Problem 2.

(a)

* The proportion of “Good” to “Bad” cases: **7:3**

> freq = table(German.Credit$RESPONSE)/nrow(German.Credit)

> freq

0 1

0.3 0.7

* Predictor variables: Descriptions and the Relationship with the target variable

1. CHK\_ACCT

> table(German.Credit$CHK\_ACCT)

0 1 2 3

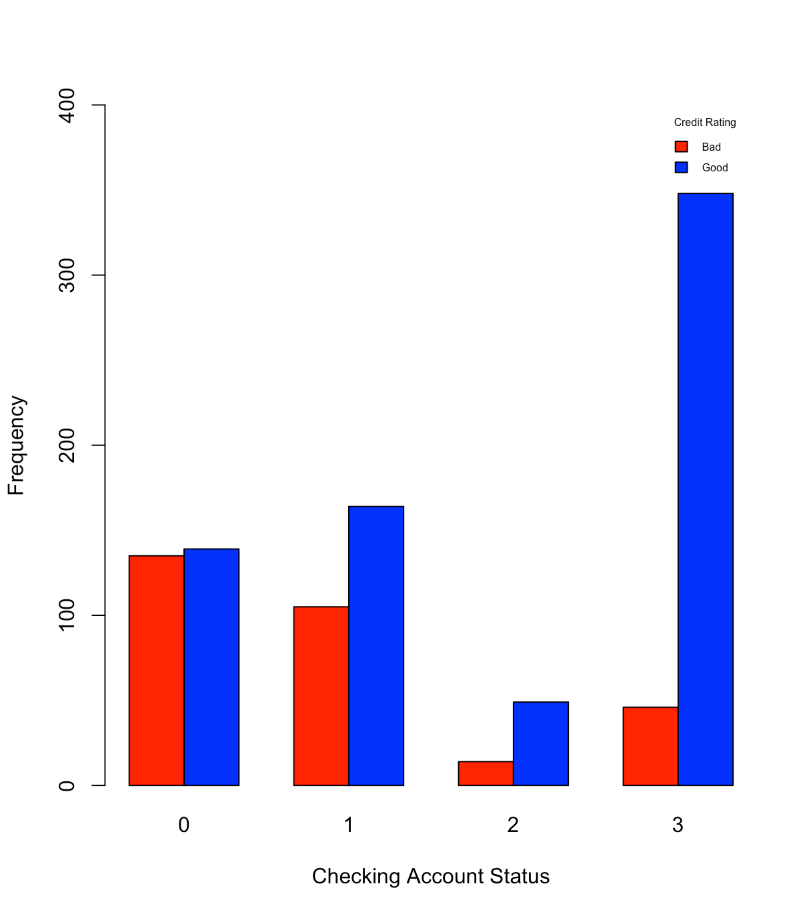
274 269 63 394

> table(German.Credit$CHK\_ACCT)/nrow(German.Credit)

0 1 2 3

0.274 0.269 0.063 0.394

> barplot(table(German.Credit$HISTORY, German.Credit$CHK\_ACCT), beside = T, xlab = "Checking Account Status", ylab = "Frequency", col= c("red", "blue") ,legend = c("Bad", "Good"), args.legend = list(title = "Credit Rating", x = "topright", cex = 0.5, bty = "n"), ylim = c(0,550))



1. DURATION

> summary(German.Credit$DURATION)

