Modeling Credit Scoring / Credit Rating / Consumer Risk

#### ***Ariful Mondal***

#### ***20 September 2016***

1. Introduction

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

1.1 Setting up the environment and packages

# Set working directory  
setwd("C:/creditscoring")  
# For error handling  
x <- tryCatch(simpleError("eror mesid"), error = function(e) e)

# List of required libraries - packages  
library(lattice) # for visualization  
library(knitr) # for kable  
library(gplots)

##   
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':  
##   
## lowess

library(ggplot2) # for data visualization  
library(ClustOfVar) # for variable clustering  
#library(GPArotation)   
library(ape) # for as.phylo  
library(Information)  
library(ROCR) # for ROC  
library(caret)   
library(rpart) # for Traditional recursive Partitioning - Bayesian  
library(rpart.utils)  
library(rpart.plot)  
library(randomForest) # for random forest

## randomForest 4.6-12

## Type rfNews() to see new features/changes/bug fixes.

##   
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':  
##   
## margin

library(party) # Conditional inference Trees

## Loading required package: grid

## Loading required package: mvtnorm

## Loading required package: modeltools

## Loading required package: stats4

## Loading required package: strucchange

## Loading required package: zoo

##   
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric

## Loading required package: sandwich

##   
## Attaching package: 'party'

## The following object is masked from 'package:ape':  
##   
## where

library(bnlearn) # Bayesian Network

##   
## Attaching package: 'bnlearn'

## The following object is masked from 'package:stats':  
##   
## sigma

library(DAAG) #load Data Analysis And Graphics Package for R (DAAG)  
library(vcd) #the `vcd' package is required for CD plots  
#library(neuralnet) # Neural Network - call when you run neural network \_ predition function will be different.  
library(kernlab) # SVM

##   
## Attaching package: 'kernlab'

## The following object is masked from 'package:modeltools':  
##   
## prior

## The following object is masked from 'package:ggplot2':  
##   
## alpha

1.2 Reading raw data into R

# Read data into R (tab delimitted)  
cdata<-read.table("data.txt", h=T, sep="")  
  
# Col names  
  
# chk\_ac\_status\_1 duration\_month\_2 credit\_history\_3 purpose\_4 credit\_amount\_5 savings\_ac\_bond\_6 p\_employment\_since\_7 installment\_pct\_disp\_inc\_8 personal\_status\_9 other\_debtors\_or\_grantors\_10 present\_residence\_since\_11 property\_type\_12 age\_in\_yrs\_13 other\_installment\_type\_14 housing\_type\_15 number\_cards\_this\_bank\_16 job\_17 no\_people\_liable\_for\_mntnance\_18 telephone\_19 foreign\_worker\_20 good\_bad\_21  
  
colnames(cdata)

## [1] "chk\_ac\_status\_1" "duration\_month\_2" "credit\_history\_3" "purpose\_4" "credit\_amount\_5"   
## [6] "savings\_ac\_bond\_6" "p\_employment\_since\_7" "installment\_pct\_disp\_inc\_8" "personal\_status\_9" "other\_debtors\_or\_grantors\_10"   
## [11] "present\_residence\_since\_11" "property\_type\_12" "age\_in\_yrs\_13" "other\_installment\_type\_14" "housing\_type\_15"   
## [16] "number\_cards\_this\_bank\_16" "job\_17" "no\_people\_liable\_for\_mntnance\_18" "telephone\_19" "foreign\_worker\_20"   
## [21] "good\_bad\_21"

y <- c(0,1) # for abline  
x <- c(0,1) # for abline

1.3 Get first hand feeling of the data

# Get first hand feeling of the data  
str(cdata)

## 'data.frame': 1000 obs. of 21 variables:  
## $ chk\_ac\_status\_1 : Factor w/ 4 levels "A11","A12","A13",..: 1 2 4 1 1 4 4 2 4 2 ...  
## $ duration\_month\_2 : int 6 48 12 42 24 36 24 36 12 30 ...  
## $ credit\_history\_3 : Factor w/ 5 levels "A30","A31","A32",..: 5 3 5 3 4 3 3 3 3 5 ...  
## $ purpose\_4 : Factor w/ 10 levels "A40","A41","A410",..: 5 5 8 4 1 8 4 2 5 1 ...  
## $ credit\_amount\_5 : int 1169 5951 2096 7882 4870 9055 2835 6948 3059 5234 ...  
## $ savings\_ac\_bond\_6 : Factor w/ 5 levels "A61","A62","A63",..: 5 1 1 1 1 5 3 1 4 1 ...  
## $ p\_employment\_since\_7 : Factor w/ 5 levels "A71","A72","A73",..: 5 3 4 4 3 3 5 3 4 1 ...  
## $ installment\_pct\_disp\_inc\_8 : int 4 2 2 2 3 2 3 2 2 4 ...  
## $ personal\_status\_9 : Factor w/ 4 levels "A91","A92","A93",..: 3 2 3 3 3 3 3 3 1 4 ...  
## $ other\_debtors\_or\_grantors\_10 : Factor w/ 3 levels "A101","A102",..: 1 1 1 3 1 1 1 1 1 1 ...  
## $ present\_residence\_since\_11 : int 4 2 3 4 4 4 4 2 4 2 ...  
## $ property\_type\_12 : Factor w/ 4 levels "A121","A122",..: 1 1 1 2 4 4 2 3 1 3 ...  
## $ age\_in\_yrs\_13 : int 67 22 49 45 53 35 53 35 61 28 ...  
## $ other\_installment\_type\_14 : Factor w/ 3 levels "A141","A142",..: 3 3 3 3 3 3 3 3 3 3 ...  
## $ housing\_type\_15 : Factor w/ 3 levels "A151","A152",..: 2 2 2 3 3 3 2 1 2 2 ...  
## $ number\_cards\_this\_bank\_16 : int 2 1 1 1 2 1 1 1 1 2 ...  
## $ job\_17 : Factor w/ 4 levels "A171","A172",..: 3 3 2 3 3 2 3 4 2 4 ...  
## $ no\_people\_liable\_for\_mntnance\_18: int 1 1 2 2 2 2 1 1 1 1 ...  
## $ telephone\_19 : Factor w/ 2 levels "A191","A192": 2 1 1 1 1 2 1 2 1 1 ...  
## $ foreign\_worker\_20 : Factor w/ 2 levels "A201","A202": 1 1 1 1 1 1 1 1 1 1 ...  
## $ good\_bad\_21 : int 1 2 1 1 2 1 1 1 1 2 ...

summary(cdata)

## chk\_ac\_status\_1 duration\_month\_2 credit\_history\_3 purpose\_4 credit\_amount\_5 savings\_ac\_bond\_6 p\_employment\_since\_7 installment\_pct\_disp\_inc\_8 personal\_status\_9 other\_debtors\_or\_grantors\_10  
## A11:274 Min. : 4.0 A30: 40 A43 :280 Min. : 250 A61:603 A71: 62 Min. :1.000 A91: 50 A101:907   
## A12:269 1st Qu.:12.0 A31: 49 A40 :234 1st Qu.: 1366 A62:103 A72:172 1st Qu.:2.000 A92:310 A102: 41   
## A13: 63 Median :18.0 A32:530 A42 :181 Median : 2320 A63: 63 A73:339 Median :3.000 A93:548 A103: 52   
## A14:394 Mean :20.9 A33: 88 A41 :103 Mean : 3271 A64: 48 A74:174 Mean :2.973 A94: 92   
## 3rd Qu.:24.0 A34:293 A49 : 97 3rd Qu.: 3972 A65:183 A75:253 3rd Qu.:4.000   
## Max. :72.0 A46 : 50 Max. :18424 Max. :4.000   
## (Other): 55   
## present\_residence\_since\_11 property\_type\_12 age\_in\_yrs\_13 other\_installment\_type\_14 housing\_type\_15 number\_cards\_this\_bank\_16 job\_17 no\_people\_liable\_for\_mntnance\_18 telephone\_19  
## Min. :1.000 A121:282 Min. :19.00 A141:139 A151:179 Min. :1.000 A171: 22 Min. :1.000 A191:596   
## 1st Qu.:2.000 A122:232 1st Qu.:27.00 A142: 47 A152:713 1st Qu.:1.000 A172:200 1st Qu.:1.000 A192:404   
## Median :3.000 A123:332 Median :33.00 A143:814 A153:108 Median :1.000 A173:630 Median :1.000   
## Mean :2.845 A124:154 Mean :35.55 Mean :1.407 A174:148 Mean :1.155   
## 3rd Qu.:4.000 3rd Qu.:42.00 3rd Qu.:2.000 3rd Qu.:1.000   
## Max. :4.000 Max. :75.00 Max. :4.000 Max. :2.000   
##   
## foreign\_worker\_20 good\_bad\_21   
## A201:963 Min. :1.0   
## A202: 37 1st Qu.:1.0   
## Median :1.0   
## Mean :1.3   
## 3rd Qu.:2.0   
## Max. :2.0   
##

# print few observations  
kable(head(cdata), format="pandoc", padding=0, caption="How may the data look like?")

How may the data look like?

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| chk\_ac\_status\_1 | duration\_month\_2 | credit\_history\_3 | purpose\_4 | credit\_amount\_5 | savings\_ac\_bond\_6 | p\_employment\_since\_7 | installment\_pct\_disp\_inc\_8 | personal\_status\_9 | other\_debtors\_or\_grantors\_10 | present\_residence\_since\_11 | property\_type\_12 | age\_in\_yrs\_13 | other\_installment\_type\_14 | housing\_type\_15 | number\_cards\_this\_bank\_16 | job\_17 | no\_people\_liable\_for\_mntnance\_18 | telephone\_19 | foreign\_worker\_20 | good\_bad\_21 |
| A11 | 6 | A34 | A43 | 1169 | A65 | A75 | 4 | A93 | A101 | 4 | A121 | 67 | A143 | A152 | 2 | A173 | 1 | A192 | A201 | 1 |
| A12 | 48 | A32 | A43 | 5951 | A61 | A73 | 2 | A92 | A101 | 2 | A121 | 22 | A143 | A152 | 1 | A173 | 1 | A191 | A201 | 2 |
| A14 | 12 | A34 | A46 | 2096 | A61 | A74 | 2 | A93 | A101 | 3 | A121 | 49 | A143 | A152 | 1 | A172 | 2 | A191 | A201 | 1 |
| A11 | 42 | A32 | A42 | 7882 | A61 | A74 | 2 | A93 | A103 | 4 | A122 | 45 | A143 | A153 | 1 | A173 | 2 | A191 | A201 | 1 |
| A11 | 24 | A33 | A40 | 4870 | A61 | A73 | 3 | A93 | A101 | 4 | A124 | 53 | A143 | A153 | 2 | A173 | 2 | A191 | A201 | 2 |
| A14 | 36 | A32 | A46 | 9055 | A65 | A73 | 2 | A93 | A101 | 4 | A124 | 35 | A143 | A153 | 1 | A172 | 2 | A192 | A201 | 1 |

# convert integers to numeric  
cdata$duration\_month\_2 <- as.numeric(cdata$duration\_month\_2)   
cdata$credit\_amount\_5 <- as.numeric(cdata$credit\_amount\_5 )   
cdata$installment\_pct\_disp\_inc\_8 <- as.numeric(cdata$installment\_pct\_disp\_inc\_8)   
cdata$present\_residence\_since\_11 <- as.numeric(cdata$present\_residence\_since\_11)   
cdata$age\_in\_yrs\_13 <- as.numeric(cdata$age\_in\_yrs\_13)   
cdata$number\_cards\_this\_bank\_16 <- as.numeric(cdata$number\_cards\_this\_bank\_16)   
cdata$no\_people\_liable\_for\_mntnance\_18 <- as.numeric(cdata$no\_people\_liable\_for\_mntnance\_18)

2.Data analysis and variable creation

2.0 Create your own functions for analysis and modeling

# Function 1: Create function to calculate percent distribution for factors  
pct <- function(x){  
 tbl <- table(x)  
 tbl\_pct <- cbind(tbl,round(prop.table(tbl)\*100,2))  
 colnames(tbl\_pct) <- c('Count','Percentage')  
 kable(tbl\_pct)  
}  
  
#pct(cdata$good\_bad\_21)  
  
  
# Function 2: to calculate bad rates by groups - IV, WOE and Eefficiency   
gbpct <- function(x){  
 mt <- as.matrix(table(as.factor(x), as.factor(cdata$good\_bad\_21)))  
 Total <- mt[,1] + mt[,2]  
 Total\_Pct <- round(Total/sum(mt)\*100, 2)  
 Bad\_pct <- round((mt[,1]/sum(mt[,1]))\*100, 2)  
 Good\_pct <- round((mt[,2]/sum(mt[,2]))\*100, 2)  
 Bad\_Rate <- round((mt[,1]/(mt[,1]+mt[,2]))\*100, 2)  
 grp\_score <- round((Good\_pct/(Good\_pct + Bad\_pct))\*10, 2)  
 WOE <- round(log(Good\_pct/Bad\_pct)\*10, 2)  
 g\_b\_comp <- ifelse(mt[,1] == mt[,2], 0, 1)  
 IV <- ifelse(g\_b\_comp == 0, 0, (Good\_pct - Bad\_pct)\*(WOE/10))  
 Efficiency <- abs(Good\_pct - Bad\_pct)/2  
 otb<-as.data.frame(cbind(mt, Good\_pct, Bad\_pct, Total,   
 Total\_Pct, Bad\_Rate, grp\_score,   
 WOE, IV, Efficiency ))  
 otb$Names <- rownames(otb)  
 rownames(otb) <- NULL  
 otb[,c(12,2,1,3:11)]  
}  
  
  
# Function 3: Print the rules from part trees  
  
listrules<-function(model)  
{  
 if (!inherits(model, "rpart")) stop("Not a legitimate  
rpart tree")  
 #  
 # Get some information.  
 #  
 frm <- model$frame  
 names <- row.names(frm)  
 ylevels <- attr(model, "ylevels")  
 ds.size <- model$frame[1,]$n  
 #  
 # Print each leaf node as a rule.  
 #  
 for (i in 1:nrow(frm))  
 {  
 if (frm[i,1] == "<leaf>" & ylevels[frm[i,]$yval]=='bad')  
 {  
 # The following [,5] is hardwired - needs work!  
 cat("\n")  
 cat(sprintf(" Rule number: %s ", names[i]))  
 cat(sprintf("[yval=%s cover=%d N=%.0f Y=%.0f (%.0f%%)  
prob=%0.2f]\n",  
ylevels[frm[i,]$yval], frm[i,]$n,  
formatC(frm[i,]$yval2[,2], format = "f", digits = 2),  
formatC(frm[i,]$n-frm[i,]$yval2[,2], format = "f", digits  
 = 2),  
round(100\*frm[i,]$n/ds.size), frm[i,]  
$yval2[,5]))  
 pth <- path.rpart(model, nodes=as.numeric(names[i]),  
 print.it=FALSE)  
 cat(sprintf(" %s\n", unlist(pth)[-1]), sep="")  
 pth  
 }  
 }  
}  
#listrules(m2)

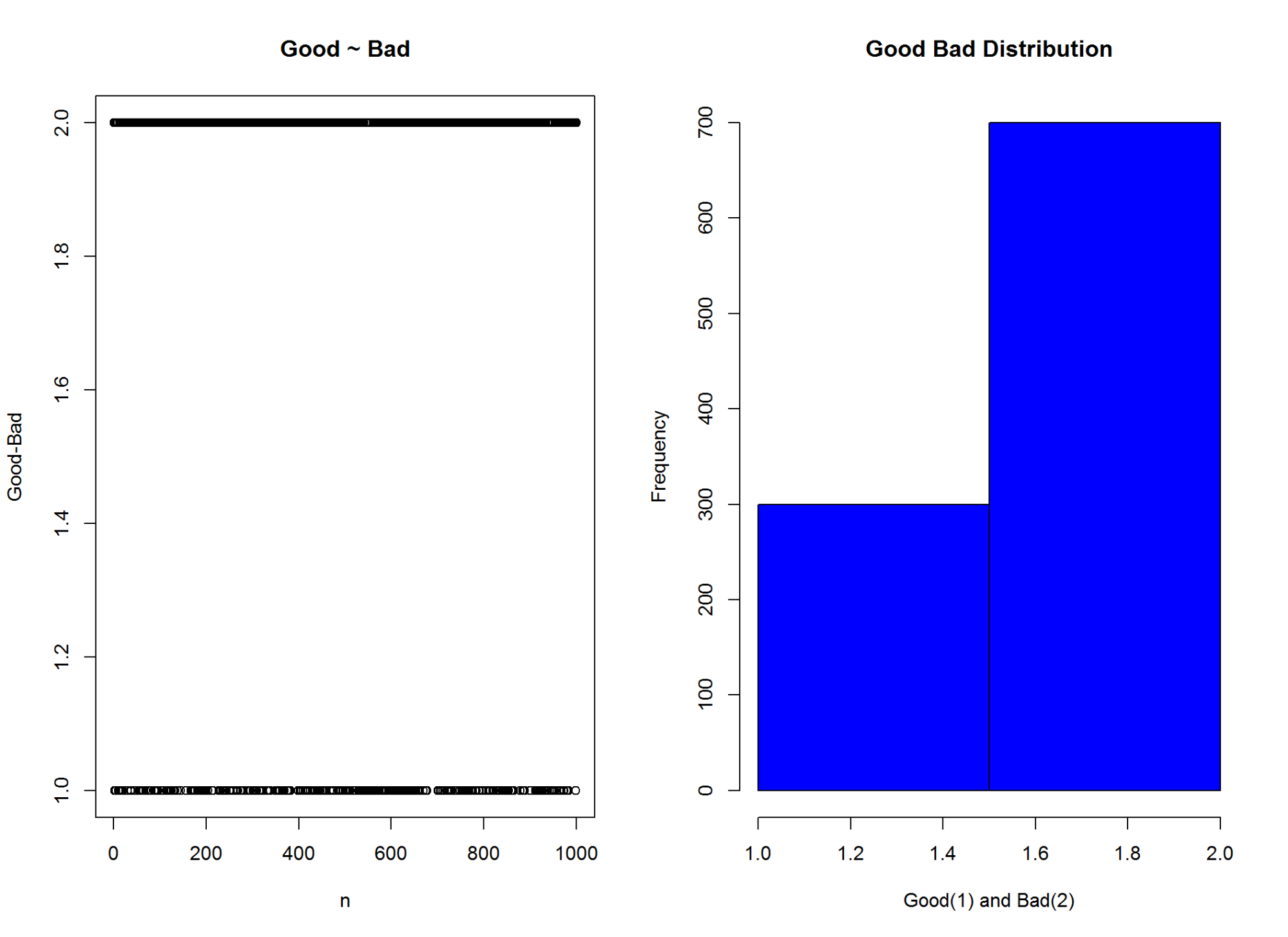
2.1 Good-Bad and understand relationships between variables:

2.1.1 Analyse good\_bad(1-good, 2-bad)

cdata$good\_bad\_21<-as.factor(ifelse(cdata$good\_bad\_21 == 1, "Good", "Bad"))  
pct(cdata$good\_bad\_21)

|  |  |  |
| --- | --- | --- |
|  | Count | Percentage |
| Bad | 300 | 30 |
| Good | 700 | 70 |

op<-par(mfrow=c(1,2), new=TRUE)  
  
plot(as.numeric(cdata$good\_bad\_21), ylab="Good-Bad", xlab="n", main="Good ~ Bad")  
hist(as.numeric(cdata$good\_bad\_21), breaks=2,   
 xlab="Good(1) and Bad(2)", col="blue",   
 main="Good Bad Distribution")

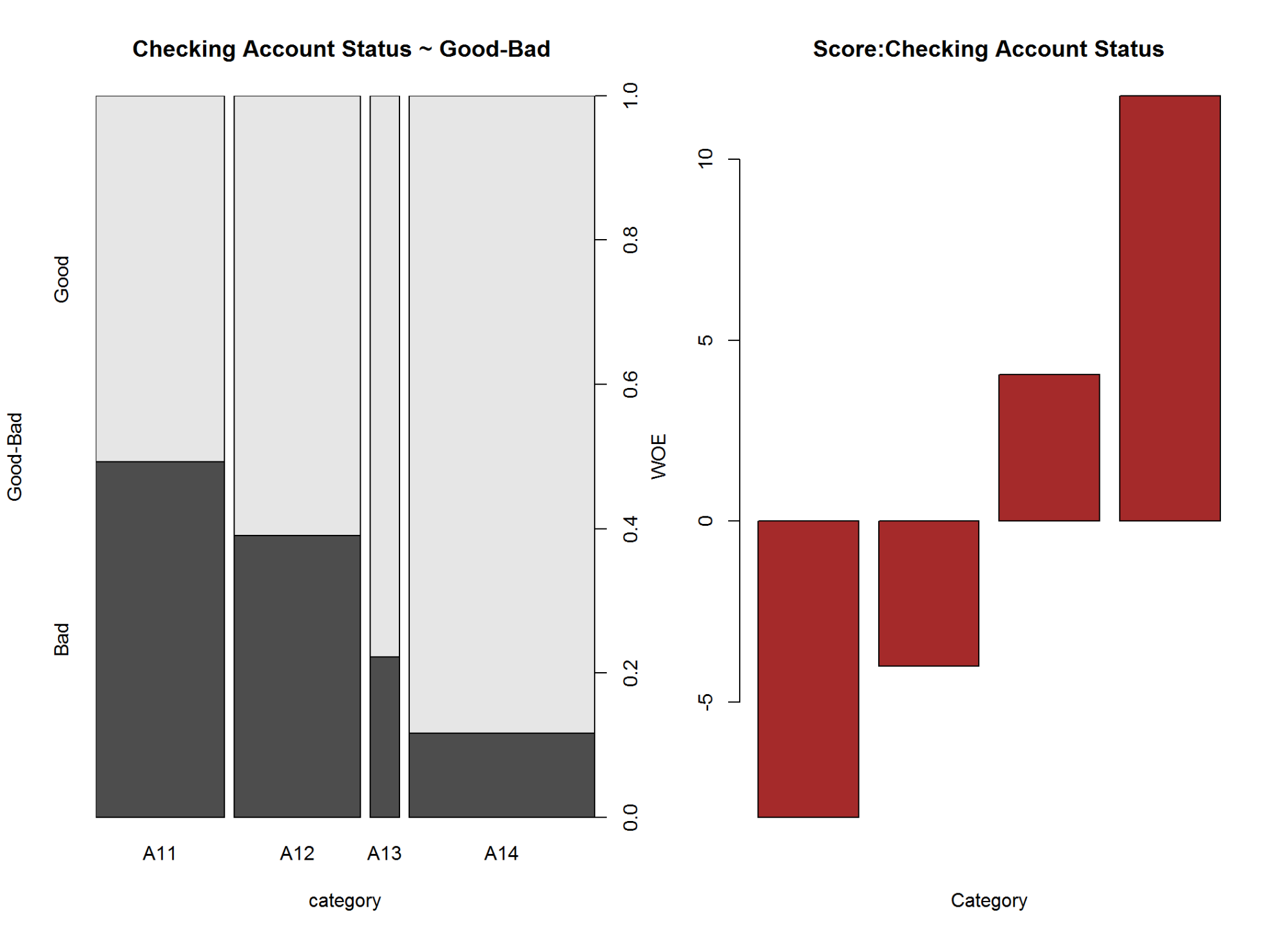


par(op)

2.2 Detail Analysis of variables and variable reduction:

2.2.1 Checking account status

# Attribute 1: (qualitative)  
#-----------------------------------------------------------  
# Checking account status  
  
# Status of existing checking account  
# A11 : ... < 0 DM  
# A12 : 0 <= ... < 200 DM  
# A13 : ... >= 200 DM /  
# salary assignments for at least 1 year  
# A14 : no checking account  
  
  
A1 <- gbpct(cdata$chk\_ac\_status\_1)  
  
op1<-par(mfrow=c(1,2), new=TRUE)  
  
plot(cdata$chk\_ac\_status\_1, cdata$good\_bad\_21,   
 ylab="Good-Bad", xlab="category",   
 main="Checking Account Status ~ Good-Bad ")  
  
barplot(A1$WOE, col="brown", names.arg=c(A1$Levels),   
 main="Score:Checking Account Status",  
 xlab="Category",  
 ylab="WOE"  
)



par(op1)  
  
kable(A1, caption = 'Checking Account Status ~ Good-Bad')

Checking Account Status ~ Good-Bad

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Names | Good | Bad | Good\_pct | Bad\_pct | Total | Total\_Pct | Bad\_Rate | grp\_score | WOE | IV | Efficiency |
| A11 | 139 | 135 | 19.86 | 45.00 | 274 | 27.4 | 49.27 | 3.06 | -8.18 | 20.56452 | 12.570 |
| A12 | 164 | 105 | 23.43 | 35.00 | 269 | 26.9 | 39.03 | 4.01 | -4.01 | 4.63957 | 5.785 |
| A13 | 49 | 14 | 7.00 | 4.67 | 63 | 6.3 | 22.22 | 6.00 | 4.05 | 0.94365 | 1.165 |
| A14 | 348 | 46 | 49.71 | 15.33 | 394 | 39.4 | 11.68 | 7.64 | 11.76 | 40.43088 | 17.190 |

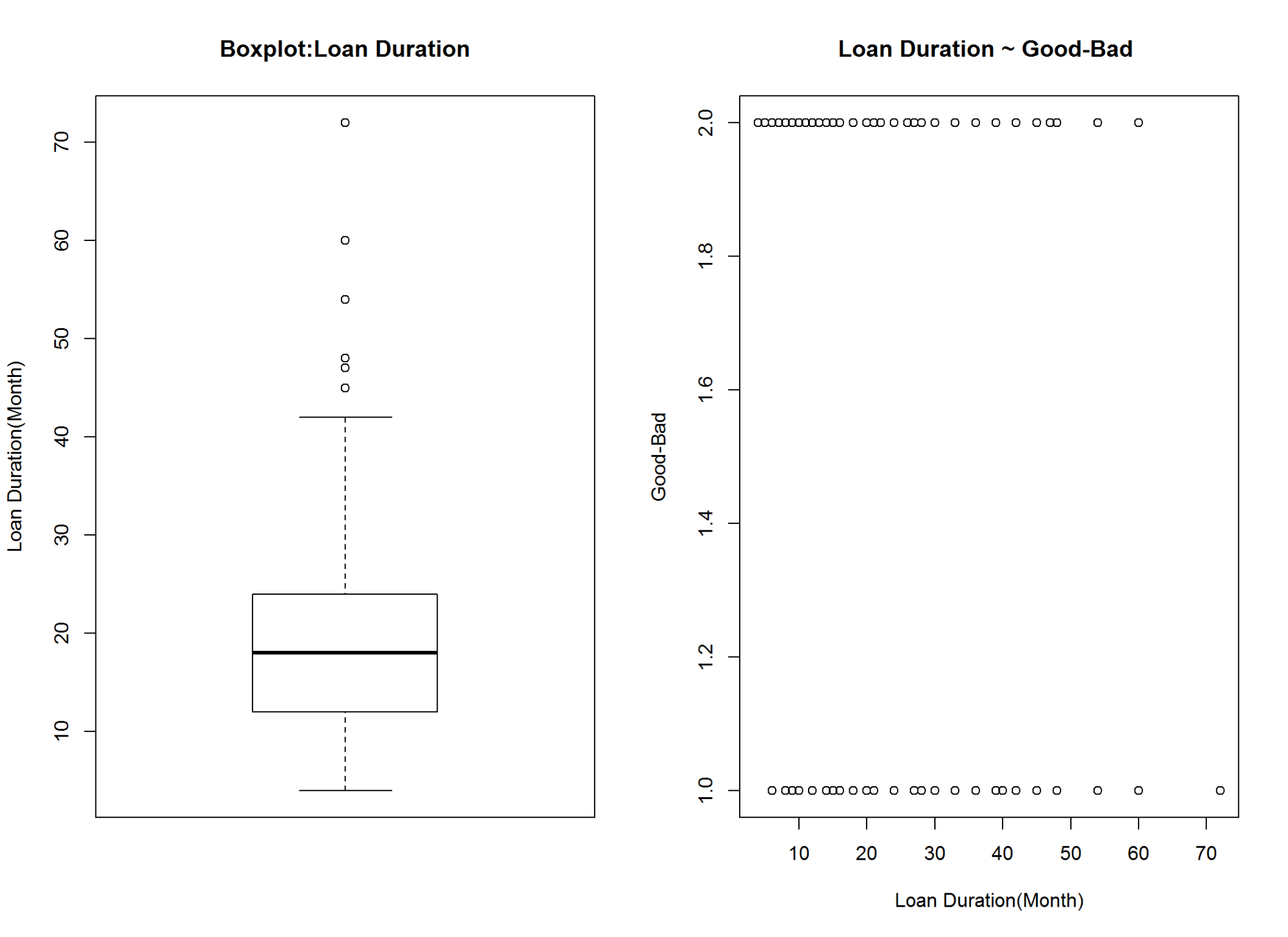
Information Value is  **66.58**  and Efficiency is  **36.71** .

2.2.2 Loan Duration

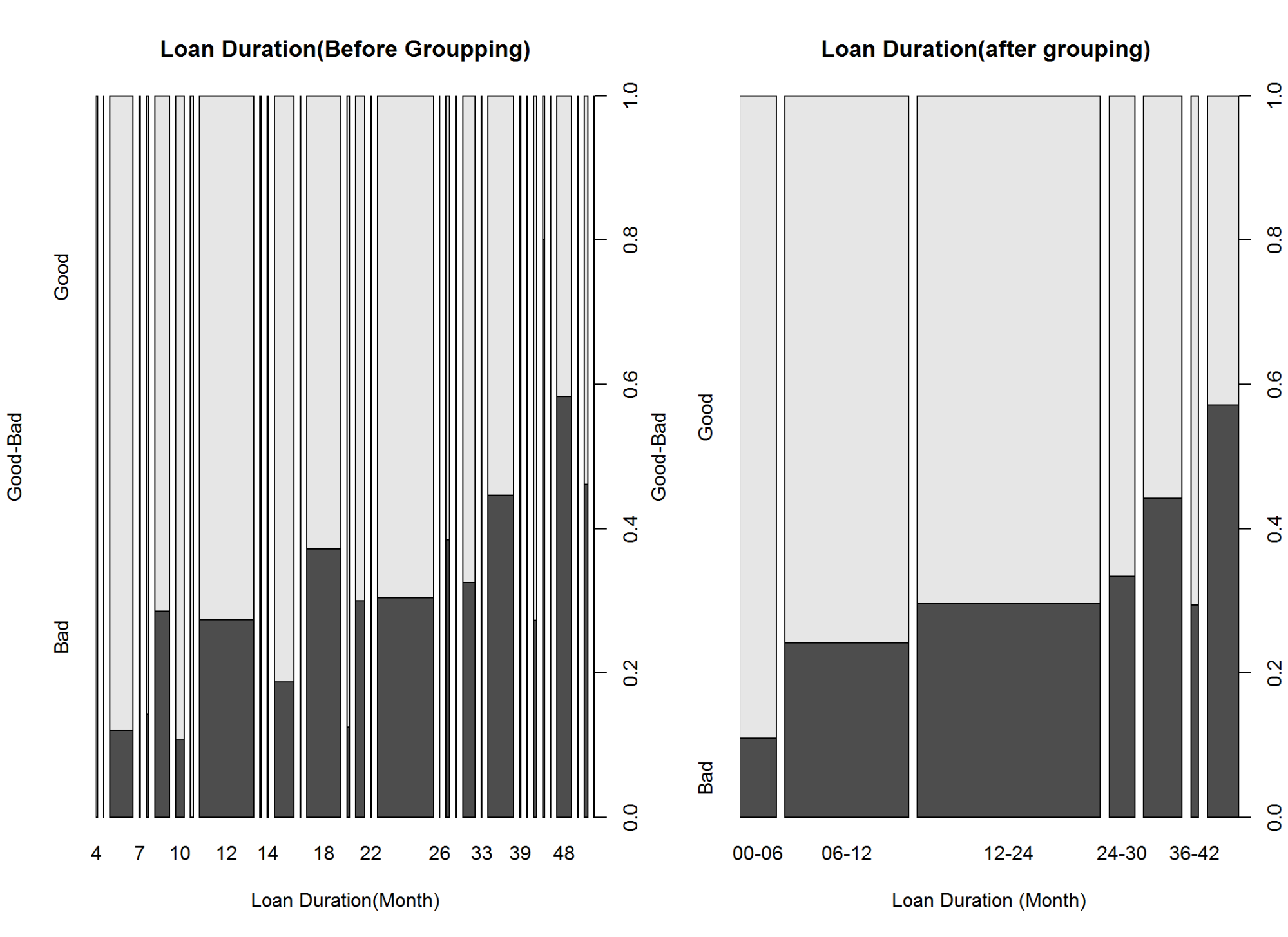
# Attribute 2: (numerical)  
#-----------------------------------------------------------  
# Loan Duration (Tenure) in Month  
  
summary(cdata$duration\_month\_2)

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

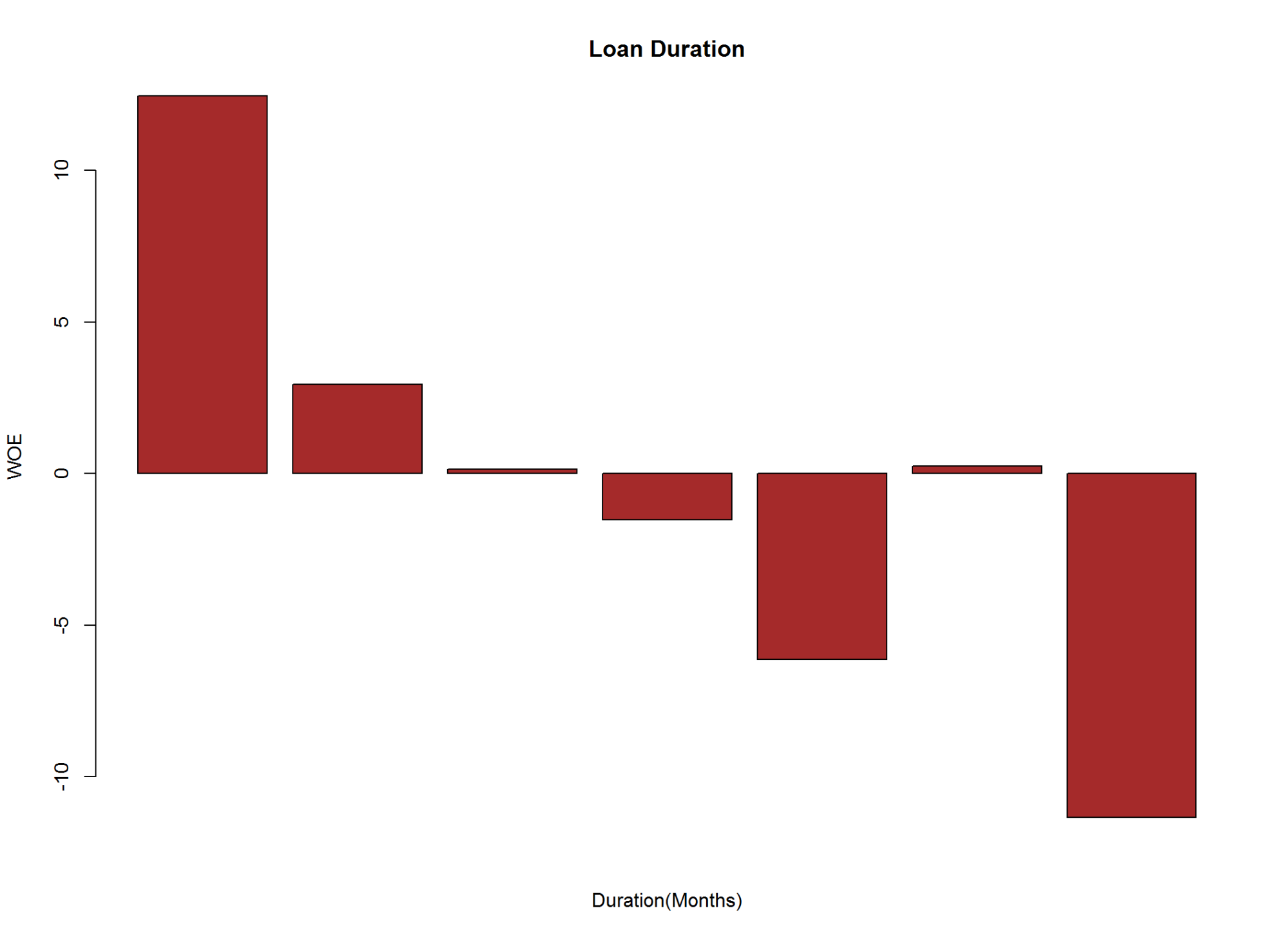
op2<-par(mfrow=c(1,2))  
boxplot(cdata$duration\_month\_2, ylab="Loan Duration(Month)", main="Boxplot:Loan Duration")  
  
plot(cdata$duration\_month\_2, cdata$good\_bad\_21,   
 ylab="Good-Bad", xlab="Loan Duration(Month)",  
 main="Loan Duration ~ Good-Bad ")



plot(as.factor(cdata$duration\_month\_2), cdata$good\_bad\_21,   
 ylab="Good-Bad", xlab="Loan Duration(Month)",  
 main="Loan Duration(Before Groupping)")  
  
  
cdata$duration\_month\_2 <-as.factor(ifelse(cdata$duration\_month\_2<=6,'00-06',  
 ifelse(cdata$duration\_month\_2<=12,'06-12',  
 ifelse(cdata$duration\_month\_2<=24,'12-24',   
 ifelse(cdata$duration\_month\_2<=30,'24-30',  
 ifelse(cdata$duration\_month\_2<=36,'30-36',  
 ifelse(cdata$duration\_month\_2<=42,'36-42','42+')))))))  
   
   
  
  
plot(cdata$duration\_month\_2, cdata$good\_bad\_21,  
 main="Loan Duration(after grouping) ",  
 xlab="Loan Duration (Month)",  
 ylab="Good-Bad")



par(op2)  
  
A2<-gbpct(cdata$duration\_month\_2)  
  
barplot(A2$WOE, col="brown", names.arg=c(A2$Levels),  
 main="Loan Duration",  
 xlab="Duration(Months)",  
 ylab="WOE"  
)



kable(A2, caption = 'Loan Duration ~ Good-Bad')

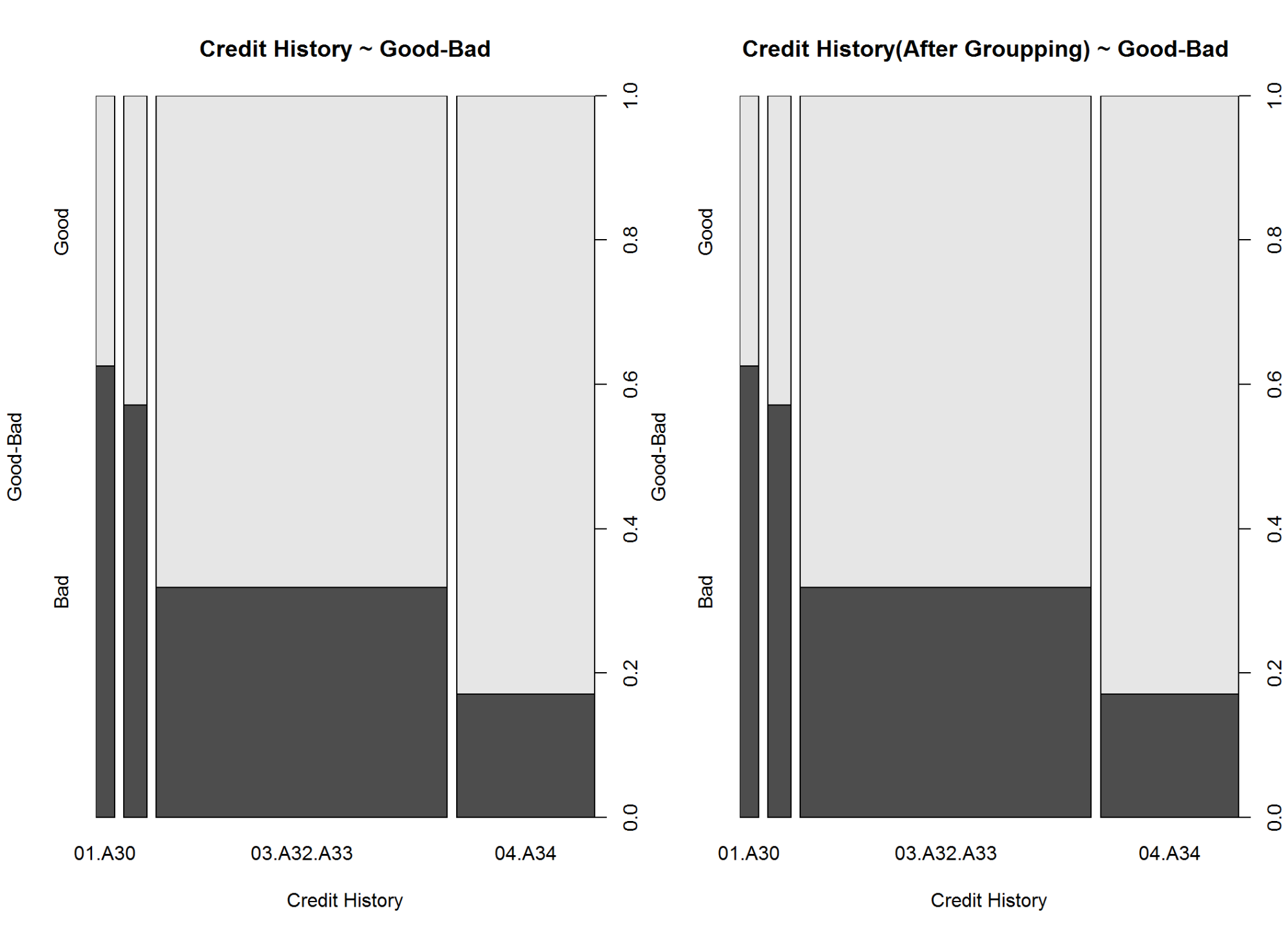
Loan Duration ~ Good-Bad

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Names | Good | Bad | Good\_pct | Bad\_pct | Total | Total\_Pct | Bad\_Rate | grp\_score | WOE | IV | Efficiency |
| 00-06 | 73 | 9 | 10.43 | 3.00 | 82 | 8.2 | 10.98 | 7.77 | 12.46 | 9.25778 | 3.715 |
| 06-12 | 210 | 67 | 30.00 | 22.33 | 277 | 27.7 | 24.19 | 5.73 | 2.95 | 2.26265 | 3.835 |
| 12-24 | 289 | 122 | 41.29 | 40.67 | 411 | 41.1 | 29.68 | 5.04 | 0.15 | 0.00930 | 0.310 |
| 24-30 | 38 | 19 | 5.43 | 6.33 | 57 | 5.7 | 33.33 | 4.62 | -1.53 | 0.13770 | 0.450 |
| 30-36 | 48 | 38 | 6.86 | 12.67 | 86 | 8.6 | 44.19 | 3.51 | -6.14 | 3.56734 | 2.905 |
| 36-42 | 12 | 5 | 1.71 | 1.67 | 17 | 1.7 | 29.41 | 5.06 | 0.24 | 0.00096 | 0.020 |
| 42+ | 30 | 40 | 4.29 | 13.33 | 70 | 7.0 | 57.14 | 2.43 | -11.34 | 10.25136 | 4.520 |

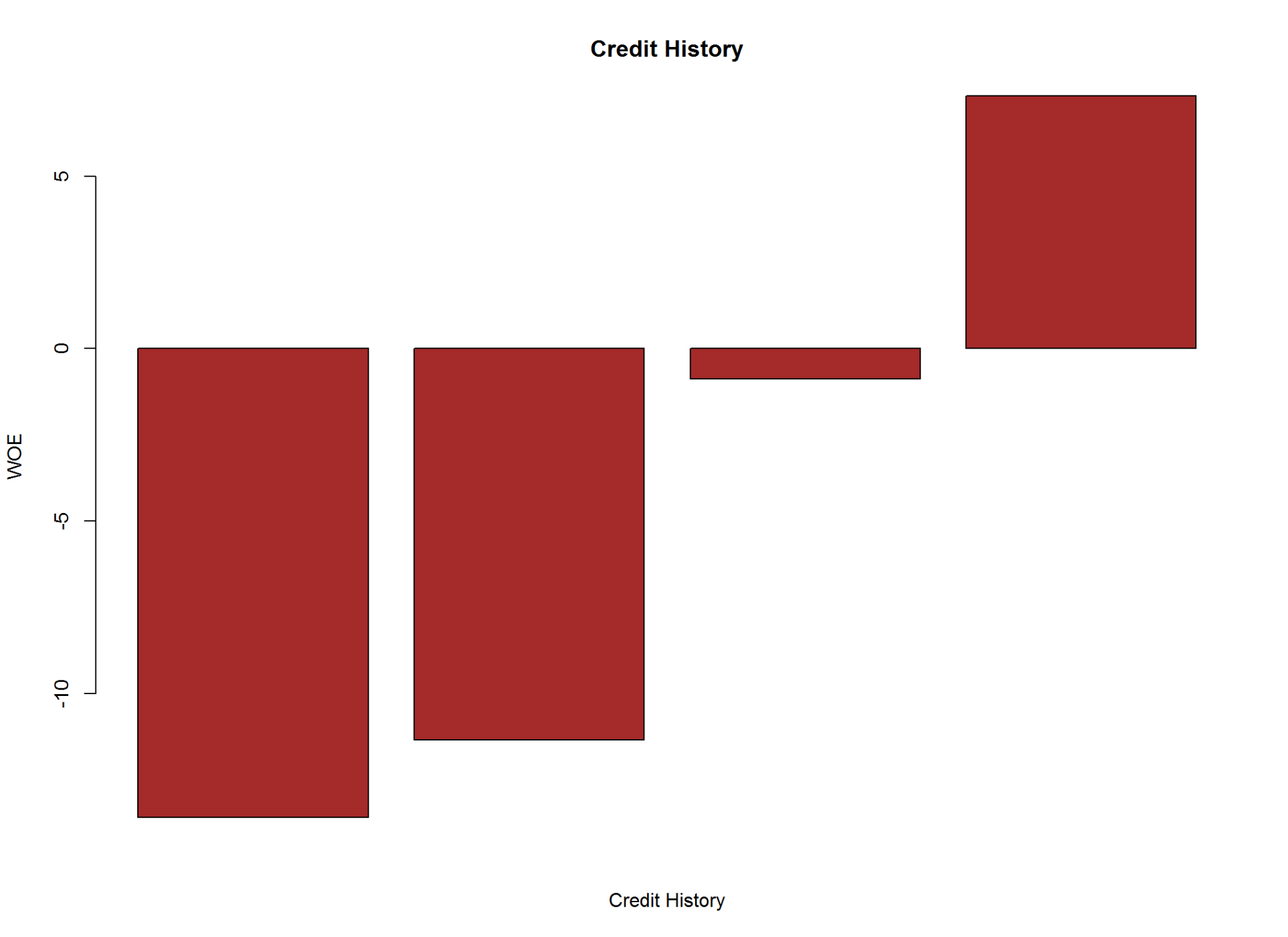
Information Value is  **25.49**  and Efficiency is  **15.75** .

