- qplot(dep.var, indep.var, geom = "boxplot", xlab = deparse(substitute(dep.var)), ylab = depars

  grid.arrange(pl1, pl2, ncol = 2)
}

## Bar charts
visualize.barchart <- function(indep.var) {

  qplot(indep.var, geom = "bar", fill = I("gray"), col = I("black"), xlab = deparse(substitute(indep.va
}

## Mosaic plots
visualize.contingency.table <- function(dep.var, indep.var) {

  if(names(dev.cur()) != "null device") {
    dev.off()
  }

  mosaicplot(dep.var ~ indep.var, color = T, main = "Contingency Table Plot")
}
```

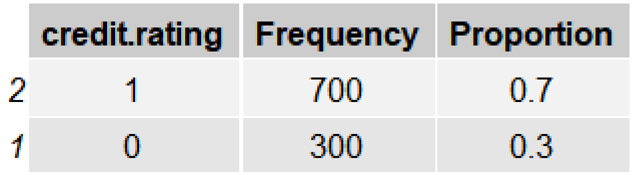
```
# Attach data to enable direct calls
attach(credit.df)
```

## 4 Descriptive Analysis

The actual descriptive analysis will start from here and will build on all previously declared principles.

## 4.1 Credit Rating

```
# Credit rating
get.categorical.variable.stats(credit.rating)
visualize.barchart(credit.rating)
```



| | credit.rating | Frequency | Proportion |
|---|---|---|---|
| 2 | 1 | 700 | 0.7 |
| *1* | 0 | 300 | 0.3 |

Figure 1: Credit Rating statistics

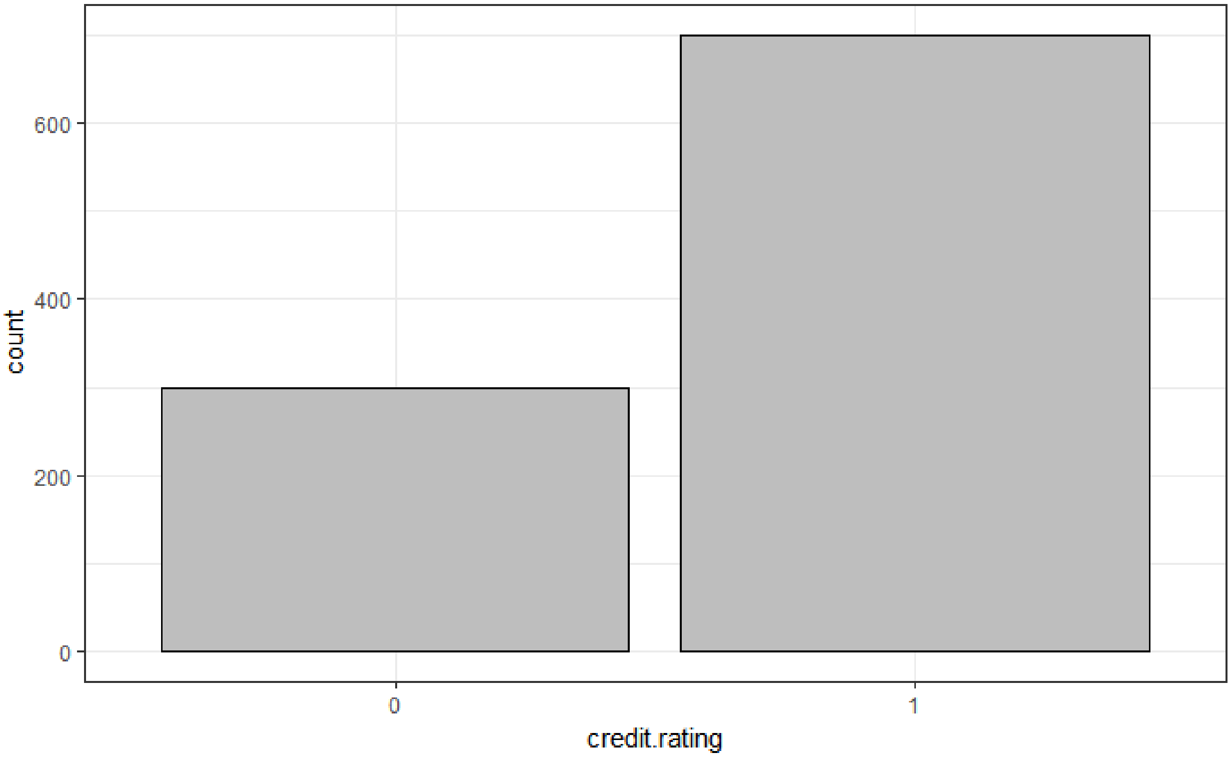


Figure 2: Visualisation of Credit Rating

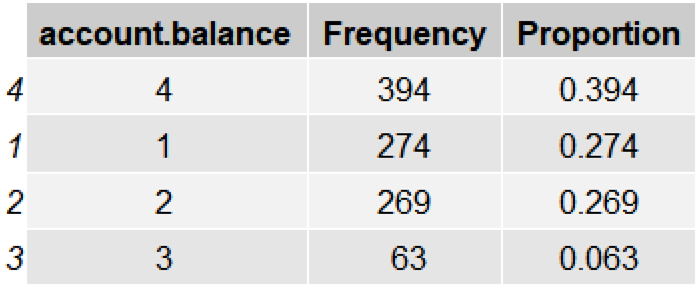
## 4.2 Account Balance

```
# Account balance
get.categorical.variable.stats(account.balance)
visualize.barchart(account.balance)

## Combine class 3 and 4: new class indicaties positive balance in account
```

```r
new.account.balance <- recode(account.balance, "1=1;2=2;3=3;4=3")
credit.df$account.balance <- new.account.balance

# Relationship: credit rating ~ new account balance and visualisation
get.contingency.table(credit.rating, new.account.balance, stat.tests = T)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Col Total |
## |-------------------------|
##
##
## Total Observations in Table:  1000
##
##
##              | indep.var
##      dep.var |         1 |         2 |         3 | 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
##
##
##
## Fisher's Exact Test for Count Data
## ------------------------------------------------------------
## Alternative hypothesis: two.sided
## p =  3.400743e-28
##
##
```