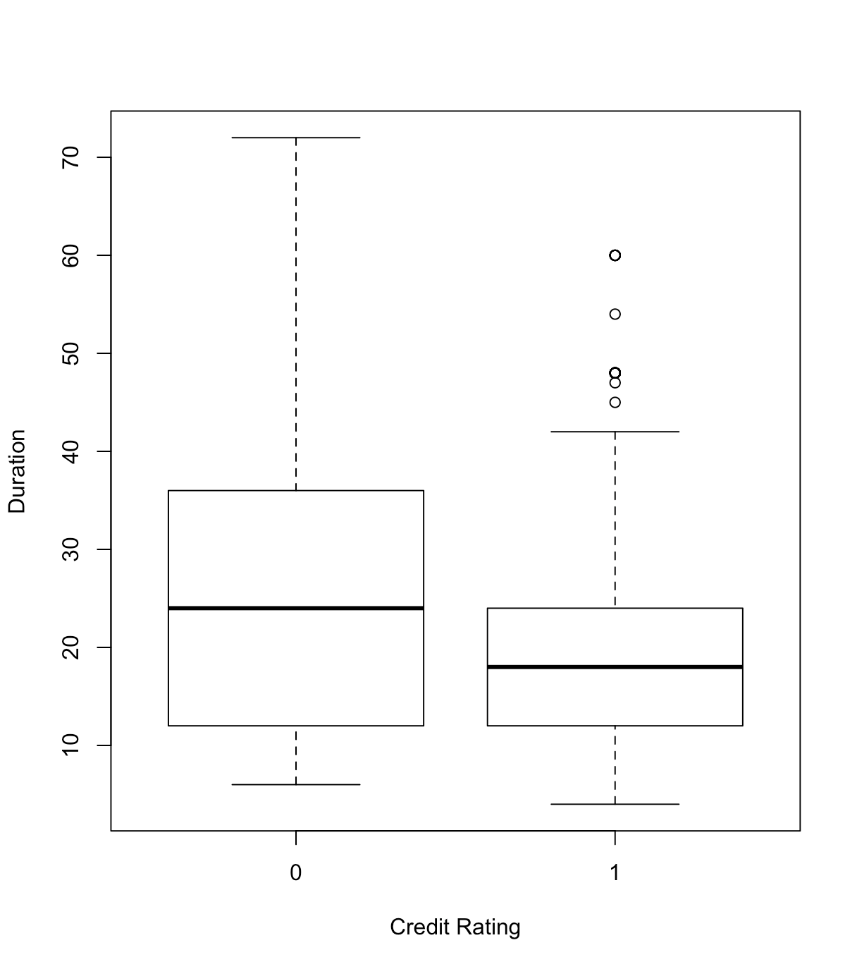
Min. 1st Qu. Median Mean 3rd Qu. Max.

4.0 12.0 18.0 20.9 24.0 72.0

> sd(German.Credit$DURATION)

[1] 12.05881

> boxplot(German.Credit$DURATION ~ German.Credit$RESPONSE, Data = German.Credit, xlab="Credit Rating", ylab="Duration")



1. HISTORY

> table(German.Credit$HISTORY)