2.2.3 Credit History

# Attribute 3: (qualitative)  
#-----------------------------------------------------------  
# Credit History  
  
# A30 : no credits taken/  
# all credits paid back duly  
# A31 : all credits at this bank paid back duly  
# A32 : existing credits paid back duly till now  
# A33 : delay in paying off in the past  
# A34 : critical account/  
# other credits existing (not at this bank)  
  
cdata$credit\_history\_3<-as.factor(ifelse(cdata$credit\_history\_3 == "A30", "01.A30",  
 ifelse(cdata$credit\_history\_3 == "A31","02.A31",  
 ifelse(cdata$credit\_history\_3 == "A32","03.A32.A33",  
 ifelse(cdata$credit\_history\_3 == "A33","03.A32.A33",  
 "04.A34")))))  
  
op3<-par(mfrow=c(1,2))  
  
plot(cdata$credit\_history\_3, cdata$good\_bad\_21,   
 main = "Credit History ~ Good-Bad",  
 xlab = "Credit History",  
 ylab = "Good-Bad")  
  
plot(cdata$credit\_history\_3, cdata$good\_bad\_21,   
 main = "Credit History(After Groupping) ~ Good-Bad ",  
 xlab = "Credit History",  
 ylab = "Good-Bad")



par(op3)  
  
A3<-gbpct(cdata$credit\_history\_3)  
  
barplot(A3$WOE, col="brown", names.arg=c(A3$Levels),  
 main="Credit History",  
 xlab="Credit History",  
 ylab="WOE"  
)



kable(A3, caption = "Credit History~ Good-Bad")

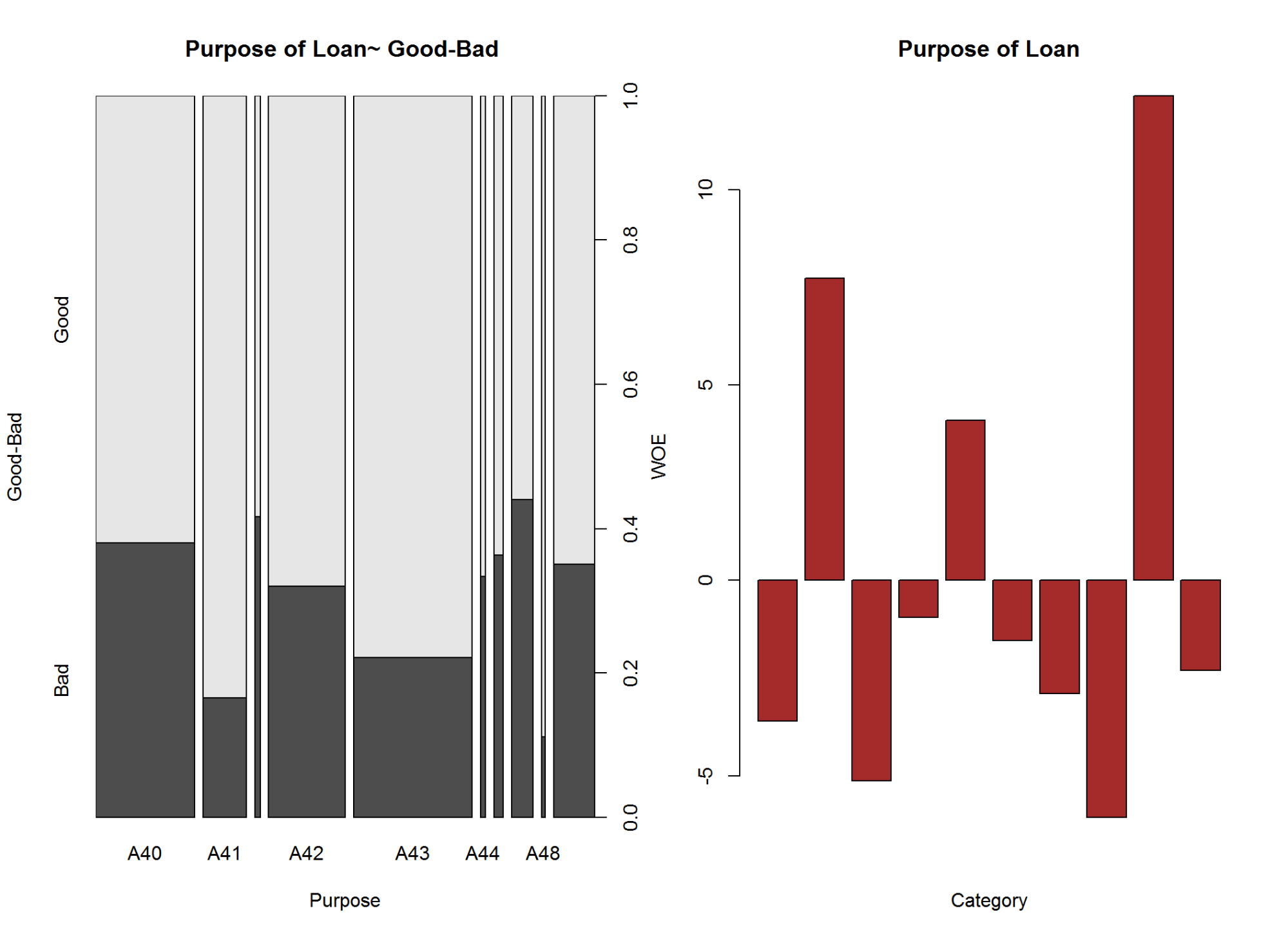
Credit History~ Good-Bad

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Names | Good | Bad | Good\_pct | Bad\_pct | Total | Total\_Pct | Bad\_Rate | grp\_score | WOE | IV | Efficiency |
| 01.A30 | 15 | 25 | 2.14 | 8.33 | 40 | 4.0 | 62.50 | 2.04 | -13.59 | 8.41221 | 3.095 |
| 02.A31 | 21 | 28 | 3.00 | 9.33 | 49 | 4.9 | 57.14 | 2.43 | -11.35 | 7.18455 | 3.165 |
| 03.A32.A33 | 421 | 197 | 60.14 | 65.67 | 618 | 61.8 | 31.88 | 4.78 | -0.88 | 0.48664 | 2.765 |
| 04.A34 | 243 | 50 | 34.71 | 16.67 | 293 | 29.3 | 17.06 | 6.76 | 7.33 | 13.22332 | 9.020 |

Information Value is  **29.31**  and Efficiency is  **18.05** .

2.2.4 Purpose of the loan

# Attribute 4: (qualitative)  
#-----------------------------------------------------------  
# Purpose of the loan  
  
#   
# A40 : car (new)  
# A41 : car (used)  
# A42 : furniture/equipment  
# A43 : radio/television  
# A44 : domestic appliances  
# A45 : repairs  
# A46 : education  
# A47 : (vacation - does not exist?)  
# A48 : retraining  
# A49 : business  
# A410 : others  
  
  
A4<-gbpct(cdata$purpose\_4)  
  
  
op4<-par(mfrow=c(1,2))  
plot(cdata$purpose\_4, cdata$good\_bad\_21,   
 main="Purpose of Loan~ Good-Bad ",  
 xlab="Purpose",  
 ylab="Good-Bad")  
  
barplot(A4$WOE, col="brown", names.arg=c(A4$Levels),  
 main="Purpose of Loan",  
 xlab="Category",  
 ylab="WOE")



par(op4)  
  
kable(A4, caption = "Purpose of Loan~ Good-Bad")

Purpose of Loan~ Good-Bad

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Names | Good | Bad | Good\_pct | Bad\_pct | Total | Total\_Pct | Bad\_Rate | grp\_score | WOE | IV | Efficiency |
| A40 | 145 | 89 | 20.71 | 29.67 | 234 | 23.4 | 38.03 | 4.11 | -3.60 | 3.22560 | 4.480 |
| A41 | 86 | 17 | 12.29 | 5.67 | 103 | 10.3 | 16.50 | 6.84 | 7.74 | 5.12388 | 3.310 |
| A410 | 7 | 5 | 1.00 | 1.67 | 12 | 1.2 | 41.67 | 3.75 | -5.13 | 0.34371 | 0.335 |
| A42 | 123 | 58 | 17.57 | 19.33 | 181 | 18.1 | 32.04 | 4.76 | -0.95 | 0.16720 | 0.880 |
| A43 | 218 | 62 | 31.14 | 20.67 | 280 | 28.0 | 22.14 | 6.01 | 4.10 | 4.29270 | 5.235 |
| A44 | 8 | 4 | 1.14 | 1.33 | 12 | 1.2 | 33.33 | 4.62 | -1.54 | 0.02926 | 0.095 |
| A45 | 14 | 8 | 2.00 | 2.67 | 22 | 2.2 | 36.36 | 4.28 | -2.89 | 0.19363 | 0.335 |
| A46 | 28 | 22 | 4.00 | 7.33 | 50 | 5.0 | 44.00 | 3.53 | -6.06 | 2.01798 | 1.665 |
| A48 | 8 | 1 | 1.14 | 0.33 | 9 | 0.9 | 11.11 | 7.76 | 12.40 | 1.00440 | 0.405 |
| A49 | 63 | 34 | 9.00 | 11.33 | 97 | 9.7 | 35.05 | 4.43 | -2.30 | 0.53590 | 1.165 |

Information Value is  **16.93**  and Efficiency is  **17.9** .

2.2.5 Credit Amount

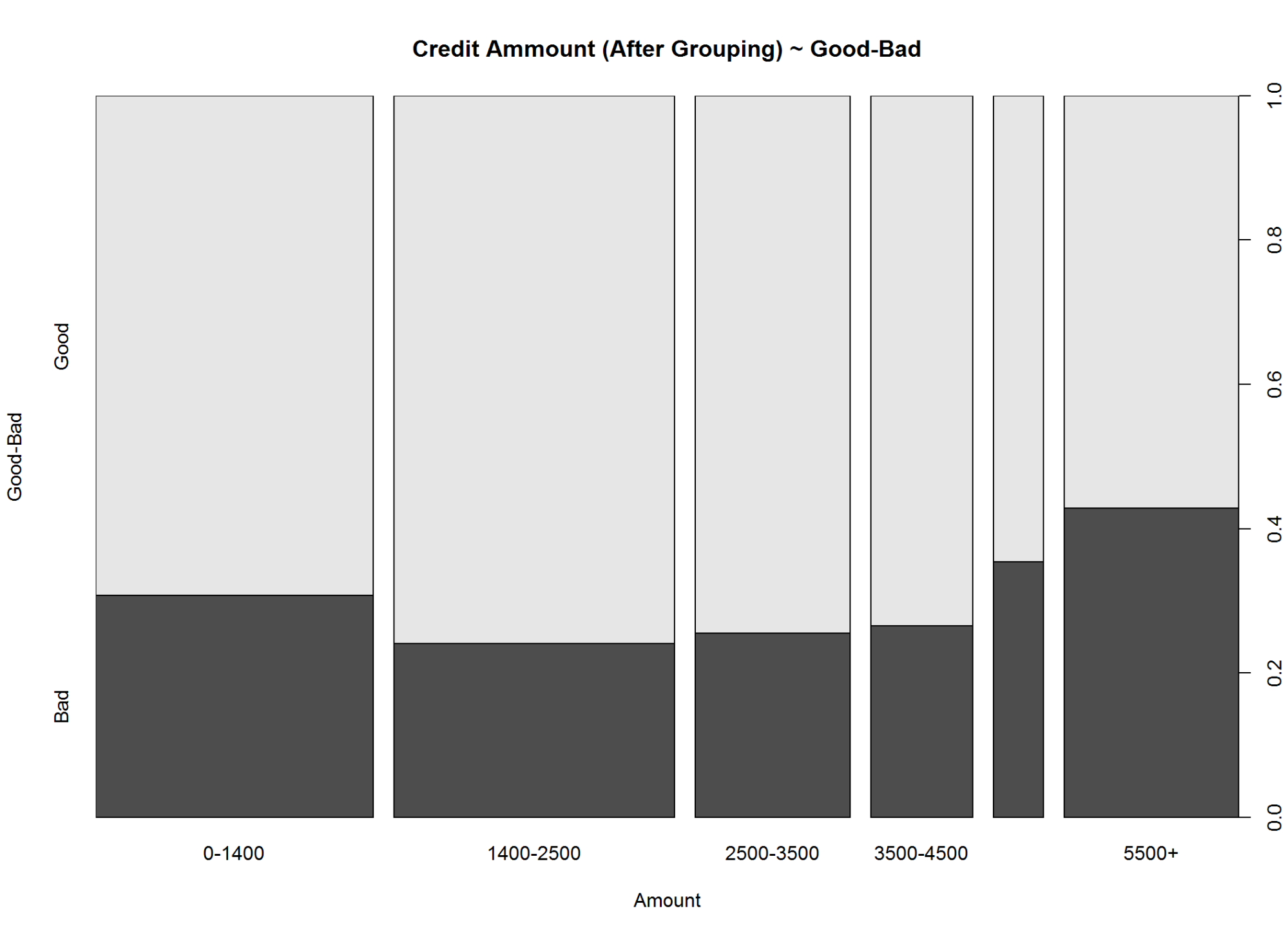
# Attribute 5: (numerical)  
#-----------------------------------------------------------  
# Credit (Loan) Amount  
  
cdata$credit\_amount\_5 <- as.double(cdata$credit\_amount\_5)  
summary(cdata$credit\_amount\_5)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 250 1366 2320 3271 3972 18420

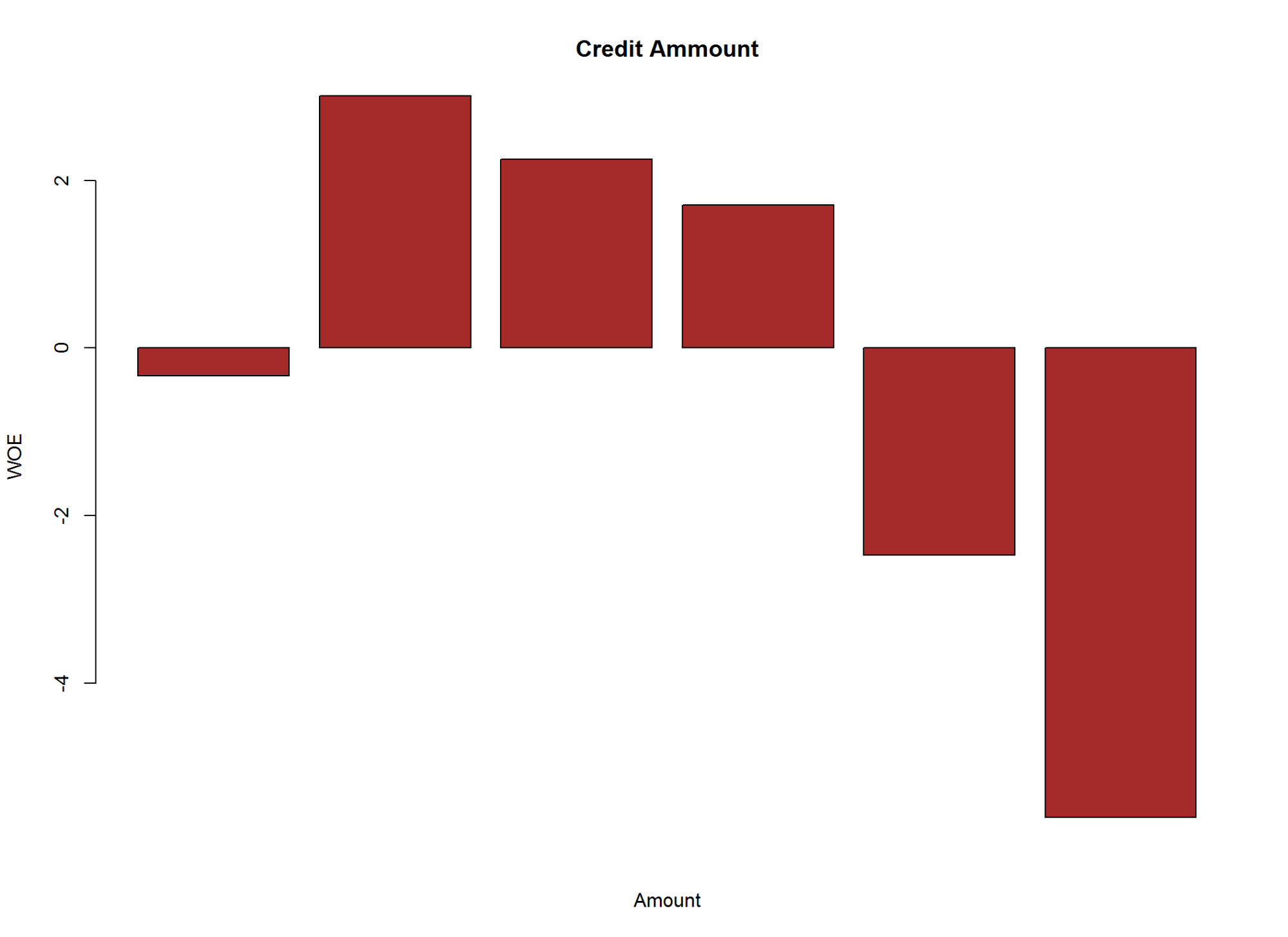
boxplot(cdata$credit\_amount\_5)



cdata$credit\_amount\_5<-as.factor(ifelse(cdata$credit\_amount\_5<=1400,'0-1400',  
 ifelse(cdata$credit\_amount\_5<=2500,'1400-2500',  
 ifelse(cdata$credit\_amount\_5<=3500,'2500-3500',   
 ifelse(cdata$credit\_amount\_5<=4500,'3500-4500',  
 ifelse(cdata$credit\_amount\_5<=5500,'4500-5500','5500+'))))))  
  
  
A5<-gbpct(cdata$credit\_amount\_5)  
  
  
  
plot(cdata$credit\_amount\_5, cdata$good\_bad\_21,   
 main="Credit Ammount (After Grouping) ~ Good-Bad",  
 xlab="Amount",  
 ylab="Good-Bad")



barplot(A5$WOE, col="brown", names.arg=c(A5$Levels),  
 main="Credit Ammount",  
 xlab="Amount",  
 ylab="WOE")



kable(A5, caption = "Credit Ammount ~ Good-Bad")

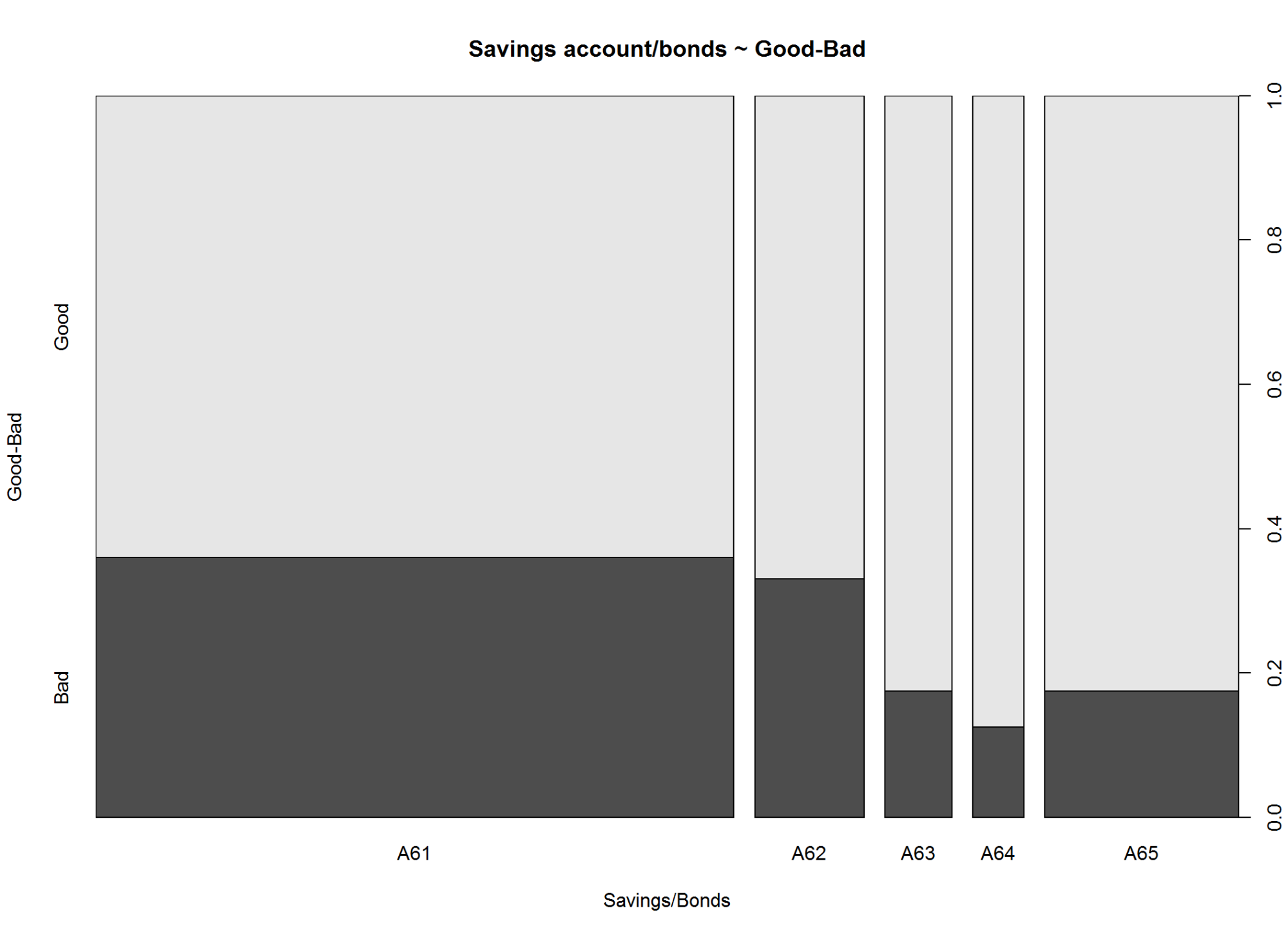
Credit Ammount ~ Good-Bad

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Names | Good | Bad | Good\_pct | Bad\_pct | Total | Total\_Pct | Bad\_Rate | grp\_score | WOE | IV | Efficiency |
| 0-1400 | 185 | 82 | 26.43 | 27.33 | 267 | 26.7 | 30.71 | 4.92 | -0.33 | 0.02970 | 0.450 |
| 1400-2500 | 205 | 65 | 29.29 | 21.67 | 270 | 27.0 | 24.07 | 5.75 | 3.01 | 2.29362 | 3.810 |
| 2500-3500 | 111 | 38 | 15.86 | 12.67 | 149 | 14.9 | 25.50 | 5.56 | 2.25 | 0.71775 | 1.595 |
| 3500-4500 | 72 | 26 | 10.29 | 8.67 | 98 | 9.8 | 26.53 | 5.43 | 1.71 | 0.27702 | 0.810 |
| 4500-5500 | 31 | 17 | 4.43 | 5.67 | 48 | 4.8 | 35.42 | 4.39 | -2.47 | 0.30628 | 0.620 |
| 5500+ | 96 | 72 | 13.71 | 24.00 | 168 | 16.8 | 42.86 | 3.64 | -5.60 | 5.76240 | 5.145 |

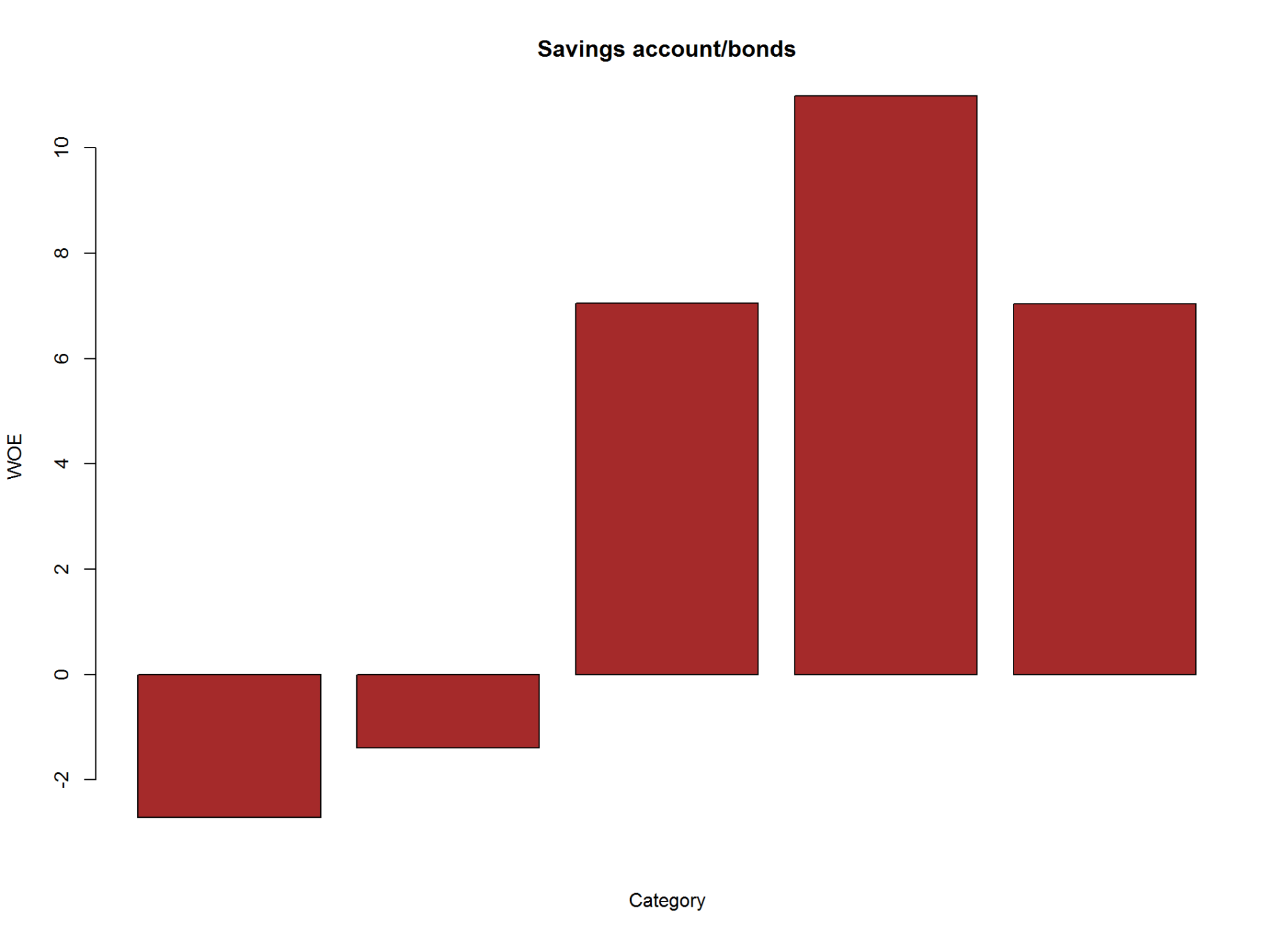
Information Value is  **9.39**  and Efficiency is  **12.43** .

2.2.6 Savings account/bonds

# Attibute 6: (qualitative)  
#-----------------------------------------------------------  
# Savings account/bonds  
  
# A61 : ... < 100 DM  
# A62 : 100 <= ... < 500 DM  
# A63 : 500 <= ... < 1000 DM  
# A64 : .. >= 1000 DM  
# A65 : unknown/ no savings account  
  
A6<-gbpct(cdata$savings\_ac\_bond\_6)  
  
  
plot(cdata$savings\_ac\_bond\_6, cdata$good\_bad\_21,   
 main="Savings account/bonds ~ Good-Bad",  
 xlab="Savings/Bonds",  
 ylab="Good-Bad")



barplot(A6$WOE, col="brown", names.arg=c(A6$Levels),  
 main="Savings account/bonds",  
 xlab="Category",  
 ylab="WOE")



kable(A6, caption = "Savings account/bonds ~ Good-Bad" )

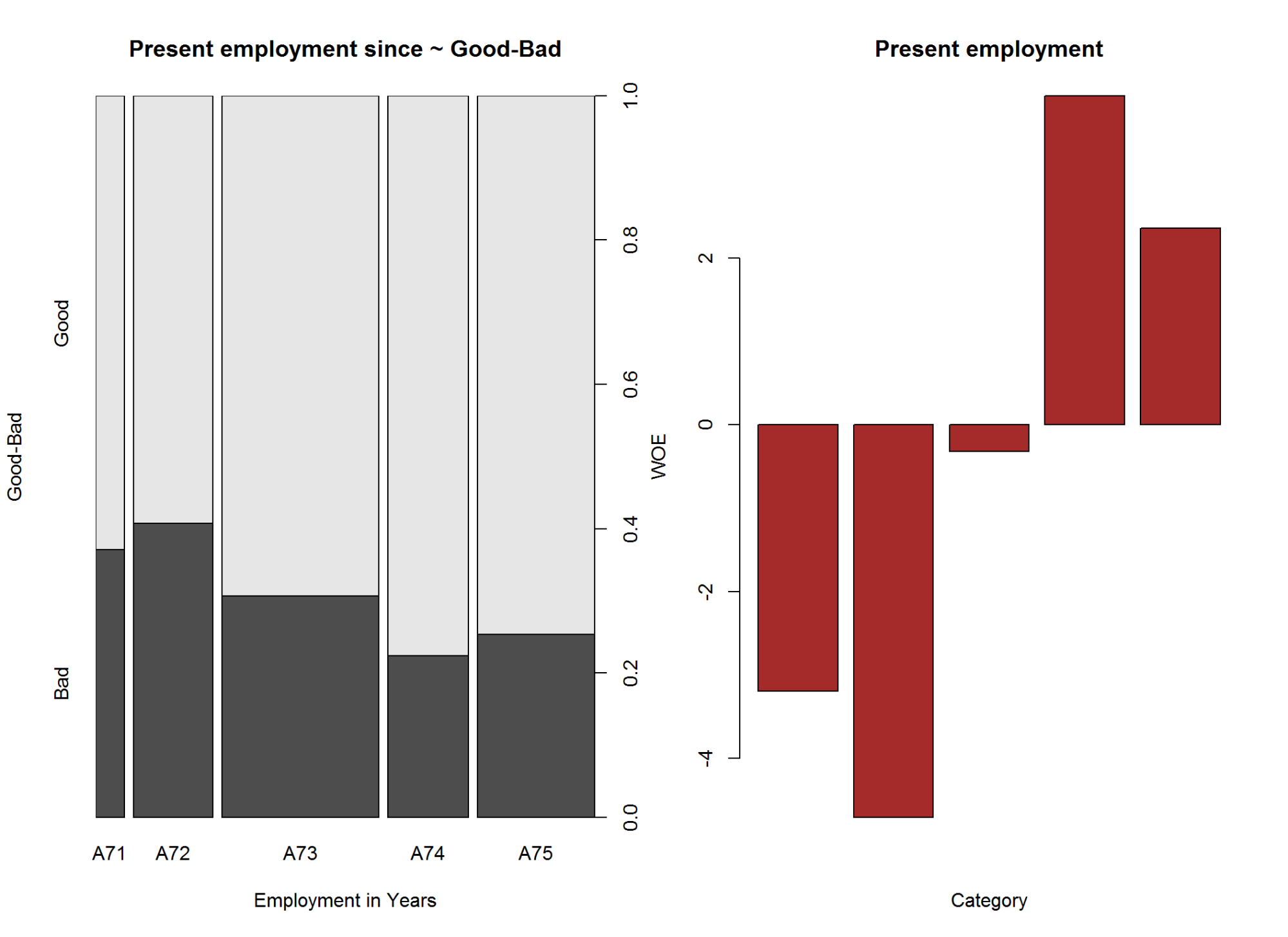
Savings account/bonds ~ Good-Bad

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Names | Good | Bad | Good\_pct | Bad\_pct | Total | Total\_Pct | Bad\_Rate | grp\_score | WOE | IV | Efficiency |
| A61 | 386 | 217 | 55.14 | 72.33 | 603 | 60.3 | 35.99 | 4.33 | -2.71 | 4.65849 | 8.595 |
| A62 | 69 | 34 | 9.86 | 11.33 | 103 | 10.3 | 33.01 | 4.65 | -1.39 | 0.20433 | 0.735 |
| A63 | 52 | 11 | 7.43 | 3.67 | 63 | 6.3 | 17.46 | 6.69 | 7.05 | 2.65080 | 1.880 |
| A64 | 42 | 6 | 6.00 | 2.00 | 48 | 4.8 | 12.50 | 7.50 | 10.99 | 4.39600 | 2.000 |
| A65 | 151 | 32 | 21.57 | 10.67 | 183 | 18.3 | 17.49 | 6.69 | 7.04 | 7.67360 | 5.450 |

Information Value is  **19.58**  and Efficiency is  **18.66** .

2.2.7 Present employment since

# Attribute 7: (qualitative)  
#-----------------------------------------------------------  
# Present employment since  
  
# A71 : unemployed  
# A72 : ... < 1 year  
# A73 : 1 <= ... < 4 years  
# A74 : 4 <= ... < 7 years  
# A75 : .. >= 7 years  
  
A7<-gbpct(cdata$p\_employment\_since\_7)  
  
op7<-par(mfrow=c(1,2))  
plot(cdata$p\_employment\_since\_7, cdata$good\_bad\_21,  
 main="Present employment since ~ Good-Bad",  
 xlab="Employment in Years",  
 ylab="Good-Bad")  
  
barplot(A7$WOE, col="brown", names.arg=c(A7$Levels),  
 main="Present employment",  
 xlab="Category",  
 ylab="WOE")



par(op7)  
  
kable(A7, caption ="Present employment since ~ Good-Bad")

Present employment since ~ Good-Bad

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Names | Good | Bad | Good\_pct | Bad\_pct | Total | Total\_Pct | Bad\_Rate | grp\_score | WOE | IV | Efficiency |
| A71 | 39 | 23 | 5.57 | 7.67 | 62 | 6.2 | 37.10 | 4.21 | -3.20 | 0.67200 | 1.050 |
| A72 | 102 | 70 | 14.57 | 23.33 | 172 | 17.2 | 40.70 | 3.84 | -4.71 | 4.12596 | 4.380 |
| A73 | 235 | 104 | 33.57 | 34.67 | 339 | 33.9 | 30.68 | 4.92 | -0.32 | 0.03520 | 0.550 |
| A74 | 135 | 39 | 19.29 | 13.00 | 174 | 17.4 | 22.41 | 5.97 | 3.95 | 2.48455 | 3.145 |
| A75 | 189 | 64 | 27.00 | 21.33 | 253 | 25.3 | 25.30 | 5.59 | 2.36 | 1.33812 | 2.835 |

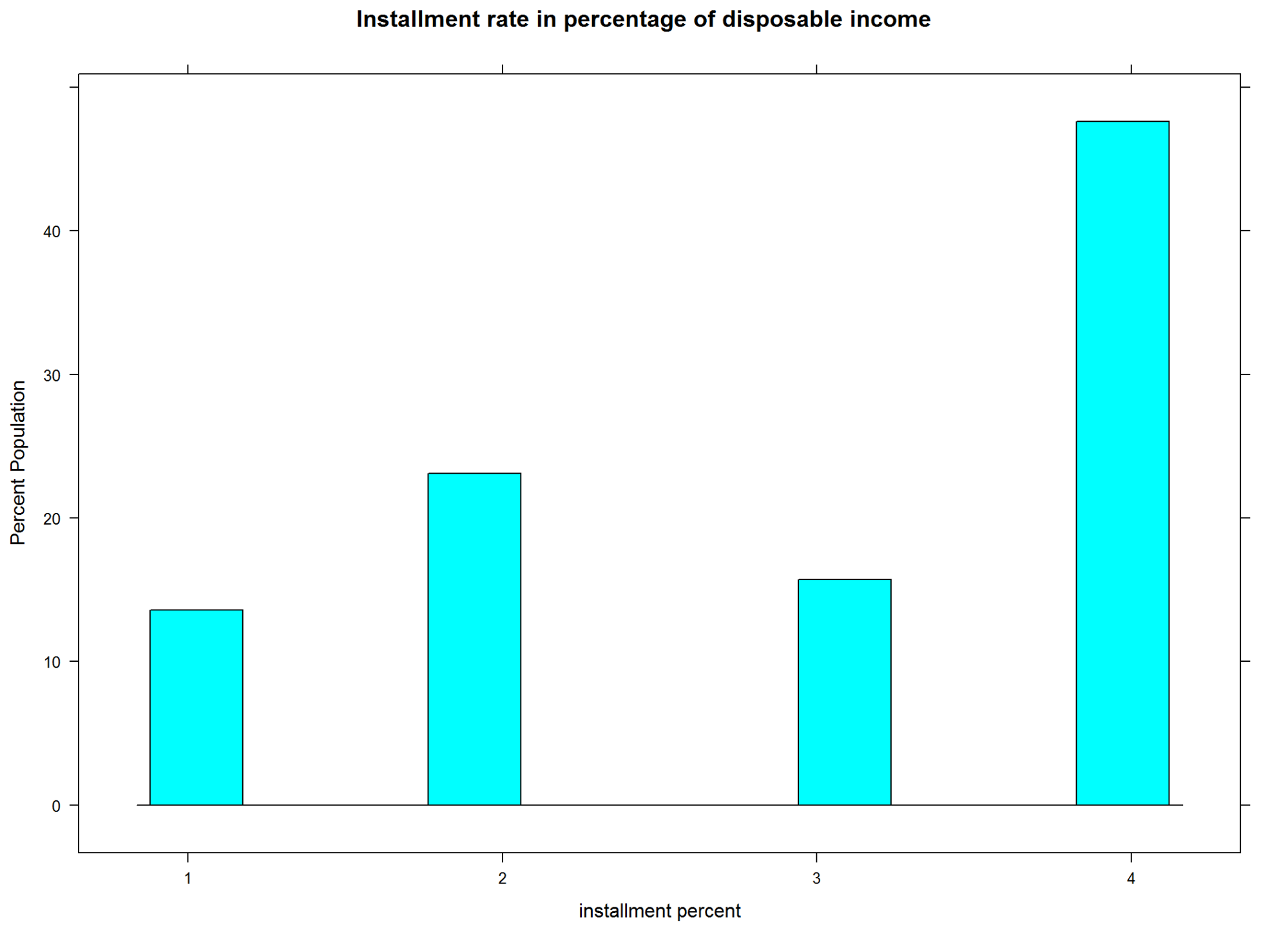
Information Value is  **8.66**  and Efficiency is  **11.96** .