```r
visualize.contingency.table(credit.rating, new.account.balance)
```



| | account.balance | Frequency | Proportion |
|---|---|---|---|
| *4* | 4 | 394 | 0.394 |
| *1* | 1 | 274 | 0.274 |
| *2* | 2 | 269 | 0.269 |
| *3* | 3 | 63 | 0.063 |

Figure 3: Account Balance statistics

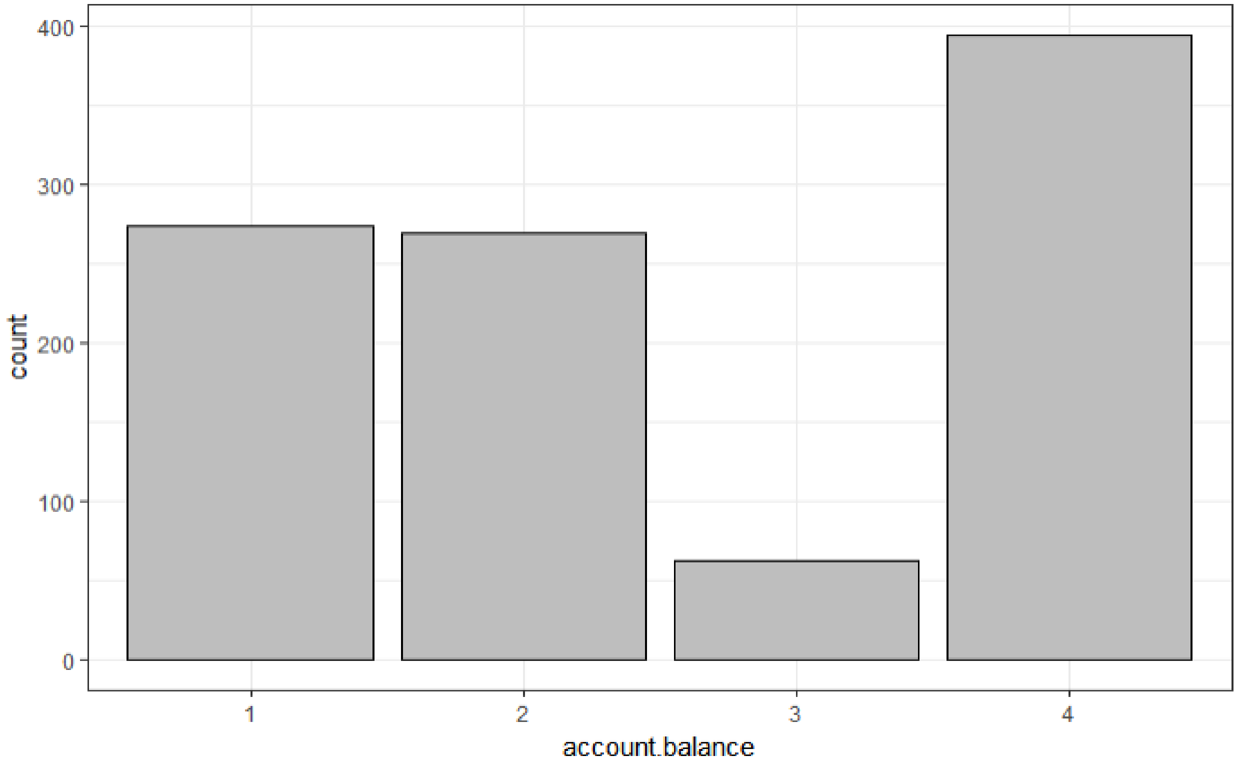


Figure 4: Visualisation of Account Balance classes

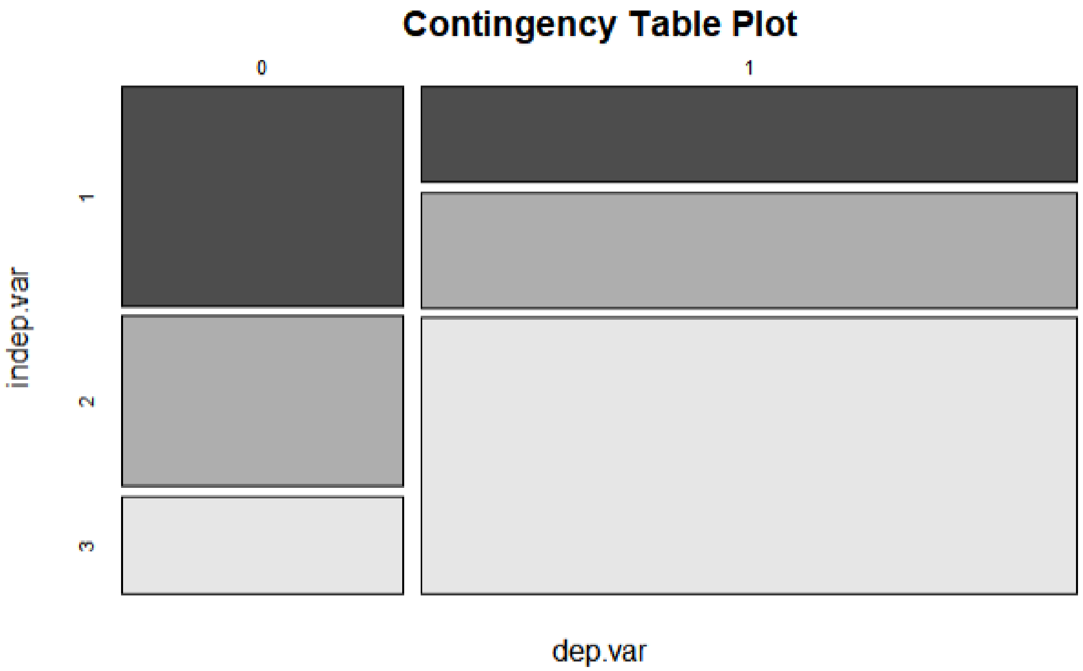


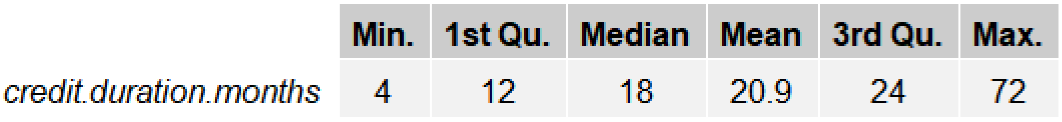
Figure 5: Visualisation of Credit Rating ~ Account Balance contingency table

### 4.3 Credit Duration

```
# Credit duration months
## Summary
get.numeric.variable.stats(credit.duration.months)