0 1 2 3 4

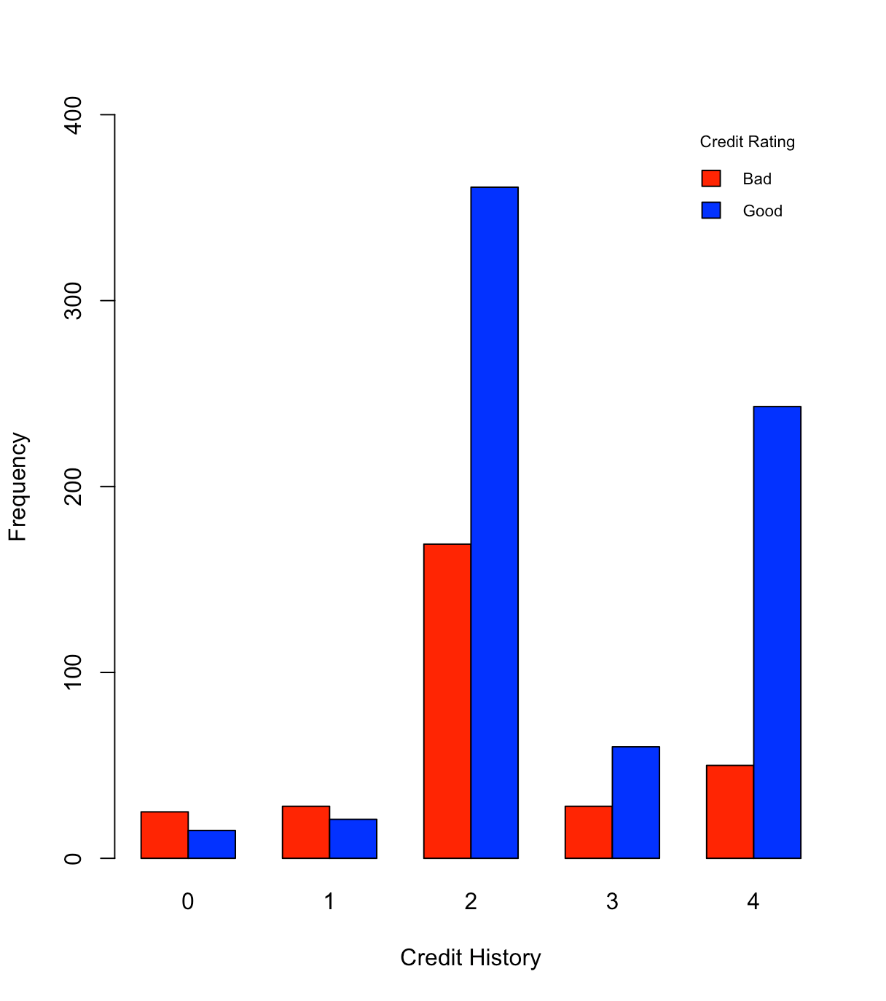
40 49 530 88 293

> table(German.Credit$HISTORY)/nrow(German.Credit)

0 1 2 3 4

0.040 0.049 0.530 0.088 0.293

> barplot(table(German.Credit$RESPONSE, German.Credit$HISTORY), beside = T, xlab = "Credit History", ylab = "Frequency", col= c("red", "blue") ,legend = c("Bad", "Good"), args.legend = list(title = "Credit Rating", x = "topright", cex = 0.7, bty = "n"), ylim = c(0,400))



1. NEW-CAR

> table(German.Credit$NEW\_CAR)

0 1

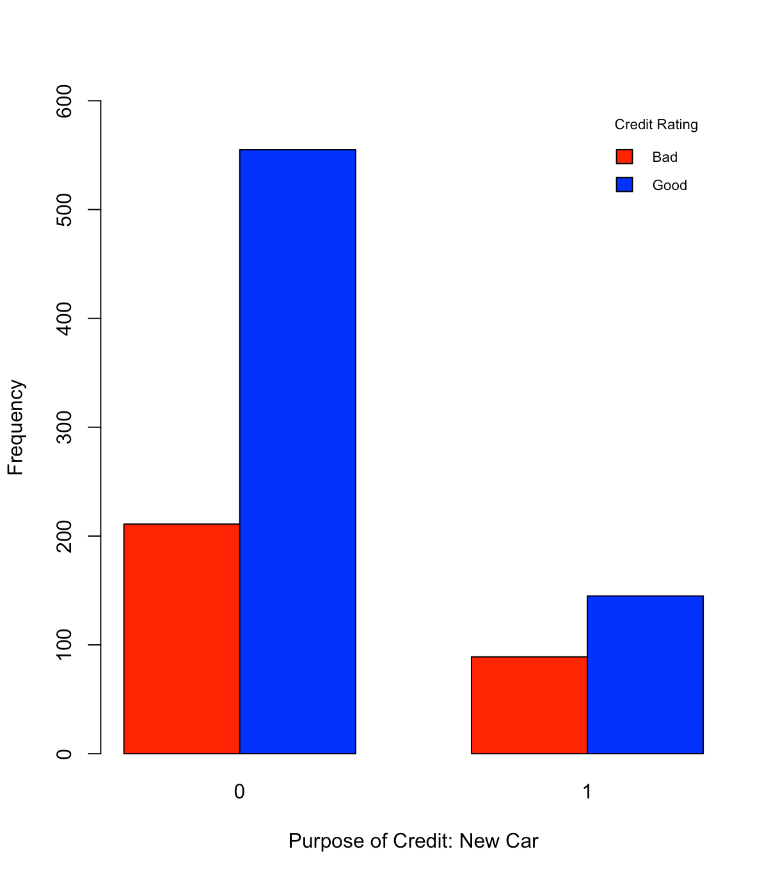
766 234

> table(German.Credit$NEW\_CAR)/nrow(German.Credit)

0 1

0.766 0.234

> barplot(table(German.Credit$RESPONSE, German.Credit$NEW\_CAR), beside = T, xlab = "Purpose of Credit: New Car", ylab = "Frequency", col= c("red", "blue") ,legend = c("Bad", "Good"), args.legend = list(title = "Credit Rating", x = "topright", cex = 0.7, bty = "n"), ylim = c(0,600))



1. USED-CAR

> table(German.Credit$USED\_CAR)