2.2.8 Installment rate in percentage of disposable income

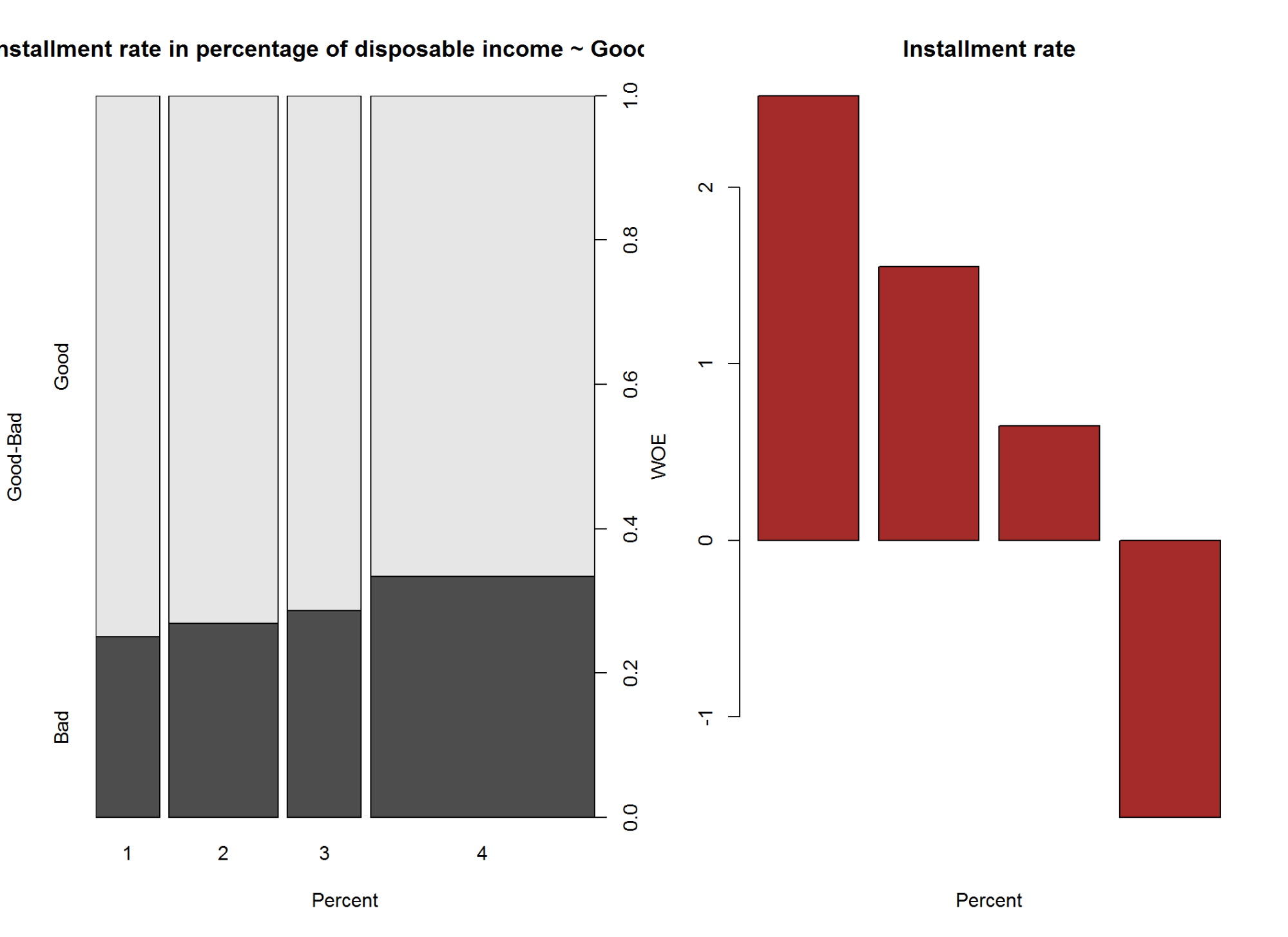
# Attribute 8: (numerical)  
#-----------------------------------------------------------  
# Installment rate in percentage of disposable income  
  
summary(cdata$installment\_pct\_disp\_inc\_8)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 2.000 3.000 2.973 4.000 4.000

op8<-par(mfrow=c(1,2))  
boxplot(cdata$installment\_pct\_disp\_inc\_8)  
histogram(cdata$installment\_pct\_disp\_inc\_8,  
 main = "Installment rate in percentage of disposable income",   
 xlab = "installment percent",  
 ylab = "Percent Population")  
par(op8)



A8<-gbpct(cdata$installment\_pct\_disp\_inc\_8)  
  
op8\_1<-par(mfrow=c(1,2))  
plot(as.factor(cdata$installment\_pct\_disp\_inc\_8), cdata$good\_bad\_21,   
 main="Installment rate in percentage of disposable income ~ Good-Bad",  
 xlab="Percent",  
 ylab="Good-Bad")  
  
barplot(A8$WOE, col="brown", names.arg=c(A8$Levels),  
 main="Installment rate",  
 xlab="Percent",  
 ylab="WOE")



par(op8\_1)  
  
kable(A8, caption = "Installment rate in percentage of disposable income ~ Good-Bad")

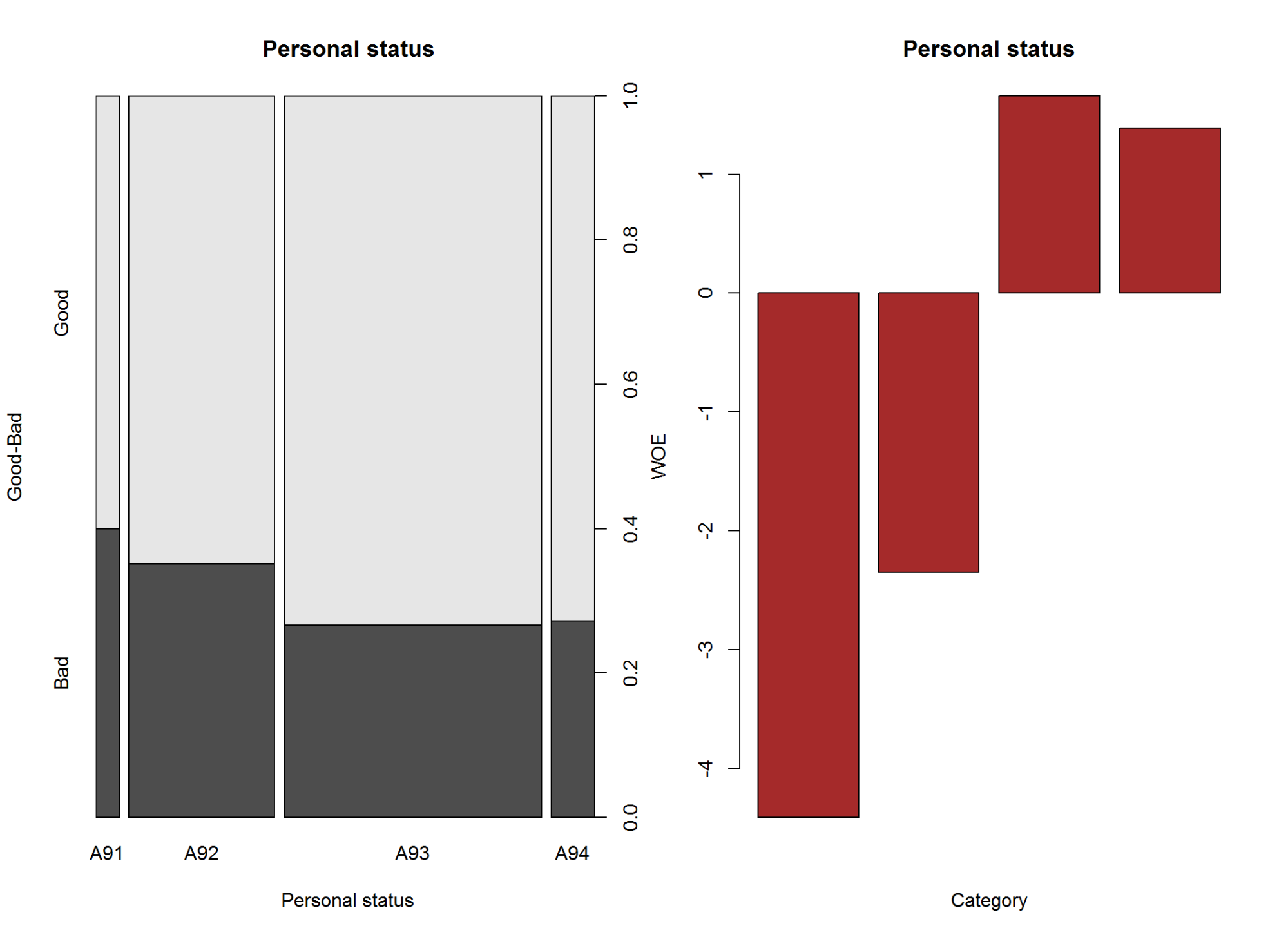
Installment rate in percentage of disposable income ~ Good-Bad

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Names | Good | Bad | Good\_pct | Bad\_pct | Total | Total\_Pct | Bad\_Rate | grp\_score | WOE | IV | Efficiency |
| 1 | 102 | 34 | 14.57 | 11.33 | 136 | 13.6 | 25.00 | 5.63 | 2.52 | 0.81648 | 1.620 |
| 2 | 169 | 62 | 24.14 | 20.67 | 231 | 23.1 | 26.84 | 5.39 | 1.55 | 0.53785 | 1.735 |
| 3 | 112 | 45 | 16.00 | 15.00 | 157 | 15.7 | 28.66 | 5.16 | 0.65 | 0.06500 | 0.500 |
| 4 | 317 | 159 | 45.29 | 53.00 | 476 | 47.6 | 33.40 | 4.61 | -1.57 | 1.21047 | 3.855 |

Information Value is  **2.63**  and Efficiency is  **7.71** .

2.2.9 Personal status and sex

# Attribute 9: (qualitative)  
#-----------------------------------------------------------  
# Personal status and sex - you may not use for some country due to regulations  
  
# A91 : male : divorced/separated  
# A92 : female : divorced/separated/married  
# A93 : male : single  
# A94 : male : married/widowed  
# A95 : female : single  
  
A9<-gbpct(cdata$personal\_status\_9)  
  
op9<-par(mfrow=c(1,2))  
plot(cdata$personal\_status\_9, cdata$good\_bad\_21,   
 main=" Personal status",  
 xlab=" Personal status",  
 ylab="Good-Bad")  
  
  
barplot(A9$WOE, col="brown", names.arg=c(A9$Levels),  
 main="Personal status",  
 xlab="Category",  
 ylab="WOE")



par(op9)  
  
kable(A9, caption = "Personal status ~ Good-Bad")

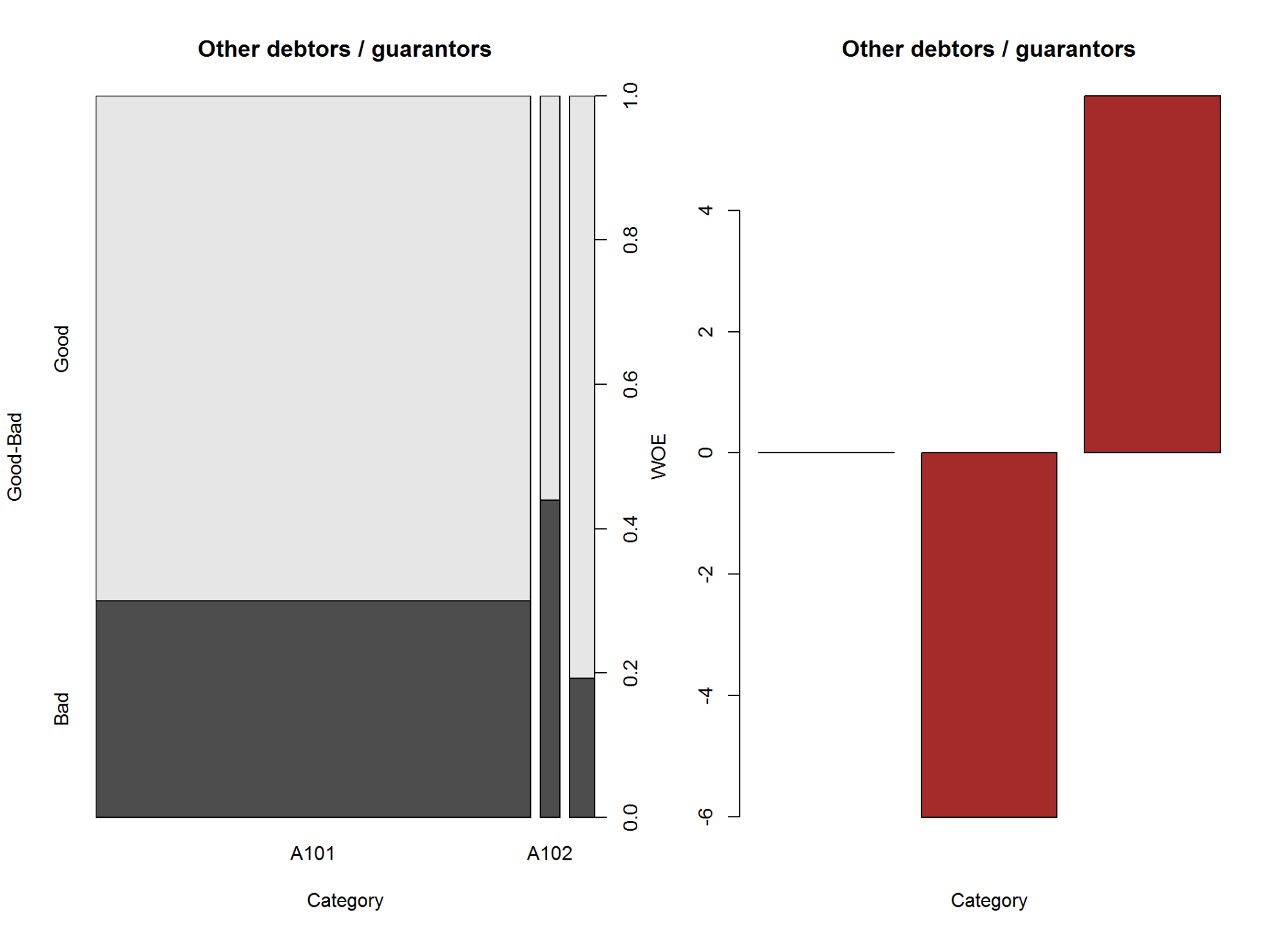
Personal status ~ Good-Bad

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Names | Good | Bad | Good\_pct | Bad\_pct | Total | Total\_Pct | Bad\_Rate | grp\_score | WOE | IV | Efficiency |
| A91 | 30 | 20 | 4.29 | 6.67 | 50 | 5.0 | 40.00 | 3.91 | -4.41 | 1.04958 | 1.19 |
| A92 | 201 | 109 | 28.71 | 36.33 | 310 | 31.0 | 35.16 | 4.41 | -2.35 | 1.79070 | 3.81 |
| A93 | 402 | 146 | 57.43 | 48.67 | 548 | 54.8 | 26.64 | 5.41 | 1.66 | 1.45416 | 4.38 |
| A94 | 67 | 25 | 9.57 | 8.33 | 92 | 9.2 | 27.17 | 5.35 | 1.39 | 0.17236 | 0.62 |

Information Value is  **4.47**  and Efficiency is  **10** .

2.2.10 Other debtors / guarantors

# Attribute 10: (qualitative)   
#-----------------------------------------------------------  
# Other debtors / guarantors  
  
# A101 : none  
# A102 : co-applicant  
# A103 : guarantor  
  
A10<-gbpct(cdata$other\_debtors\_or\_grantors\_10)  
  
op10<-par(mfrow=c(1,2))  
  
plot(cdata$other\_debtors\_or\_grantors\_10, cdata$good\_bad\_21,   
 main="Other debtors / guarantors",  
 xlab="Category",  
 ylab="Good-Bad")  
  
barplot(A10$WOE, col="brown", names.arg=c(A10$Levels),  
 main="Other debtors / guarantors",  
 xlab="Category",  
 ylab="WOE")



par(op10)  
  
kable(A10, caption = "Other debtors / guarantors ~ Good-Bad")

Other debtors / guarantors ~ Good-Bad

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Names | Good | Bad | Good\_pct | Bad\_pct | Total | Total\_Pct | Bad\_Rate | grp\_score | WOE | IV | Efficiency |
| A101 | 635 | 272 | 90.71 | 90.67 | 907 | 90.7 | 29.99 | 5.00 | 0.00 | 0.00000 | 0.020 |
| A102 | 23 | 18 | 3.29 | 6.00 | 41 | 4.1 | 43.90 | 3.54 | -6.01 | 1.62871 | 1.355 |
| A103 | 42 | 10 | 6.00 | 3.33 | 52 | 5.2 | 19.23 | 6.43 | 5.89 | 1.57263 | 1.335 |

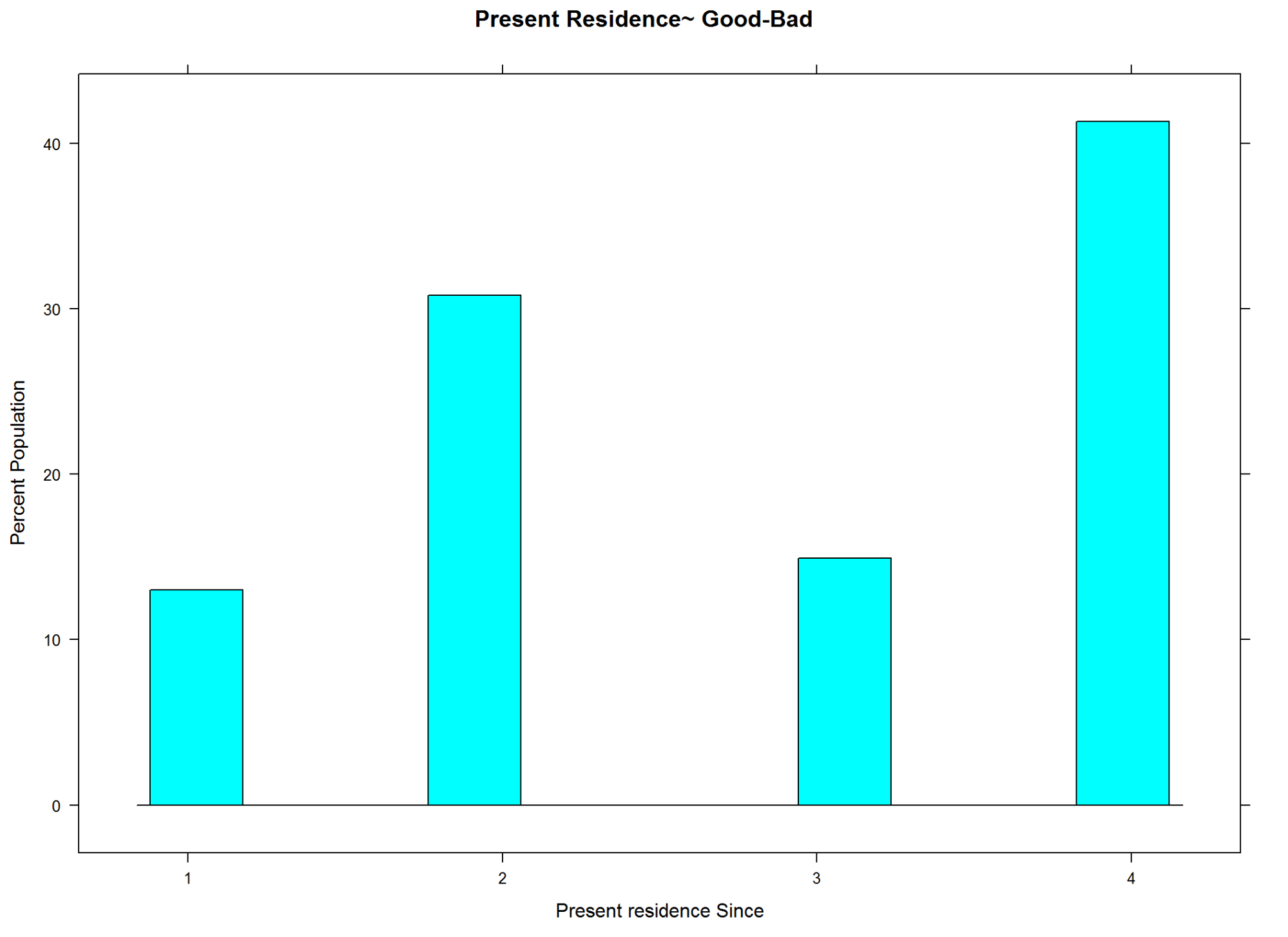
Information Value is  **3.2**  and Efficiency is  **2.71** .

2.2.11 Present residence since

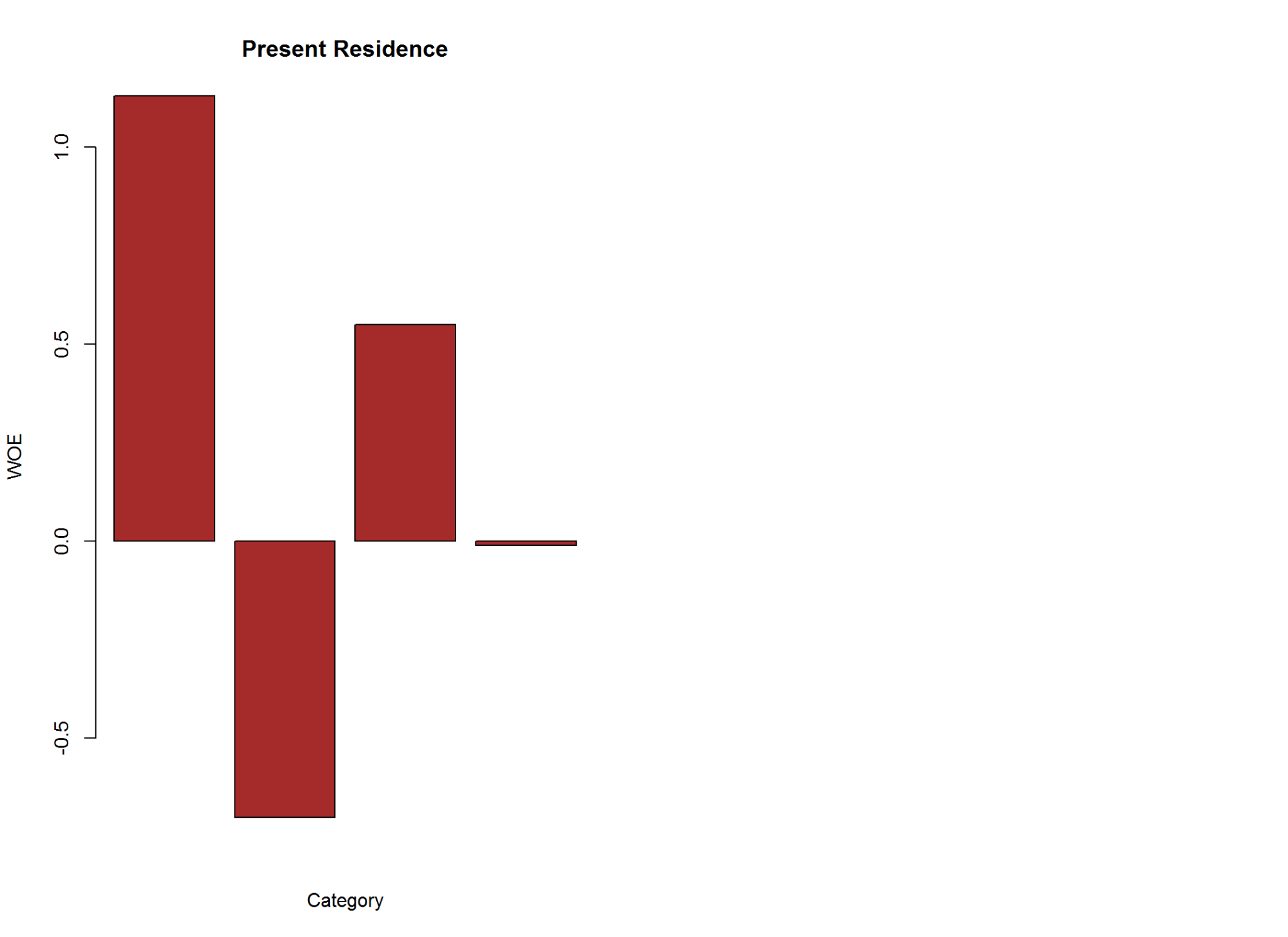
# Attribute 11: (numerical)  
#-----------------------------------------------------------  
# Present residence since  
summary(cdata$present\_residence\_since\_11)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 2.000 3.000 2.845 4.000 4.000

A11<-gbpct(cdata$present\_residence\_since\_11)  
  
op11<-par(mfrow=c(1,2))  
histogram(cdata$present\_residence\_since\_11,  
 main="Present Residence~ Good-Bad",  
 xlab="Present residence Since",   
 ylab="Percent Population")



barplot(A11$WOE, col="brown", names.arg=c(A11$Levels),  
 main="Present Residence",  
 xlab="Category",  
 ylab="WOE")  
par(op11)



kable(A11, caption = "Present Residence~ Good-Bad")

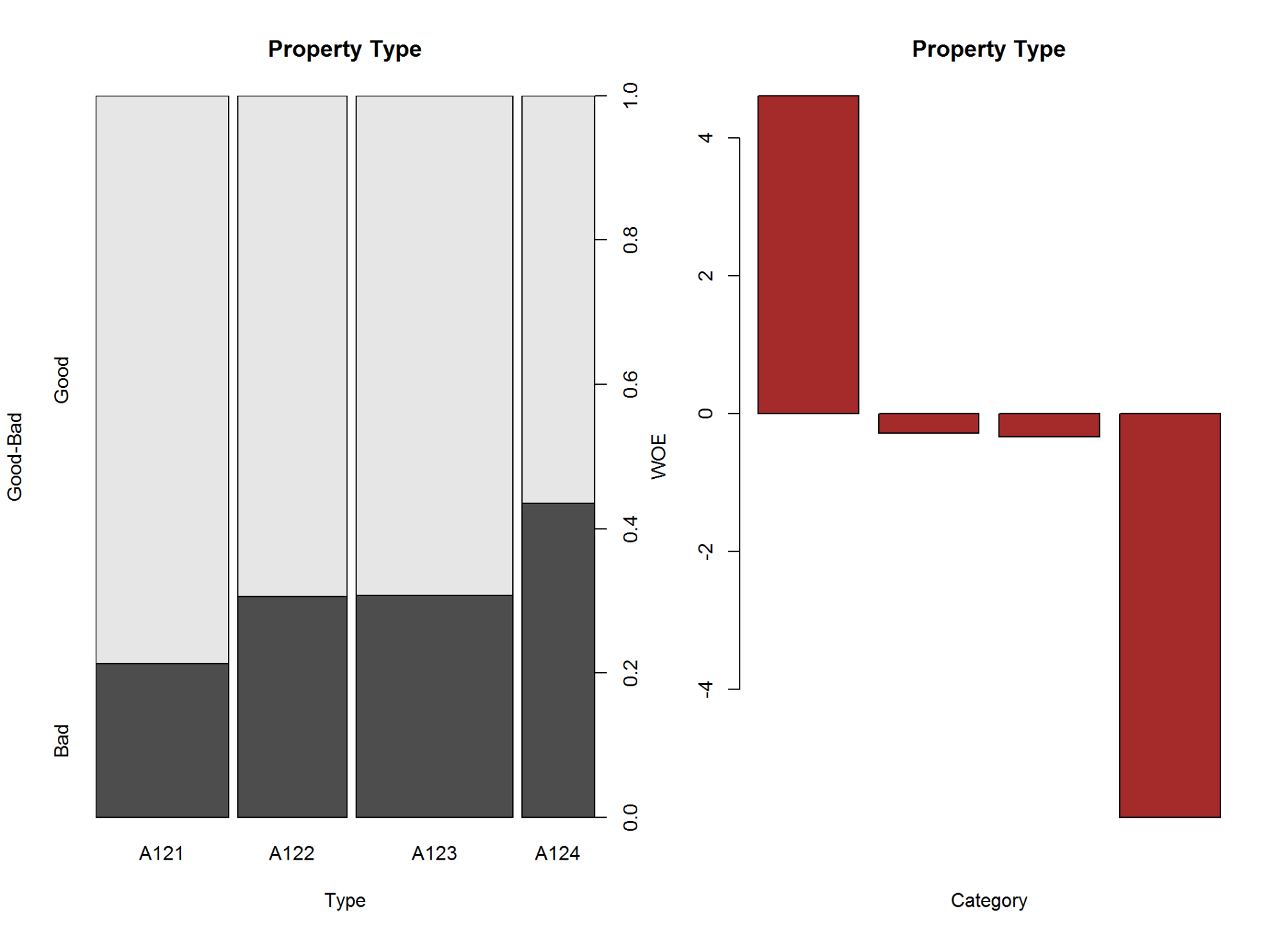
Present Residence~ Good-Bad

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Names | Good | Bad | Good\_pct | Bad\_pct | Total | Total\_Pct | Bad\_Rate | grp\_score | WOE | IV | Efficiency |
| 1 | 94 | 36 | 13.43 | 12.00 | 130 | 13.0 | 27.69 | 5.28 | 1.13 | 0.16159 | 0.715 |
| 2 | 211 | 97 | 30.14 | 32.33 | 308 | 30.8 | 31.49 | 4.82 | -0.70 | 0.15330 | 1.095 |
| 3 | 106 | 43 | 15.14 | 14.33 | 149 | 14.9 | 28.86 | 5.14 | 0.55 | 0.04455 | 0.405 |
| 4 | 289 | 124 | 41.29 | 41.33 | 413 | 41.3 | 30.02 | 5.00 | -0.01 | 0.00004 | 0.020 |

Information Value is  **0.36**  and Efficiency is  **2.23** .

2.2.12 Property Type

# Attribute 12: (qualitative)  
#-----------------------------------------------------------  
# Property  
# A121 : real estate  
# A122 : if not A121 : building society savings agreement/  
# life insurance  
# A123 : if not A121/A122 : car or other, not in attribute 6  
# A124 : unknown / no property  
  
A12 <- gbpct(cdata$property\_type\_12)  
  
op12 <- par(mfrow = c(1,2))  
plot(cdata$property\_type\_12, cdata$good\_bad\_21,   
 main = "Property Type",  
 xlab="Type",  
 ylab="Good-Bad")   
  
barplot(A12$WOE, col="brown", names.arg=c(A12$Levels),  
 main="Property Type",  
 xlab="Category",  
 ylab="WOE")



par(op12)  
  
kable(A12, caption = "Property Type")

Property Type

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Names | Good | Bad | Good\_pct | Bad\_pct | Total | Total\_Pct | Bad\_Rate | grp\_score | WOE | IV | Efficiency |
| A121 | 222 | 60 | 31.71 | 20.00 | 282 | 28.2 | 21.28 | 6.13 | 4.61 | 5.39831 | 5.855 |
| A122 | 161 | 71 | 23.00 | 23.67 | 232 | 23.2 | 30.60 | 4.93 | -0.29 | 0.01943 | 0.335 |
| A123 | 230 | 102 | 32.86 | 34.00 | 332 | 33.2 | 30.72 | 4.91 | -0.34 | 0.03876 | 0.570 |
| A124 | 87 | 67 | 12.43 | 22.33 | 154 | 15.4 | 43.51 | 3.58 | -5.86 | 5.80140 | 4.950 |

Information Value is  **11.26**  and Efficiency is  **11.71** .

2.2.13 Age in Years

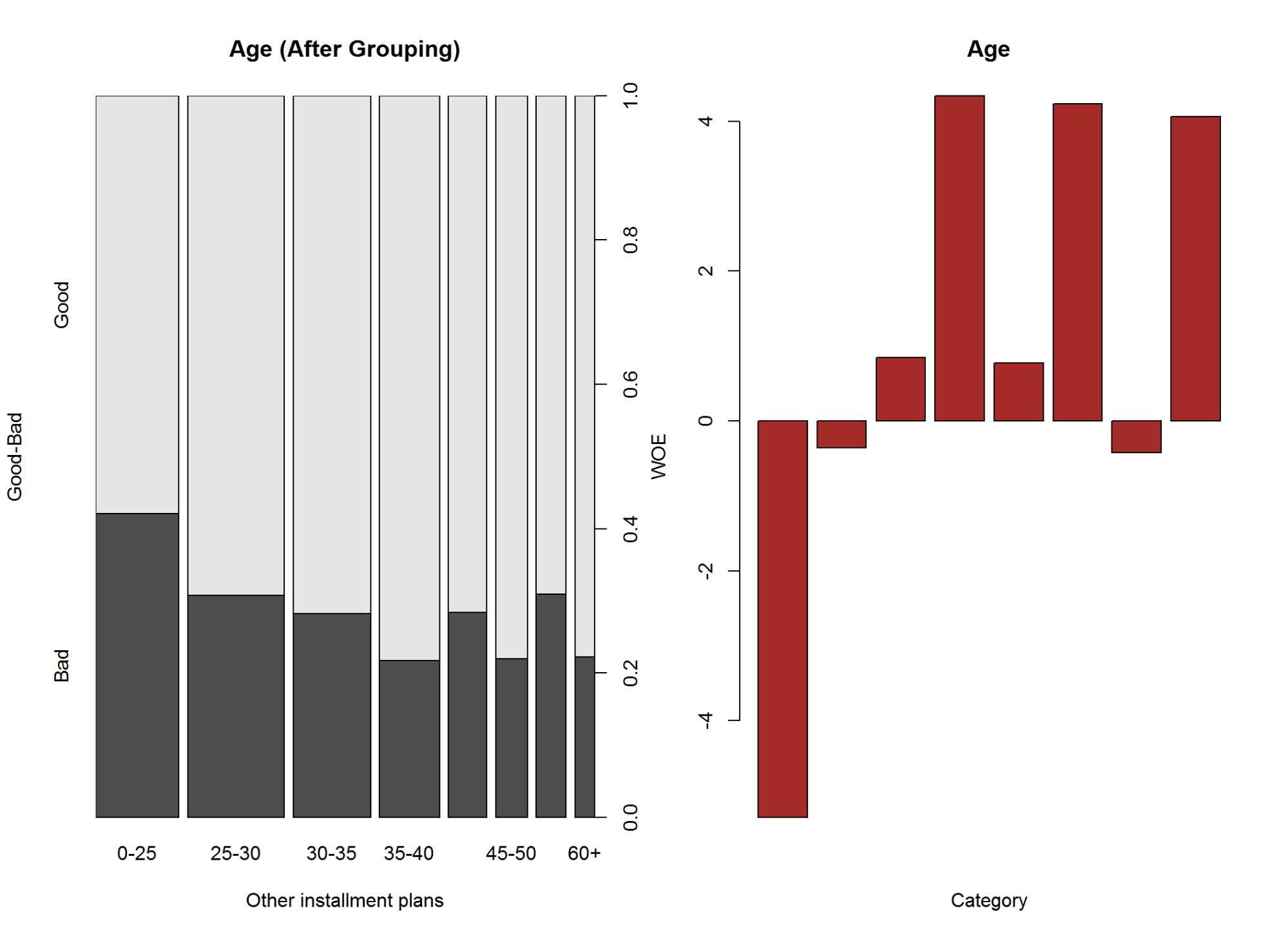
# Attribute 13: (numerical)  
#-----------------------------------------------------------  
# Age in Years  
  
summary(cdata$age\_in\_yrs\_13)

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

op13 <- par(mfrow = c(1,2))  
boxplot(cdata$age\_in\_yrs\_13)  
  
plot(as.factor(cdata$age\_in\_yrs\_13), cdata$good\_bad\_21,  
 main = "Age",  
 xlab = "Age in Years",  
 ylab = "Good-Bad")



par(op13)  
  
cdata$age\_in\_yrs\_13 <- as.factor(ifelse(cdata$age\_in\_yrs\_13<=25, '0-25',  
 ifelse(cdata$age\_in\_yrs\_13<=30, '25-30',  
 ifelse(cdata$age\_in\_yrs\_13<=35, '30-35',   
 ifelse(cdata$age\_in\_yrs\_13<=40, '35-40',   
 ifelse(cdata$age\_in\_yrs\_13<=45, '40-45',   
 ifelse(cdata$age\_in\_yrs\_13<=50, '45-50',  
 ifelse(cdata$age\_in\_yrs\_13<=60, '50-60',  
 '60+'))))))))  
  
  
A13<-gbpct(cdata$age\_in\_yrs\_13)  
  
op13\_1<-par(mfrow=c(1,2))  
plot(as.factor(cdata$age\_in\_yrs\_13), cdata$good\_bad\_21,   
 main="Age (After Grouping)",  
 xlab="Other installment plans",  
 ylab="Good-Bad")  
  
  
barplot(A13$WOE, col="brown", names.arg=c(A13$Levels),  
 main="Age",  
 xlab="Category",  
 ylab="WOE")



par(op13\_1)  
  
kable(A13, caption = "Age (After Grouping) ~ Good-Bad")

Age (After Grouping) ~ Good-Bad

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Names | Good | Bad | Good\_pct | Bad\_pct | Total | Total\_Pct | Bad\_Rate | grp\_score | WOE | IV | Efficiency |
| 0-25 | 110 | 80 | 15.71 | 26.67 | 190 | 19.0 | 42.11 | 3.71 | -5.29 | 5.79784 | 5.480 |
| 25-30 | 153 | 68 | 21.86 | 22.67 | 221 | 22.1 | 30.77 | 4.91 | -0.36 | 0.02916 | 0.405 |
| 30-35 | 127 | 50 | 18.14 | 16.67 | 177 | 17.7 | 28.25 | 5.21 | 0.85 | 0.12495 | 0.735 |
| 35-40 | 108 | 30 | 15.43 | 10.00 | 138 | 13.8 | 21.74 | 6.07 | 4.34 | 2.35662 | 2.715 |
| 40-45 | 63 | 25 | 9.00 | 8.33 | 88 | 8.8 | 28.41 | 5.19 | 0.77 | 0.05159 | 0.335 |
| 45-50 | 57 | 16 | 8.14 | 5.33 | 73 | 7.3 | 21.92 | 6.04 | 4.23 | 1.18863 | 1.405 |
| 50-60 | 47 | 21 | 6.71 | 7.00 | 68 | 6.8 | 30.88 | 4.89 | -0.42 | 0.01218 | 0.145 |
| 60+ | 35 | 10 | 5.00 | 3.33 | 45 | 4.5 | 22.22 | 6.00 | 4.06 | 0.67802 | 0.835 |

Information Value is  **10.24**  and Efficiency is  **12.06** .