## Histogram and density plots
visualize.distribution(credit.duration.months)
```

```
## Warning: Ignoring unknown parameters: binwidth
```

```
# Credit duration months ~ credit rating
## Box plots
visualize.boxplot(credit.duration.months, credit.rating)
```



| | Min. | 1st Qu. | Median | Mean | 3rd Qu. | Max. |
|---|---|---|---|---|---|---|
| *credit.duration.months* | 4 | 12 | 18 | 20.9 | 24 | 72 |

Figure 6: Credit Duration statistics

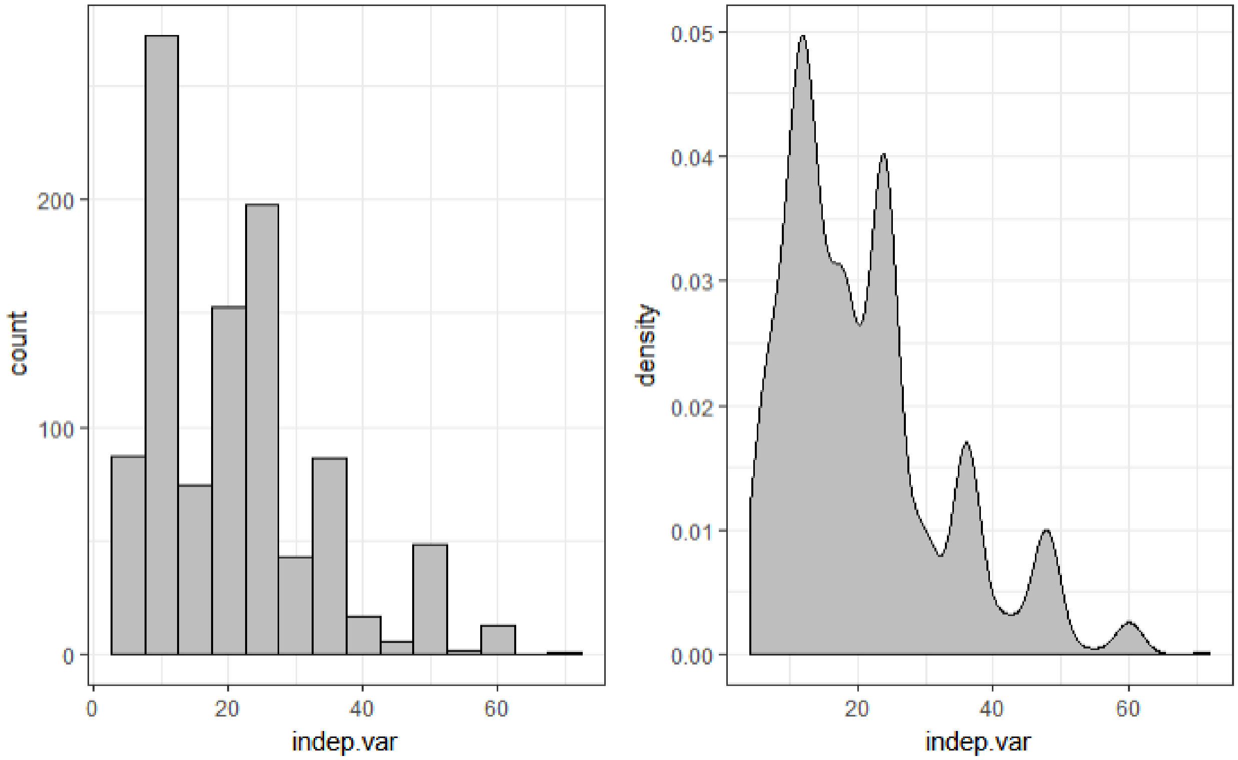


Figure 7: Histogram and density visualisation of Credit Duration

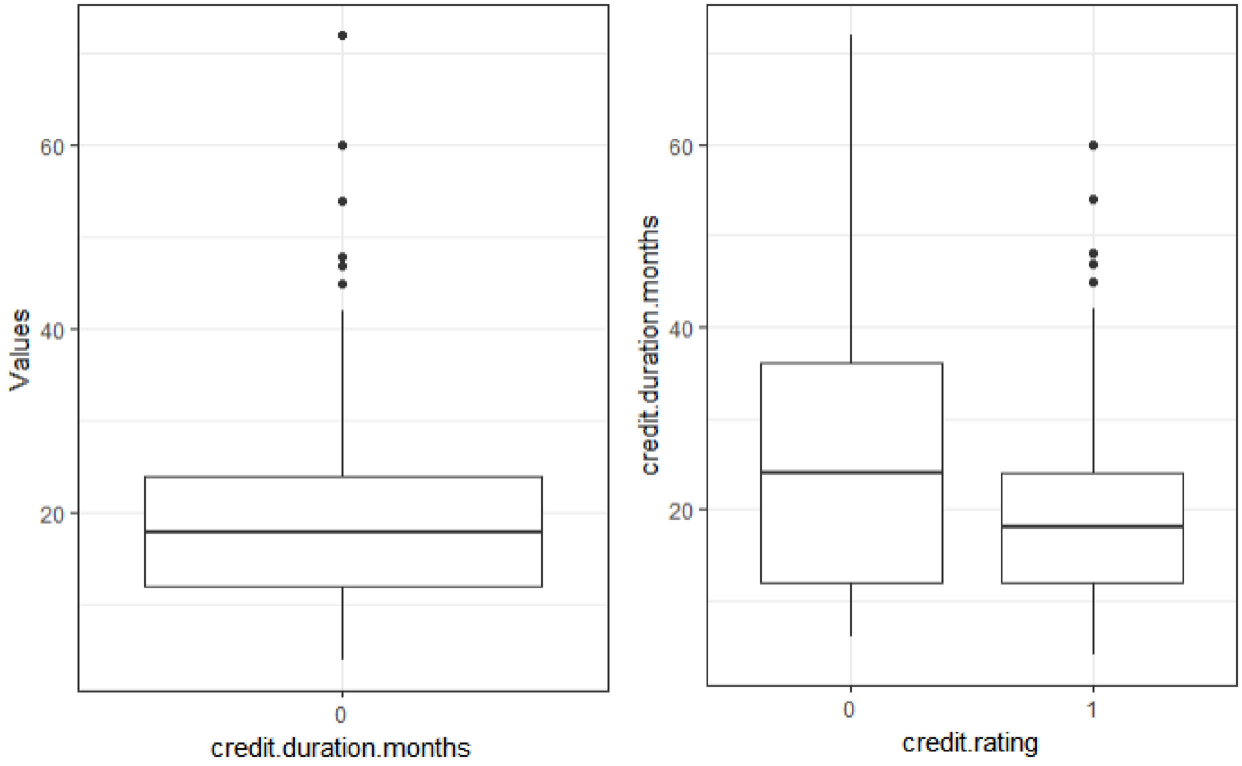


Figure 8: Boxplots for Credit Duration and Credit Rating

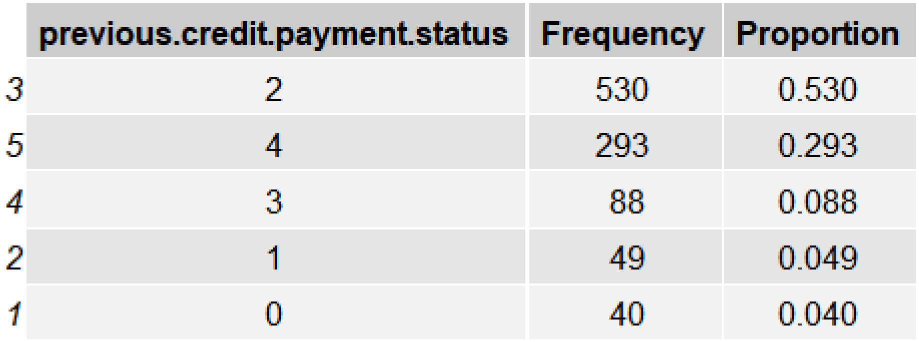
## 4.4 Previous Credit Payment Status

```
# Previous credit payment status
get.categorical.variable.stats(previous.credit.payment.status)
visualize.barchart(previous.credit.payment.status)

# Combine semantics: 0 + 1 and 3 + 4
new.previous.credit.payment.status <- recode(previous.credit.payment.status, "0=1;1=1;2=2;3=3;4=3")
credit.df$previous.credit.payment.status <- new.previous.credit.payment.status