0 1

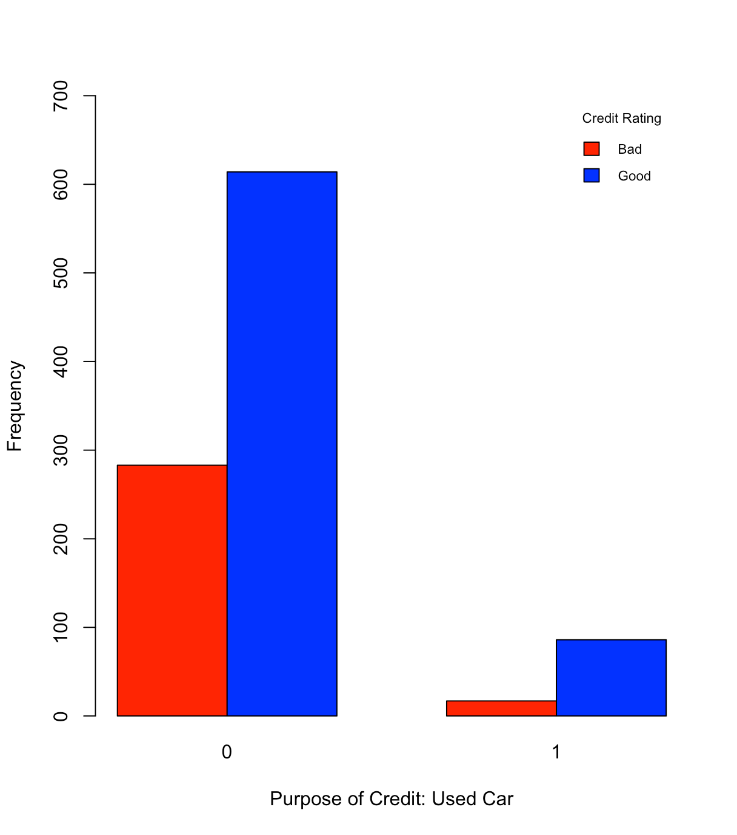
897 103

> table(German.Credit$USED\_CAR)/nrow(German.Credit)

0 1

0.897 0.103

> barplot(table(German.Credit$RESPONSE, German.Credit$USED\_CAR), beside = T, xlab = "Purpose of Credit: Used Car", ylab = "Frequency", col= c("red", "blue") ,legend = c("Bad", "Good"), args.legend = list(title = "Credit Rating", x = "topright", cex = 0.7, bty = "n"), ylim = c(0,700))



1. FURNITURE

> table(German.Credit$FURNITURE)

0 1

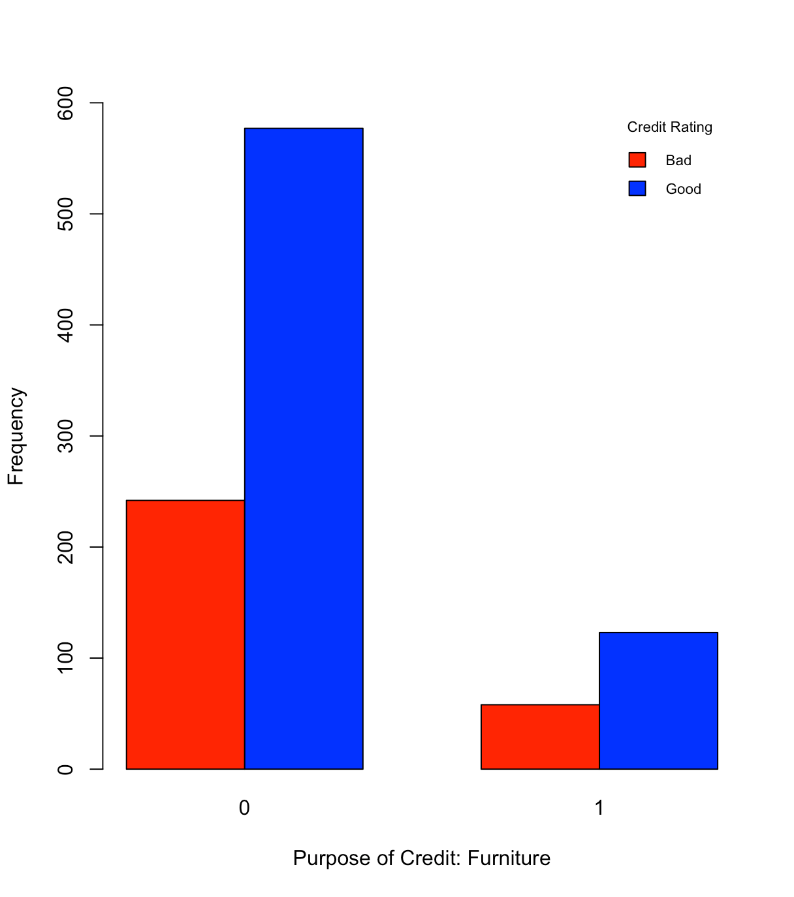
819 181

> table(German.Credit$FURNITURE)/nrow(German.Credit)