2.2.14 Other installment plans

# Attribute 14: (qualitative)  
#-----------------------------------------------------------  
# Other installment plans   
# A141 : bank  
# A142 : stores  
# A143 : none  
  
A14<-gbpct(cdata$other\_installment\_type\_14)  
  
op14<-par(mfrow=c(1,2))  
  
plot(cdata$other\_installment\_type\_14, cdata$good\_bad\_21,   
 main="Other installment plans ~ Good-Bad",  
 xlab="Other installment plans",  
 ylab="Good-Bad")  
  
barplot(A14$WOE, col="brown", names.arg=c(A14$Levels),  
 main="Other installment plans",  
 xlab="Category",  
 ylab="WOE")



par(op14)  
  
kable(A14, caption = "Other installment plans ~ Good-Bad")

Other installment plans ~ Good-Bad

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Names | Good | Bad | Good\_pct | Bad\_pct | Total | Total\_Pct | Bad\_Rate | grp\_score | WOE | IV | Efficiency |
| A141 | 82 | 57 | 11.71 | 19.00 | 139 | 13.9 | 41.01 | 3.81 | -4.84 | 3.52836 | 3.645 |
| A142 | 28 | 19 | 4.00 | 6.33 | 47 | 4.7 | 40.43 | 3.87 | -4.59 | 1.06947 | 1.165 |
| A143 | 590 | 224 | 84.29 | 74.67 | 814 | 81.4 | 27.52 | 5.30 | 1.21 | 1.16402 | 4.810 |

# cdata$other\_installment\_type\_14<-as.factor(ifelse(cdata$other\_installment\_type\_14 == "A143", "None", "banknstore"))  
#   
# A14\_1<-gbpct(cdata$other\_installment\_type\_14)  
#   
# plot(cdata$other\_installment\_type\_14, cdata$good\_bad\_21,   
# ylab="Good-Bad", xlab="Other installment plans",  
# main="Other installment plans (after grouping) ~ Good-Bad")   
#   
# barplot(A14\_1$WOE, col="brown", names.arg=c(A14\_1$Levels),  
# main="Other installment plans",  
# xlab="Category",  
# ylab="WOE")  
#   
# kable(A14\_1)

Information Value is  **5.76**  and Efficiency is  **9.62** .

2.2.15 Housing Type

# Attribute 15: (qualitative)  
#-----------------------------------------------------------  
# Housing  
# A151 : rent  
# A152 : own  
# A153 : for free  
  
A15<-gbpct(cdata$housing\_type\_15)  
  
op15<-par(mfrow=c(1,2))  
plot(cdata$housing\_type\_15, cdata$good\_bad\_21,   
 main="Home Ownership Type",  
 xlab="Type",  
 ylab="Good-Bad")  
  
barplot(A15$WOE, col="brown", names.arg=c(A15$Levels),  
 main="Home Ownership Type",  
 xlab="Type",  
 ylab="WOE")



par(op15)  
  
kable(A15, caption = "Home Ownership Type ~ Good-Bad")

Home Ownership Type ~ Good-Bad

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Names | Good | Bad | Good\_pct | Bad\_pct | Total | Total\_Pct | Bad\_Rate | grp\_score | WOE | IV | Efficiency |
| A151 | 109 | 70 | 15.57 | 23.33 | 179 | 17.9 | 39.11 | 4.00 | -4.04 | 3.13504 | 3.880 |
| A152 | 527 | 186 | 75.29 | 62.00 | 713 | 71.3 | 26.09 | 5.48 | 1.94 | 2.57826 | 6.645 |
| A153 | 64 | 44 | 9.14 | 14.67 | 108 | 10.8 | 40.74 | 3.84 | -4.73 | 2.61569 | 2.765 |

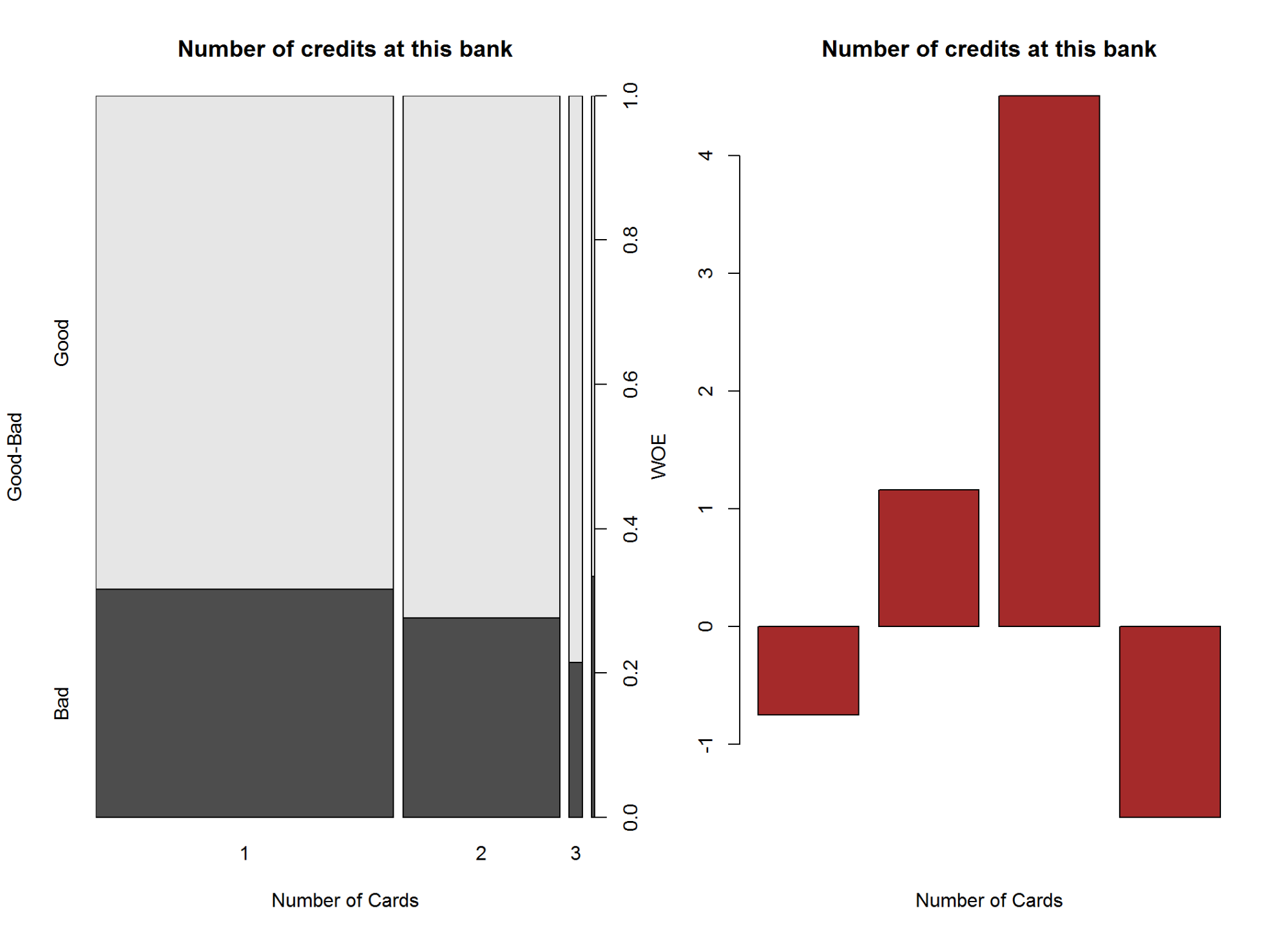
Information Value is  **8.33**  and Efficiency is  **13.29** .

2.2.16 Number of existing credits at this bank

# Attribute 16: (numerical)  
#-----------------------------------------------------------  
# Number of existing credits at this bank  
  
summary(cdata$number\_cards\_this\_bank\_16)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 1.000 1.407 2.000 4.000

A16<-gbpct(cdata$number\_cards\_this\_bank\_16)  
  
op16<-par(mfrow=c(1,2))  
plot(as.factor(cdata$number\_cards\_this\_bank\_16), cdata$good\_bad\_21,  
 main="Number of credits at this bank",  
 xlab="Number of Cards",  
 ylab="Good-Bad")  
  
barplot(A16$WOE, col="brown", names.arg=c(A16$Levels),  
 main="Number of credits at this bank",  
 xlab="Number of Cards",  
 ylab="WOE")



par(op16)  
  
kable(A16, caption = "Number of credits at this bank ~ Good-Bad")

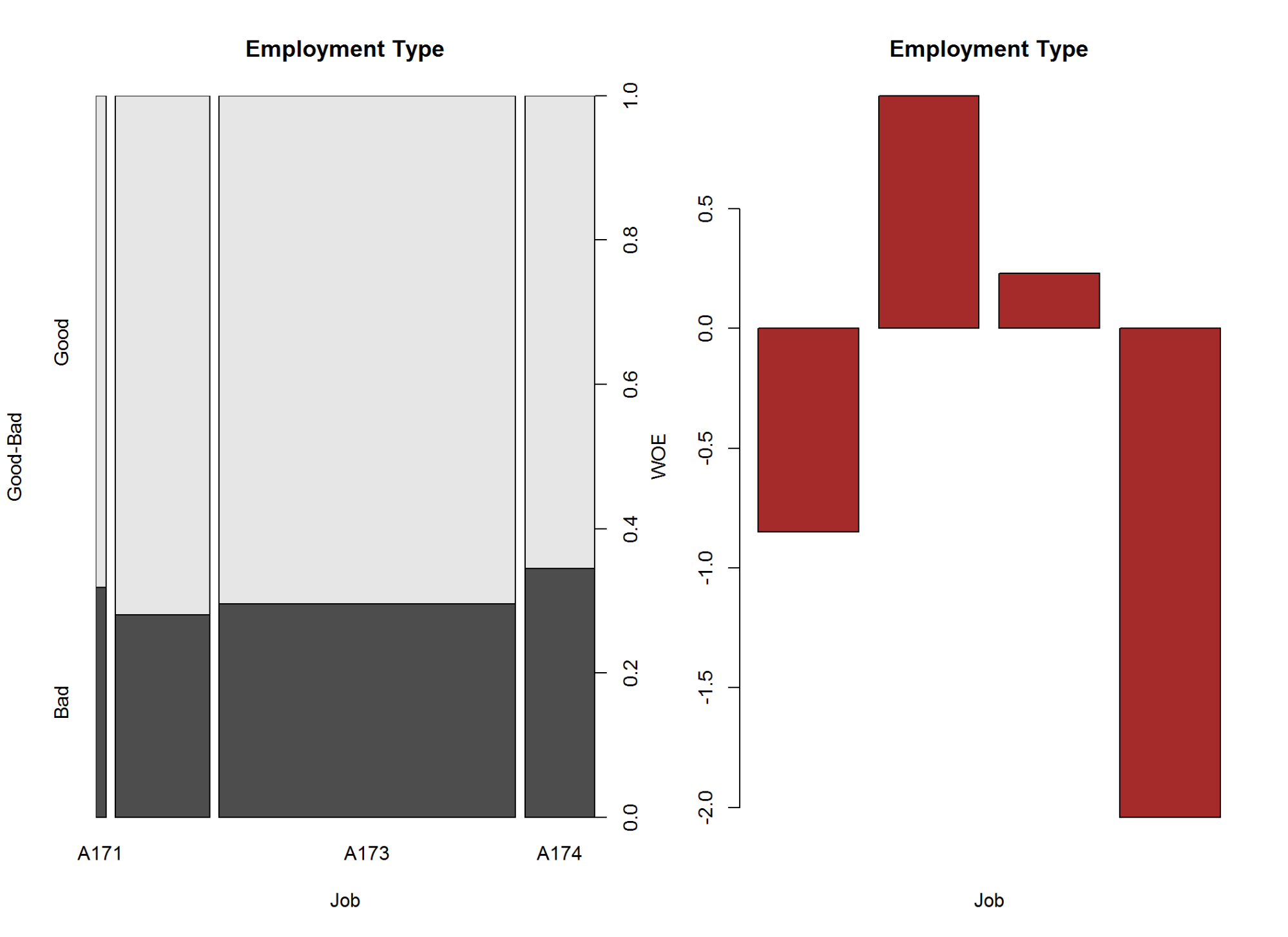
Number of credits at this bank ~ Good-Bad

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Names | Good | Bad | Good\_pct | Bad\_pct | Total | Total\_Pct | Bad\_Rate | grp\_score | WOE | IV | Efficiency |
| 1 | 433 | 200 | 61.86 | 66.67 | 633 | 63.3 | 31.60 | 4.81 | -0.75 | 0.36075 | 2.405 |
| 2 | 241 | 92 | 34.43 | 30.67 | 333 | 33.3 | 27.63 | 5.29 | 1.16 | 0.43616 | 1.880 |
| 3 | 22 | 6 | 3.14 | 2.00 | 28 | 2.8 | 21.43 | 6.11 | 4.51 | 0.51414 | 0.570 |
| 4 | 4 | 2 | 0.57 | 0.67 | 6 | 0.6 | 33.33 | 4.60 | -1.62 | 0.01620 | 0.050 |

Information Value is  **1.33**  and Efficiency is  **4.91** .

2.2.17 Job Status

# Attribute 17: (qualitative)  
#-----------------------------------------------------------  
# Job  
# A171 : unemployed/ unskilled - non-resident  
# A172 : unskilled - resident  
# A173 : skilled employee / official  
# A174 : management/ self-employed/  
# highly qualified employee/ officer  
  
A17<-gbpct(cdata$job\_17)  
  
op17<-par(mfrow=c(1,2))  
  
plot(cdata$job\_17, cdata$good\_bad\_21,   
 main="Employment Type",  
 xlab="Job",  
 ylab="Good-Bad")  
  
barplot(A17$WOE, col="brown", names.arg=c(A17$Levels),  
 main="Employment Type",  
 xlab="Job",  
 ylab="WOE")



par(op17)  
  
kable(A17, caption = "Employment Type ~ Good-Bad")

Employment Type ~ Good-Bad

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Names | Good | Bad | Good\_pct | Bad\_pct | Total | Total\_Pct | Bad\_Rate | grp\_score | WOE | IV | Efficiency |
| A171 | 15 | 7 | 2.14 | 2.33 | 22 | 2.2 | 31.82 | 4.79 | -0.85 | 0.01615 | 0.095 |
| A172 | 144 | 56 | 20.57 | 18.67 | 200 | 20.0 | 28.00 | 5.24 | 0.97 | 0.18430 | 0.950 |
| A173 | 444 | 186 | 63.43 | 62.00 | 630 | 63.0 | 29.52 | 5.06 | 0.23 | 0.03289 | 0.715 |
| A174 | 97 | 51 | 13.86 | 17.00 | 148 | 14.8 | 34.46 | 4.49 | -2.04 | 0.64056 | 1.570 |

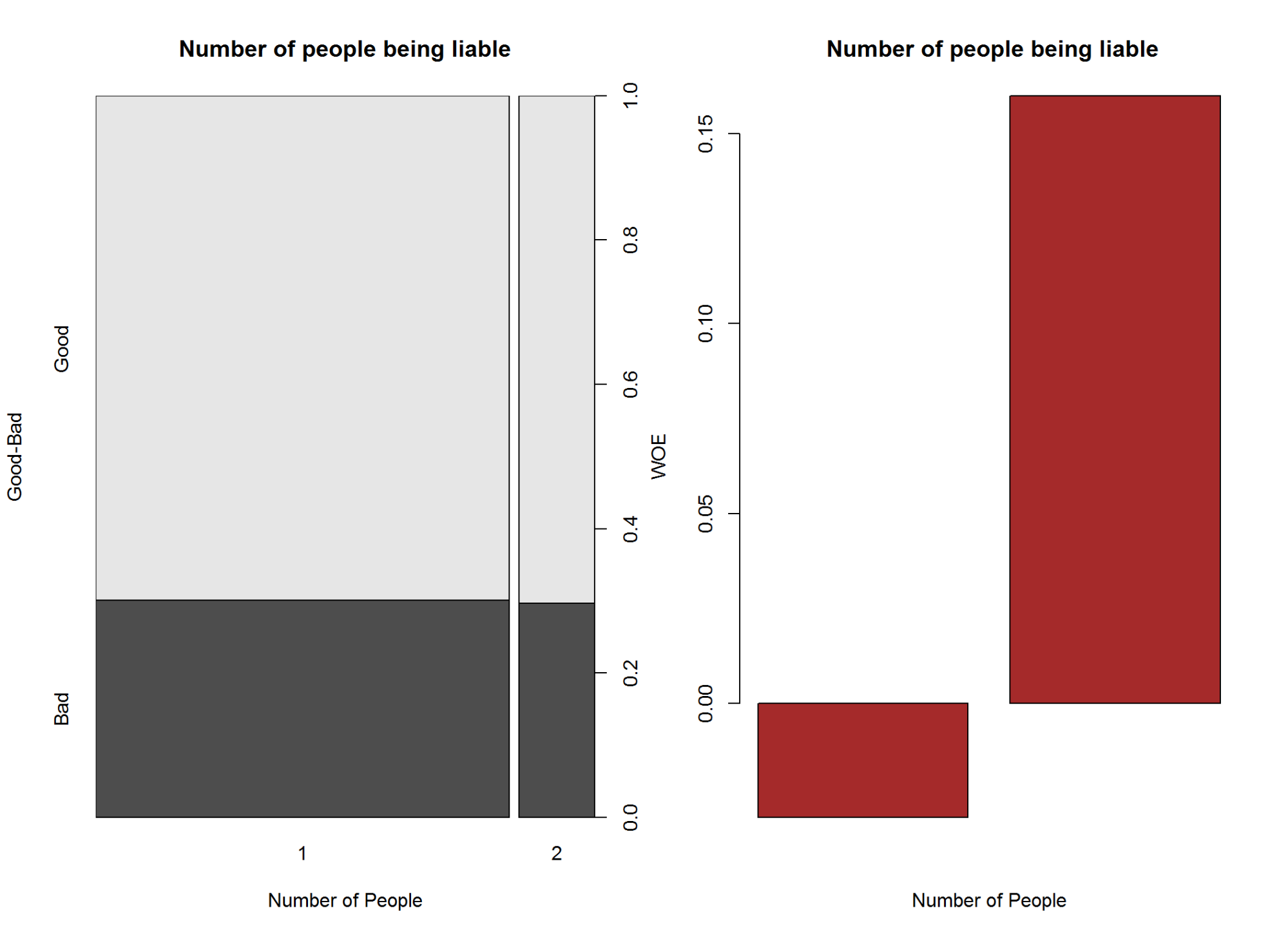
Information Value is  **0.87**  and Efficiency is  **3.33** .

2.2.18 Number of people being liable to provide maintenance for

# Attribute 18: (numerical)  
#-----------------------------------------------------------  
# Number of people being liable to provide maintenance for  
  
summary(cdata$no\_people\_liable\_for\_mntnance\_18)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 1.000 1.155 1.000 2.000

A18<-gbpct(cdata$no\_people\_liable\_for\_mntnance\_18)  
  
op18<-par(mfrow = c(1,2))  
  
plot(as.factor(cdata$no\_people\_liable\_for\_mntnance\_18), cdata$good\_bad\_21,   
 main = "Number of people being liable",  
 xlab = "Number of People",  
 ylab = "Good-Bad")  
  
barplot(A18$WOE, col = "brown", names.arg=c(A18$Levels),  
 main = " Number of people being liable",  
 xlab = "Number of People",  
 ylab = "WOE")



par(op18)  
  
kable(A18, caption = "Number of people being liable ~ Good-Bad")

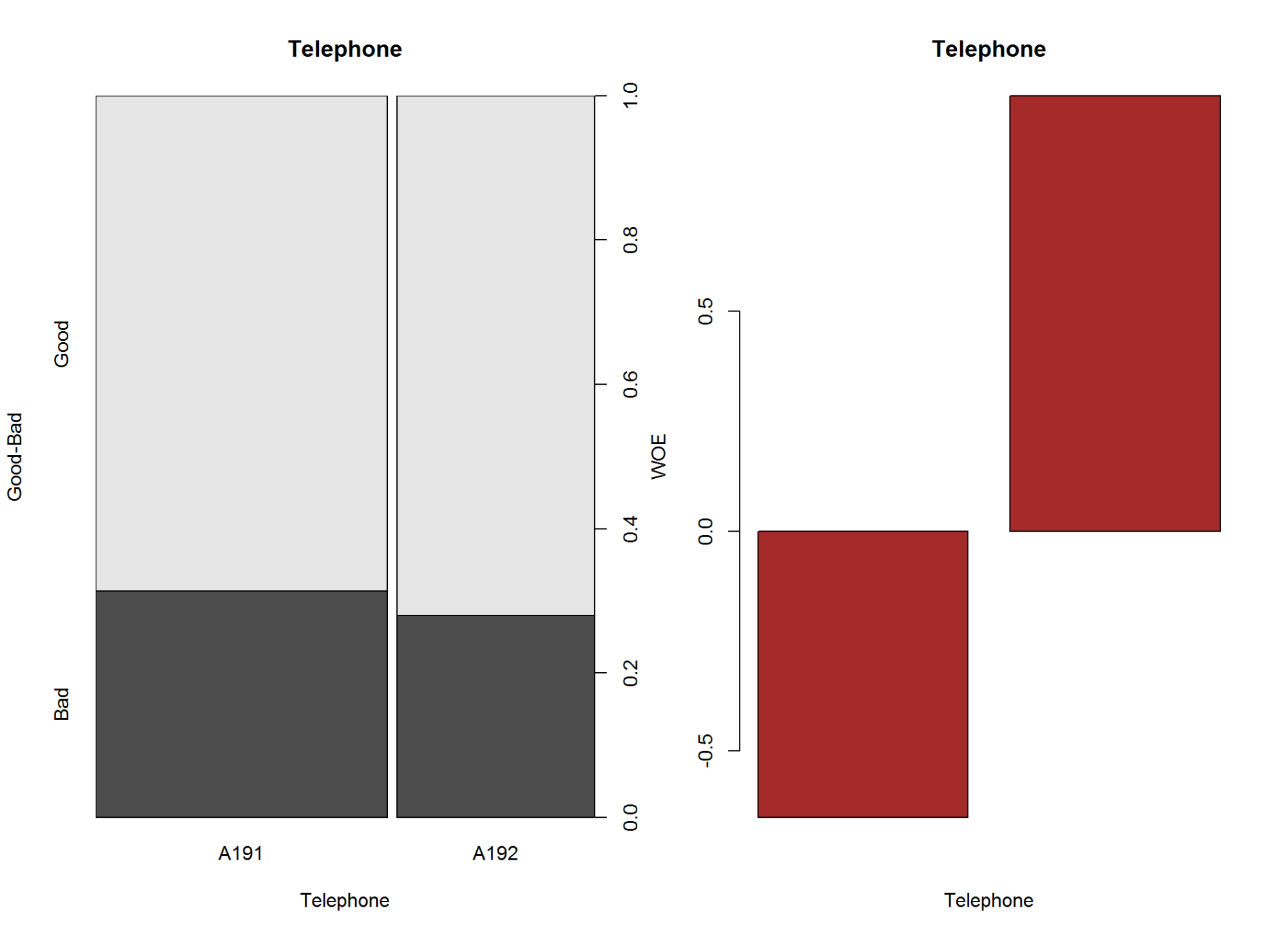
Number of people being liable ~ Good-Bad

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Names | Good | Bad | Good\_pct | Bad\_pct | Total | Total\_Pct | Bad\_Rate | grp\_score | WOE | IV | Efficiency |
| 1 | 591 | 254 | 84.43 | 84.67 | 845 | 84.5 | 30.06 | 4.99 | -0.03 | 0.00072 | 0.12 |
| 2 | 109 | 46 | 15.57 | 15.33 | 155 | 15.5 | 29.68 | 5.04 | 0.16 | 0.00384 | 0.12 |

Information Value is  **0**  and Efficiency is  **0.24** .

2.2.19 Telephone

# Attribute 19: (qualitative)  
#-----------------------------------------------------------  
# Telephone  
# A191 : none  
# A192 : yes, registered under the customers name  
  
A19<-gbpct(cdata$telephone\_19)  
  
op19<-par(mfrow=c(1,2))  
  
plot(cdata$telephone\_19, cdata$good\_bad\_21,   
 main="Telephone",  
 xlab="Telephone",  
 ylab="Good-Bad")  
  
barplot(A19$WOE, col="brown", names.arg=c(A19$Levels),  
 main="Telephone",  
 xlab="Telephone",  
 ylab="WOE")



par(op19)  
  
kable(A19, caption = "Telephone ~ Good-Bad")

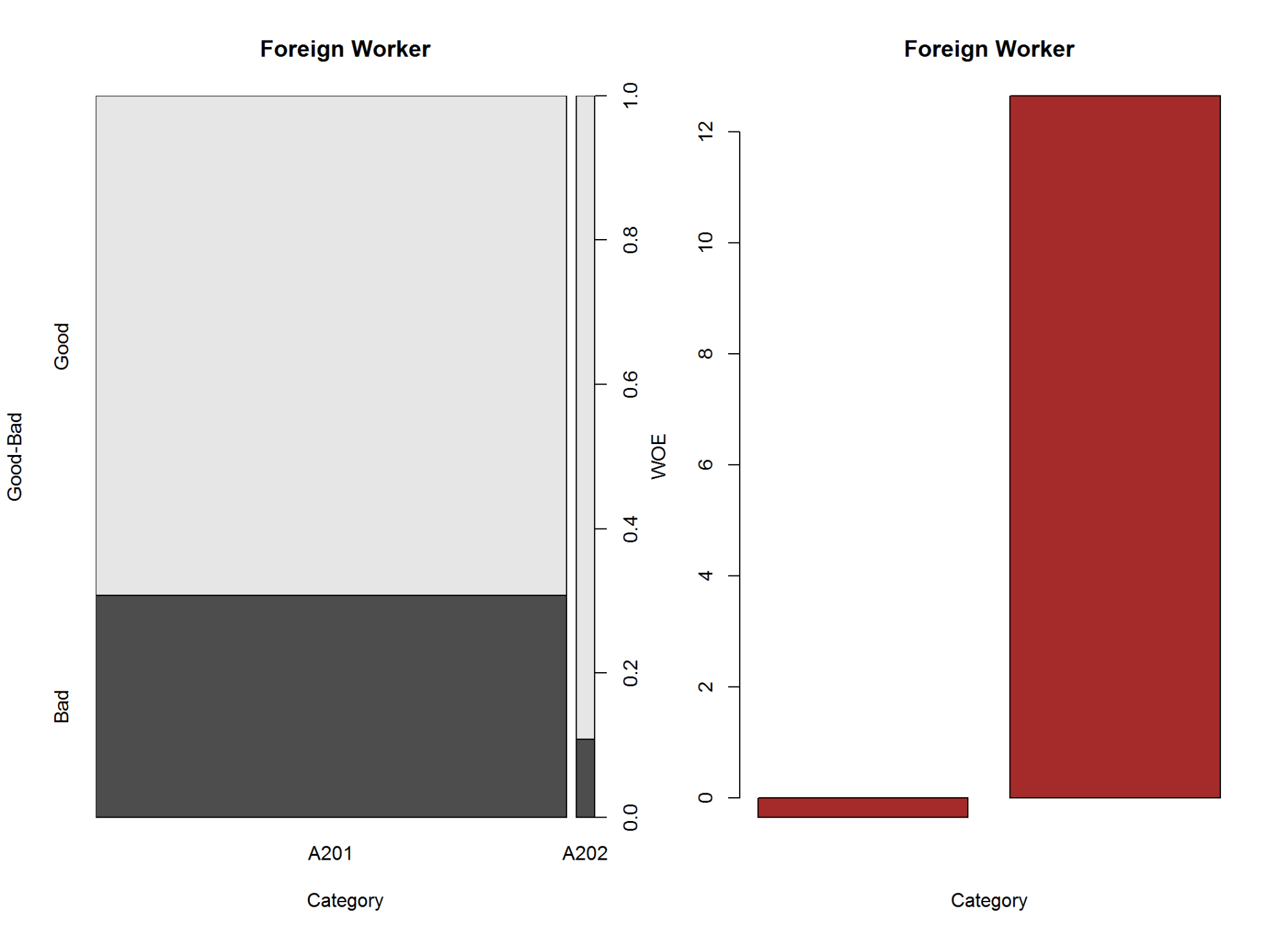
Telephone ~ Good-Bad

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Names | Good | Bad | Good\_pct | Bad\_pct | Total | Total\_Pct | Bad\_Rate | grp\_score | WOE | IV | Efficiency |
| A191 | 409 | 187 | 58.43 | 62.33 | 596 | 59.6 | 31.38 | 4.84 | -0.65 | 0.2535 | 1.95 |
| A192 | 291 | 113 | 41.57 | 37.67 | 404 | 40.4 | 27.97 | 5.25 | 0.99 | 0.3861 | 1.95 |

Information Value is  **0.64**  and Efficiency is  **3.9** .

2.2.20 foreign worker

# Attribute 20: (qualitative)  
#-----------------------------------------------------------  
# foreign worker  
# A201 : yes  
# A202 : no  
  
  
A20<-gbpct(cdata$foreign\_worker\_20)  
  
op20<-par(mfrow=c(1,2))  
  
plot(cdata$foreign\_worker\_20, cdata$good\_bad\_21,   
 main="Foreign Worker",  
 xlab="Category",  
 ylab="Good-Bad")  
  
barplot(A20$WOE, col="brown", names.arg=c(A20$Levels),  
 main="Foreign Worker",  
 xlab="Category",  
 ylab="WOE")



par(op20)  
  
kable(A20, caption = "Foreign Worker ~ Good-Bad")

Foreign Worker ~ Good-Bad

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Names | Good | Bad | Good\_pct | Bad\_pct | Total | Total\_Pct | Bad\_Rate | grp\_score | WOE | IV | Efficiency |
| A201 | 667 | 296 | 95.29 | 98.67 | 963 | 96.3 | 30.74 | 4.91 | -0.35 | 0.1183 | 1.69 |
| A202 | 33 | 4 | 4.71 | 1.33 | 37 | 3.7 | 10.81 | 7.80 | 12.65 | 4.2757 | 1.69 |

Information Value is  **4.39**  and Efficiency is  **3.38** .

2.2.21 IV and WOE

cdata$good\_bad\_21<-as.numeric(ifelse(cdata$good\_bad\_21 == "Good", 0, 1))  
IV <- Information::create\_infotables(data=cdata, NULL, y="good\_bad\_21", 10)  
IV$Summary$IV <- round(IV$Summary$IV\*100,2)  
  
IV$Tables

## $chk\_ac\_status\_1  
## chk\_ac\_status\_1 N Percent WOE IV  
## 1 A11 274 0.274 0.8180987 0.2056934  
## 2 A12 269 0.269 0.4013918 0.2521402  
## 3 A13 63 0.063 -0.4054651 0.2616010  
## 4 A14 394 0.394 -1.1762632 0.6660115  
##   
## $duration\_month\_2  
## duration\_month\_2 N Percent WOE IV  
## 1 00-06 82 0.082 -1.24593700 0.09255532  
## 2 06-12 277 0.277 -0.29511705 0.11518096  
## 3 12-24 411 0.411 -0.01510778 0.11527449  
## 4 24-30 57 0.057 0.15415068 0.11666918  
## 5 30-36 86 0.086 0.61368301 0.15232124  
## 6 36-42 17 0.017 -0.02817088 0.15233466  
## 7 42+ 70 0.070 1.13497993 0.25502332  
##   
## $credit\_history\_3  
## credit\_history\_3 N Percent WOE IV  
## 1 01.A30 40 0.040 1.35812348 0.08407431  
## 2 02.A31 49 0.049 1.13497993 0.15595637  
## 3 03.A32.A33 618 0.618 0.08786876 0.16081008  
## 4 04.A34 293 0.293 -0.73374058 0.29323278  
##   
## $purpose\_4  
## purpose\_4 N Percent WOE IV  
## 1 A40 234 0.234 0.35920049 0.03215700  
## 2 A41 103 0.103 -0.77383609 0.08337758  
## 3 A410 12 0.012 0.51082562 0.08678308  
## 4 A42 181 0.181 0.09555652 0.08846669  
## 5 A43 280 0.280 -0.41006282 0.13142566  
## 6 A44 12 0.012 0.15415068 0.13171928  
## 7 A45 22 0.022 0.28768207 0.13363716  
## 8 A46 50 0.050 0.60613580 0.15384168  
## 9 A48 9 0.009 -1.23214368 0.16381618  
## 10 A49 97 0.097 0.23052366 0.16919507  
##   
## $credit\_amount\_5  
## credit\_amount\_5 N Percent WOE IV  
## 1 0-1400 267 0.267 0.03366128 0.0003045545  
## 2 1400-2500 270 0.270 -0.30132485 0.0232626382  
## 3 2500-3500 149 0.149 -0.22464618 0.0304299211  
## 4 3500-4500 98 0.098 -0.17127172 0.0332028918  
## 5 4500-5500 48 0.048 0.24652400 0.0362550937  
## 6 5500+ 168 0.168 0.55961579 0.0938155748  
##   
## $savings\_ac\_bond\_6  
## savings\_ac\_bond\_6 N Percent WOE IV  
## 1 A61 603 0.603 0.2713578 0.04664771  
## 2 A62 103 0.103 0.1395519 0.04870776  
## 3 A63 63 0.063 -0.7060506 0.07526871  
## 4 A64 48 0.048 -1.0986123 0.11921320  
## 5 A65 183 0.183 -0.7042461 0.19600956  
##   
## $p\_employment\_since\_7  
## p\_employment\_since\_7 N Percent WOE IV  
## 1 A71 62 0.062 0.31923043 0.006688638  
## 2 A72 172 0.172 0.47082029 0.047941463  
## 3 A73 339 0.339 0.03210325 0.048293070  
## 4 A74 174 0.174 -0.39441527 0.073084887  
## 5 A75 253 0.253 -0.23556607 0.086433631  
##   
## $installment\_pct\_disp\_inc\_8  
## installment\_pct\_disp\_inc\_8 N Percent WOE IV  
## 1 [1,1] 136 0.136 -0.25131443 0.008137801  
## 2 [2,2] 231 0.231 -0.15546647 0.013542111  
## 3 [3,3] 157 0.157 -0.06453852 0.014187496  
## 4 [4,4] 476 0.476 0.15730029 0.026322090  
##   
## $personal\_status\_9  
## personal\_status\_9 N Percent WOE IV  
## 1 A91 50 0.050 0.4418328 0.01051983  
## 2 A92 310 0.310 0.2353408 0.02845056  
## 3 A93 548 0.548 -0.1655476 0.04295568  
## 4 A94 92 0.092 -0.1385189 0.04467068  
##   
## $other\_debtors\_or\_grantors\_10  
## other\_debtors\_or\_grantors\_10 N Percent WOE IV  
## 1 A101 907 0.907 -0.0005250722 2.500344e-07  
## 2 A102 41 0.041 0.6021754024 1.634501e-02  
## 3 A103 52 0.052 -0.5877866649 3.201932e-02  
##   
## $present\_residence\_since\_11  
## present\_residence\_since\_11 N Percent WOE IV  
## 1 [1,1] 130 0.130 -0.112477983 0.001606828  
## 2 [2,2] 308 0.308 0.070150705 0.003143463  
## 3 [3,3] 149 0.149 -0.054941118 0.003588224  
## 4 [4,4] 413 0.413 0.001152738 0.003588773  
##   
## $property\_type\_12  
## property\_type\_12 N Percent WOE IV  
## 1 A121 282 0.282 -0.46103496 0.05400695  
## 2 A122 232 0.232 0.02857337 0.05419744  
## 3 A123 332 0.332 0.03419136 0.05458820  
## 4 A124 154 0.154 0.58608236 0.11263826  
##   
## $age\_in\_yrs\_13  
## age\_in\_yrs\_13 N Percent WOE IV  
## 1 0-25 190 0.190 0.52884413 0.05792102  
## 2 25-30 221 0.221 0.03636764 0.05821543  
## 3 30-35 177 0.177 -0.08486622 0.05946822  
## 4 35-40 138 0.138 -0.43363599 0.08300845  
## 5 40-45 88 0.088 -0.07696104 0.08352153  
## 6 45-50 73 0.073 -0.42316469 0.09541044  
## 7 50-60 68 0.068 0.04167270 0.09552951  
## 8 60+ 45 0.045 -0.40546511 0.10228726  
##   
## $other\_installment\_type\_14  
## other\_installment\_type\_14 N Percent WOE IV  
## 1 A141 139 0.139 0.4836299 0.03523589  
## 2 A142 47 0.047 0.4595323 0.04595831  
## 3 A143 814 0.814 -0.1211786 0.05761454  
##   
## $housing\_type\_15  
## housing\_type\_15 N Percent WOE IV  
## 1 A151 179 0.179 0.4044452 0.03139265  
## 2 A152 713 0.713 -0.1941560 0.05718767  
## 3 A153 108 0.108 0.4726044 0.08329343  
##   
## $number\_cards\_this\_bank\_16  
## number\_cards\_this\_bank\_16 N Percent WOE IV  
## 1 [1,1] 633 0.633 0.0748775 0.003601251  
## 2 [2,4] 367 0.367 -0.1347806 0.010083557  
##   
## $job\_17  
## job\_17 N Percent WOE IV  
## 1 A171 22 0.022 0.08515781 0.0001622053  
## 2 A172 200 0.200 -0.09716375 0.0020129434  
## 3 A173 630 0.630 -0.02278003 0.0023383724  
## 4 A174 148 0.148 0.20441251 0.0087627657  
##   
## $no\_people\_liable\_for\_mntnance\_18  
## no\_people\_liable\_for\_mntnance\_18 N Percent WOE IV  
## 1 [1,1] 845 0.845 0.00281611 6.705024e-06  
## 2 [2,2] 155 0.155 -0.01540863 4.339223e-05  
##   
## $telephone\_19  
## telephone\_19 N Percent WOE IV  
## 1 A191 596 0.596 0.06469132 0.002526042  
## 2 A192 404 0.404 -0.09863759 0.006377605  
##   
## $foreign\_worker\_20  
## foreign\_worker\_20 N Percent WOE IV  
## 1 A201 963 0.963 0.03486727 0.001178846  
## 2 A202 37 0.037 -1.26291534 0.043877412

kable(IV$Summary)

|  |  |  |
| --- | --- | --- |
|  | Variable | IV |
| 1 | chk\_ac\_status\_1 | 66.60 |
| 3 | credit\_history\_3 | 29.32 |
| 2 | duration\_month\_2 | 25.50 |
| 6 | savings\_ac\_bond\_6 | 19.60 |
| 4 | purpose\_4 | 16.92 |
| 12 | property\_type\_12 | 11.26 |
| 13 | age\_in\_yrs\_13 | 10.23 |
| 5 | credit\_amount\_5 | 9.38 |
| 7 | p\_employment\_since\_7 | 8.64 |
| 15 | housing\_type\_15 | 8.33 |
| 14 | other\_installment\_type\_14 | 5.76 |
| 9 | personal\_status\_9 | 4.47 |
| 20 | foreign\_worker\_20 | 4.39 |
| 10 | other\_debtors\_or\_grantors\_10 | 3.20 |
| 8 | installment\_pct\_disp\_inc\_8 | 2.63 |
| 16 | number\_cards\_this\_bank\_16 | 1.01 |
| 17 | job\_17 | 0.88 |
| 19 | telephone\_19 | 0.64 |
| 11 | present\_residence\_since\_11 | 0.36 |
| 18 | no\_people\_liable\_for\_mntnance\_18 | 0.00 |

cdata$good\_bad\_21<-as.factor(ifelse(cdata$good\_bad\_21 == 0, "Good", "Bad"))

I. Following variables do not have prediction power - very very weak predictor (IV< 2%), hence we shall exclude them from modeling

Position Variable IV 16 number\_cards\_this\_bank\_16 1.01 17 job\_17 0.88 19 telephone\_19 0.64 11 present\_residence\_since\_11, 0.36 18 no\_people\_liable\_for\_mntnance\_18, 0.00

1. Following variables are very weak predictors (2%<=IV< 10%), hence we may or may not include them while modeling

* Position Variable IV
* 7 p\_employment\_since\_7 8.64
* 15 housing\_type\_15 8.33
* 14 other\_installment\_type\_14 5.76
* 9 personal\_status\_9 4.47
* 20 foreign\_worker\_20 4.39
* 10 other\_debtors\_or\_grantors\_10 3.20
* 8 installment\_pct\_disp\_inc\_8 2.63

1. Following variables have medium prediction power (10%<=IV< 30%), hence we will include them in modeling as we have less number of variables

* Position Variable IV
* 3 credit\_history\_3 29.32
* 2 duration\_month\_2 27.79
* 6 savings\_ac\_bond\_6 19.60
* 4 purpose\_4 16.92
* 13 age\_in\_yrs\_13 12.12
* 12 property\_type\_12 11.26
* 5 credit\_amount\_5 11.18

1. There is no strong predictor with IV between 30% to 50%
2. chk\_ac\_status\_1 has a very high prediction power (IV > 50%), it could be suspicious and require further investigation

* Position Variable IV
* 1 chk\_ac\_status\_1 66.60

2.2.21.1 Subset Data - 1

var\_list\_1 <- IV$Summary[IV$Summary$IV > 2,] # 15 variables  
cdata\_reduced\_1<-cdata[, c(var\_list\_1$Variable,"good\_bad\_21")] #16 variables