# Contingency table tranformed semantics: credit rating ~ updated previous credit payment status
get.contingency.table(credit.rating, new.previous.credit.payment.status)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Col Total |
## |-------------------------|
##
##
## Total Observations in Table:  1000
##
##
##              | indep.var
##      dep.var |         1 |         2 |         3 | Row Total |
## -------------|-----------|-----------|-----------|-----------|
##            0 |        53 |       169 |        78 |       300 |
##              |       0.6 |       0.3 |       0.2 |           |
## -------------|-----------|-----------|-----------|-----------|
##            1 |        36 |       361 |       303 |       700 |
##              |       0.4 |       0.7 |       0.8 |           |
## -------------|-----------|-----------|-----------|-----------|
## Column Total |        89 |       530 |       381 |      1000 |
##              |       0.1 |       0.5 |       0.4 |           |
## -------------|-----------|-----------|-----------|-----------|
##
##
```

## 4.5 Credit Purpose

```
# Credit purpose: statistics and visualisation
get.categorical.variable.stats(credit.purpose)
visualize.barchart(credit.purpose)

# Feature engineering on semantics: 1 = new car, 2 = used car, 3 = furniture items, 4 = others {contain
new.credit.purpose <- recode(credit.purpose, "0=4;1=1;2=2;3=3;4=3;5=3;6=3;7=4;8=4;9=4;10=4")
credit.df$credit.purpose <- new.credit.purpose

# Contingency table: credit rating ~ updated credit purpose
get.contingency.table(credit.rating, new.credit.purpose, stat.tests = TRUE)
```



| | previous.credit.payment.status | Frequency | Proportion |
|---|---|---|---|
| 3 | 2 | 530 | 0.530 |
| 5 | 4 | 293 | 0.293 |
| 4 | 3 | 88 | 0.088 |
| 2 | 1 | 49 | 0.049 |
| 1 | 0 | 40 | 0.040 |

Figure 9: Previous Credit Payment Status statistics

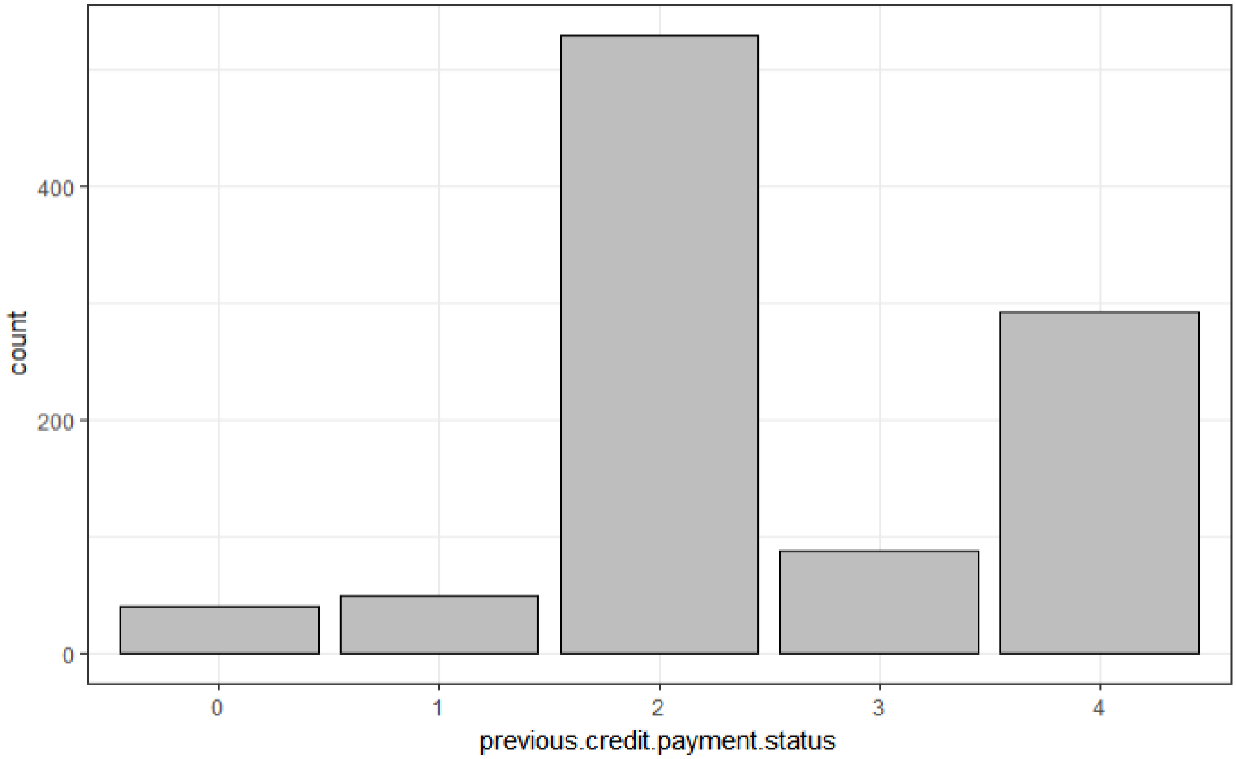


Figure 10: Visualisation of Previous Credit Payment Status classes

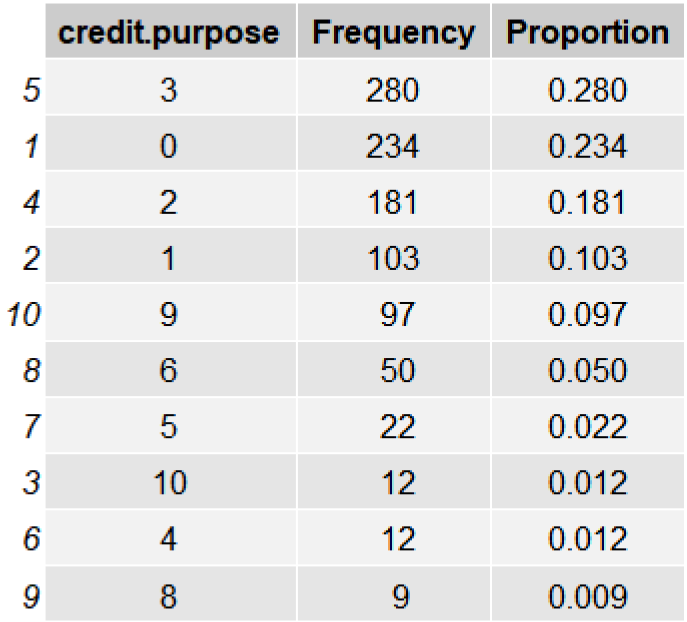
```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Col Total |
## |-------------------------|
##
##
## Total Observations in Table:  1000
##
##
##              | indep.var
##      dep.var |         1 |         2 |         3 |         4 | Row Total |
## -------------|-----------|-----------|-----------|-----------|-----------|
##            0 |        17 |        58 |        96 |       129 |       300 |
##              |       0.2 |       0.3 |       0.3 |       0.4 |           |
## -------------|-----------|-----------|-----------|-----------|-----------|
##            1 |        86 |       123 |       268 |       223 |       700 |
##              |       0.8 |       0.7 |       0.7 |       0.6 |           |
## -------------|-----------|-----------|-----------|-----------|-----------|
## Column Total |       103 |       181 |       364 |       352 |      1000 |
##              |       0.1 |       0.2 |       0.4 |       0.4 |           |
## -------------|-----------|-----------|-----------|-----------|-----------|
##
##
## Statistics for All Table Factors
##
##
## Pearson's Chi-squared test
## ------------------------------------------------------------
## Chi^2 =  19     d.f. =  3     p =  0.00028
##
##
##
## Fisher's Exact Test for Count Data
## ------------------------------------------------------------
## Alternative hypothesis: two.sided
## p =  0.00021
##
##
```

## 4.6 Credit Amount