0 1

0.819 0.181

> barplot(table(German.Credit$RESPONSE, German.Credit$FURNITURE), beside = T, xlab = "Purpose of Credit: Furniture", ylab = "Frequency", col= c("red", "blue") ,legend = c("Bad", "Good"), args.legend = list(title = "Credit Rating", x = "topright", cex = 0.7, bty = "n"), ylim = c(0,600))



1. RADIO/TV

> table(German.Credit$RADIO.TV)

0 1

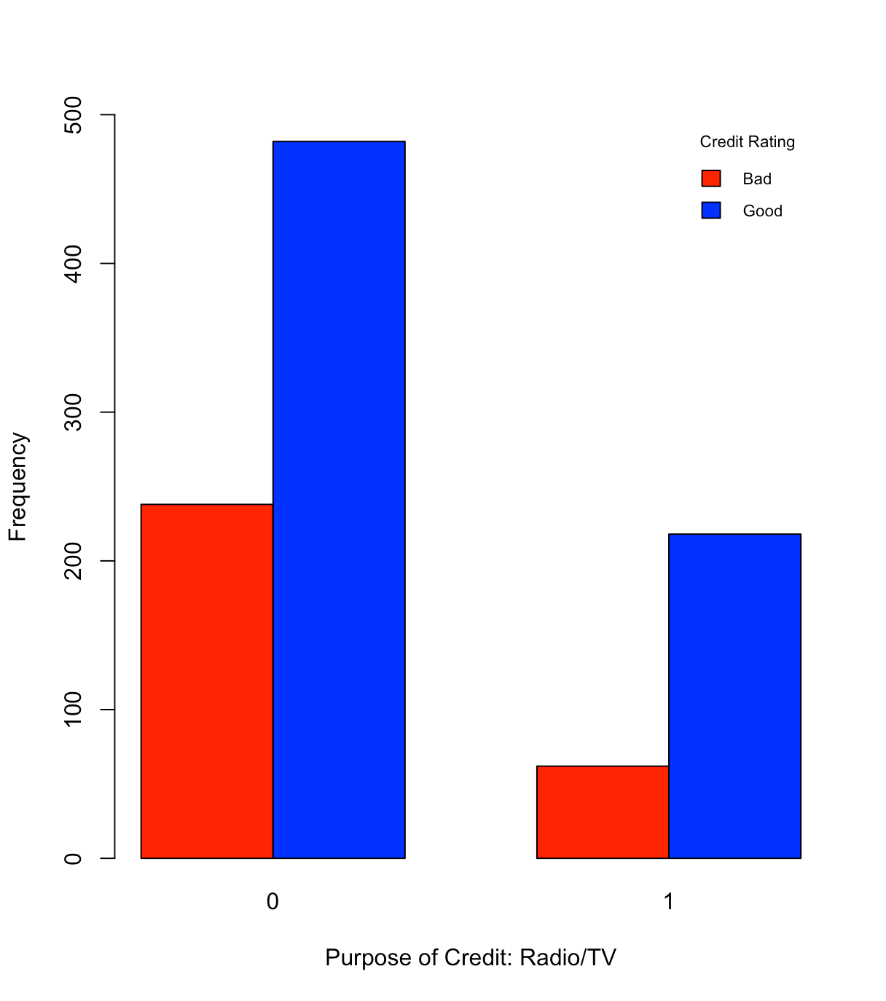
720 280

> table(German.Credit$RADIO.TV)/nrow(German.Credit)

0 1

0.72 0.28

> barplot(table(German.Credit$RESPONSE, German.Credit$RADIO.TV), beside = T, xlab = "Purpose of Credit: Radio/TV", ylab = "Frequency", col= c("red", "blue") ,legend = c("Bad", "Good"), args.legend = list(title = "Credit Rating", x = "topright", cex = 0.7, bty = "n"), ylim = c(0,500))