2.2.22 Variable Reduction using Variable Clustering

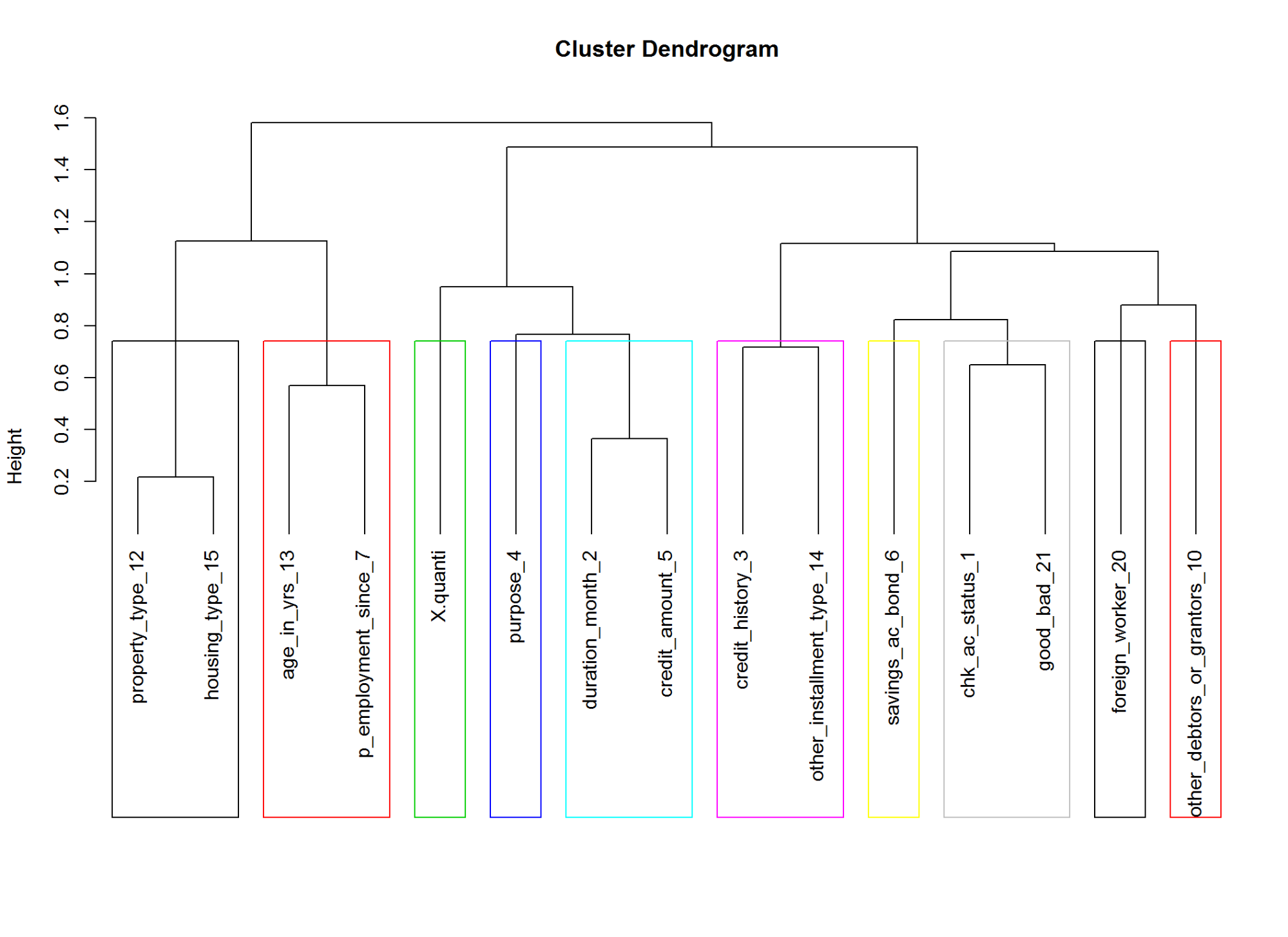
# ClusterOfVariables  
# Step 1:  
factors<-sapply(cdata\_reduced\_1, is.factor)  
vars\_quali<- cdata\_reduced\_1[,factors]  
#vars\_quali$good\_bad\_21<-vars\_quali$good\_bad\_21[drop=TRUE] # remove empty factors  
str(vars\_quali)

## 'data.frame': 1000 obs. of 15 variables:  
## $ chk\_ac\_status\_1 : Factor w/ 4 levels "A11","A12","A13",..: 1 2 4 1 1 4 4 2 4 2 ...  
## $ credit\_history\_3 : Factor w/ 4 levels "01.A30","02.A31",..: 4 3 4 3 3 3 3 3 3 4 ...  
## $ duration\_month\_2 : Factor w/ 7 levels "00-06","06-12",..: 1 7 2 6 3 5 3 5 2 4 ...  
## $ savings\_ac\_bond\_6 : Factor w/ 5 levels "A61","A62","A63",..: 5 1 1 1 1 5 3 1 4 1 ...  
## $ purpose\_4 : Factor w/ 10 levels "A40","A41","A410",..: 5 5 8 4 1 8 4 2 5 1 ...  
## $ property\_type\_12 : Factor w/ 4 levels "A121","A122",..: 1 1 1 2 4 4 2 3 1 3 ...  
## $ age\_in\_yrs\_13 : Factor w/ 8 levels "0-25","25-30",..: 8 1 6 5 7 3 7 3 8 2 ...  
## $ credit\_amount\_5 : Factor w/ 6 levels "0-1400","1400-2500",..: 1 6 2 6 5 6 3 6 3 5 ...  
## $ p\_employment\_since\_7 : Factor w/ 5 levels "A71","A72","A73",..: 5 3 4 4 3 3 5 3 4 1 ...  
## $ housing\_type\_15 : Factor w/ 3 levels "A151","A152",..: 2 2 2 3 3 3 2 1 2 2 ...  
## $ other\_installment\_type\_14 : Factor w/ 3 levels "A141","A142",..: 3 3 3 3 3 3 3 3 3 3 ...  
## $ personal\_status\_9 : Factor w/ 4 levels "A91","A92","A93",..: 3 2 3 3 3 3 3 3 1 4 ...  
## $ foreign\_worker\_20 : Factor w/ 2 levels "A201","A202": 1 1 1 1 1 1 1 1 1 1 ...  
## $ other\_debtors\_or\_grantors\_10: Factor w/ 3 levels "A101","A102",..: 1 1 1 3 1 1 1 1 1 1 ...  
## $ good\_bad\_21 : Factor w/ 2 levels "Bad","Good": 2 1 2 2 1 2 2 2 2 1 ...

vars\_quanti <- cdata\_reduced\_1[,!factors]  
  
str(vars\_quanti)

## num [1:1000] 4 2 2 2 3 2 3 2 2 4 ...

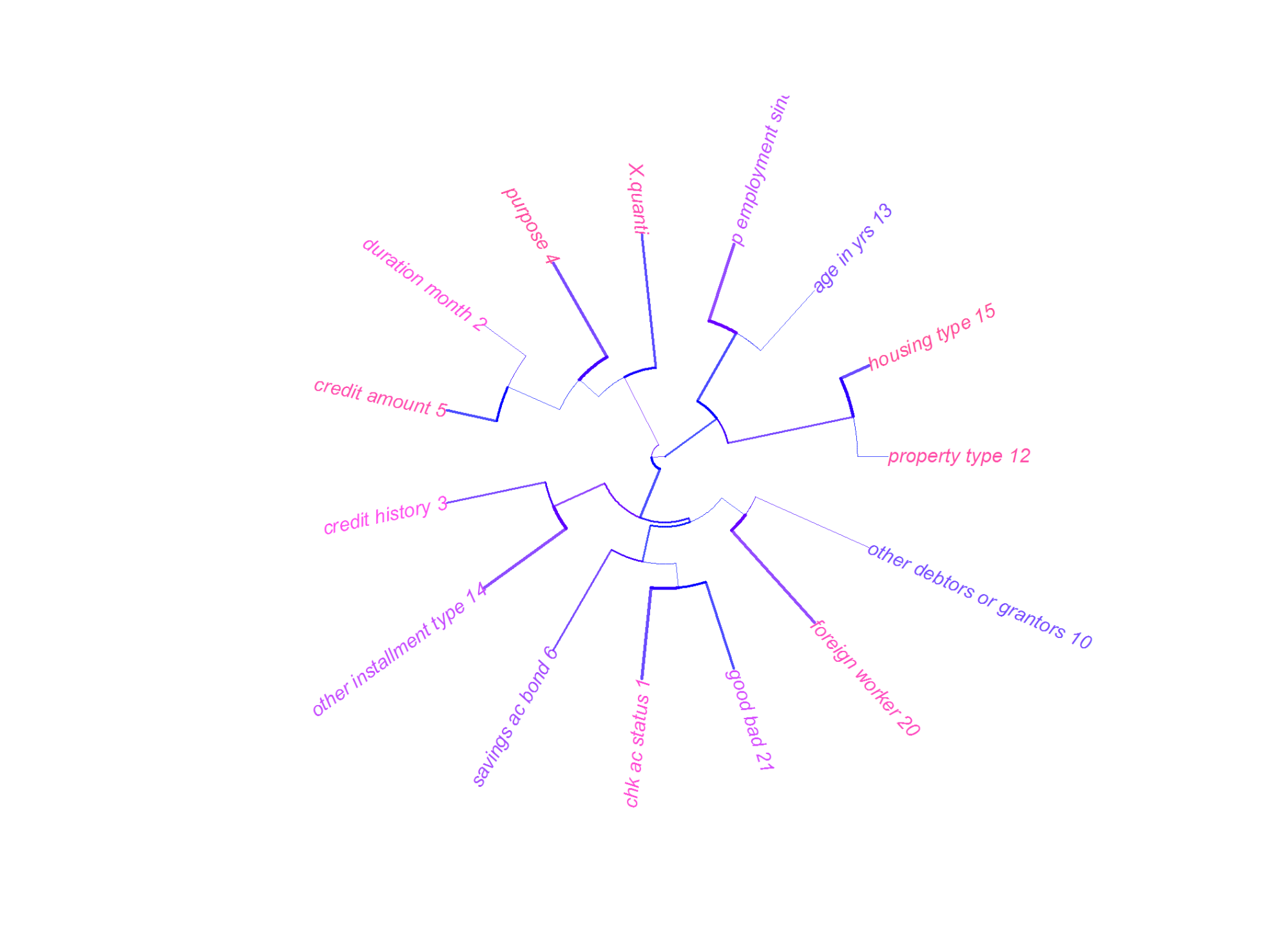
tree <- hclustvar(X.quanti=vars\_quanti,X.quali=vars\_quali[,-c(12)])  
plot(tree)  
rect.hclust(tree, k=10, border = 1:10)



summary(tree)

## Length Class Mode   
## call 3 -none- call   
## rec 14 -none- list   
## merge 28 -none- numeric   
## height 14 -none- numeric   
## order 15 -none- numeric   
## labels 15 -none- character  
## clusmat 225 -none- numeric   
## X.quanti 1 data.frame list   
## X.quali 14 data.frame list

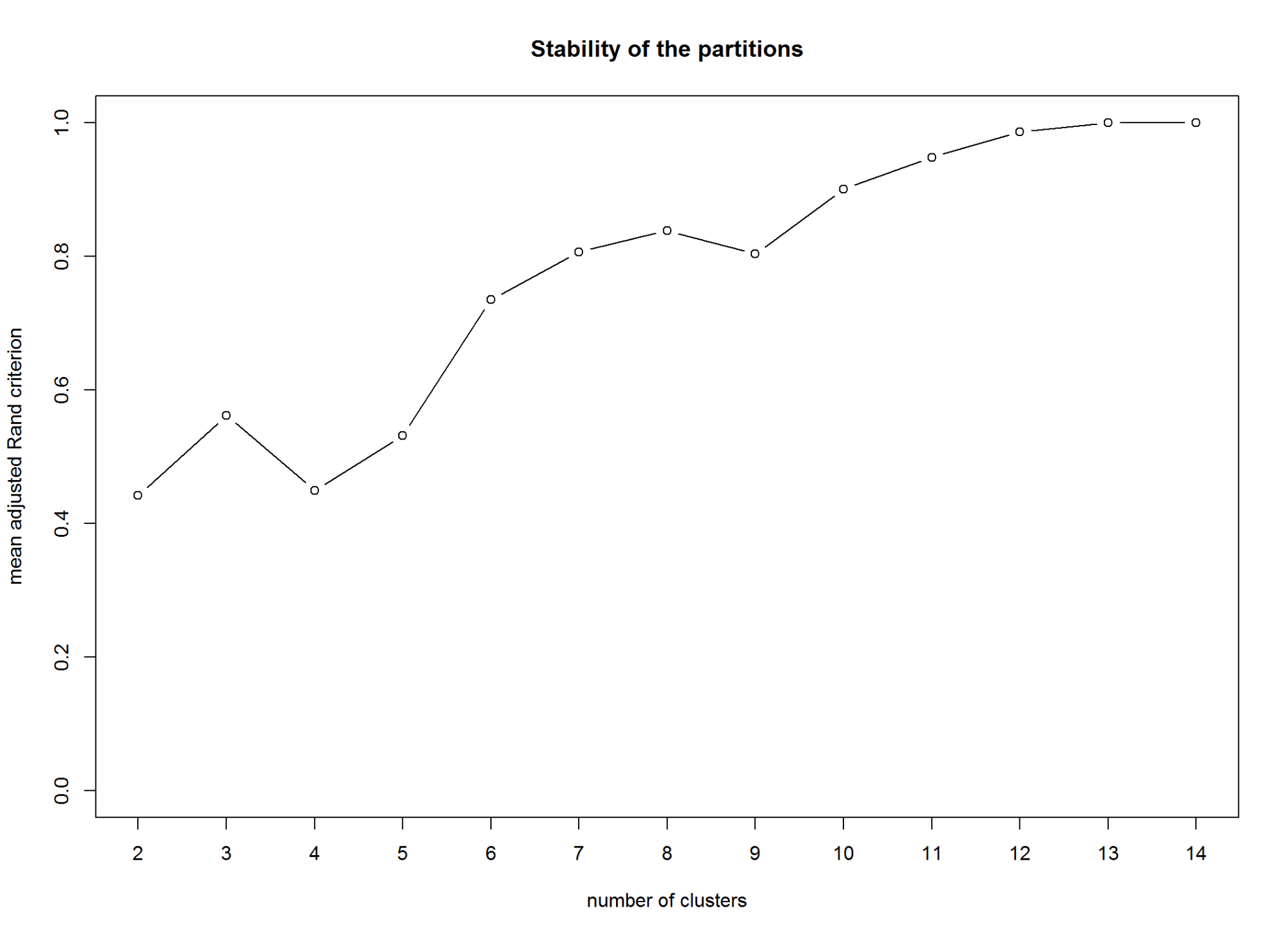
# add colors randomly  
plot(as.phylo(tree), type = "fan",  
 tip.color = hsv(runif(15, 0.65, 0.95), 1, 1, 0.7),  
 edge.color = hsv(runif(10, 0.65, 0.75), 1, 1, 0.7),   
 edge.width = runif(20, 0.5, 3), use.edge.length = TRUE, col = "gray80")



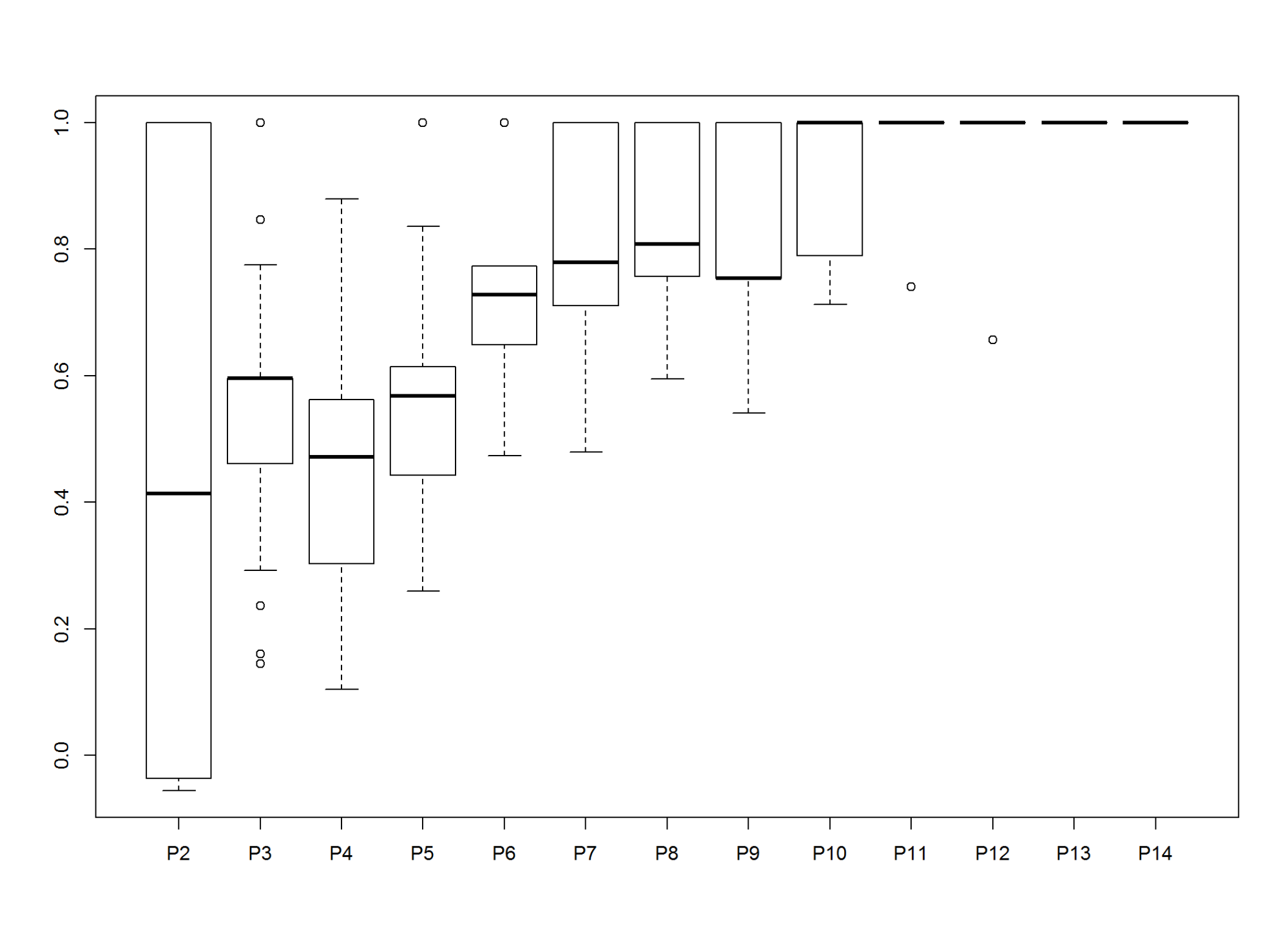
summary.phylo(as.phylo(tree))

##   
## Phylogenetic tree: as.phylo(tree)   
##   
## Number of tips: 15   
## Number of nodes: 14   
## Branch lengths:  
## mean: 0.2484595   
## variance: 0.01806587   
## distribution summary:  
## Min. 1st Qu. Median 3rd Qu. Max.   
## 0.01508 0.12620 0.24880 0.35860 0.47440   
## No root edge.  
## First ten tip labels: X.quanti   
## chk\_ac\_status\_1  
## credit\_history\_3  
## duration\_month\_2  
## savings\_ac\_bond\_6  
## purpose\_4  
## property\_type\_12  
## age\_in\_yrs\_13  
## credit\_amount\_5  
## p\_employment\_since\_7  
## No node labels.

stab<-stability(tree,B=50) # Bootstrap 50 times



#plot(stab,main="Stability of the partitions")  
boxplot(stab$matCR)



part<-cutreevar(tree,10)  
print(part)

##   
## Call:  
## cutreevar(obj = tree, k = 10)  
##   
##   
##   
## name description   
## "$var" "list of variables in each cluster"  
## "$sim" "similarity matrix in each cluster"  
## "$cluster" "cluster memberships"   
## "$wss" "within-cluster sum of squares"   
## "$E" "gain in cohesion (in %)"   
## "$size" "size of each cluster"   
## "$scores" "score of each cluster"

summary(part)

##   
## Call:  
## cutreevar(obj = tree, k = 10)  
##   
##   
##   
## Data:   
## number of observations: 1000  
## number of variables: 15  
## number of numerical variables: 1  
## number of categorical variables: 14  
## number of clusters: 10  
##   
## Cluster 1 :   
## squared loading  
## X.quanti 1  
##   
##   
## Cluster 2 :   
## squared loading  
## chk\_ac\_status\_1 0.68  
## good\_bad\_21 0.68  
##   
##   
## Cluster 3 :   
## squared loading  
## credit\_history\_3 0.64  
## other\_installment\_type\_14 0.64  
##   
##   
## Cluster 4 :   
## squared loading  
## duration\_month\_2 0.82  
## credit\_amount\_5 0.82  
##   
##   
## Cluster 5 :   
## squared loading  
## savings\_ac\_bond\_6 1  
##   
##   
## Cluster 6 :   
## squared loading  
## purpose\_4 1  
##   
##   
## Cluster 7 :   
## squared loading  
## property\_type\_12 0.89  
## housing\_type\_15 0.89  
##   
##   
## Cluster 8 :   
## squared loading  
## age\_in\_yrs\_13 0.72  
## p\_employment\_since\_7 0.72  
##   
##   
## Cluster 9 :   
## squared loading  
## foreign\_worker\_20 1  
##   
##   
## Cluster 10 :   
## squared loading  
## other\_debtors\_or\_grantors\_10 1  
##   
##   
## Gain in cohesion (in %): 79.58

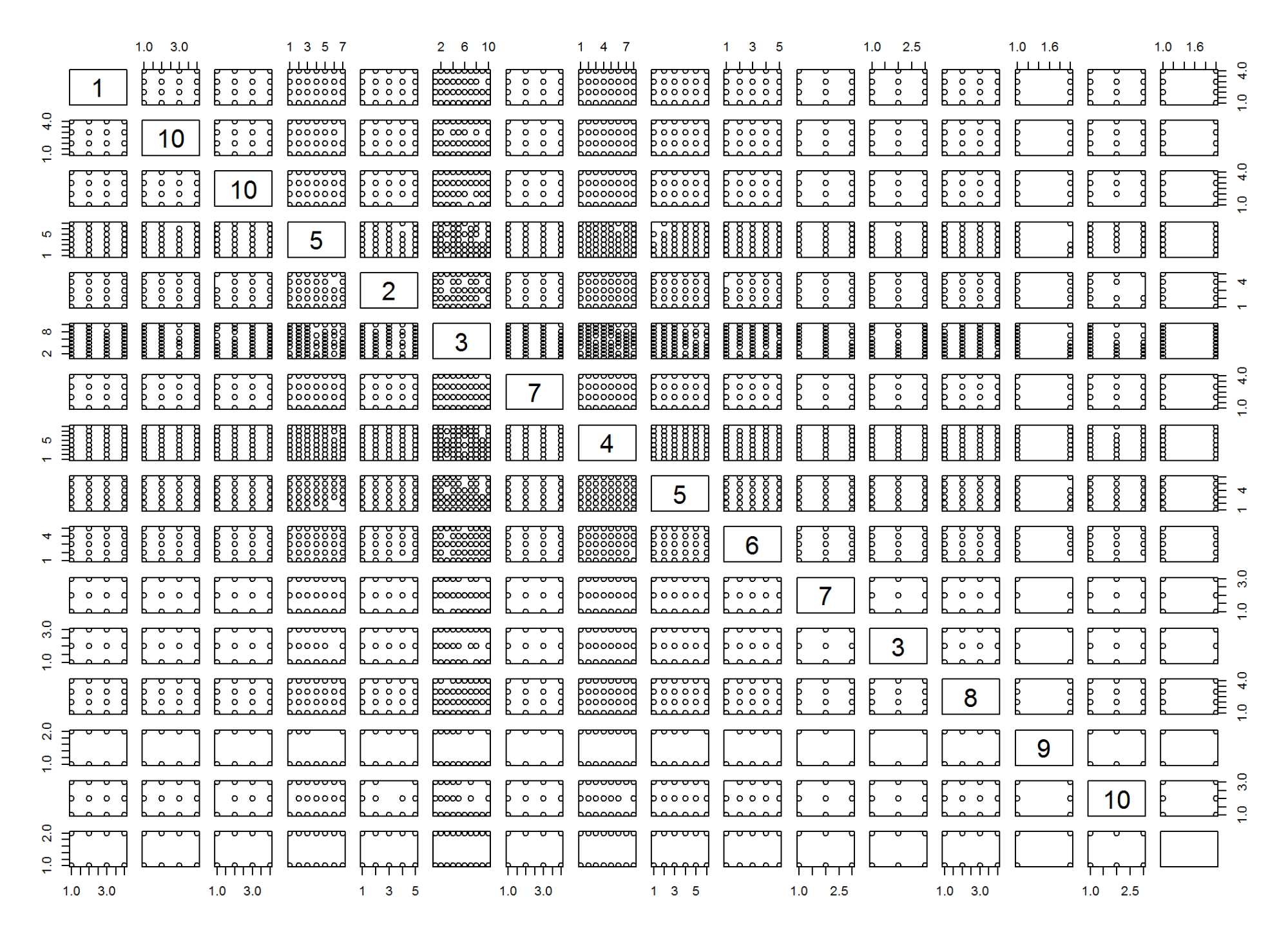
#head(part$scores)

We may also cross check using Kmeansvar clustering

kfit<-kmeansvar(X.quanti = vars\_quanti, X.quali = vars\_quali[,-c(12)], init=10,  
 iter.max = 150, nstart = 1, matsim = TRUE)  
summary(kfit)

##   
## Call:  
## kmeansvar(X.quanti = vars\_quanti, X.quali = vars\_quali[, -c(12)], init = 10, iter.max = 150, nstart = 1, matsim = TRUE)  
##   
## number of iterations: 1  
##   
## Data:   
## number of observations: 1000  
## number of variables: 15  
## number of numerical variables: 1  
## number of categorical variables: 14  
## number of clusters: 10  
##   
## Cluster 1 :   
## squared loading  
## X.quanti 1  
##   
##   
## Cluster 2 :   
## squared loading  
## savings\_ac\_bond\_6 1  
##   
##   
## Cluster 3 :   
## squared loading  
## purpose\_4 0.59  
## other\_installment\_type\_14 0.59  
##   
##   
## Cluster 4 :   
## squared loading  
## age\_in\_yrs\_13 1  
##   
##   
## Cluster 5 :   
## squared loading  
## duration\_month\_2 0.82  
## credit\_amount\_5 0.82  
##   
##   
## Cluster 6 :   
## squared loading  
## p\_employment\_since\_7 1  
##   
##   
## Cluster 7 :   
## squared loading  
## property\_type\_12 0.89  
## housing\_type\_15 0.89  
##   
##   
## Cluster 8 :   
## squared loading  
## foreign\_worker\_20 1  
##   
##   
## Cluster 9 :   
## squared loading  
## other\_debtors\_or\_grantors\_10 1  
##   
##   
## Cluster 10 :   
## squared loading  
## chk\_ac\_status\_1 0.54  
## credit\_history\_3 0.40  
## good\_bad\_21 0.60  
##   
##   
## Gain in cohesion (in %): 76.76

plot(cbind(vars\_quanti, vars\_quali), as.factor(kfit$cluster))



kfit$E

## [1] 76.76116

We will model first ten tip labels from the varclus:

* 1. duration\_month\_2
  2. age\_in\_yrs\_13
  3. credit\_amount\_5
  4. installment\_pct\_disp\_inc\_8
  5. chk\_ac\_status\_1
  6. credit\_history\_3
  7. savings\_ac\_bond\_6
  8. purpose\_4
  9. property\_type\_12
  10. p\_employment\_since\_7

2.2.22.1 Subset data -2

keep<- c(1:8,12,13,21)  
cdata\_reduced\_2 <- cdata[,keep]  
str(cdata\_reduced\_2)

## 'data.frame': 1000 obs. of 11 variables:  
## $ chk\_ac\_status\_1 : Factor w/ 4 levels "A11","A12","A13",..: 1 2 4 1 1 4 4 2 4 2 ...  
## $ duration\_month\_2 : Factor w/ 7 levels "00-06","06-12",..: 1 7 2 6 3 5 3 5 2 4 ...  
## $ credit\_history\_3 : Factor w/ 4 levels "01.A30","02.A31",..: 4 3 4 3 3 3 3 3 3 4 ...  
## $ purpose\_4 : Factor w/ 10 levels "A40","A41","A410",..: 5 5 8 4 1 8 4 2 5 1 ...  
## $ credit\_amount\_5 : Factor w/ 6 levels "0-1400","1400-2500",..: 1 6 2 6 5 6 3 6 3 5 ...  
## $ savings\_ac\_bond\_6 : Factor w/ 5 levels "A61","A62","A63",..: 5 1 1 1 1 5 3 1 4 1 ...  
## $ p\_employment\_since\_7 : Factor w/ 5 levels "A71","A72","A73",..: 5 3 4 4 3 3 5 3 4 1 ...  
## $ installment\_pct\_disp\_inc\_8: num 4 2 2 2 3 2 3 2 2 4 ...  
## $ property\_type\_12 : Factor w/ 4 levels "A121","A122",..: 1 1 1 2 4 4 2 3 1 3 ...  
## $ age\_in\_yrs\_13 : Factor w/ 8 levels "0-25","25-30",..: 8 1 6 5 7 3 7 3 8 2 ...  
## $ good\_bad\_21 : Factor w/ 2 levels "Bad","Good": 2 1 2 2 1 2 2 2 2 1 ...

2.3. Random Sampling (Train and Test)

We may split the data (given population) into random samples with 50-50, 60-40 or 70-30 ratios for **Training** (Development Sample on which model will be developed or trained) and **Test** (validation/holdout sample on which model will be tested) based on population size. In this exercise we will split the sample into 70-30.

2.3.1 Simple Random Sampling

div\_part <- sort(sample(nrow(cdata\_reduced\_2), nrow(cdata\_reduced\_2)\*.7))  
  
#select training sample   
train<-cdata\_reduced\_2[div\_part,] # 70% here  
pct(train$good\_bad\_21)

|  |  |  |
| --- | --- | --- |
|  | Count | Percentage |
| Bad | 220 | 31.43 |
| Good | 480 | 68.57 |

# put remaining into test sample  
test<-cdata\_reduced\_2[-div\_part,] # rest of the 30% data goes here  
pct(test$good\_bad\_21)

|  |  |  |
| --- | --- | --- |
|  | Count | Percentage |
| Bad | 80 | 26.67 |
| Good | 220 | 73.33 |

2.3.2 Stratified Random Sampling

# Required "caret" package  
# considering good\_bad variable as strata  
  
pct(cdata\_reduced\_2$good\_bad\_21)

|  |  |  |
| --- | --- | --- |
|  | Count | Percentage |
| Bad | 300 | 30 |
| Good | 700 | 70 |

div\_part\_1 <- createDataPartition(y = cdata\_reduced\_2$good\_bad\_21, p = 0.7, list = F)  
  
# Training Sample  
train\_1 <- cdata\_reduced\_2[div\_part\_1,] # 70% here  
pct(train\_1$good\_bad\_21)

|  |  |  |
| --- | --- | --- |
|  | Count | Percentage |
| Bad | 210 | 30 |
| Good | 490 | 70 |

# Test Sample  
test\_1 <- cdata\_reduced\_2[-div\_part\_1,] # rest of the 30% data goes here  
pct(test\_1$good\_bad\_21)

|  |  |  |
| --- | --- | --- |
|  | Count | Percentage |
| Bad | 90 | 30 |
| Good | 210 | 70 |

Clearly stratified sampling is more accurate than simple random sampling.

3 Model Development

3.1 Logistic Regression

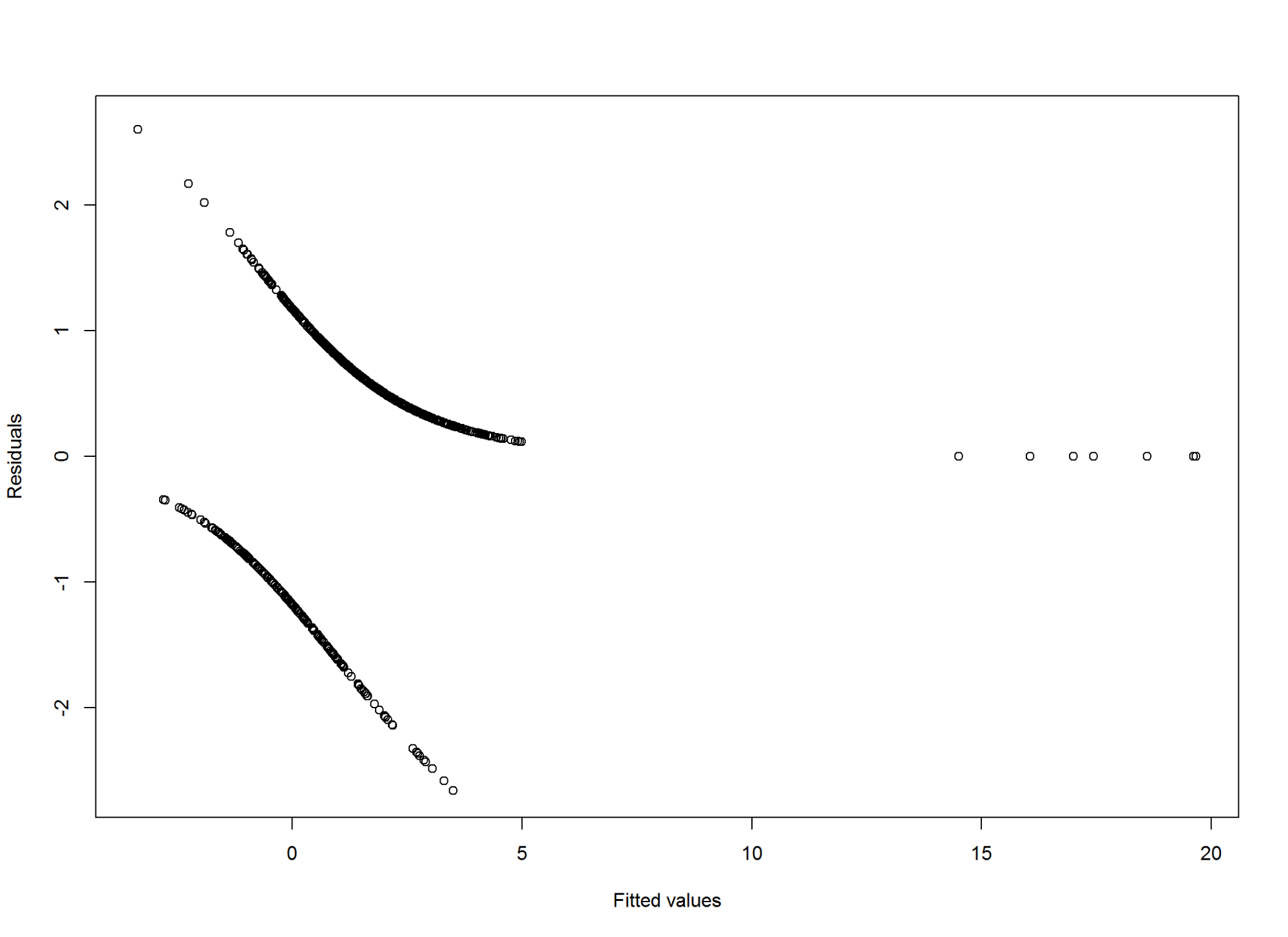
# Logistic Regression Model  
m1<-glm(good\_bad\_21~.,data=train\_1,family=binomial())  
m1<-step(m1)

## Start: AIC=688.86  
## good\_bad\_21 ~ chk\_ac\_status\_1 + duration\_month\_2 + credit\_history\_3 +   
## purpose\_4 + credit\_amount\_5 + savings\_ac\_bond\_6 + p\_employment\_since\_7 +   
## installment\_pct\_disp\_inc\_8 + property\_type\_12 + age\_in\_yrs\_13  
##   
## Df Deviance AIC  
## - property\_type\_12 3 628.62 686.62  
## <none> 624.86 688.86  
## - p\_employment\_since\_7 4 634.45 690.45  
## - credit\_amount\_5 1 628.99 690.99  
## - age\_in\_yrs\_13 1 629.10 691.10  
## - installment\_pct\_disp\_inc\_8 1 630.43 692.43  
## - duration\_month\_2 1 632.48 694.48  
## - purpose\_4 9 650.86 696.86  
## - credit\_history\_3 4 646.09 702.09  
## - savings\_ac\_bond\_6 4 646.28 702.28  
## - chk\_ac\_status\_1 3 684.92 742.92  
##   
## Step: AIC=686.62  
## good\_bad\_21 ~ chk\_ac\_status\_1 + duration\_month\_2 + credit\_history\_3 +   
## purpose\_4 + credit\_amount\_5 + savings\_ac\_bond\_6 + p\_employment\_since\_7 +   
## installment\_pct\_disp\_inc\_8 + age\_in\_yrs\_13  
##   
## Df Deviance AIC  
## <none> 628.62 686.62  
## - age\_in\_yrs\_13 1 632.08 688.08  
## - p\_employment\_since\_7 4 638.88 688.88  
## - credit\_amount\_5 1 633.76 689.76  
## - installment\_pct\_disp\_inc\_8 1 635.31 691.31  
## - purpose\_4 9 653.56 693.56  
## - duration\_month\_2 1 637.94 693.94  
## - savings\_ac\_bond\_6 4 650.19 700.19  
## - credit\_history\_3 4 651.67 701.67  
## - chk\_ac\_status\_1 3 688.68 740.68

summary(m1)

##   
## Call:  
## glm(formula = good\_bad\_21 ~ chk\_ac\_status\_1 + duration\_month\_2 +   
## credit\_history\_3 + purpose\_4 + credit\_amount\_5 + savings\_ac\_bond\_6 +   
## p\_employment\_since\_7 + installment\_pct\_disp\_inc\_8 + age\_in\_yrs\_13,   
## family = binomial(), data = train\_1)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.6591 -0.7195 0.3698 0.6914 2.6031   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -4.792e-01 8.300e-01 -0.577 0.563714   
## chk\_ac\_status\_1A12 3.819e-01 2.514e-01 1.519 0.128782   
## chk\_ac\_status\_1A13 1.133e+00 4.150e-01 2.729 0.006353 \*\*   
## chk\_ac\_status\_1A14 1.896e+00 2.754e-01 6.884 5.82e-12 \*\*\*  
## duration\_month\_2 -3.220e-02 1.065e-02 -3.022 0.002509 \*\*   
## credit\_history\_3A31 -1.098e+00 6.538e-01 -1.680 0.093029 .   
## credit\_history\_3A32 5.126e-01 4.682e-01 1.095 0.273621   
## credit\_history\_3A33 7.664e-01 5.482e-01 1.398 0.162081   
## credit\_history\_3A34 1.103e+00 4.890e-01 2.255 0.024105 \*   
## purpose\_4A41 1.377e+00 4.349e-01 3.165 0.001551 \*\*   
## purpose\_4A410 1.098e+00 9.224e-01 1.191 0.233844   
## purpose\_4A42 5.872e-01 2.978e-01 1.972 0.048657 \*   
## purpose\_4A43 6.872e-01 2.854e-01 2.408 0.016047 \*   
## purpose\_4A44 -5.106e-01 9.574e-01 -0.533 0.593772   
## purpose\_4A45 4.025e-01 6.934e-01 0.580 0.561610   
## purpose\_4A46 -5.325e-01 4.743e-01 -1.123 0.261554   
## purpose\_4A48 1.623e+01 7.303e+02 0.022 0.982267   
## purpose\_4A49 5.292e-01 4.036e-01 1.311 0.189771   
## credit\_amount\_5 -1.146e-04 5.072e-05 -2.260 0.023821 \*   
## savings\_ac\_bond\_6A62 8.459e-02 3.290e-01 0.257 0.797095   
## savings\_ac\_bond\_6A63 1.194e+00 6.403e-01 1.865 0.062131 .   
## savings\_ac\_bond\_6A64 1.444e+00 6.449e-01 2.239 0.025167 \*   
## savings\_ac\_bond\_6A65 1.121e+00 3.155e-01 3.553 0.000381 \*\*\*  
## p\_employment\_since\_7A72 4.357e-02 4.751e-01 0.092 0.926934   
## p\_employment\_since\_7A73 3.950e-01 4.429e-01 0.892 0.372411   
## p\_employment\_since\_7A74 1.077e+00 4.900e-01 2.198 0.027925 \*   
## p\_employment\_since\_7A75 4.458e-01 4.549e-01 0.980 0.327161   
## installment\_pct\_disp\_inc\_8 -2.539e-01 9.936e-02 -2.556 0.010599 \*   
## age\_in\_yrs\_13 1.881e-02 1.024e-02 1.837 0.066276 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 855.21 on 699 degrees of freedom  
## Residual deviance: 628.62 on 671 degrees of freedom  
## AIC: 686.62  
##   
## Number of Fisher Scoring iterations: 15

prob <- predict(m1, type = "response")  
res <- residuals(m1, type = "deviance")  
  
#Plot Residuals  
plot(predict(m1), res,  
 xlab="Fitted values", ylab = "Residuals",  
 ylim = max(abs(res)) \* c(-1,1))



## CIs using profiled log-likelihood  
confint(m1)

## Waiting for profiling to be done...

## 2.5 % 97.5 %  
## (Intercept) -2.119994e+00 1.1415170731  
## chk\_ac\_status\_1A12 -1.096189e-01 0.8774595105  
## chk\_ac\_status\_1A13 3.472578e-01 1.9847943295  
## chk\_ac\_status\_1A14 1.367491e+00 2.4496986367  
## duration\_month\_2 -5.333701e-02 -0.0114784042  
## credit\_history\_3A31 -2.401937e+00 0.1732388341  
## credit\_history\_3A32 -3.979225e-01 1.4488094892  
## credit\_history\_3A33 -2.940089e-01 1.8639791391  
## credit\_history\_3A34 1.529542e-01 2.0794538991  
## purpose\_4A41 5.492183e-01 2.2604903941  
## purpose\_4A410 -6.911907e-01 3.0269883017  
## purpose\_4A42 7.293058e-03 1.1768847341  
## purpose\_4A43 1.304682e-01 1.2512811571  
## purpose\_4A44 -2.504641e+00 1.3193925744  
## purpose\_4A45 -9.148990e-01 1.8595355408  
## purpose\_4A46 -1.464795e+00 0.4028982921  
## purpose\_4A48 -4.900114e+01 NA  
## purpose\_4A49 -2.501200e-01 1.3373434208  
## credit\_amount\_5 -2.151751e-04 -0.0000155696  
## savings\_ac\_bond\_6A62 -5.515885e-01 0.7424391052  
## savings\_ac\_bond\_6A63 6.539219e-02 2.6177531459  
## savings\_ac\_bond\_6A64 2.874101e-01 2.8627731769  
## savings\_ac\_bond\_6A65 5.204425e-01 1.7615316107  
## p\_employment\_since\_7A72 -8.900684e-01 0.9786828164  
## p\_employment\_since\_7A73 -4.771549e-01 1.2657141733  
## p\_employment\_since\_7A74 1.195437e-01 2.0468690959  
## p\_employment\_since\_7A75 -4.533117e-01 1.3363877234  
## installment\_pct\_disp\_inc\_8 -4.512246e-01 -0.0610506239  
## age\_in\_yrs\_13 -9.907029e-04 0.0392460053