```
# Credit amount: statistics and visualisation
get.numeric.variable.stats(credit.amount)
visualize.distribution(credit.amount)
```

```
## Warning: Ignoring unknown parameters: binwidth
```

```
# Relationship: credit amount ~ credit rating
visualize.boxplot(credit.amount, credit.rating)
```



|  | credit.purpose | Frequency | Proportion |
|---|---|---|---|
| *5* | 3 | 280 | 0.280 |
| *1* | 0 | 234 | 0.234 |
| *4* | 2 | 181 | 0.181 |
| *2* | 1 | 103 | 0.103 |
| *10* | 9 | 97 | 0.097 |
| *8* | 6 | 50 | 0.050 |
| *7* | 5 | 22 | 0.022 |
| *3* | 10 | 12 | 0.012 |
| *6* | 4 | 12 | 0.012 |
| *9* | 8 | 9 | 0.009 |

Figure 11: Credit Purpose statistics

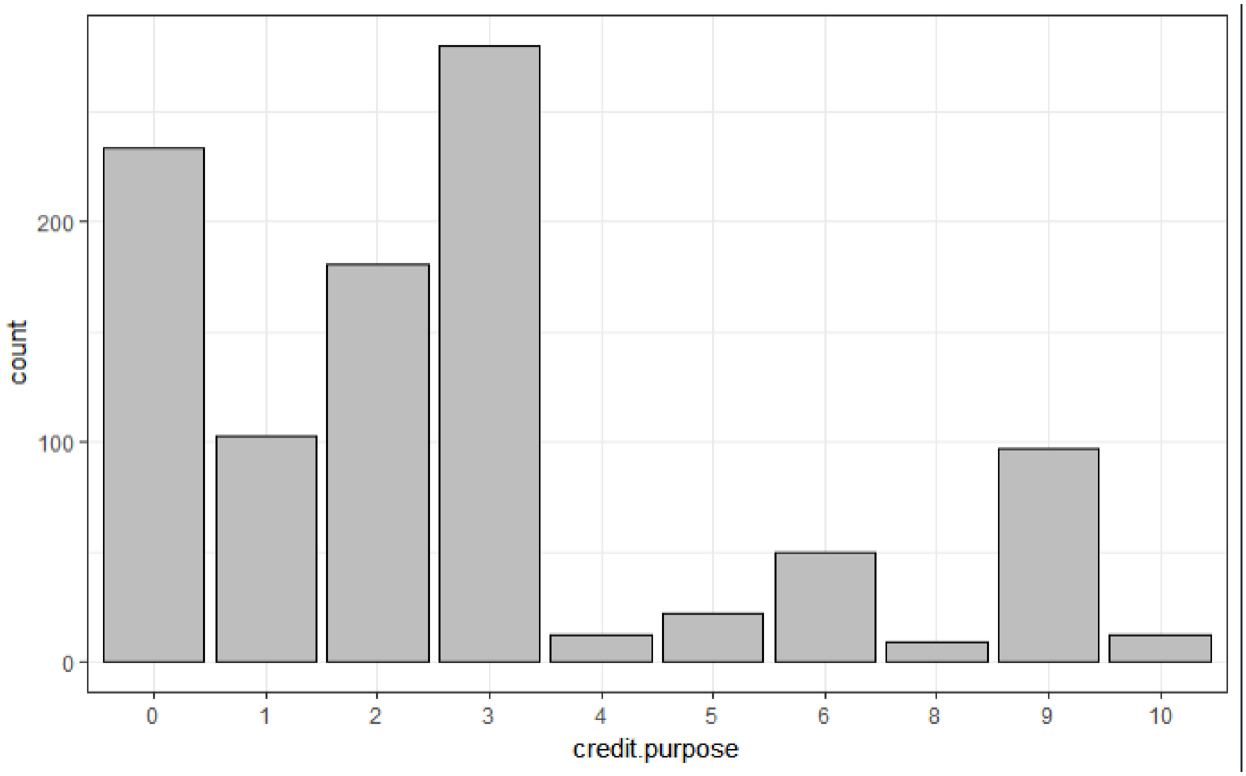
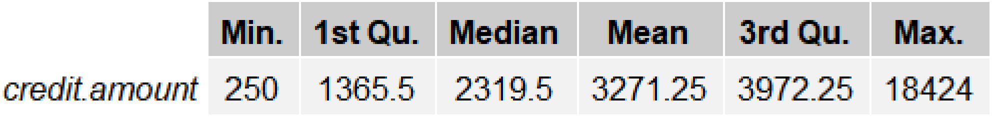


Figure 12: Visualisation of Credit Purpose classes



| | Min. | 1st Qu. | Median | Mean | 3rd Qu. | Max. |
|---|---|---|---|---|---|---|
| *credit.amount* | 250 | 1365.5 | 2319.5 | 3271.25 | 3972.25 | 18424 |

Figure 13: Credit Amount statistics

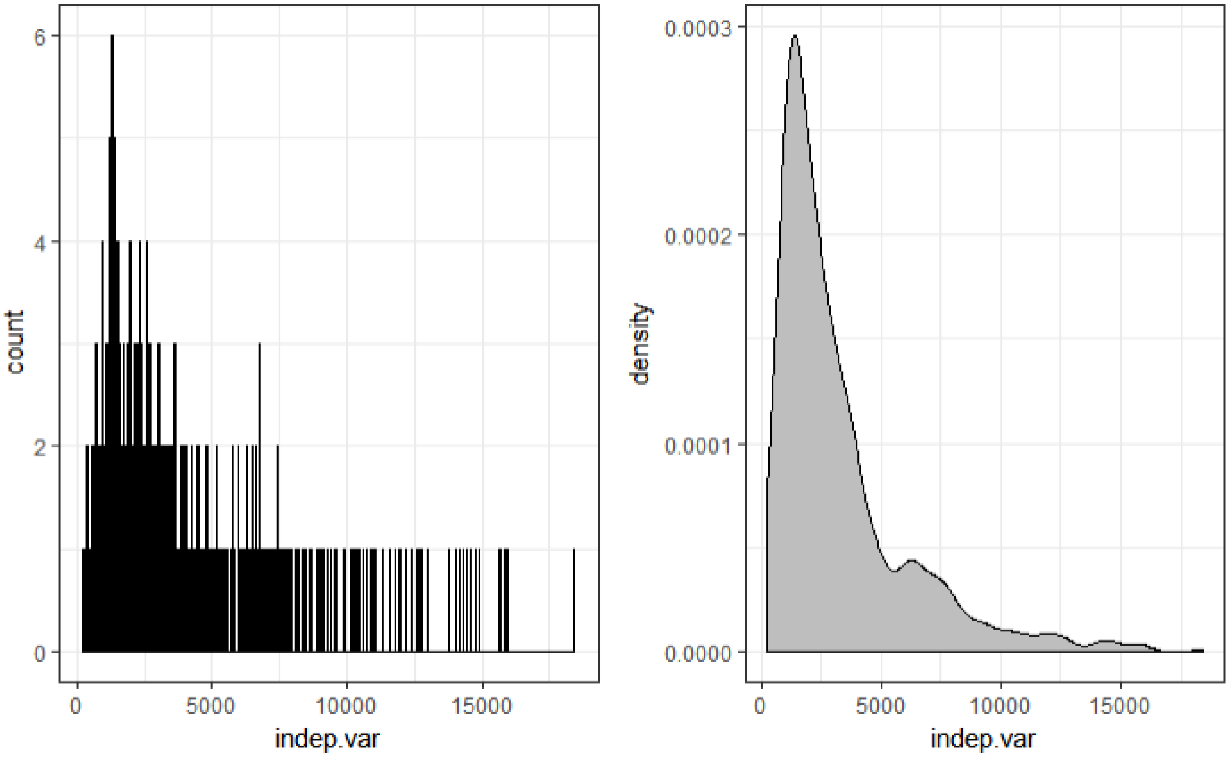


Figure 14: Histogram and density visualisation of Credit Amount

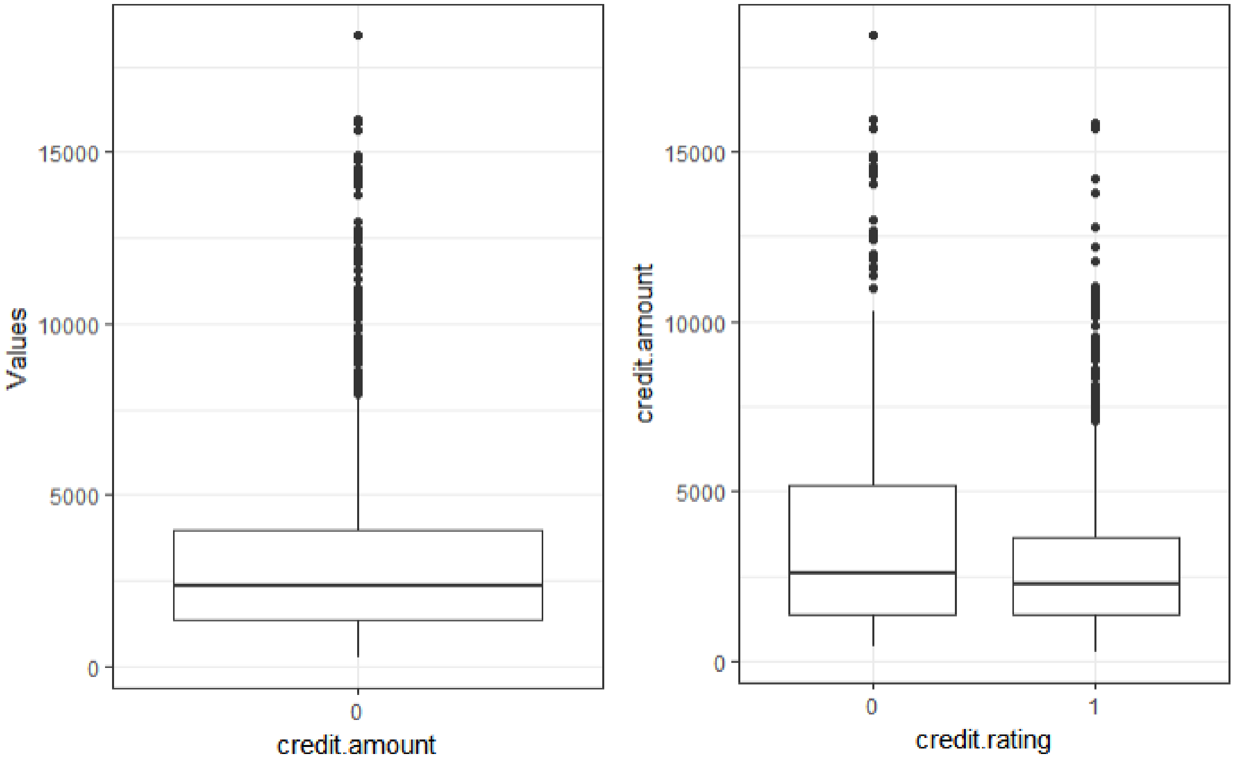


Figure 15: Boxplots for Credit Amount and Credit Rating

## 4.7 Savings