1. EDUCATION

> table(German.Credit$EDUCATION)

0 1

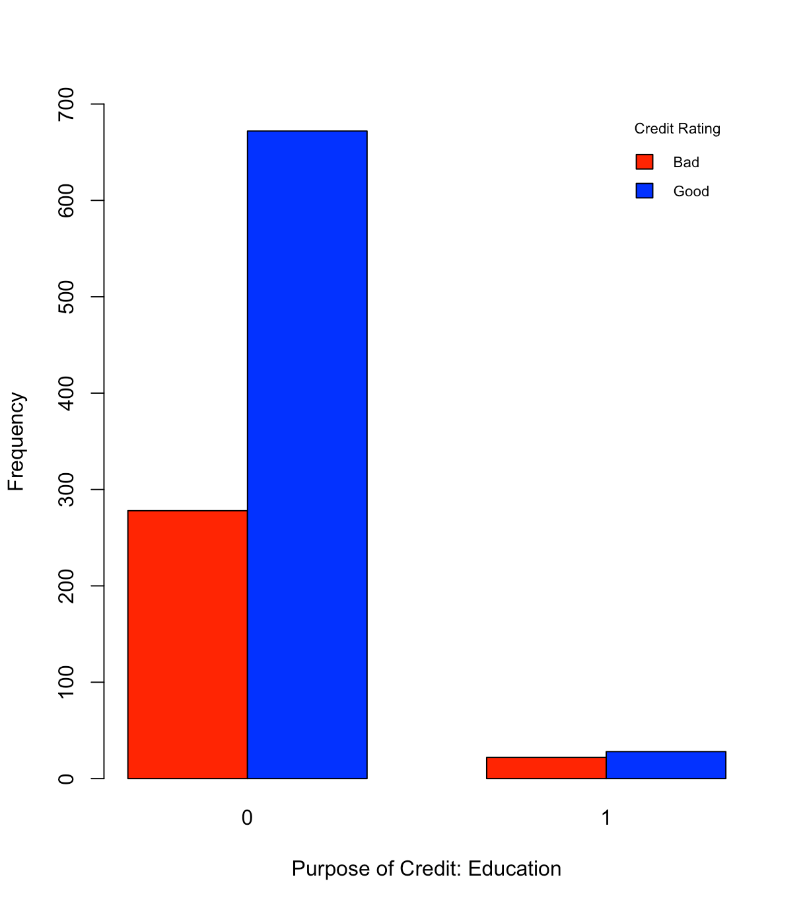
950 50

> table(German.Credit$EDUCATION)/nrow(German.Credit)

0 1

0.95 0.05

> barplot(table(German.Credit$RESPONSE, German.Credit$EDUCATION), beside = T, xlab = "Purpose of Credit: Education", ylab = "Frequency", col= c("red", "blue") ,legend = c("Bad", "Good"), args.legend = list(title = "Credit Rating", x = "topright", cex = 0.7, bty = "n"), ylim = c(0,700))



1. RETRAINING

> table(German.Credit$RETRAINING)