## CIs using standard errors  
confint.default(m1)

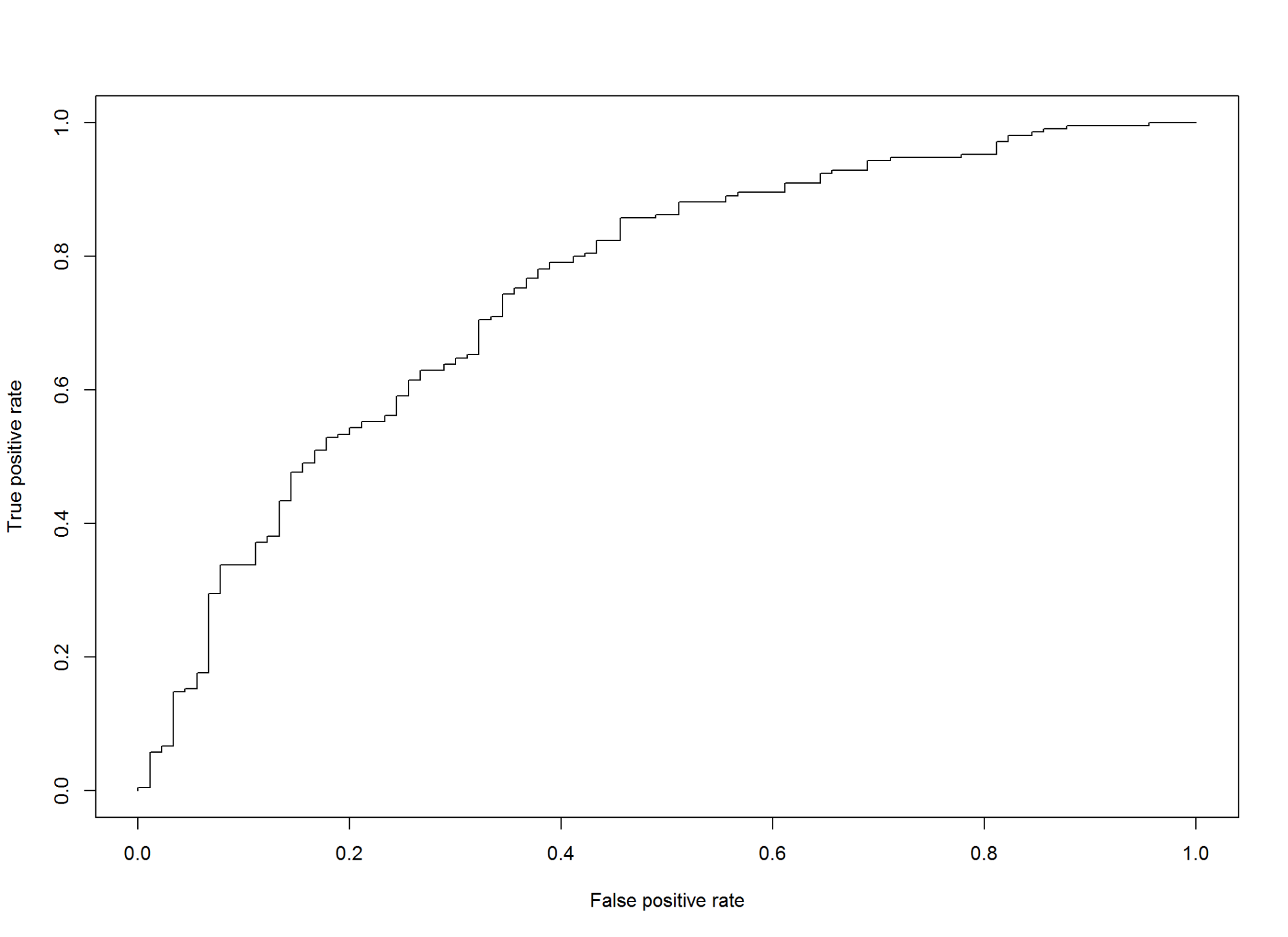
## 2.5 % 97.5 %  
## (Intercept) -2.105964e+00 1.147587e+00  
## chk\_ac\_status\_1A12 -1.108892e-01 8.746777e-01  
## chk\_ac\_status\_1A13 3.191519e-01 1.945968e+00  
## chk\_ac\_status\_1A14 1.355955e+00 2.435428e+00  
## duration\_month\_2 -5.307684e-02 -1.131664e-02  
## credit\_history\_3A31 -2.379746e+00 1.832932e-01  
## credit\_history\_3A32 -4.050823e-01 1.430202e+00  
## credit\_history\_3A33 -3.080110e-01 1.840895e+00  
## credit\_history\_3A34 1.444749e-01 2.061124e+00  
## purpose\_4A41 5.240773e-01 2.228982e+00  
## purpose\_4A410 -7.097615e-01 2.906094e+00  
## purpose\_4A42 3.460649e-03 1.170987e+00  
## purpose\_4A43 1.278348e-01 1.246649e+00  
## purpose\_4A44 -2.387067e+00 1.365780e+00  
## purpose\_4A45 -9.565221e-01 1.761466e+00  
## purpose\_4A46 -1.462096e+00 3.970959e-01  
## purpose\_4A48 -1.415097e+03 1.447562e+03  
## purpose\_4A49 -2.618018e-01 1.320183e+00  
## credit\_amount\_5 -2.140488e-04 -1.521882e-05  
## savings\_ac\_bond\_6A62 -5.602364e-01 7.294130e-01  
## savings\_ac\_bond\_6A63 -6.057341e-02 2.449333e+00  
## savings\_ac\_bond\_6A64 1.798357e-01 2.707769e+00  
## savings\_ac\_bond\_6A65 5.025098e-01 1.739214e+00  
## p\_employment\_since\_7A72 -8.875826e-01 9.747164e-01  
## p\_employment\_since\_7A73 -4.729749e-01 1.263008e+00  
## p\_employment\_since\_7A74 1.168030e-01 2.037553e+00  
## p\_employment\_since\_7A75 -4.458900e-01 1.337427e+00  
## installment\_pct\_disp\_inc\_8 -4.486712e-01 -5.918639e-02  
## age\_in\_yrs\_13 -1.263775e-03 3.887796e-02

#  
## odds ratios and 95% CI  
exp(cbind(OR = coef(m1), confint(m1)))

## Waiting for profiling to be done...

## OR 2.5 % 97.5 %  
## (Intercept) 6.192858e-01 1.200324e-01 3.1315155  
## chk\_ac\_status\_1A12 1.465057e+00 8.961756e-01 2.4047826  
## chk\_ac\_status\_1A13 3.103591e+00 1.415181e+00 7.2775505  
## chk\_ac\_status\_1A14 6.657150e+00 3.925491e+00 11.5848549  
## duration\_month\_2 9.683161e-01 9.480605e-01 0.9885872  
## credit\_history\_3A31 3.334620e-01 9.054243e-02 1.1891501  
## credit\_history\_3A32 1.669560e+00 6.717141e-01 4.2580423  
## credit\_history\_3A33 2.152096e+00 7.452699e-01 6.4493486  
## credit\_history\_3A34 3.012588e+00 1.165272e+00 8.0000989  
## purpose\_4A41 3.961131e+00 1.731899e+00 9.5877898  
## purpose\_4A410 2.998663e+00 5.009792e-01 20.6349925  
## purpose\_4A42 1.798987e+00 1.007320e+00 3.2442517  
## purpose\_4A43 1.988225e+00 1.139362e+00 3.4948175  
## purpose\_4A44 6.001090e-01 8.170496e-02 3.7411482  
## purpose\_4A45 1.495517e+00 4.005571e-01 6.4207539  
## purpose\_4A46 5.871351e-01 2.311253e-01 1.4961547  
## purpose\_4A48 1.120952e+07 5.236931e-22 NA  
## purpose\_4A49 1.697557e+00 7.787073e-01 3.8089114  
## credit\_amount\_5 9.998854e-01 9.997848e-01 0.9999844  
## savings\_ac\_bond\_6A62 1.088269e+00 5.760341e-01 2.1010540  
## savings\_ac\_bond\_6A63 3.301509e+00 1.067578e+00 13.7048961  
## savings\_ac\_bond\_6A64 4.236774e+00 1.332971e+00 17.5100180  
## savings\_ac\_bond\_6A65 3.067496e+00 1.682772e+00 5.8213466  
## p\_employment\_since\_7A72 1.044530e+00 4.106277e-01 2.6609490  
## p\_employment\_since\_7A73 1.484409e+00 6.205464e-01 3.5456240  
## p\_employment\_since\_7A74 2.936381e+00 1.126982e+00 7.7436186  
## p\_employment\_since\_7A75 1.561690e+00 6.355200e-01 3.8052730  
## installment\_pct\_disp\_inc\_8 7.757470e-01 6.368478e-01 0.9407756  
## age\_in\_yrs\_13 1.018985e+00 9.990098e-01 1.0400263

#score test data set  
test\_1$m1\_score<-predict(m1,type='response',test\_1)  
m1\_pred<-prediction(test\_1$m1\_score, test\_1$good\_bad\_21)  
m1\_perf <- performance(m1\_pred,"tpr","fpr")  
plot(m1\_perf)



#KS  
m1\_KS<-max(attr(m1\_perf,'y.values')[[1]]-attr(m1\_perf,'x.values')[[1]])\*100  
m1\_KS

## [1] 40.31746

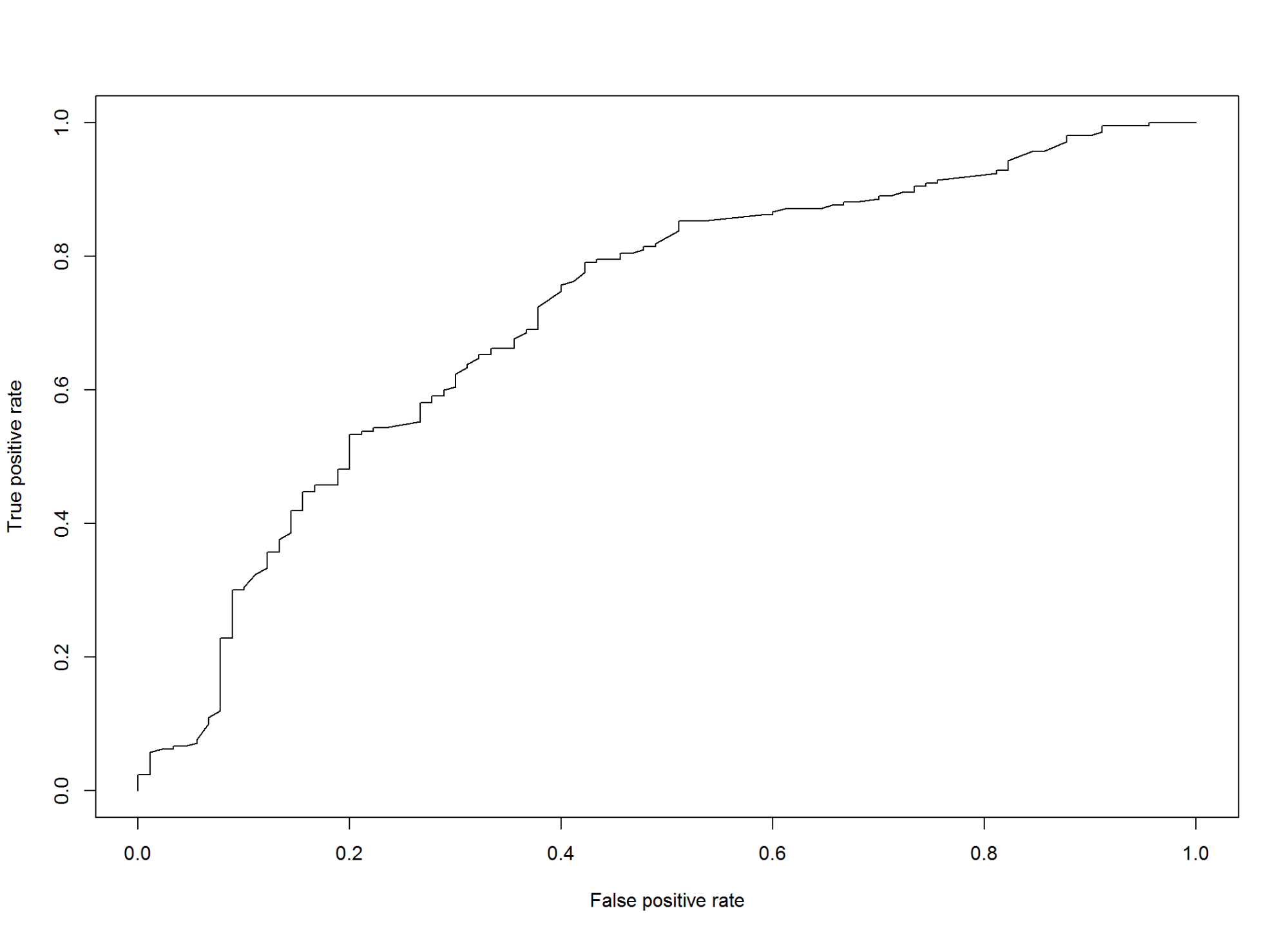
# Cross Validatio  
#load Data Analysis And Graphics Package for R (DAAG)  
#library(DAAG)  
#calculate accuracy over 100 random folds of data for simple logit  
m1\_h<-CVbinary(obj=m1, rand=NULL, nfolds=100, print.details=TRUE)

##   
## Fold: 66 78 9 91 79 93 35 75 85 21 82 52 57 15 19 86 16 48 55 99 92 31 84 100  
## 22 90 71 17 46 42 60  
## 81 51 3 29 72 94 59 69 40  
## 68 70 37 89 7 77 39 80  
## 45 24 27 6 47 49 34 8 67 50 36 44 53 28 95 58 83 76 41 26 18 20  
## 4 96 5 74 38 56 33 10 11 43 62 64 14 2 97 65 87 88 23 54 63 30 98 12 13 1 25 73 61 32  
## Internal estimate of accuracy = 0.779  
## Cross-validation estimate of accuracy = 0.751

m1\_1<-glm(good\_bad\_21~chk\_ac\_status\_1+duration\_month\_2  
 +savings\_ac\_bond\_6+installment\_pct\_disp\_inc\_8,  
 data=train\_1,family=binomial())  
  
summary(m1\_1)

##   
## Call:  
## glm(formula = good\_bad\_21 ~ chk\_ac\_status\_1 + duration\_month\_2 +   
## savings\_ac\_bond\_6 + installment\_pct\_disp\_inc\_8, family = binomial(),   
## data = train\_1)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.5209 -0.9211 0.4523 0.8000 1.7616   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 1.064255 0.341518 3.116 0.001832 \*\*   
## chk\_ac\_status\_1A12 0.324719 0.225514 1.440 0.149894   
## chk\_ac\_status\_1A13 1.131316 0.380268 2.975 0.002929 \*\*   
## chk\_ac\_status\_1A14 2.066268 0.257707 8.018 1.08e-15 \*\*\*  
## duration\_month\_2 -0.037833 0.007796 -4.853 1.22e-06 \*\*\*  
## savings\_ac\_bond\_6A62 -0.103309 0.301677 -0.342 0.732013   
## savings\_ac\_bond\_6A63 1.191392 0.560246 2.127 0.033457 \*   
## savings\_ac\_bond\_6A64 1.136891 0.574750 1.978 0.047922 \*   
## savings\_ac\_bond\_6A65 0.966230 0.285263 3.387 0.000706 \*\*\*  
## installment\_pct\_disp\_inc\_8 -0.140427 0.084984 -1.652 0.098455 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 855.21 on 699 degrees of freedom  
## Residual deviance: 696.28 on 690 degrees of freedom  
## AIC: 716.28  
##   
## Number of Fisher Scoring iterations: 5

test\_1$m1\_1\_score<-predict(m1\_1,type='response',test\_1)  
m1\_1\_pred<-prediction(test\_1$m1\_1\_score,test\_1$good\_bad\_21)  
m1\_1\_perf <- performance(m1\_1\_pred,"tpr","fpr")  
  
plot(m1\_1\_perf)



AUCRF=performance(m1\_pred, measure = "auc")@y.values[[1]]  
cat("AUC: ",AUCRF,"\n")

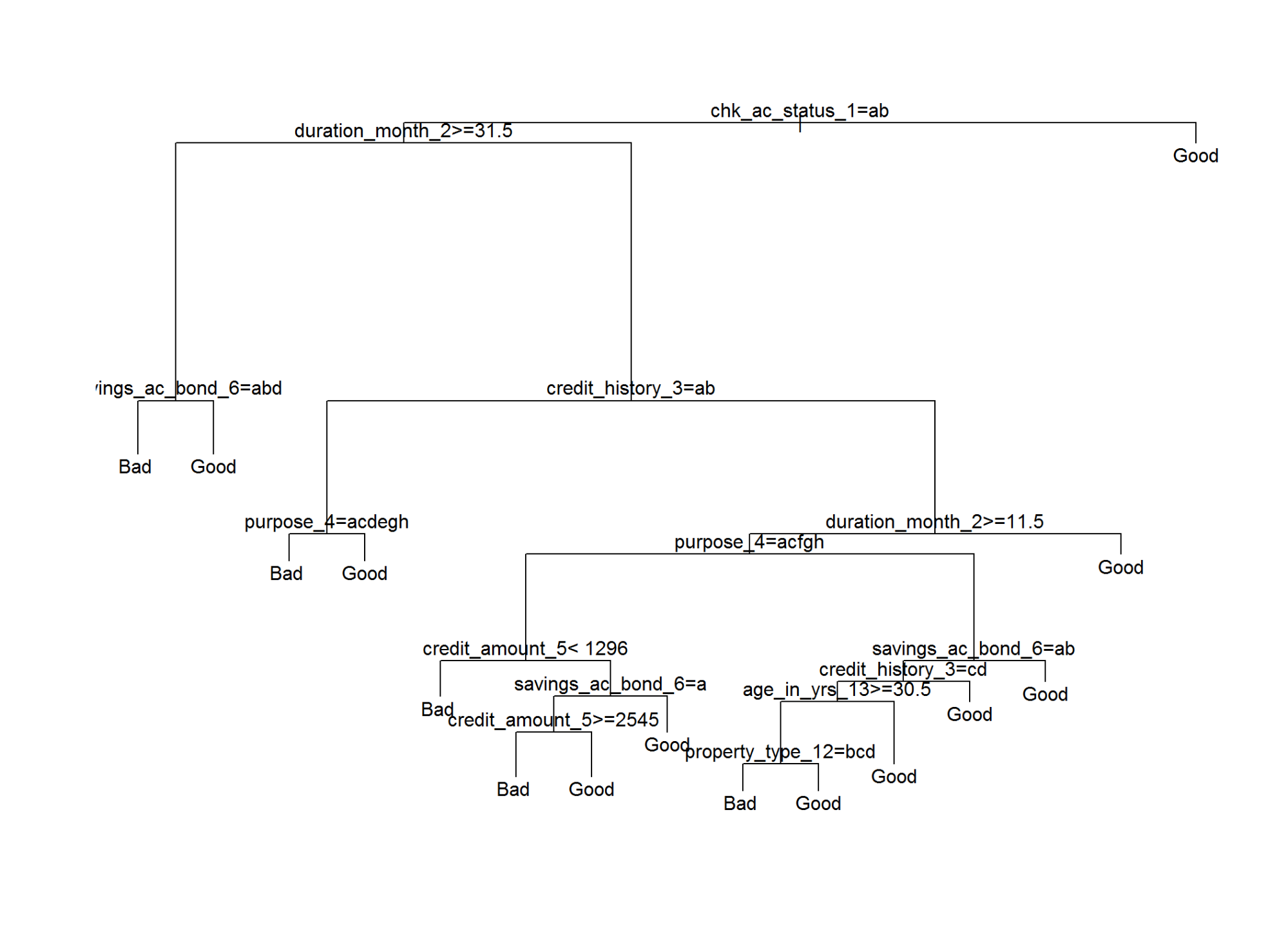
## AUC: 0.7506878

AUCRF=performance(m1\_1\_pred, measure = "auc")@y.values[[1]]  
cat("AUC: ",AUCRF,"\n")

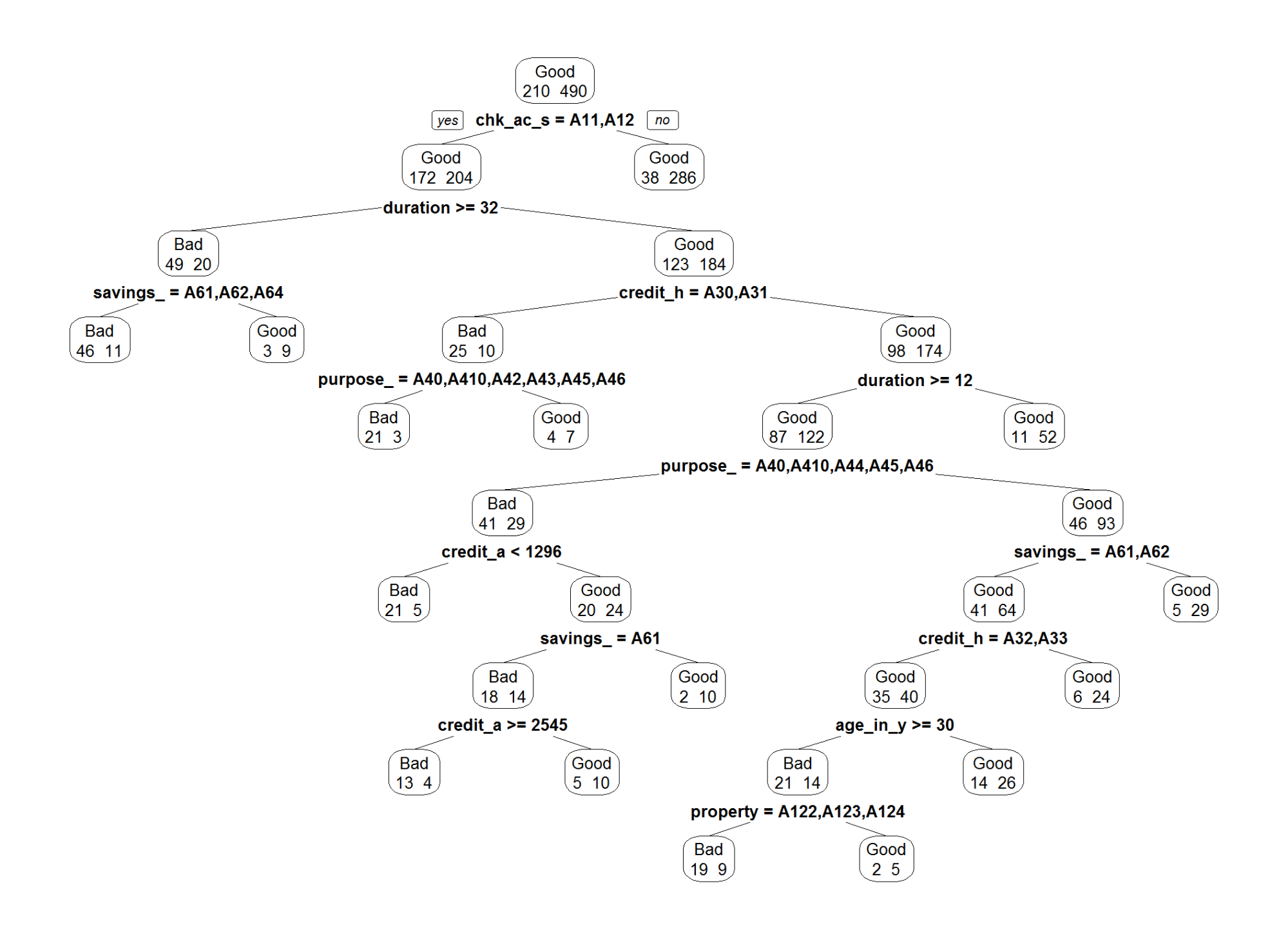
## AUC: 0.7136772

3.2 Using Bayesian N Using Traditional recursive Partitioning

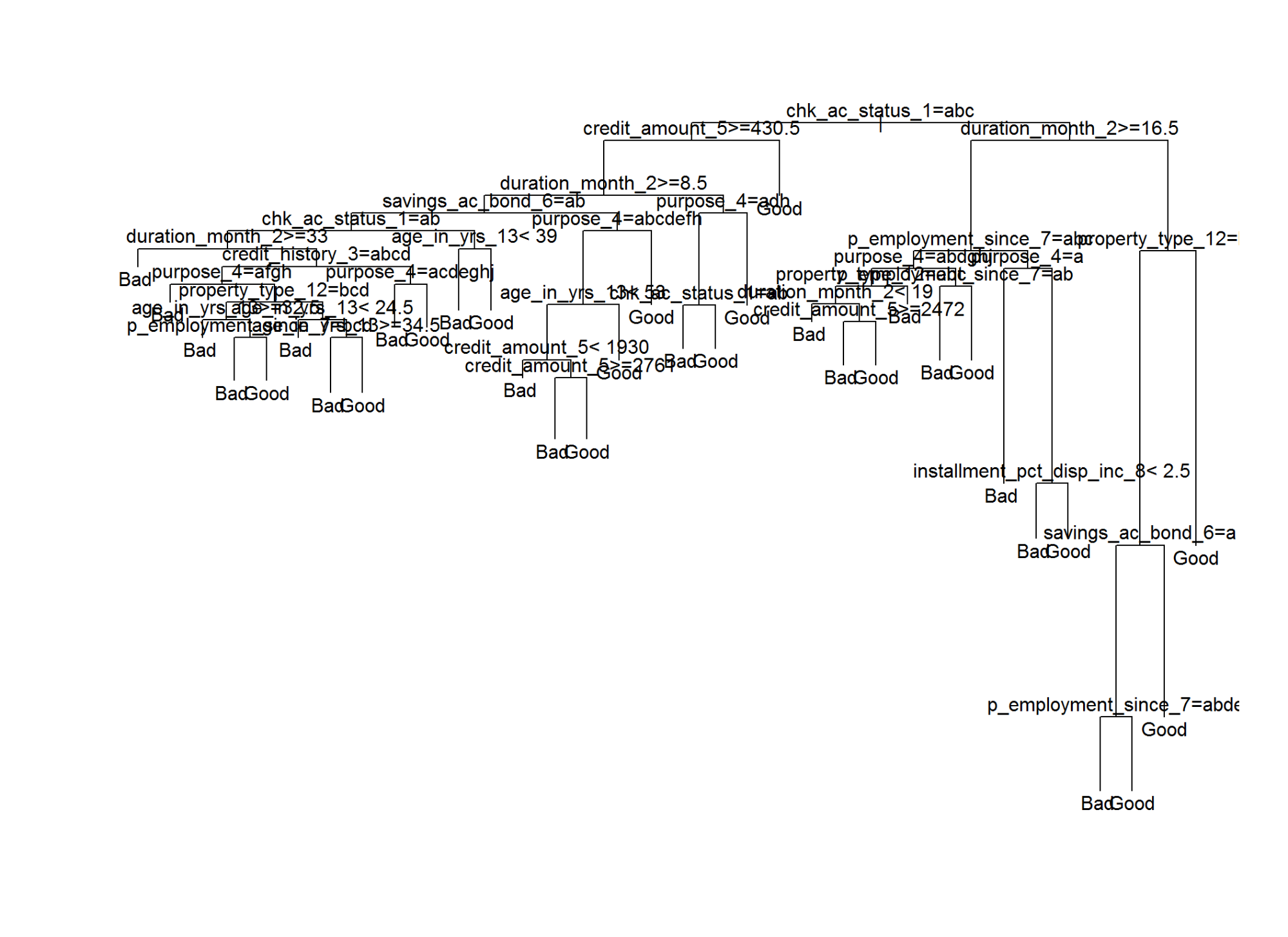
m2 <- rpart(good\_bad\_21~.,data=train\_1)  
plot(m2);text(m2);



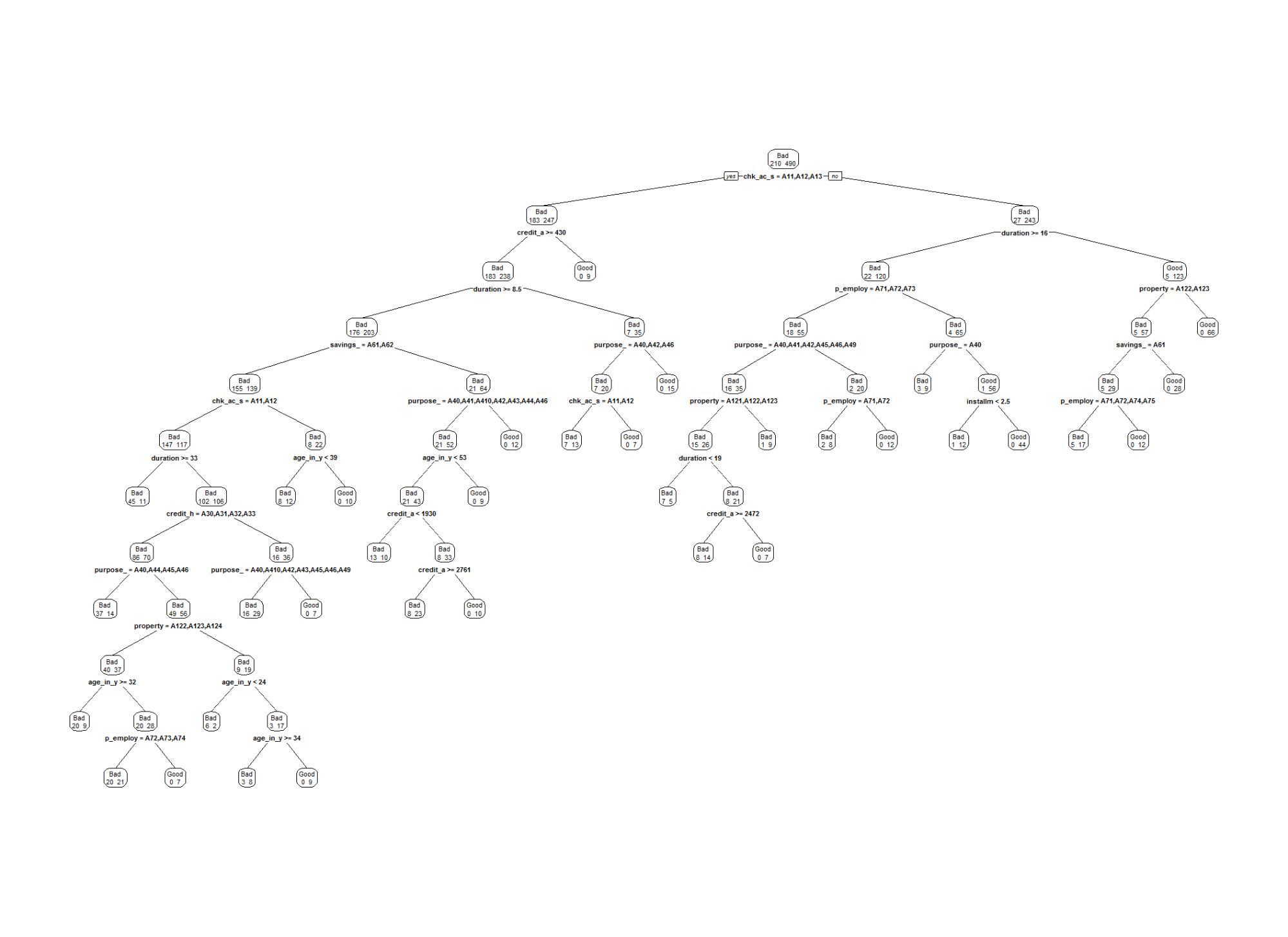
prp(m2,type=2,extra=1)



#score test data  
test\_1$m2\_score <- predict(m2,type='prob',test\_1)  
m2\_pred <- prediction(test\_1$m2\_score[,2],test\_1$good\_bad\_21)  
m2\_perf <- performance(m2\_pred,"tpr","fpr")  
  
#build model using 90% 10% priors  
#with smaller complexity parameter to allow more complextrees  
# for tuning complexity vs. pruning see Thernau 1997  
m2\_1<-rpart(good\_bad\_21~.,data=train\_1,parms=list(prior=c(.9,.1)),cp=.0002)  
plot(m2\_1);text(m2\_1);



prp(m2\_1,type=2,extra=1)



test\_1$m2\_1\_score <- predict(m2\_1,type='prob',test\_1)  
  
m2\_1\_pred<-prediction(test\_1$m2\_1\_score[,2],test\_1$good\_bad\_21)  
m2\_1\_perf<- performance(m2\_1\_pred,"tpr","fpr")  
  
AUCRF=performance(m2\_pred, measure = "auc")@y.values[[1]]  
cat("AUC: ",AUCRF,"\n")

## AUC: 0.698836

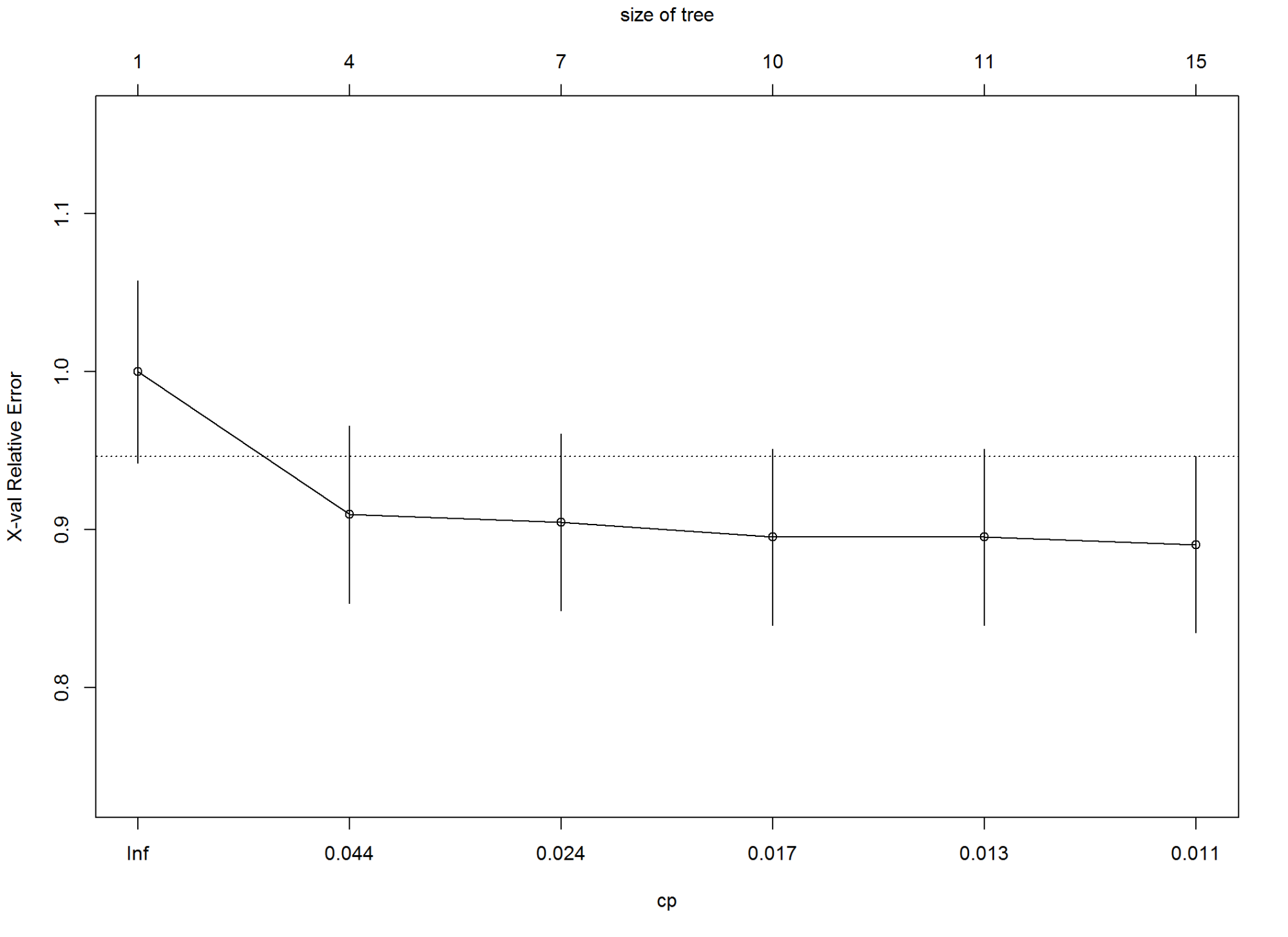
AUCRF=performance(m2\_1\_pred, measure = "auc")@y.values[[1]]  
cat("AUC: ",AUCRF,"\n")

## AUC: 0.659709

#prints complexity and out of sample error  
printcp(m2)

##   
## Classification tree:  
## rpart(formula = good\_bad\_21 ~ ., data = train\_1)  
##   
## Variables actually used in tree construction:  
## [1] age\_in\_yrs\_13 chk\_ac\_status\_1 credit\_amount\_5 credit\_history\_3 duration\_month\_2 property\_type\_12 purpose\_4 savings\_ac\_bond\_6  
##   
## Root node error: 210/700 = 0.3  
##   
## n= 700   
##   
## CP nsplit rel error xerror xstd  
## 1 0.069048 0 1.00000 1.00000 0.057735  
## 2 0.028571 3 0.79048 0.90952 0.056119  
## 3 0.020635 6 0.70476 0.90476 0.056027  
## 4 0.014286 9 0.64286 0.89524 0.055840  
## 5 0.011111 10 0.62857 0.89524 0.055840  
## 6 0.010000 14 0.58095 0.89048 0.055746

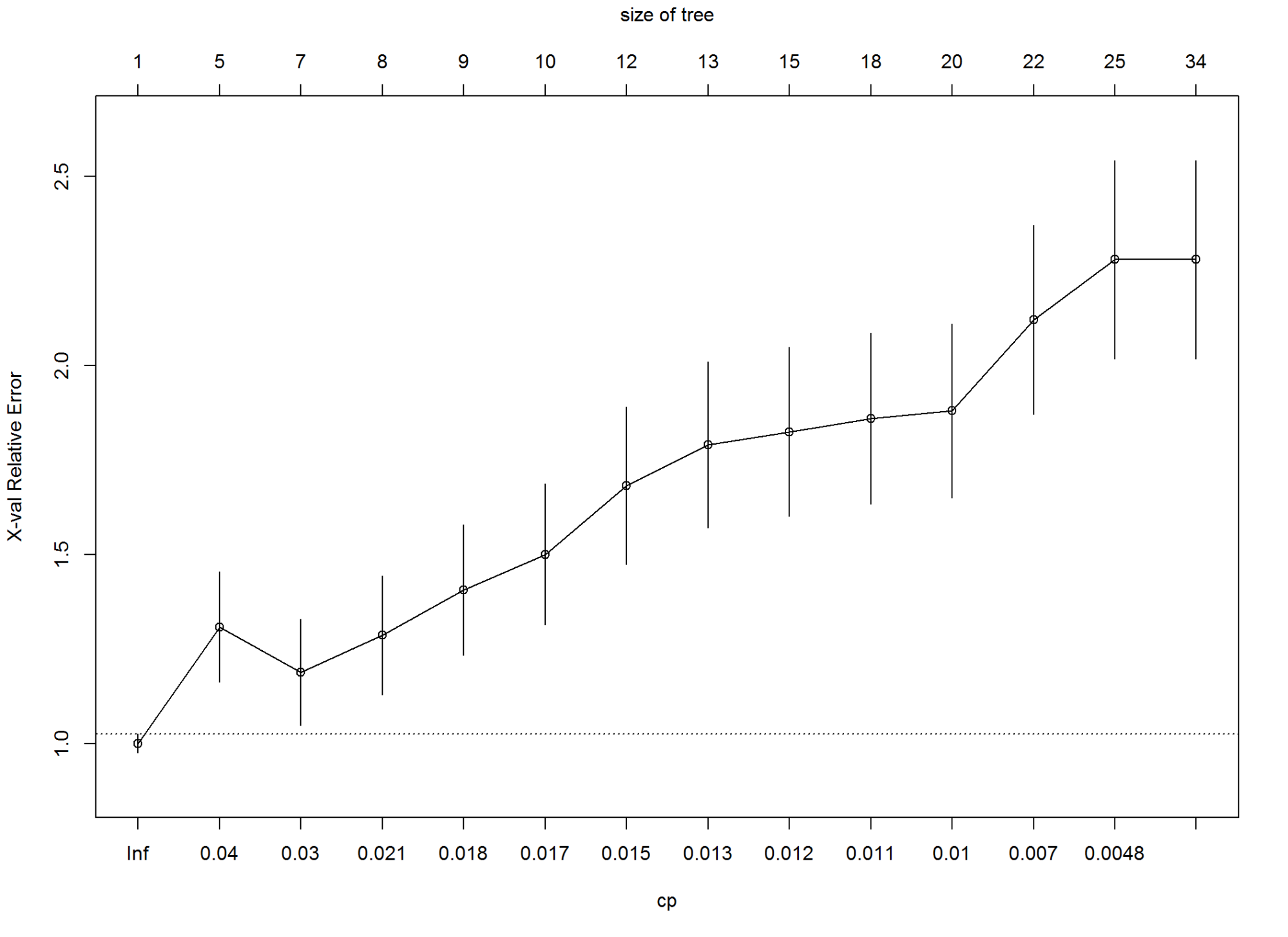
#plots complexity vs. error  
plotcp(m2)