```
# Savings: recode semantics
new.savings <- recode(savings, "1=1;2=2;3=3;4=3;5=4")
credit.df$savings <- new.savings

# Contingency table: credit rating ~ savings
get.contingency.table(credit.rating, savings, stat.tests = FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Col Total |
## |-------------------------|
##
##
## Total Observations in Table:  1000
##
##
##              | indep.var
##      dep.var |         1 |         2 |         3 |         4 |         5 | Row Total |
## -------------|-----------|-----------|-----------|-----------|-----------|-----------|
##            0 |       217 |        34 |        11 |         6 |        32 |       300 |
##              |       0.4 |       0.3 |       0.2 |       0.1 |       0.2 |           |
## -------------|-----------|-----------|-----------|-----------|-----------|-----------|
##            1 |       386 |        69 |        52 |        42 |       151 |       700 |
##              |       0.6 |       0.7 |       0.8 |       0.9 |       0.8 |           |
## -------------|-----------|-----------|-----------|-----------|-----------|-----------|
## Column Total |       603 |       103 |        63 |        48 |       183 |      1000 |
##              |       0.6 |       0.1 |       0.1 |       0.0 |       0.2 |           |
## -------------|-----------|-----------|-----------|-----------|-----------|-----------|
##
##
```

## 4.8 Employment Duration & Installment Rate

```r
# Employment duration: recode semantics
new.employment.duration <- recode(employment.duration, "1=1;2=1;3=2;4=3;5=4")
credit.df$employment.duration <- new.employment.duration

# Contingency table: credit rating ~ updated employment duration
get.contingency.table(credit.rating, new.employment.duration, stat.tests = TRUE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Col Total |
## |-------------------------|
##
##
## Total Observations in Table:  1000
##
##
##              | indep.var
##      dep.var |         1 |         2 |         3 |         4 | Row Total |
## -------------|-----------|-----------|-----------|-----------|-----------|
##            0 |        93 |       104 |        39 |        64 |       300 |
##              |       0.4 |       0.3 |       0.2 |       0.3 |           |
## -------------|-----------|-----------|-----------|-----------|-----------|
##            1 |       141 |       235 |       135 |       189 |       700 |
##              |       0.6 |       0.7 |       0.8 |       0.7 |           |
## -------------|-----------|-----------|-----------|-----------|-----------|
## Column Total |       234 |       339 |       174 |       253 |      1000 |
##              |       0.2 |       0.3 |       0.2 |       0.3 |           |
## -------------|-----------|-----------|-----------|-----------|-----------|
##
##
## Statistics for All Table Factors
##
##
## Pearson's Chi-squared test
## -----------------------------------------------------------
## Chi^2 =  18     d.f. =  3     p =  0.00042
##
##
```

```
##
## Fisher's Exact Test for Count Data
## ------------------------------------------------------------
## Alternative hypothesis: two.sided
## p =  0.00048
##
##
```