#prints complexity and out of sample error  
printcp(m2\_1)

##   
## Classification tree:  
## rpart(formula = good\_bad\_21 ~ ., data = train\_1, parms = list(prior = c(0.9,   
## 0.1)), cp = 2e-04)  
##   
## Variables actually used in tree construction:  
## [1] age\_in\_yrs\_13 chk\_ac\_status\_1 credit\_amount\_5 credit\_history\_3 duration\_month\_2 installment\_pct\_disp\_inc\_8 p\_employment\_since\_7   
## [8] property\_type\_12 purpose\_4 savings\_ac\_bond\_6   
##   
## Root node error: 70/700 = 0.1  
##   
## n= 700   
##   
## CP nsplit rel error xerror xstd  
## 1 0.0448980 0 1.00000 1.0000 0.024744  
## 2 0.0357143 4 0.80816 1.3082 0.145658  
## 3 0.0244898 6 0.73673 1.1878 0.140181  
## 4 0.0183673 7 0.71224 1.2857 0.157328  
## 5 0.0183673 8 0.69388 1.4061 0.172577  
## 6 0.0153061 9 0.67551 1.5000 0.186634  
## 7 0.0142857 11 0.64490 1.6816 0.207663  
## 8 0.0122449 12 0.63061 1.7898 0.219217  
## 9 0.0122449 14 0.60612 1.8245 0.222928  
## 10 0.0102041 17 0.56327 1.8592 0.226571  
## 11 0.0102041 19 0.54286 1.8796 0.230253  
## 12 0.0047619 21 0.52245 2.1204 0.250255  
## 13 0.0047619 24 0.50816 2.2796 0.262581  
## 14 0.0002000 33 0.46122 2.2796 0.262581

#plots complexity vs. error  
plotcp(m2\_1)



#KS m1  
m2\_KS<-max(attr(m2\_perf,'y.values')[[1]]-attr(m2\_perf,'x.values')[[1]])\*100  
m2\_KS

## [1] 32.69841

#KS m2  
m2\_1\_KS<-max(attr(m2\_1\_perf,'y.values')[[1]]-attr(m2\_1\_perf,'x.values')[[1]])\*100  
m2\_1\_KS

## [1] 28.09524

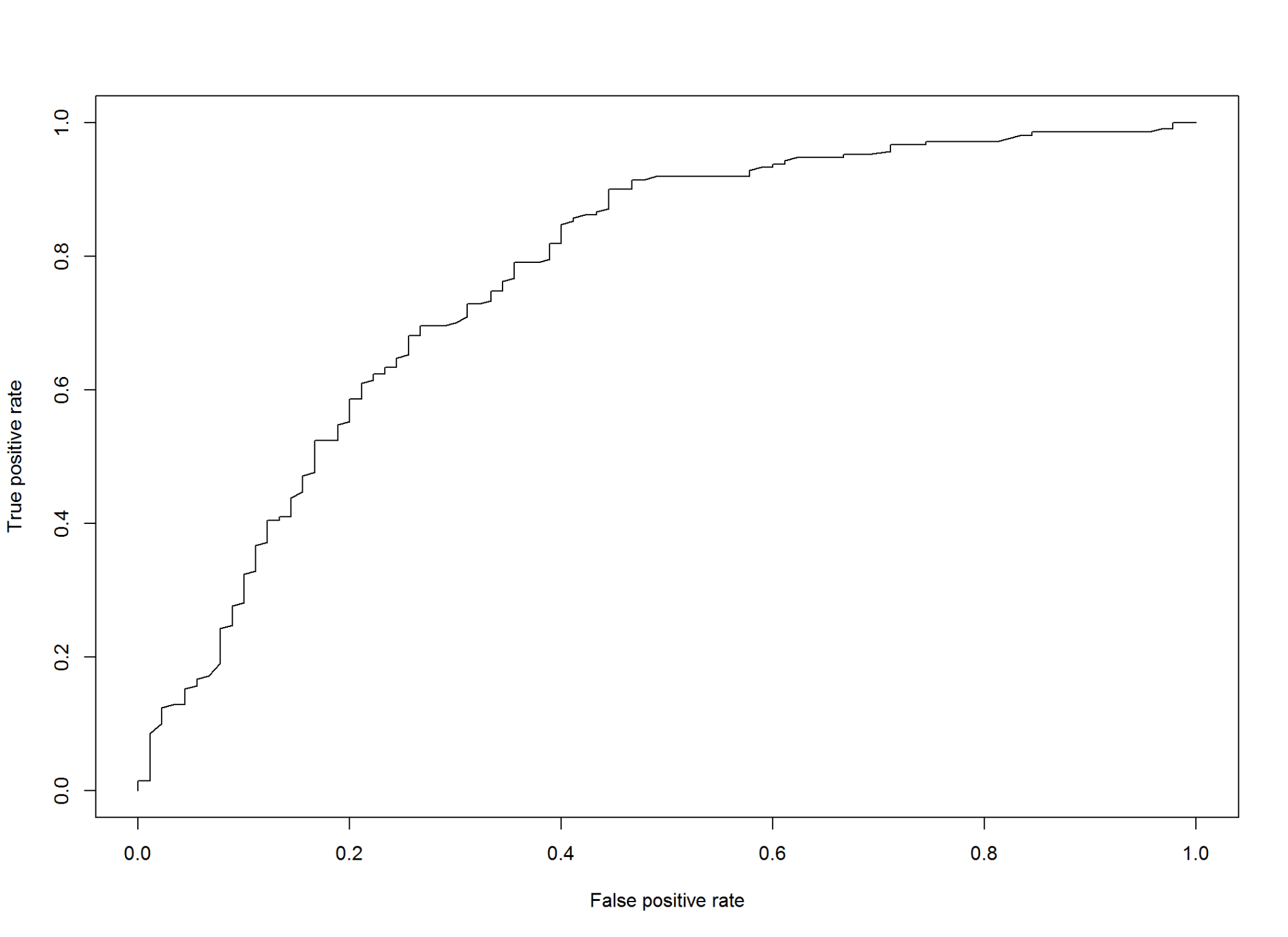
Print tree rules

#print rules for all classes  
#rpart.lists(m2)  
#rpart.rules(m2)  
#rpart.lists(m2\_1)  
#rpart.rules.table(m2\_1)

3.3 Random Forest

3.3.1 General Randmon Forest

m3 <- randomForest(good\_bad\_21 ~ ., data =train\_1)  
m3\_fitForest <- predict(m3, newdata=test\_1, type="prob")[,2]  
m3\_pred <- prediction( m3\_fitForest, test\_1$good\_bad\_21)  
m3\_perf <- performance(m3\_pred, "tpr", "fpr")  
plot(m3\_perf)



#plot variable importance  
varImpPlot(m3, main="Random Forest: Variable Importance")



# Model Performance  
m3\_AUCRF <- performance(m3\_pred, measure = "auc")@y.values[[1]]  
cat("AUC: ",m3\_AUCRF,"\n")

## AUC: 0.771455

#KS m3  
m3\_KS<-max(attr(m3\_perf,'y.values')[[1]]-attr(m3\_perf,'x.values')[[1]])\*100  
m3\_KS

## [1] 45.55556

3.3.2 Conditional Random Forest

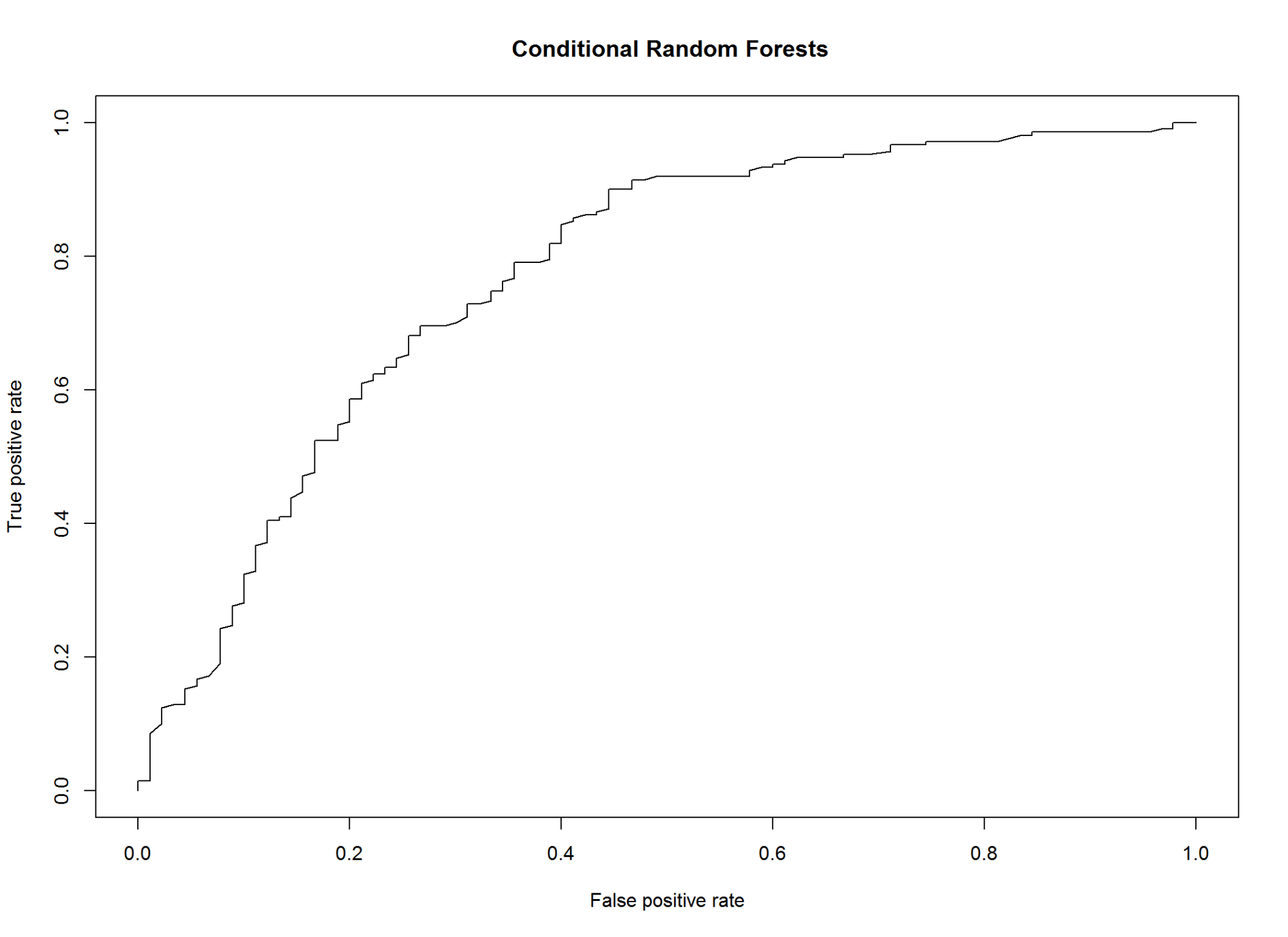
#library(party)  
  
set.seed(42)  
m3\_1<-cforest(good\_bad\_21~.,control = cforest\_unbiased(mtry = 2, ntree = 50), data=train\_1)  
  
# Variable Importance  
kable(as.data.frame(varimp(m3\_1)))

|  |  |
| --- | --- |
|  | varimp(m3\_1) |
| chk\_ac\_status\_1 | 0.0340856 |
| duration\_month\_2 | 0.0237354 |
| credit\_history\_3 | 0.0092607 |
| purpose\_4 | 0.0070817 |
| credit\_amount\_5 | 0.0060700 |
| savings\_ac\_bond\_6 | 0.0140078 |
| p\_employment\_since\_7 | -0.0004669 |
| installment\_pct\_disp\_inc\_8 | 0.0003891 |
| property\_type\_12 | -0.0010117 |
| age\_in\_yrs\_13 | 0.0010895 |

# Model Summary  
summary(m3\_1)

## Length Class Mode   
## 1 RandomForest S4

# Model Performance  
m3\_1\_fitForest <- predict(m3, newdata=test\_1, type="prob")[,2]  
m3\_1\_pred <- prediction(m3\_1\_fitForest, test\_1$good\_bad\_21)  
m3\_1\_perf <- performance(m3\_1\_pred, "tpr", "fpr")  
  
# Model Performance Plot  
plot(m3\_1\_perf, main = " Conditional Random Forests")



# AUC  
m3\_1\_AUCRF <- performance(m3\_1\_pred, measure = "auc")@y.values[[1]]  
cat("AUC: ",m3\_1\_AUCRF,"\n")

## AUC: 0.771455

#KS m3  
m3\_1\_KS<-max(attr(m3\_perf,'y.values')[[1]]-attr(m3\_perf,'x.values')[[1]])\*100  
m3\_1\_KS

## [1] 45.55556

3.3.3 Improve Logistic Results using Random Forest

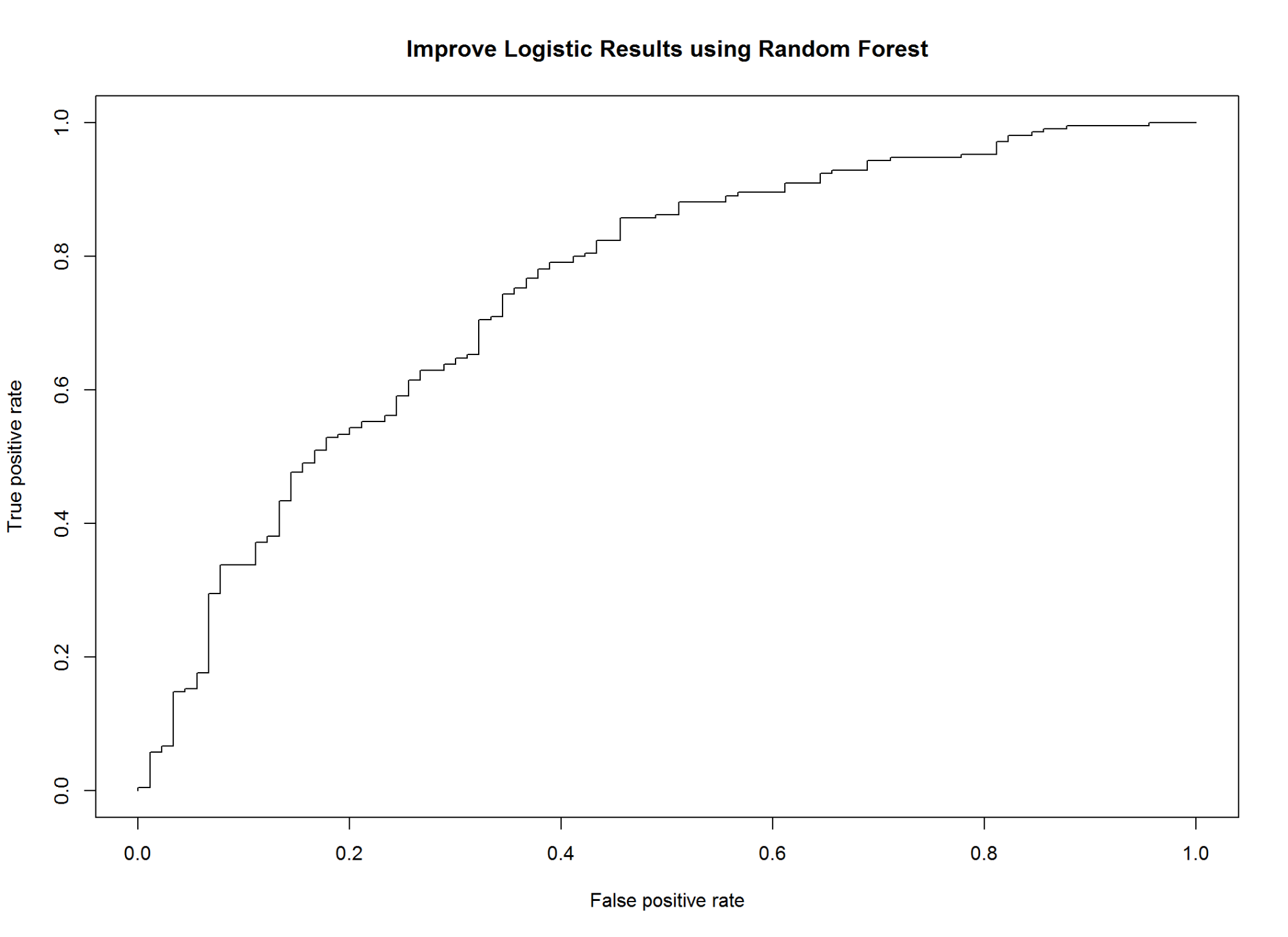
#model based on trial and error based on random forest variable importance  
#m3\_2<-glm(good\_bad\_21~.+credit\_history\_3:p\_employment\_since\_7+ credit\_history\_3:installment\_pct\_disp\_inc\_8  
# +chk\_ac\_status\_1:p\_employment\_since\_7 +chk\_ac\_status\_1:purpose\_4  
# + duration\_month\_2:credit\_amount\_5, data=train\_1,family=binomial())  
  
m3\_2<-glm(good\_bad\_21~.+credit\_history\_3:p\_employment\_since\_7  
 + credit\_history\_3:age\_in\_yrs\_13  
 + chk\_ac\_status\_1:p\_employment\_since\_7  
 + chk\_ac\_status\_1:savings\_ac\_bond\_6  
 + duration\_month\_2:purpose\_4, data=train\_1,family=binomial())  
  
  
m3\_2 <- step(m3\_2)

## Start: AIC=720.81  
## good\_bad\_21 ~ chk\_ac\_status\_1 + duration\_month\_2 + credit\_history\_3 +   
## purpose\_4 + credit\_amount\_5 + savings\_ac\_bond\_6 + p\_employment\_since\_7 +   
## installment\_pct\_disp\_inc\_8 + property\_type\_12 + age\_in\_yrs\_13 +   
## credit\_history\_3:p\_employment\_since\_7 + credit\_history\_3:age\_in\_yrs\_13 +   
## chk\_ac\_status\_1:p\_employment\_since\_7 + chk\_ac\_status\_1:savings\_ac\_bond\_6 +   
## duration\_month\_2:purpose\_4  
##   
## Df Deviance AIC  
## - credit\_history\_3:p\_employment\_since\_7 16 568.97 706.97  
## - chk\_ac\_status\_1:savings\_ac\_bond\_6 12 568.09 714.09  
## - duration\_month\_2:purpose\_4 9 563.73 715.73  
## - credit\_history\_3:age\_in\_yrs\_13 4 553.96 715.96  
## - chk\_ac\_status\_1:p\_employment\_since\_7 12 572.18 718.18  
## - property\_type\_12 3 556.80 720.80  
## <none> 550.81 720.81  
## - installment\_pct\_disp\_inc\_8 1 553.75 721.75  
## - credit\_amount\_5 1 555.48 723.48  
##   
## Step: AIC=706.97  
## good\_bad\_21 ~ chk\_ac\_status\_1 + duration\_month\_2 + credit\_history\_3 +   
## purpose\_4 + credit\_amount\_5 + savings\_ac\_bond\_6 + p\_employment\_since\_7 +   
## installment\_pct\_disp\_inc\_8 + property\_type\_12 + age\_in\_yrs\_13 +   
## credit\_history\_3:age\_in\_yrs\_13 + chk\_ac\_status\_1:p\_employment\_since\_7 +   
## chk\_ac\_status\_1:savings\_ac\_bond\_6 + duration\_month\_2:purpose\_4  
##   
## Df Deviance AIC  
## - chk\_ac\_status\_1:savings\_ac\_bond\_6 12 585.52 699.52  
## - chk\_ac\_status\_1:p\_employment\_since\_7 12 587.85 701.85  
## - duration\_month\_2:purpose\_4 9 582.99 702.99  
## - credit\_history\_3:age\_in\_yrs\_13 4 574.47 704.47  
## - property\_type\_12 3 574.49 706.49  
## <none> 568.97 706.97  
## - installment\_pct\_disp\_inc\_8 1 572.32 708.32  
## - credit\_amount\_5 1 573.54 709.54  
##   
## Step: AIC=699.52  
## good\_bad\_21 ~ chk\_ac\_status\_1 + duration\_month\_2 + credit\_history\_3 +   
## purpose\_4 + credit\_amount\_5 + savings\_ac\_bond\_6 + p\_employment\_since\_7 +   
## installment\_pct\_disp\_inc\_8 + property\_type\_12 + age\_in\_yrs\_13 +   
## credit\_history\_3:age\_in\_yrs\_13 + chk\_ac\_status\_1:p\_employment\_since\_7 +   
## duration\_month\_2:purpose\_4  
##   
## Df Deviance AIC  
## - duration\_month\_2:purpose\_4 9 599.08 695.08  
## - chk\_ac\_status\_1:p\_employment\_since\_7 12 605.16 695.16  
## - credit\_history\_3:age\_in\_yrs\_13 4 591.41 697.41  
## - property\_type\_12 3 591.34 699.34  
## <none> 585.52 699.52  
## - installment\_pct\_disp\_inc\_8 1 589.45 701.45  
## - credit\_amount\_5 1 589.47 701.47  
## - savings\_ac\_bond\_6 4 608.82 714.82  
##   
## Step: AIC=695.08  
## good\_bad\_21 ~ chk\_ac\_status\_1 + duration\_month\_2 + credit\_history\_3 +   
## purpose\_4 + credit\_amount\_5 + savings\_ac\_bond\_6 + p\_employment\_since\_7 +   
## installment\_pct\_disp\_inc\_8 + property\_type\_12 + age\_in\_yrs\_13 +   
## credit\_history\_3:age\_in\_yrs\_13 + chk\_ac\_status\_1:p\_employment\_since\_7  
##   
## Df Deviance AIC  
## - chk\_ac\_status\_1:p\_employment\_since\_7 12 619.13 691.13  
## - credit\_history\_3:age\_in\_yrs\_13 4 603.66 691.66  
## - property\_type\_12 3 603.69 693.69  
## <none> 599.08 695.08  
## - credit\_amount\_5 1 602.58 696.58  
## - installment\_pct\_disp\_inc\_8 1 603.62 697.62  
## - duration\_month\_2 1 606.89 700.89  
## - purpose\_4 9 630.19 708.19  
## - savings\_ac\_bond\_6 4 620.84 708.84  
##   
## Step: AIC=691.13  
## good\_bad\_21 ~ chk\_ac\_status\_1 + duration\_month\_2 + credit\_history\_3 +   
## purpose\_4 + credit\_amount\_5 + savings\_ac\_bond\_6 + p\_employment\_since\_7 +   
## installment\_pct\_disp\_inc\_8 + property\_type\_12 + age\_in\_yrs\_13 +   
## credit\_history\_3:age\_in\_yrs\_13  
##   
## Df Deviance AIC  
## - property\_type\_12 3 622.73 688.73  
## - credit\_history\_3:age\_in\_yrs\_13 4 624.86 688.86  
## <none> 619.13 691.13  
## - credit\_amount\_5 1 622.40 692.40  
## - p\_employment\_since\_7 4 628.95 692.95  
## - installment\_pct\_disp\_inc\_8 1 623.73 693.73  
## - duration\_month\_2 1 627.39 697.39  
## - purpose\_4 9 645.52 699.52  
## - savings\_ac\_bond\_6 4 640.14 704.14  
## - chk\_ac\_status\_1 3 681.65 747.65  
##   
## Step: AIC=688.73  
## good\_bad\_21 ~ chk\_ac\_status\_1 + duration\_month\_2 + credit\_history\_3 +   
## purpose\_4 + credit\_amount\_5 + savings\_ac\_bond\_6 + p\_employment\_since\_7 +   
## installment\_pct\_disp\_inc\_8 + age\_in\_yrs\_13 + credit\_history\_3:age\_in\_yrs\_13  
##   
## Df Deviance AIC  
## - credit\_history\_3:age\_in\_yrs\_13 4 628.62 686.62  
## <none> 622.73 688.73  
## - credit\_amount\_5 1 626.78 690.78  
## - p\_employment\_since\_7 4 633.13 691.13  
## - installment\_pct\_disp\_inc\_8 1 628.27 692.27  
## - purpose\_4 9 648.05 696.05  
## - duration\_month\_2 1 632.68 696.68  
## - savings\_ac\_bond\_6 4 643.79 701.79  
## - chk\_ac\_status\_1 3 685.17 745.17  
##   
## Step: AIC=686.62  
## good\_bad\_21 ~ chk\_ac\_status\_1 + duration\_month\_2 + credit\_history\_3 +   
## purpose\_4 + credit\_amount\_5 + savings\_ac\_bond\_6 + p\_employment\_since\_7 +   
## installment\_pct\_disp\_inc\_8 + age\_in\_yrs\_13  
##   
## Df Deviance AIC  
## <none> 628.62 686.62  
## - age\_in\_yrs\_13 1 632.08 688.08  
## - p\_employment\_since\_7 4 638.88 688.88  
## - credit\_amount\_5 1 633.76 689.76  
## - installment\_pct\_disp\_inc\_8 1 635.31 691.31  
## - purpose\_4 9 653.56 693.56  
## - duration\_month\_2 1 637.94 693.94  
## - savings\_ac\_bond\_6 4 650.19 700.19  
## - credit\_history\_3 4 651.67 701.67  
## - chk\_ac\_status\_1 3 688.68 740.68

summary(m3\_2)

##   
## Call:  
## glm(formula = good\_bad\_21 ~ chk\_ac\_status\_1 + duration\_month\_2 +   
## credit\_history\_3 + purpose\_4 + credit\_amount\_5 + savings\_ac\_bond\_6 +   
## p\_employment\_since\_7 + installment\_pct\_disp\_inc\_8 + age\_in\_yrs\_13,   
## family = binomial(), data = train\_1)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.6591 -0.7195 0.3698 0.6914 2.6031   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -4.792e-01 8.300e-01 -0.577 0.563714   
## chk\_ac\_status\_1A12 3.819e-01 2.514e-01 1.519 0.128782   
## chk\_ac\_status\_1A13 1.133e+00 4.150e-01 2.729 0.006353 \*\*   
## chk\_ac\_status\_1A14 1.896e+00 2.754e-01 6.884 5.82e-12 \*\*\*  
## duration\_month\_2 -3.220e-02 1.065e-02 -3.022 0.002509 \*\*   
## credit\_history\_3A31 -1.098e+00 6.538e-01 -1.680 0.093029 .   
## credit\_history\_3A32 5.126e-01 4.682e-01 1.095 0.273621   
## credit\_history\_3A33 7.664e-01 5.482e-01 1.398 0.162081   
## credit\_history\_3A34 1.103e+00 4.890e-01 2.255 0.024105 \*   
## purpose\_4A41 1.377e+00 4.349e-01 3.165 0.001551 \*\*   
## purpose\_4A410 1.098e+00 9.224e-01 1.191 0.233844   
## purpose\_4A42 5.872e-01 2.978e-01 1.972 0.048657 \*   
## purpose\_4A43 6.872e-01 2.854e-01 2.408 0.016047 \*   
## purpose\_4A44 -5.106e-01 9.574e-01 -0.533 0.593772   
## purpose\_4A45 4.025e-01 6.934e-01 0.580 0.561610   
## purpose\_4A46 -5.325e-01 4.743e-01 -1.123 0.261554   
## purpose\_4A48 1.623e+01 7.303e+02 0.022 0.982267   
## purpose\_4A49 5.292e-01 4.036e-01 1.311 0.189771   
## credit\_amount\_5 -1.146e-04 5.072e-05 -2.260 0.023821 \*   
## savings\_ac\_bond\_6A62 8.459e-02 3.290e-01 0.257 0.797095   
## savings\_ac\_bond\_6A63 1.194e+00 6.403e-01 1.865 0.062131 .   
## savings\_ac\_bond\_6A64 1.444e+00 6.449e-01 2.239 0.025167 \*   
## savings\_ac\_bond\_6A65 1.121e+00 3.155e-01 3.553 0.000381 \*\*\*  
## p\_employment\_since\_7A72 4.357e-02 4.751e-01 0.092 0.926934   
## p\_employment\_since\_7A73 3.950e-01 4.429e-01 0.892 0.372411   
## p\_employment\_since\_7A74 1.077e+00 4.900e-01 2.198 0.027925 \*   
## p\_employment\_since\_7A75 4.458e-01 4.549e-01 0.980 0.327161   
## installment\_pct\_disp\_inc\_8 -2.539e-01 9.936e-02 -2.556 0.010599 \*   
## age\_in\_yrs\_13 1.881e-02 1.024e-02 1.837 0.066276 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 855.21 on 699 degrees of freedom  
## Residual deviance: 628.62 on 671 degrees of freedom  
## AIC: 686.62  
##   
## Number of Fisher Scoring iterations: 15

test\_1$m3\_2\_score<-predict(m3\_2,type='response',test\_1)  
m3\_2\_pred<-prediction(test\_1$m3\_2\_score,test\_1$good\_bad\_21)  
m3\_2\_perf <- performance(m3\_2\_pred,"tpr","fpr")  
  
# Model Performance  
plot(m3\_2\_perf, main="Improve Logistic Results using Random Forest")



m3\_2\_AUCRF <- performance(m3\_2\_pred, measure = "auc")@y.values[[1]]  
cat("AUC: ",m3\_2\_AUCRF,"\n")

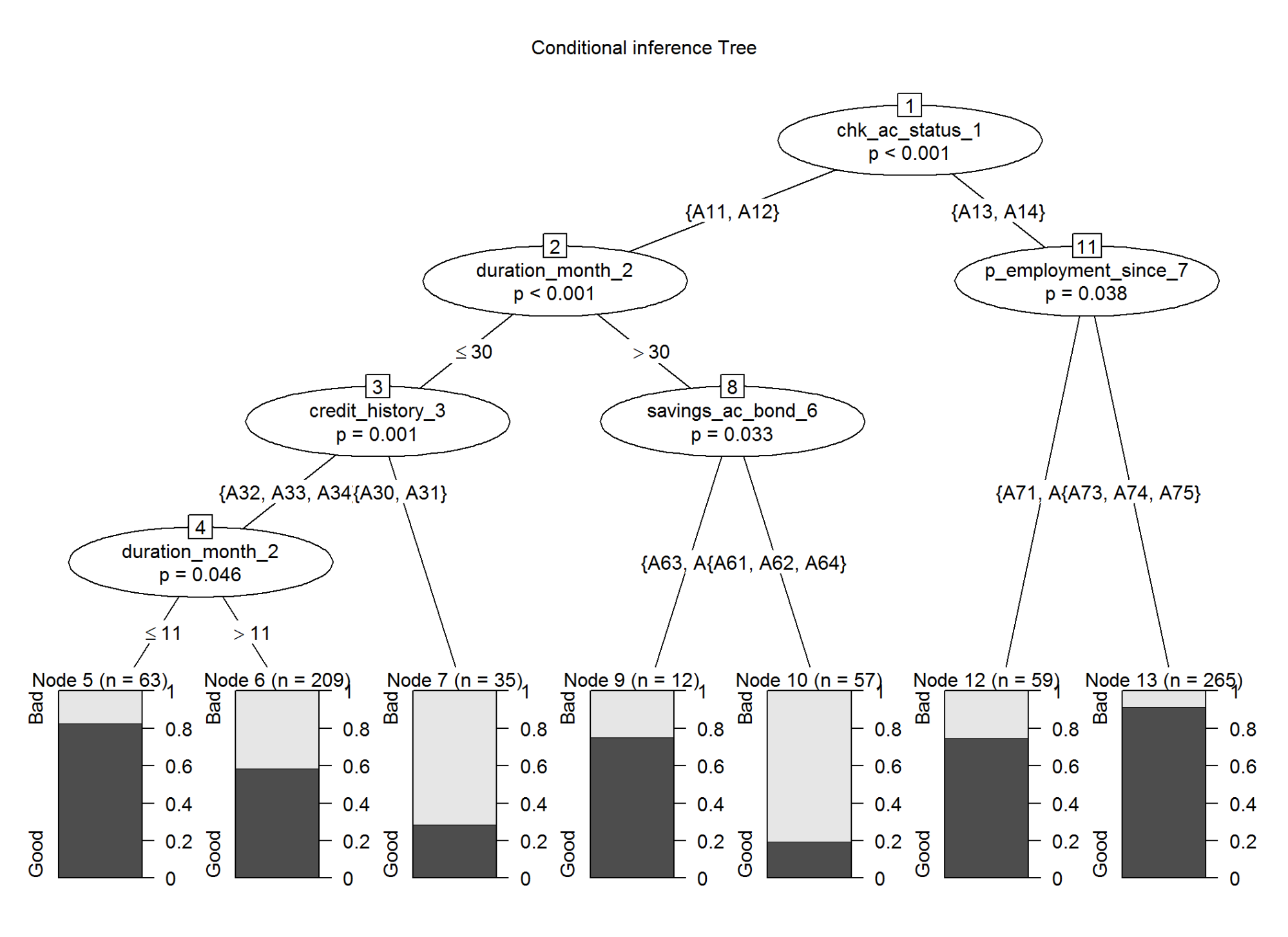
## AUC: 0.7506878

#KS m3  
m3\_2\_KS<-max(attr(m3\_2\_perf,'y.values')[[1]]-attr(m3\_2\_perf,'x.values')[[1]])\*100  
m3\_2\_KS

## [1] 40.31746

3.4 Conditional inference Trees

#library(party)  
m4 <- ctree(good\_bad\_21~.,data=train\_1)  
plot(m4, main="Conditional inference Tree");



resultdfr <- as.data.frame(do.call("rbind", treeresponse(m4, newdata = test\_1)))  
test\_1$m4\_score <- resultdfr[,2]  
m4\_pred <- prediction(test\_1$m4\_score,test\_1$good\_bad\_21)  
m4\_perf <- performance(m4\_pred,"tpr","fpr")  
  
# Model Performance  
m4\_AUCRF <- performance(m4\_pred, measure = "auc")@y.values[[1]]  
cat("AUC: ",m4\_AUCRF,"\n")

## AUC: 0.6846032

#KS m3  
m4\_KS<-max(attr(m4\_perf,'y.values')[[1]]-attr(m4\_perf,'x.values')[[1]])\*100  
m4\_KS

## [1] 30.95238

#randomForest (randomForest) and cforest (party) have same results

3.5 Bayesian Network (Computation - intensive and expensive)

#load library  
#library(bnlearn)  
train\_2<-train\_1  
train\_2$duration\_month\_2 <- as.factor(train\_2$duration\_month\_2)  
train\_2$credit\_amount\_5 <- as.factor(train\_2$credit\_amount\_5)  
train\_2$installment\_pct\_disp\_inc\_8 <- as.factor(train\_2$installment\_pct\_disp\_inc\_8)  
train\_2$age\_in\_yrs\_13 <- as.factor(train\_2$age\_in\_yrs\_13)  
   
bn.gs <- gs(train\_2)  
bn.gs

##   
## Bayesian network learned via Constraint-based methods  
##   
## model:  
## [undirected graph]  
## nodes: 11   
## arcs: 3   
## undirected arcs: 3   
## directed arcs: 0   
## average markov blanket size: 0.55   
## average neighbourhood size: 0.55   
## average branching factor: 0.00   
##   
## learning algorithm: Grow-Shrink   
## conditional independence test: Mutual Information (disc.)   
## alpha threshold: 0.05   
## tests used in the learning procedure: 168   
## optimized: TRUE

bn2 <- iamb(train\_2)  
bn2

##   
## Bayesian network learned via Constraint-based methods  
##   
## model:  
## [partially directed graph]  
## nodes: 11   
## arcs: 5   
## undirected arcs: 3   
## directed arcs: 2   
## average markov blanket size: 1.09   
## average neighbourhood size: 0.91   
## average branching factor: 0.18   
##   
## learning algorithm: IAMB   
## conditional independence test: Mutual Information (disc.)   
## alpha threshold: 0.05   
## tests used in the learning procedure: 201   
## optimized: TRUE

bn3 <- fast.iamb(train\_2)  
bn3

##   
## Bayesian network learned via Constraint-based methods  
##   
## model:  
## [undirected graph]  
## nodes: 11   
## arcs: 6   
## undirected arcs: 6   
## directed arcs: 0   
## average markov blanket size: 1.09   
## average neighbourhood size: 1.09   
## average branching factor: 0.00   
##   
## learning algorithm: Fast-IAMB   
## conditional independence test: Mutual Information (disc.)   
## alpha threshold: 0.05   
## tests used in the learning procedure: 120   
## optimized: TRUE

bn4 <- inter.iamb(train\_2)  
bn4

##   
## Bayesian network learned via Constraint-based methods  
##   
## model:  
## [undirected graph]  
## nodes: 11   
## arcs: 8   
## undirected arcs: 8   
## directed arcs: 0   
## average markov blanket size: 1.45   
## average neighbourhood size: 1.45   
## average branching factor: 0.00   
##   
## learning algorithm: Inter-IAMB   
## conditional independence test: Mutual Information (disc.)   
## alpha threshold: 0.05   
## tests used in the learning procedure: 214   
## optimized: TRUE

compare(bn.gs, bn2)

## $tp  
## [1] 1  
##   
## $fp  
## [1] 4  
##   
## $fn  
## [1] 2

compare(bn.gs, bn3)

## $tp  
## [1] 1  
##   
## $fp  
## [1] 5  
##   
## $fn  
## [1] 2

compare(bn.gs, bn4)

## $tp  
## [1] 2  
##   
## $fp  
## [1] 6  
##   
## $fn  
## [1] 1

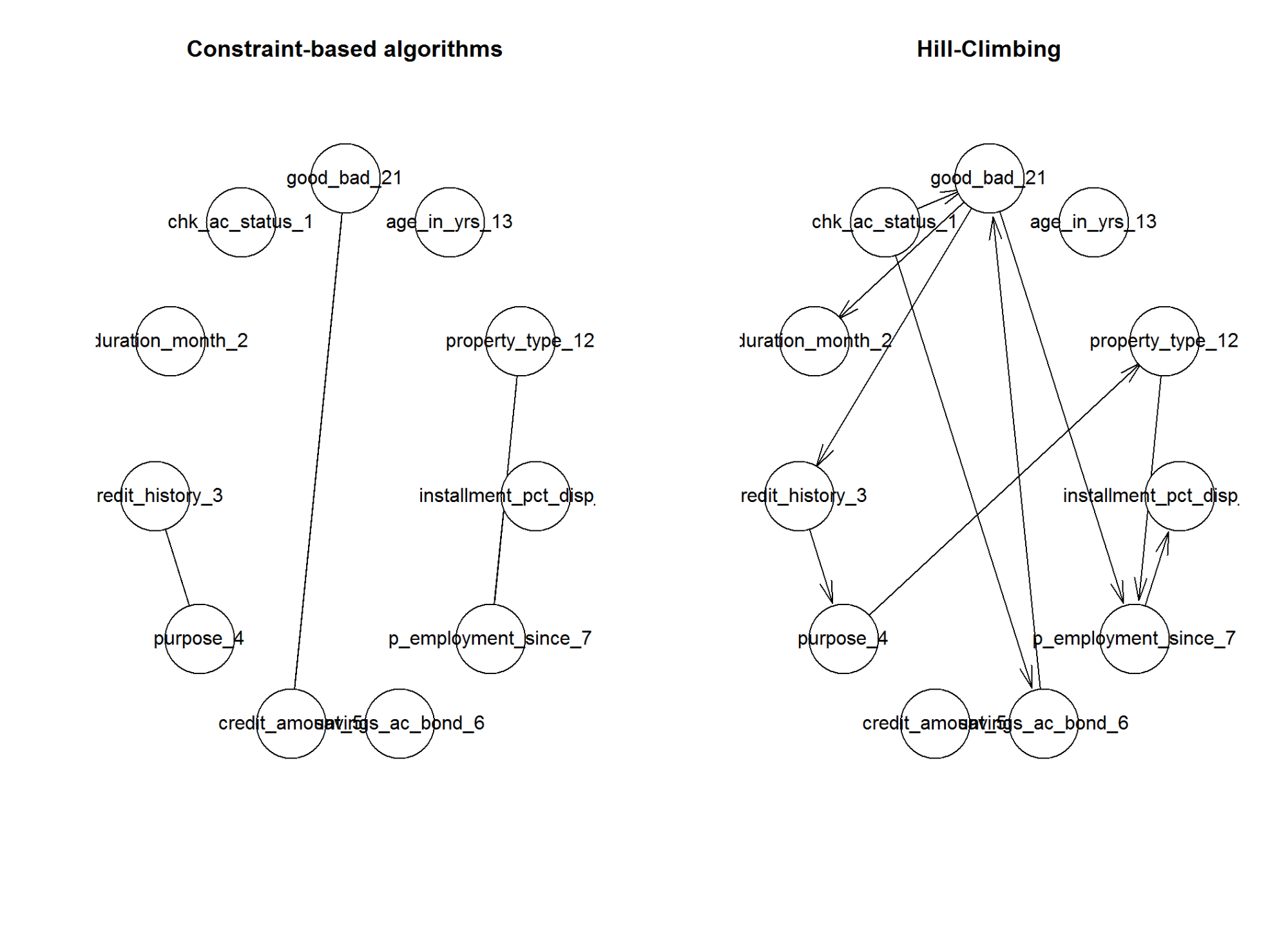
#On the other hand hill-climbing results in a completely directed network, which di  
ers from  
#the previous one because the arc between A and B is directed (A ! B instead of A B).  
bn.hc <- hc(train\_2, score = "aic")  
bn.hc

##   
## Bayesian network learned via Score-based methods  
##   
## model:  
## [chk\_ac\_status\_1][credit\_amount\_5][age\_in\_yrs\_13][savings\_ac\_bond\_6|chk\_ac\_status\_1][good\_bad\_21|chk\_ac\_status\_1:savings\_ac\_bond\_6][duration\_month\_2|good\_bad\_21][credit\_history\_3|good\_bad\_21]  
## [purpose\_4|credit\_history\_3][property\_type\_12|purpose\_4][p\_employment\_since\_7|property\_type\_12:good\_bad\_21][installment\_pct\_disp\_inc\_8|p\_employment\_since\_7]  
## nodes: 11   
## arcs: 10   
## undirected arcs: 0   
## directed arcs: 10   
## average markov blanket size: 2.00   
## average neighbourhood size: 1.82   
## average branching factor: 0.91   
##   
## learning algorithm: Hill-Climbing   
## score: AIC (disc.)   
## penalization coefficient: 1   
## tests used in the learning procedure: 155   
## optimized: TRUE

compare(bn.hc, bn.gs)

## $tp  
## [1] 0  
##   
## $fp  
## [1] 3  
##   
## $fn  
## [1] 10

opm5<-par(mfrow = c(1,2))  
plot(bn.gs, main = "Constraint-based algorithms")  
plot(bn.hc, main = "Hill-Climbing")



par(opm5)  
modelstring(bn.hc)

## [1] "[chk\_ac\_status\_1][credit\_amount\_5][age\_in\_yrs\_13][savings\_ac\_bond\_6|chk\_ac\_status\_1][good\_bad\_21|chk\_ac\_status\_1:savings\_ac\_bond\_6][duration\_month\_2|good\_bad\_21][credit\_history\_3|good\_bad\_21][purpose\_4|credit\_history\_3][property\_type\_12|purpose\_4][p\_employment\_since\_7|property\_type\_12:good\_bad\_21][installment\_pct\_disp\_inc\_8|p\_employment\_since\_7]"

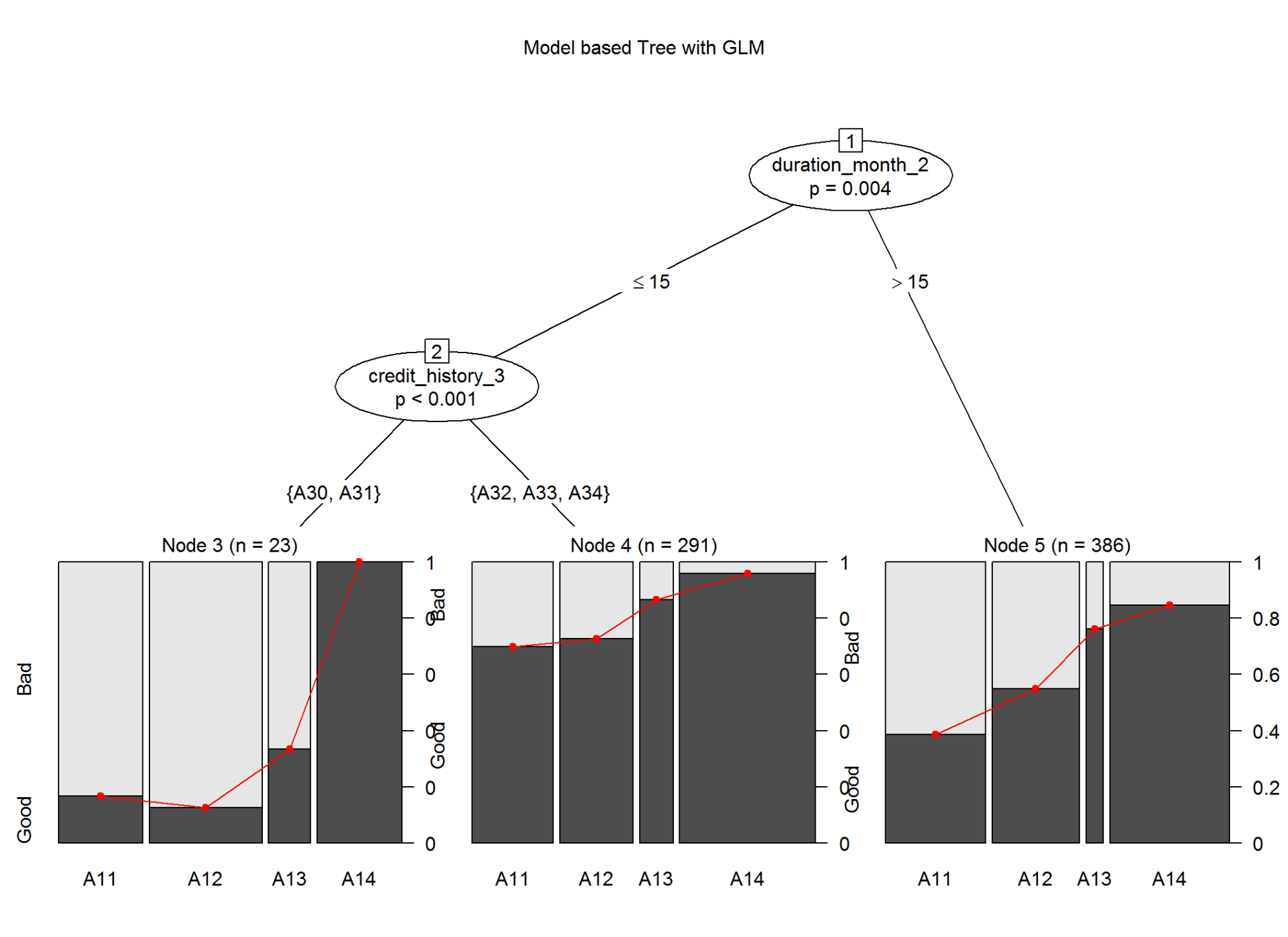
res2 = hc(train\_2)  
fitted2 = bn.fit(res2, train\_2)  
fitted2

##   
## Bayesian network parameters  
##   
## Parameters of node chk\_ac\_status\_1 (multinomial distribution)  
##   
## Conditional probability table:  
## A11 A12 A13 A14   
## 0.28285714 0.25428571 0.07714286 0.38571429   
##   
## Parameters of node duration\_month\_2 (multinomial distribution)  
##   
## Conditional probability table:  
## 4 5 6 7 8 9 10 11 12 13 14 15 16 18 20 21   
## 0.007142857 0.001428571 0.087142857 0.005714286 0.008571429 0.055714286 0.022857143 0.010000000 0.172857143 0.004285714 0.004285714 0.068571429 0.001428571 0.111428571 0.007142857 0.028571429   
## 22 24 27 30 33 36 39 40 42 45 47 48 54 60 72   
## 0.001428571 0.198571429 0.010000000 0.044285714 0.001428571 0.064285714 0.002857143 0.001428571 0.010000000 0.004285714 0.001428571 0.048571429 0.001428571 0.011428571 0.001428571   
##   
## Parameters of node credit\_history\_3 (multinomial distribution)  
##   
## Conditional probability table:  
##   
## good\_bad\_21  
## credit\_history\_3 Bad Good  
## A30 0.08571429 0.02653061  
## A31 0.09523810 0.02448980  
## A32 0.55238095 0.52244898  
## A33 0.08571429 0.07959184  
## A34 0.18095238 0.34693878  
##   
## Parameters of node purpose\_4 (multinomial distribution)  
##   
## Conditional probability table:  
## A40 A41 A410 A42 A43 A44 A45 A46 A48 A49   
## 0.23571429 0.10000000 0.01000000 0.18857143 0.28714286 0.01142857 0.02000000 0.04714286 0.01000000 0.09000000   
##   
## Parameters of node credit\_amount\_5 (multinomial distribution)  
##   
## Conditional probability table:  
## 276 338 339 343 362 368 385 392 409 426 428 433 448 454 458 484   
## 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.002857143 0.001428571 0.001428571 0.001428571 0.001428571   
## 522 571 590 601 609 618 639 640 652 660 662 666 674 682 683 684   
## 0.001428571 0.001428571 0.001428571 0.001428571 0.002857143 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571   
## 685 691 700 701 708 709 717 719 727 730 731 741 745 750 754 760   
## 0.001428571 0.001428571 0.001428571 0.002857143 0.001428571 0.002857143 0.002857143 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571   
## 763 776 781 783 790 802 836 846 866 882 886 894 900 909 918 926   
## 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571   
## 929 931 932 936 939 947 958 959 960 976 983 996 999 1024 1028 1042   
## 0.001428571 0.001428571 0.002857143 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571   
## 1047 1049 1050 1053 1055 1068 1082 1092 1101 1108 1126 1131 1136 1138 1149 1154   
## 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.002857143 0.001428571 0.001428571 0.001428571 0.002857143 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571   
## 1155 1158 1164 1168 1169 1188 1190 1193 1198 1199 1201 1203 1204 1206 1207 1213   
## 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.002857143 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571   
## 1216 1217 1221 1223 1224 1228 1231 1236 1237 1238 1239 1245 1246 1249 1255 1258   
## 0.002857143 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.002857143 0.002857143 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.004285714   
## 1262 1264 1271 1274 1275 1278 1282 1283 1285 1287 1288 1289 1297 1299 1300 1308   
## 0.004285714 0.002857143 0.001428571 0.001428571 0.004285714 0.001428571 0.002857143 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571   
## 1316 1318 1323 1330 1333 1337 1338 1343 1344 1346 1347 1352 1355 1358 1360 1361   
## 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.002857143 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571   
## 1364 1366 1371 1374 1376 1377 1382 1386 1388 1391 1393 1402 1403 1409 1410 1413   
## 0.002857143 0.001428571 0.001428571 0.002857143 0.001428571 0.001428571 0.002857143 0.001428571 0.001428571 0.001428571 0.004285714 0.001428571 0.001428571 0.001428571 0.002857143 0.001428571   
## 1414 1418 1422 1424 1433 1437 1442 1444 1445 1449 1453 1455 1459 1471 1474 1478   
## 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.002857143 0.004285714   
## 1480 1484 1493 1494 1495 1498 1501 1503 1505 1516 1521 1525 1526 1530 1532 1533   
## 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.002857143   
## 1534 1538 1542 1543 1544 1546 1549 1553 1554 1559 1567 1568 1569 1574 1577 1582   
## 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.002857143 0.001428571 0.002857143 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571   
## 1585 1591 1592 1595 1597 1602 1647 1655 1657 1659 1670 1680 1715 1736 1740 1743   
## 0.001428571 0.001428571 0.001428571 0.001428571 0.002857143 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.002857143   
## 1747 1755 1766 1768 1778 1795 1797 1800 1804 1817 1819 1820 1842 1845 1851 1864   
## 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.002857143 0.001428571 0.001428571   
## 1867 1872 1880 1881 1882 1884 1887 1893 1901 1905 1913 1919 1922 1924 1925 1928   
## 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.002857143 0.001428571 0.002857143 0.001428571 0.001428571   
## 1935 1936 1940 1943 1953 1963 1965 1967 1980 1984 1987 1995 2012 2022 2028 2030   
## 0.002857143 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571   
## 2032 2039 2051 2058 2063 2064 2073 2100 2108 2116 2118 2121 2124 2134 2141 2142   
## 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571   
## 2145 2146 2149 2150 2169 2171 2181 2186 2197 2212 2214 2221 2235 2238 2247 2251   
## 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.002857143 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571   
## 2273 2278 2279 2284 2288 2292 2299 2301 2302 2303 2315 2319 2325 2326 2327 2329   
## 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571   
## 2333 2346 2348 2353 2359 2360 2366 2375 2406 2415 2427 2442 2445 2462 2507 2511   
## 0.002857143 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571   
## 2515 2522 2528 2538 2570 2578 2579 2580 2600 2603 2611 2613 2622 2625 2631 2647   
## 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.002857143 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571   
## 2670 2671 2679 2684 2687 2697 2708 2718 2728 2746 2751 2753 2759 2760 2762 2767   
## 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571   
## 2775 2779 2788 2820 2825 2831 2848 2859 2872 2892 2896 2901 2910 2923 2924 2957   
## 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571   
## 2964 2978 3001 3016 3017 3021 3029 3049 3051 3060 3069 3074 3077 3092 3104 3105   
## 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.002857143 0.001428571 0.001428571 0.001428571   
## 3108 3114 3123 3148 3149 3160 3181 3186 3190 3213 3229 3234 3235 3343 3345 3349   
## 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.002857143   
## 3357 3368 3380 3384 3386 3394 3398 3399 3414 3422 3446 3447 3448 3485 3488 3496   
## 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571   
## 3499 3509 3512 3518 3527 3552 3556 3566 3568 3573 3578 3590 3595 3599 3612 3617   
## 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.002857143 0.001428571 0.001428571 0.001428571 0.002857143   
## 3621 3622 3632 3650 3651 3652 3656 3660 3676 3711 3763 3777 3804 3832 3835 3850   
## 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.002857143 0.001428571 0.001428571   
## 3863 3905 3915 3931 3939 3959 3965 3972 3973 3976 3979 4020 4042 4057 4139 4151   
## 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.002857143 0.001428571 0.001428571 0.001428571   
## 4153 4165 4169 4210 4221 4241 4249 4280 4281 4351 4370 4439 4454 4455 4530 4583   
## 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571   
## 4591 4594 4605 4611 4657 4675 4679 4686 4712 4716 4736 4746 4771 4788 4796 4811   
## 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571   
## 4817 4843 4870 5045 5084 5096 5129 5152 5179 5190 5248 5293 5302 5324 5507 5511   
## 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571   
## 5595 5742 5743 5771 5800 5801 5804 5842 5848 5943 5951 5954 5965 5998 6070 6110   
## 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571   
## 6143 6148 6187 6199 6224 6229 6260 6289 6304 6314 6331 6350 6361 6416 6419 6468   
## 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.002857143   
## 6527 6560 6579 6615 6681 6742 6758 6761 6836 6842 6850 6887 6948 6967 6999 7119   
## 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.002857143 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571   
## 7127 7308 7393 7408 7418 7432 7472 7485 7582 7596 7629 7678 7721 7763 7814 7865   
## 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571   
## 7882 8072 8086 8133 8335 8358 8471 8588 8648 8858 8947 8978 9034 9157 9271 9277   
## 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571   
## 9566 9629 10222 10297 10366 10623 10722 10961 11560 11590 11816 11938 12169 12204 12389 12579   
## 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571   
## 12680 12749 12976 13756 14179 14318 14555 15653 15945 18424   
## 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571 0.001428571   
##   
## Parameters of node savings\_ac\_bond\_6 (multinomial distribution)  
##   
## Conditional probability table:  
##   
## good\_bad\_21  
## savings\_ac\_bond\_6 Bad Good  
## A61 0.73809524 0.55510204  
## A62 0.12380952 0.09591837  
## A63 0.01904762 0.07551020  
## A64 0.01904762 0.05918367  
## A65 0.10000000 0.21428571  
##   
## Parameters of node p\_employment\_since\_7 (multinomial distribution)  
##   
## Conditional probability table:  
## A71 A72 A73 A74 A75   
## 0.05857143 0.17571429 0.34428571 0.18142857 0.24000000   
##   
## Parameters of node installment\_pct\_disp\_inc\_8 (multinomial distribution)  
##   
## Conditional probability table:  
## 1 2 3 4   
## 0.1371429 0.2214286 0.1514286 0.4900000   
##   
## Parameters of node property\_type\_12 (multinomial distribution)  
##   
## Conditional probability table:  
## A121 A122 A123 A124   
## 0.2857143 0.2314286 0.3228571 0.1600000   
##   
## Parameters of node age\_in\_yrs\_13 (multinomial distribution)  
##   
## Conditional probability table:  
## 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34   
## 0.002857143 0.014285714 0.012857143 0.025714286 0.051428571 0.051428571 0.038571429 0.038571429 0.045714286 0.048571429 0.035714286 0.037142857 0.042857143 0.031428571 0.032857143 0.031428571   
## 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50   
## 0.040000000 0.040000000 0.028571429 0.028571429 0.018571429 0.022857143 0.011428571 0.021428571 0.021428571 0.020000000 0.015714286 0.018571429 0.020000000 0.014285714 0.012857143 0.011428571   
## 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66   
## 0.008571429 0.010000000 0.005714286 0.010000000 0.004285714 0.004285714 0.010000000 0.007142857 0.004285714 0.005714286 0.007142857 0.001428571 0.010000000 0.004285714 0.004285714 0.004285714   
## 67 68 70 74 75   
## 0.001428571 0.001428571 0.001428571 0.004285714 0.002857143   
##   
## Parameters of node good\_bad\_21 (multinomial distribution)  
##   
## Conditional probability table:  
##   
## chk\_ac\_status\_1  
## good\_bad\_21 A11 A12 A13 A14  
## Bad 0.5050505 0.4044944 0.2037037 0.1000000  
## Good 0.4949495 0.5955056 0.7962963 0.9000000

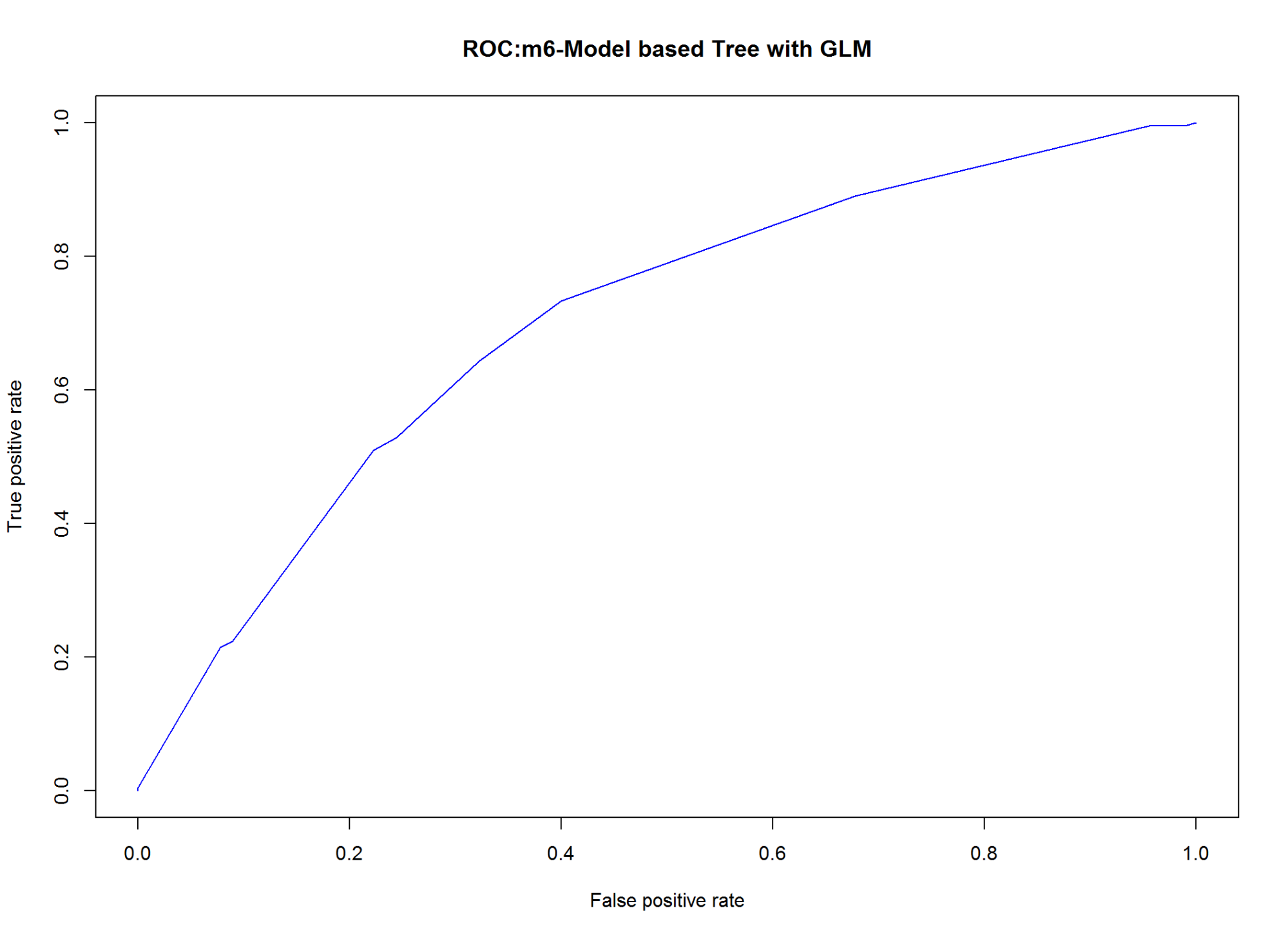
# library(gRain)

3.6 Unbiased Non parametric methods-Model Based Trees (Logistic)

#model based recursive paritioning  
#library(party)  
# iter 1  
m6<-mob(good\_bad\_21~chk\_ac\_status\_1 |  
 duration\_month\_2  
 +credit\_history\_3  
 +purpose\_4  
 +credit\_amount\_5  
 +savings\_ac\_bond\_6  
 +p\_employment\_since\_7  
 +installment\_pct\_disp\_inc\_8  
 +property\_type\_12  
 +age\_in\_yrs\_13,  
 data=train\_1,  
 model=glinearModel,family=binomial())  
  
# iter 2  
# m6<-mob(good\_bad\_21~ credit\_history\_3 +chk\_ac\_status\_1 + savings\_ac\_bond\_6 + purpose\_4 |  
# +duration\_month\_2   
# +installment\_pct\_disp\_inc\_8   
# +credit\_amount\_5   
# +p\_employment\_since\_7   
# +property\_type\_12   
# +age\_in\_yrs\_13,  
# data=train\_1,  
# model=glinearModel,family=binomial())  
  
# iter 3  
# m6<-mob(good\_bad\_21~ chk\_ac\_status\_1 + purpose\_4|  
# credit\_history\_3  
# +duration\_month\_2  
# +installment\_pct\_disp\_inc\_8  
# +credit\_amount\_5  
# +savings\_ac\_bond\_6  
# +p\_employment\_since\_7  
# +property\_type\_12  
# +age\_in\_yrs\_13,  
# data=train\_1,  
# model=glinearModel,family=binomial())  
  
  
#library(vcd) #the `vcd' package is required for CD plots  
plot(m6, main="Model based Tree with GLM")



# Scoring  
test\_1$m6\_score<-predict(m6, newdata = test\_1, type =c("response"))  
  
m6\_pred <- prediction(test\_1$m6\_score,test\_1$good\_bad\_21)  
m6\_perf <- performance(m6\_pred,"tpr","fpr")  
plot(m6\_perf, main="ROC:m6-Model based Tree with GLM", col='blue')



# Model Performance  
m6\_AUCRF <- performance(m6\_pred, measure = "auc")@y.values[[1]]  
cat("AUC: ",m6\_AUCRF,"\n")

## AUC: 0.7021429

#KS m6  
m6\_KS<-max(attr(m6\_perf,'y.values')[[1]]-attr(m6\_perf,'x.values')[[1]])\*100  
m6\_KS

## [1] 33.33333

3.7 Support Vector Machine

3.7.1 SVM - Vanilladot Kernel

#library(kernlab) #for SVM  
  
# Basic Model  
m7\_1 <- ksvm(good\_bad\_21 ~ ., data = train\_1, kernel = "vanilladot")

## Setting default kernel parameters

m7\_1\_pred <- predict(m7\_1, test\_1[,1:10], type="response")  
head(m7\_1\_pred)

## [1] Good Good Good Good Good Bad   
## Levels: Bad Good

# Model accuracy:  
table(m7\_1\_pred, test\_1$good\_bad\_21)

##   
## m7\_1\_pred Bad Good  
## Bad 39 25  
## Good 51 185

#agreement  
m7\_1\_accuracy <- (m7\_1\_pred == test\_1$good\_bad\_21)  
pct(m7\_1\_accuracy)

|  |  |  |
| --- | --- | --- |
|  | Count | Percentage |
| FALSE | 76 | 25.33 |
| TRUE | 224 | 74.67 |

# Compute at the prediction scores  
m7\_1\_score = predict(m7\_1,test\_1, type="decision")  
m7\_1\_pred <- prediction(m7\_1\_score, test\_1$good\_bad\_21)  
  
  
# Plot ROC curve  
m7\_1\_perf <- performance(m7\_1\_pred, measure = "tpr", x.measure = "fpr")  
#plot(m7\_1\_perf, main="SVM:Plot ROC curve", col="blue")  
  
# Plot precision/recall curve  
m7\_1\_perf\_precision <- performance(m7\_1\_pred, measure = "prec", x.measure = "rec")  
#plot(m7\_1\_perf\_precision, main="SVM:Plot precision/recall curve")  
  
# Plot accuracy as function of threshold  
m7\_1\_perf\_acc <- performance(m7\_1\_pred, measure = "acc")  
#plot(m7\_1\_perf\_acc, main="SVM:Plot accuracy as function of threshold")  
  
# Model Performance  
m7\_1\_AUCRF <- performance(m7\_1\_pred, measure = "auc")@y.values[[1]]  
cat("AUC: ",m7\_1\_AUCRF,"\n")

## AUC: 0.7603704

#KS m6  
m7\_1\_KS<-max(attr(m7\_1\_perf,'y.values')[[1]]-attr(m7\_1\_perf,'x.values')[[1]])\*100  
m7\_1\_KS

## [1] 44.12698

3.7.2 SVM - Gaussian RBF kernel

# Model Improvement with Gaussian RBF kernel  
m7\_2 <- ksvm(good\_bad\_21 ~ ., data = train\_1,  
 kernel = "rbfdot")  
m7\_2\_pred <- predict(m7\_2, test\_1[,1:10], type="response")  
head(m7\_2\_pred)

## [1] Good Good Good Good Good Bad   
## Levels: Bad Good

# Model accuracy:  
table(m7\_2\_pred, test\_1$good\_bad\_21)

##   
## m7\_2\_pred Bad Good  
## Bad 42 22  
## Good 48 188

#agreement  
m7\_2\_accuracy <- (m7\_2\_pred == test\_1$good\_bad\_21)  
pct(m7\_2\_accuracy)

|  |  |  |
| --- | --- | --- |
|  | Count | Percentage |
| FALSE | 70 | 23.33 |
| TRUE | 230 | 76.67 |

# Compute at the prediction scores  
m7\_2\_score = predict(m7\_2,test\_1, type="decision")  
m7\_2\_pred <- prediction(m7\_2\_score, test\_1$good\_bad\_21)  
  
  
# Plot ROC curve  
m7\_2\_perf <- performance(m7\_2\_pred, measure = "tpr", x.measure = "fpr")  
#plot(m7\_2\_perf, main="SVM:Plot ROC curve", col="blue")  
  
# Plot precision/recall curve  
m7\_2\_perf\_precision <- performance(m7\_2\_pred, measure = "prec", x.measure = "rec")  
#plot(m7\_2\_perf\_precision, main="SVM:Plot precision/recall curve")  
  
# Plot accuracy as function of threshold  
m7\_2\_perf\_acc <- performance(m7\_2\_pred, measure = "acc")  
#plot(m7\_2\_perf\_acc, main="SVM:Plot accuracy as function of threshold")  
  
# Model Performance  
m7\_2\_AUCRF <- performance(m7\_2\_pred, measure = "auc")@y.values[[1]]  
cat("AUC: ",m7\_2\_AUCRF,"\n")

## AUC: 0.7729101

#KS m6  
m7\_2\_KS<-max(attr(m7\_2\_perf,'y.values')[[1]]-attr(m7\_2\_perf,'x.values')[[1]])\*100  
m7\_2\_KS

## [1] 45.2381

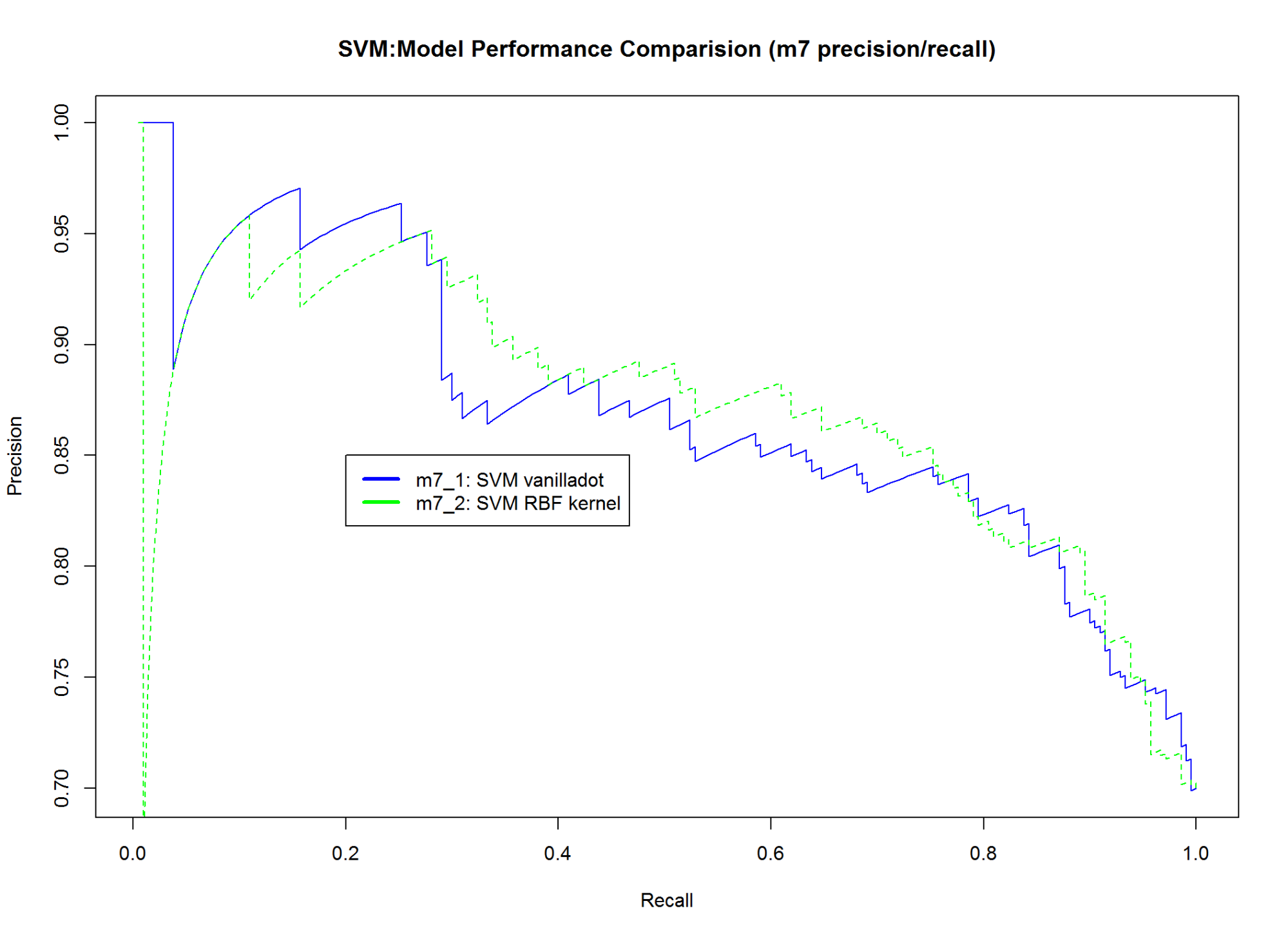
#Your results may differ from those shown here due to randomness in the ksvm RBF kernel. If you'd like them to match exactly, use set.seed(12345) prior to running the ksvm() function.

3.7.3 SVM Model Performance Comparision

# ROC Comparision  
plot(m7\_1\_perf, col='blue', lty=1, main='SVM:Model Performance Comparision (m7 ROC)')   
plot(m7\_2\_perf, col='green',lty=2, add=TRUE); # simple tree  
 legend(0.5,0.4,  
 c("m7\_1: SVM vanilladot", "m7\_2: SVM RBF kernel"),  
 col=c('blue', 'green'),  
 lwd=3);  
abline(lm(y ~x), col='red') # random line



# Precision Comparision  
plot(m7\_1\_perf\_precision, col='blue', lty=1, main='SVM:Model Performance Comparision (m7 precision/recall)')   
plot(m7\_2\_perf\_precision, col='green',lty=2, add=TRUE); # simple tree  
 legend(0.2,0.85,c("m7\_1: SVM vanilladot", "m7\_2: SVM RBF kernel"),  
 col=c('blue', 'green'),lwd=3);

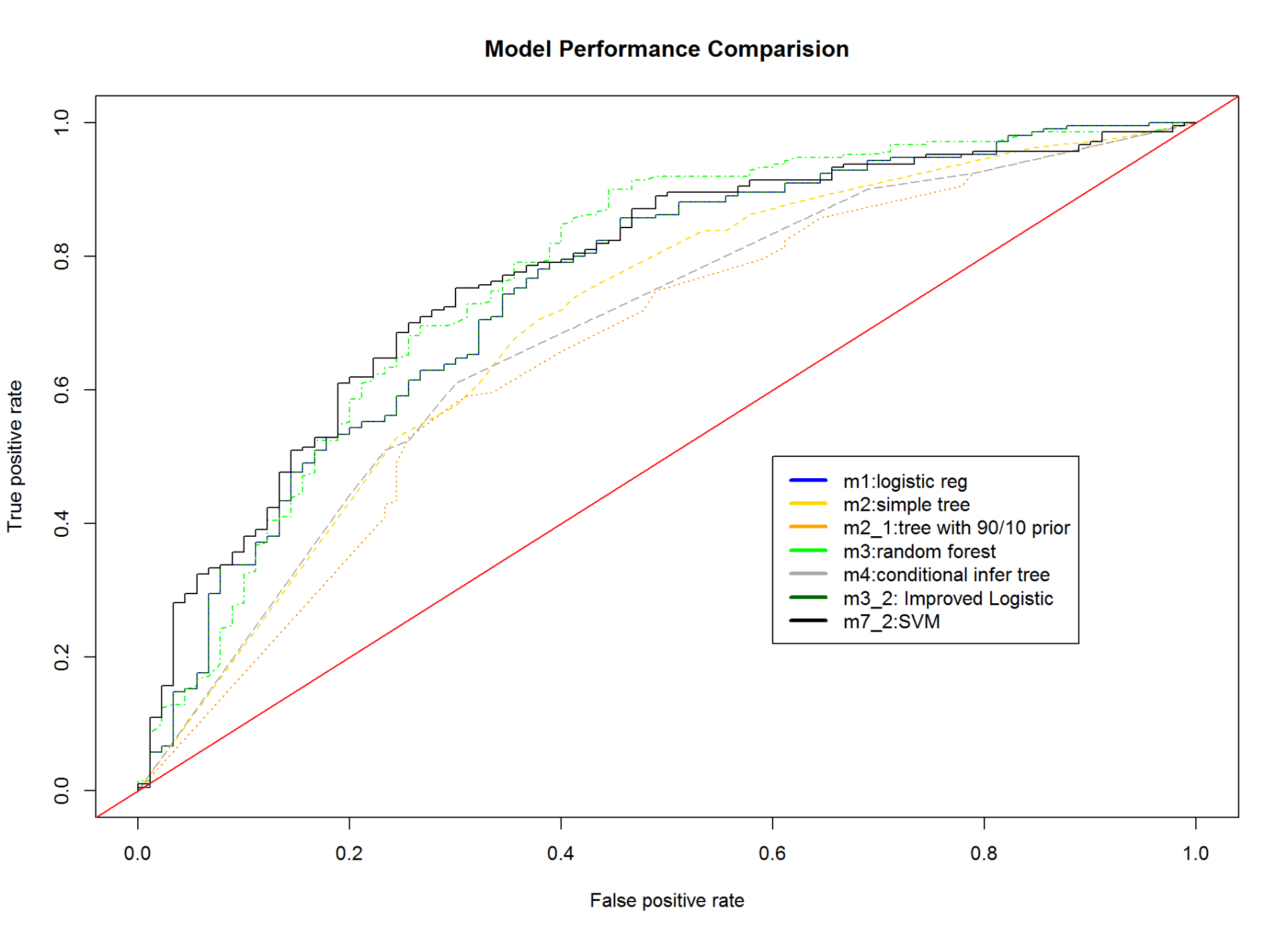


# Plot accuracy as function of threshold  
plot(m7\_1\_perf\_acc, col='blue', lty=1, main='SVM:Model accuracy as function of threshold (m7)')   
plot(m7\_2\_perf\_acc, col='green',lty=2, add=TRUE); # simple tree  
 legend(-1,0.5,c("m7\_1: SVM vanilladot", "m7\_2: SVM RBF kernel"),  
 col=c('blue', 'green'),lwd=3);



3.4 Model Comparision

#Compare ROC Performance of Models  
  
plot(m1\_perf, col='blue', lty=1, main='Model Performance Comparision') # logistic regression  
plot(m2\_perf, col='gold',lty=2, add=TRUE); # simple tree  
plot(m2\_1\_perf, col='dark orange',lty=3, add=TRUE); #tree with 90/10 prior  
plot(m3\_perf, col='green',add=TRUE,lty=4); # random forest  
plot(m4\_perf, col='dark gray',add=TRUE,lty=5); # Conditional Inference Tree  
plot(m3\_2\_perf, col='dark green',add=TRUE,lty=6);  
plot(m7\_2\_perf, col='black',add=TRUE,lty=7);  
 legend(0.6,0.5,  
 c('m1:logistic reg','m2:simple tree','m2\_1:tree with 90/10 prior',   
 'm3:random forest', "m4:conditional infer tree", "m3\_2: Improved Logistic", "m7\_2:SVM"),  
 col=c('blue','gold', 'orange','green', 'dark gray', 'dark green', "black"),  
 lwd=3);  
abline(lm(y ~x), col='red') # random line



A1: References

Credit Scoring

1. <https://sites.google.com/site/rgayler/creditscoringresources>
2. <http://forecastingsolutions.com/>
3. <http://www.rcreditscoring.com/>
4. <http://freakonometrics.hypotheses.org/48285>

Information Value and Weight of Evidence

1. <http://www.ponssard.net/wp-content/uploads/2011/02/on-the-concept-of-the-value-of-information.pdf>
2. <http://research.microsoft.com/en-us/um/people/horvitz/gev.pdf>
3. <http://ucanalytics.com/blogs/information-value-and-weight-of-evidencebanking-case/>
4. <http://www.listendata.com/2015/03/weight-of-evidence-woe-and-information.html>
5. <https://github.com/klarsen1/gampost/blob/master/compare_models.r>

Markdown

1. <https://www.rstudio.com/wp-content/uploads/2015/03/rmarkdown-reference.pdf>
2. <https://rpubs.com/gallery/cache>
3. <http://yihui.name/knitr/options/>

ClusterOfVars

1. <https://cran.r-project.org/web/packages/ClustOfVar/ClustOfVar.pdf>
2. <https://www.r-project.org/conferences/useR-2011/TalkSlides/Contributed/16Aug_1600_FocusII_5-DimReduction_1-Chavent.pdf>
3. <https://arxiv.org/pdf/1112.0295.pdf>
4. <https://stat.ethz.ch/R-manual/R-devel/library/stats/html/rect.hclust.html>

Bootstrap

1.<https://www.r-bloggers.com/bootstrap-evaluation-of-clusters/>

Random Sampling

1.<https://cran.r-project.org/web/packages/sampling/sampling.pdf>

Logistic Regression

1. <https://cran.r-project.org/web/packages/HSAUR/vignettes/Ch_logistic_regression_glm.pdf>
2. <http://www.ats.ucla.edu/stat/r/dae/logit.htm>

rpart

1. <https://cran.r-project.org/web/packages/rpart.utils/rpart.utils.pdf>

Bayesian networks

1. <https://cran.r-project.org/web/packages/gRain/vignettes/gRain-intro.pdf>
2. <https://arxiv.org/pdf/0908.3817.pdf>
3. <http://www.bnlearn.com/examples/score/>
4. <https://stat.ethz.ch/pipermail/r-help//2012-September/336359.html>
5. <https://cran.r-project.org/web/packages/bnlearn/bnlearn.pdf>
6. <https://www.library.ln.edu.hk/eresources/etext/hkibs/hkws_0053.pdf>
7. <https://www.r-project.org/conferences/DSC-2003/Proceedings/BottcherDethlefsen.pdf>
8. <https://www.r-project.org/conferences/DSC-2003/Drafts/>
9. <http://www.vetepi.uzh.ch/dam/jcr:00000000-2b43-78bb-ffff-ffffb89c13f1/abn.pdf>
10. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.114.3548&rep=rep1&type=pdf>
11. <http://www.cs.uu.nl/research/techreps/repo/CS-2001/2001-58.pdf>
12. <http://bayesian-intelligence.com/publications/TR2010_1_zonneveldt_korb_nicholson_bn_class_credit_data.pdf>
13. <http://pure.au.dk/portal-asb-student/files/47799695/Thesis.pdf>

Plots and Visualization

1. <http://www.harding.edu/fmccown/r/>
2. <https://rstudio.github.io/dygraphs/gallery-upper-lower-bars.html>
3. <https://stat.ethz.ch/R-manual/R-devel/library/graphics/html/abline.html>

germanCredit

1. <https://ocw.mit.edu/courses/sloan-school-of-management/15-062-data-mining-spring-2003/assignments/GermanCredit.pdf>

Neural Network

1. <https://www.r-bloggers.com/using-neural-networks-for-credit-scoring-a-simple-example/>
2. <http://www.ijstm.com/images/short_pdf/1431276484_P_26-31.pdf>

Machine Learning

1. Data: <http://archive.ics.uci.edu/ml/>

SVM:

1. <https://www.jstatsoft.org/article/view/v011i09>
2. <https://escience.rpi.edu/data/DA/svmbasic_notes.pdf>
3. <https://cran.r-project.org/web/packages/e1071/vignettes/svmdoc.pdf>
4. <https://cran.r-project.org/web/packages/kernlab/kernlab.pdf> \*\*\*

Note that the echo = FALSE parameter was added to the code chunk to prevent printing of the R code that generated the plot.