```
# Contingency table: credit rating ~ installment.rate
get.contingency.table(credit.rating, installment.rate, stat.tests = TRUE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Col Total |
## |-------------------------|
##
##
## Total Observations in Table:  1000
##
##
##              | indep.var
##      dep.var |         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 =  5.5     d.f. =  3     p =  0.14
##
##
##
## Fisher's Exact Test for Count Data
## ------------------------------------------------------------
## Alternative hypothesis: two.sided
## p =  0.15
##
##
```

### 4.9 Marital Status & Guarantor

```
# Marital status: recode semantics
new.marital.status <- recode(marital.status, "1=1;2=1;3=2;4=3")
credit.df$marital.status <- new.marital.status

# Contingency table: credit rating ~ updated marital status
get.contingency.table(credit.rating, new.marital.status, stat.tests = TRUE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Col Total |
## |-------------------------|
##
##
## Total Observations in Table:  1000
##
##
##              | indep.var
##      dep.var |         1 |         2 |         3 | Row Total |
## -------------|-----------|-----------|-----------|-----------|
##            0 |       129 |       146 |        25 |       300 |
##              |       0.4 |       0.3 |       0.3 |           |
## -------------|-----------|-----------|-----------|-----------|
##            1 |       231 |       402 |        67 |       700 |
##              |       0.6 |       0.7 |       0.7 |           |
## -------------|-----------|-----------|-----------|-----------|
## Column Total |       360 |       548 |        92 |      1000 |
##              |       0.4 |       0.5 |       0.1 |           |
## -------------|-----------|-----------|-----------|-----------|
##
##
## Statistics for All Table Factors
##
##
## Pearson's Chi-squared test
## ------------------------------------------------------------
## Chi^2 =  9.1     d.f. =  2     p =  0.01
##
##
##
## Fisher's Exact Test for Count Data
## ------------------------------------------------------------
## Alternative hypothesis: two.sided
## p =  0.011
##
##
```

```
# Guarantor: recode semantics and relationship tests
new.guarantor <- recode(guarantor, "1=1;2=2;3=2")
credit.df$marital.status <- new.guarantor
```

```r
get.contingency.table(credit.rating, new.guarantor, stat.tests = TRUE)
```

```
## 
## 
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Col Total |
## |-------------------------|
## 
## 
## Total Observations in Table:  1000 
## 
## 
##              | indep.var 
##      dep.var |         1 |         2 | 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 =  0.00056     d.f. =  1     p =  0.98 
## 
## Pearson's Chi-squared test with Yates' continuity correction 
## ------------------------------------------------------------
## Chi^2 =  0.0000000000000000000000000000061     d.f. =  1     p =  1 
## 
## 
## Fisher's Exact Test for Count Data
## ------------------------------------------------------------
## Sample estimate odds ratio:  0.99 
## 
## Alternative hypothesis: true odds ratio is not equal to 1
## p =  1 
## 95% confidence interval:  0.61 1.6 
## 
## Alternative hypothesis: true odds ratio is less than 1
## p =  0.53 
## 95% confidence interval:  0 1.5 
## 
## Alternative hypothesis: true odds ratio is greater than 1
## p =  0.56 
## 95% confidence interval:  0.66 Inf
##
##
##
```

### 4.10 Residence Duration & Current Assets

```
# Relationship: credit rating ~ residence duration
fisher.test(credit.rating, residence.duration)
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  credit.rating and residence.duration
## p-value = 0.9
## alternative hypothesis: two.sided
```

```
chisq.test(credit.rating, residence.duration)
```

```
##
##  Pearson's Chi-squared test
##
## data:  credit.rating and residence.duration
## X-squared = 0.7, df = 3, p-value = 0.9
```

```
# Relationship: credit rating ~ current assets
fisher.test(credit.rating, current.assets)
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  credit.rating and current.assets
## p-value = 0.00003
## alternative hypothesis: two.sided
```

```
chisq.test(credit.rating, current.assets)
```

```
##
##  Pearson's Chi-squared test
##
## data:  credit.rating and current.assets
## X-squared = 20, df = 3, p-value = 0.00003
```

### 4.11 Remaining features

This section includes visualisations for Age, recoding and testing the relationship of credit rating with Other Credits, Apartment Type, Bank Credits, Occupation, Dependents, Telephone and Foreign Worker.

```
# Age: stats and visualisation
get.numeric.variable.stats(age)
visualize.distribution(age)
```

```
## Warning: Ignoring unknown parameters: binwidth
```

```
visualize.boxplot(age, credit.rating)
```

```r
# Other credits: recode semantics and investigate relationship
new.other.credits <- recode(other.credits, "1=1;2=1;3=2")
credit.df$other.credits <- new.other.credits
fisher.test(credit.rating, new.other.credits)
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  credit.rating and new.other.credits
## p-value = 0.0005
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  1.3 2.6
## sample estimates:
## odds ratio
##        1.8
```

```r
chisq.test(credit.rating, new.other.credits)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  credit.rating and new.other.credits
## X-squared = 10, df = 1, p-value = 0.0005
```

```r
# Apartment type: relationship
fisher.test(credit.rating, apartment.type)
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  credit.rating and apartment.type
## p-value = 0.0001
## alternative hypothesis: two.sided
```

```r
chisq.test(credit.rating, apartment.type)
```

```
##
##  Pearson's Chi-squared test
##
## data:  credit.rating and apartment.type
## X-squared = 20, df = 2, p-value = 0.00009
```

```r
# Bank credits: recode semantics and test relationship
new.bank.credits <- recode(bank.credits, "1=1;2=2;3=2;4=2")
credit.df$bank.credits <- new.bank.credits
fisher.test(credit.rating, new.bank.credits)
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  credit.rating and new.bank.credits
## p-value = 0.2
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  0.92 1.66
## sample estimates:
```

```
## odds ratio
##        1.2
```

```r
chisq.test(credit.rating, new.bank.credits)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  credit.rating and new.bank.credits
## X-squared = 2, df = 1, p-value = 0.2
```

```r
# Occupation: relationships
fisher.test(credit.rating, occupation)
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  credit.rating and occupation
## p-value = 0.6
## alternative hypothesis: two.sided
```

```r
chisq.test(credit.rating, occupation)
```

```
##
##  Pearson's Chi-squared test
##
## data:  credit.rating and occupation
## X-squared = 2, df = 3, p-value = 0.6
```

```r
# Relationships for: Dependents, Telephone and Foreign Worker
fisher.test(credit.rating, dependents)
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  credit.rating and dependents
## p-value = 1
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  0.69 1.52
## sample estimates:
## odds ratio
##          1
```

```r
chisq.test(credit.rating, dependents)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  credit.rating and dependents
## X-squared = 0, df = 1, p-value = 1
```

```r
fisher.test(credit.rating, telephone)
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  credit.rating and telephone
## p-value = 0.3
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  0.88 1.57
## sample estimates:
## odds ratio
##        1.2
```

```r
chisq.test(credit.rating, telephone)
```

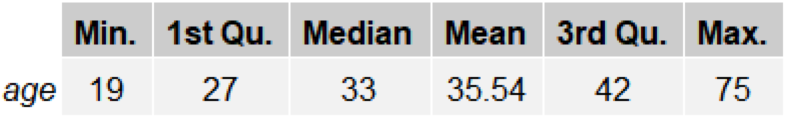
```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  credit.rating and telephone
## X-squared = 1, df = 1, p-value = 0.3
```

```r
fisher.test(credit.rating, foreign.worker)
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  credit.rating and foreign.worker
## p-value = 0.009
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##   1.3 14.3
## sample estimates:
## odds ratio
##        3.7
```

```r
chisq.test(credit.rating, foreign.worker)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  credit.rating and foreign.worker
## X-squared = 6, df = 1, p-value = 0.02
```



| | Min. | 1st Qu. | Median | Mean | 3rd Qu. | Max. |
|---|---|---|---|---|---|---|
| *age* | 19 | 27 | 33 | 35.54 | 42 | 75 |

Figure 16: Age statistics

# 5 Export

Finally, all transformed data is merged into a new dataset and saved to disk.

```r
# Save the new, tranformed data set
write.csv(file = "credit_dataset_transformed.csv", x = credit.df, row.names = F)
```

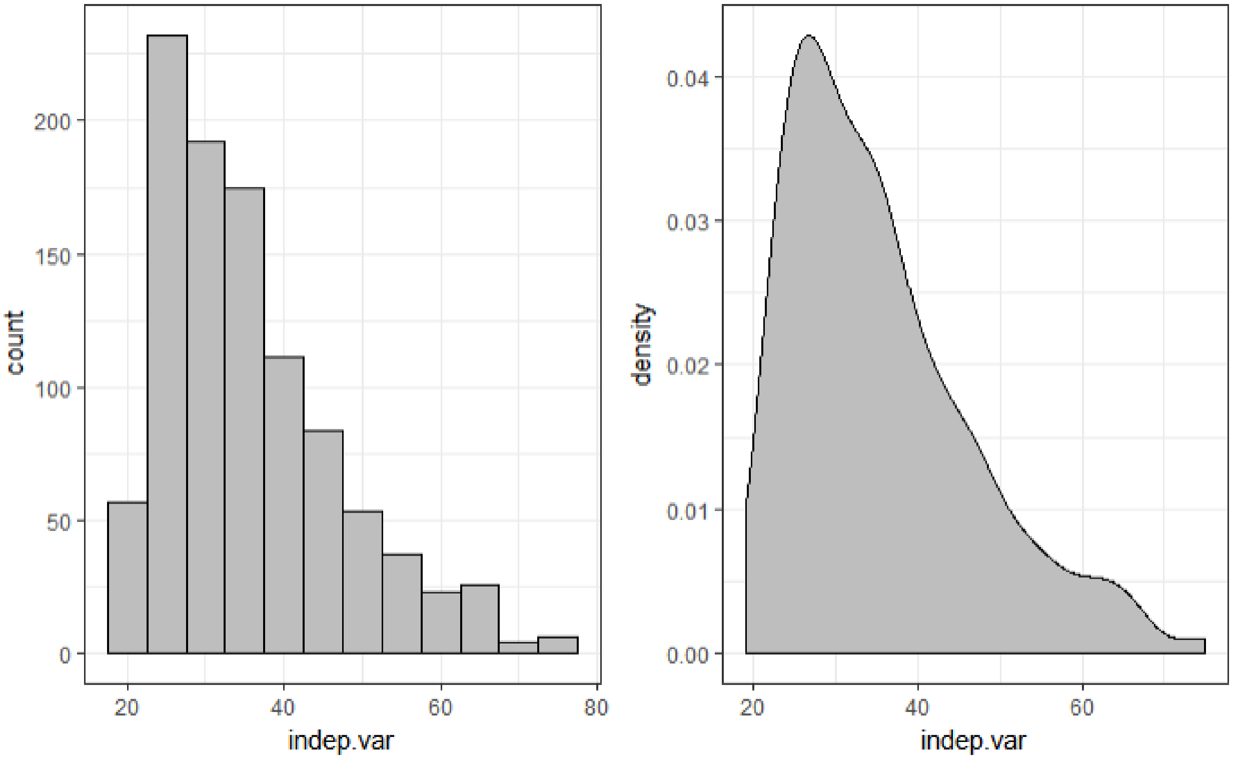


Figure 17: Histogram and density visualisations for Age

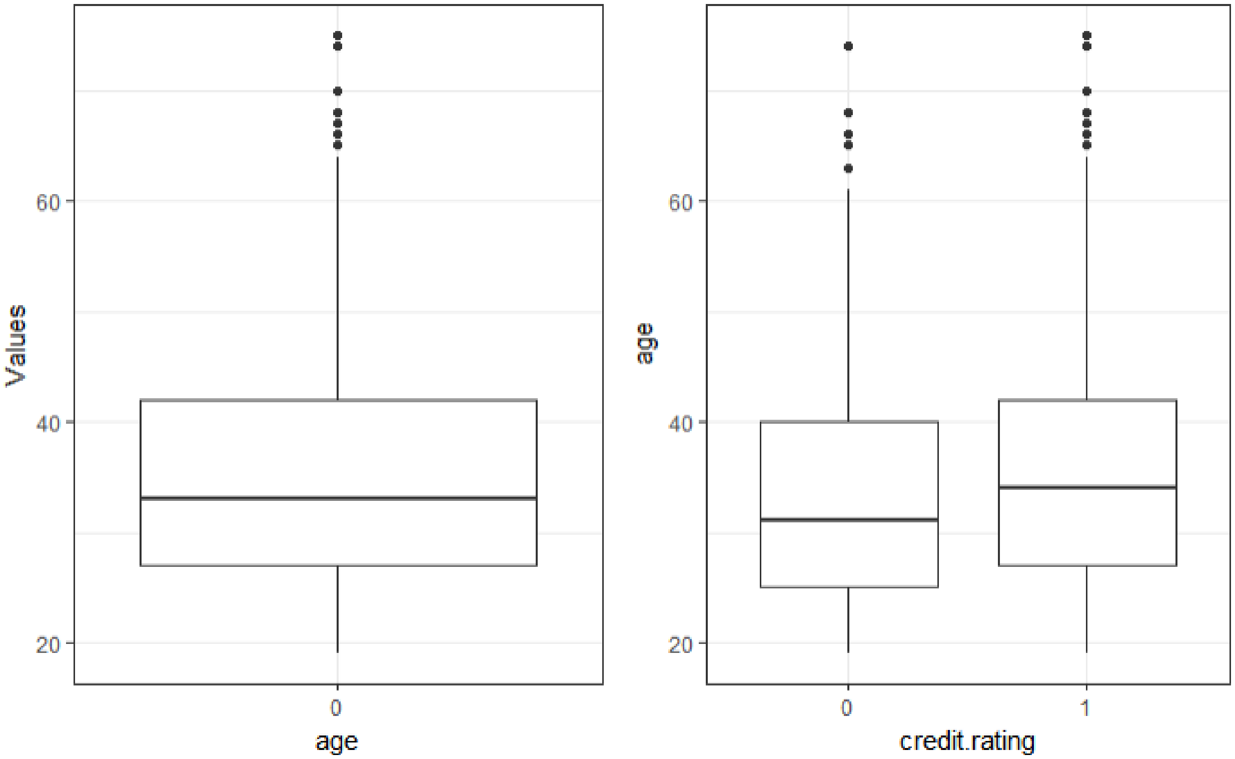


Figure 18: Boxplots for Age and Credit Rating