0 1

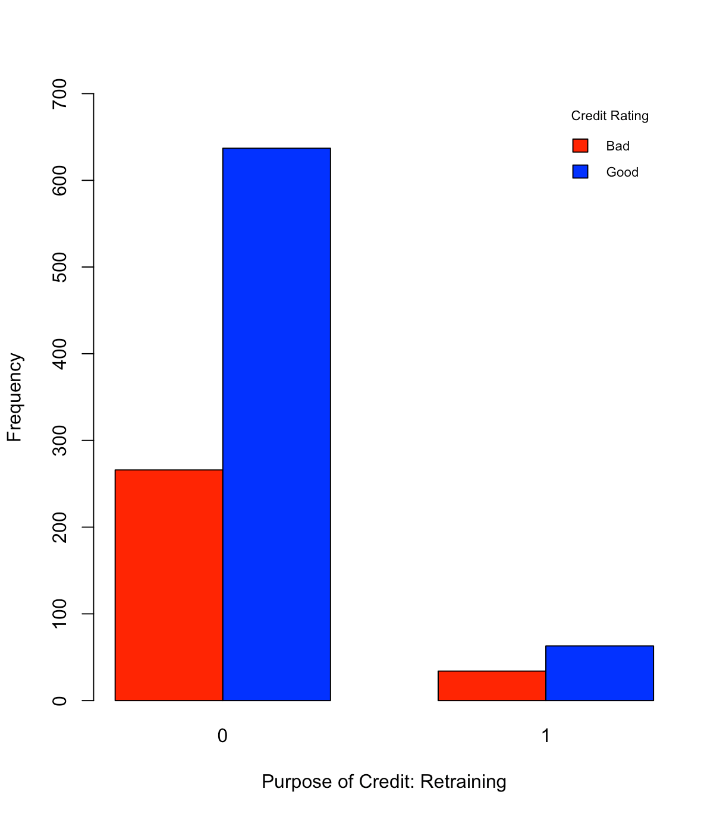
903 97

> table(German.Credit$RETRAINING)/nrow(German.Credit)

0 1

0.903 0.097

> barplot(table(German.Credit$RESPONSE, German.Credit$RETRAINING), beside = T, xlab = "Purpose of Credit: Retraining", ylab = "Frequency", col= c("red", "blue") ,legend = c("Bad", "Good"), args.legend = list(title = "Credit Rating", x = "topright", cex = 0.7, bty = "n"), ylim = c(0,700))



1. AMOUNT

> summary(German.Credit$AMOUNT)

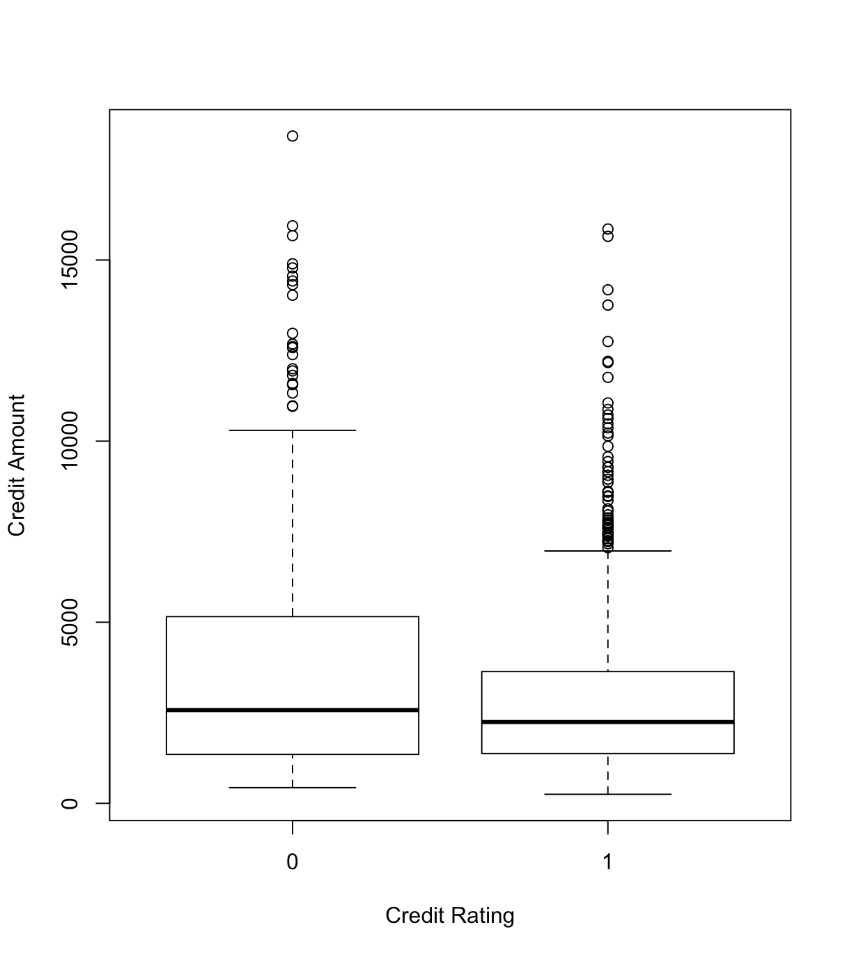
Min. 1st Qu. Median Mean 3rd Qu. Max.

250 1366 2320 3271 3972 18420

> sd(German.Credit$AMOUNT)

[1] 2822.737

> boxplot(German.Credit$AMOUNT ~ German.Credit$RESPONSE, Data = German.Credit, xlab="Credit Rating", ylab="Credit Amount")



1. SAV\_ACCT

> table(German.Credit$SAV\_ACCT)

0 1 2 3 4

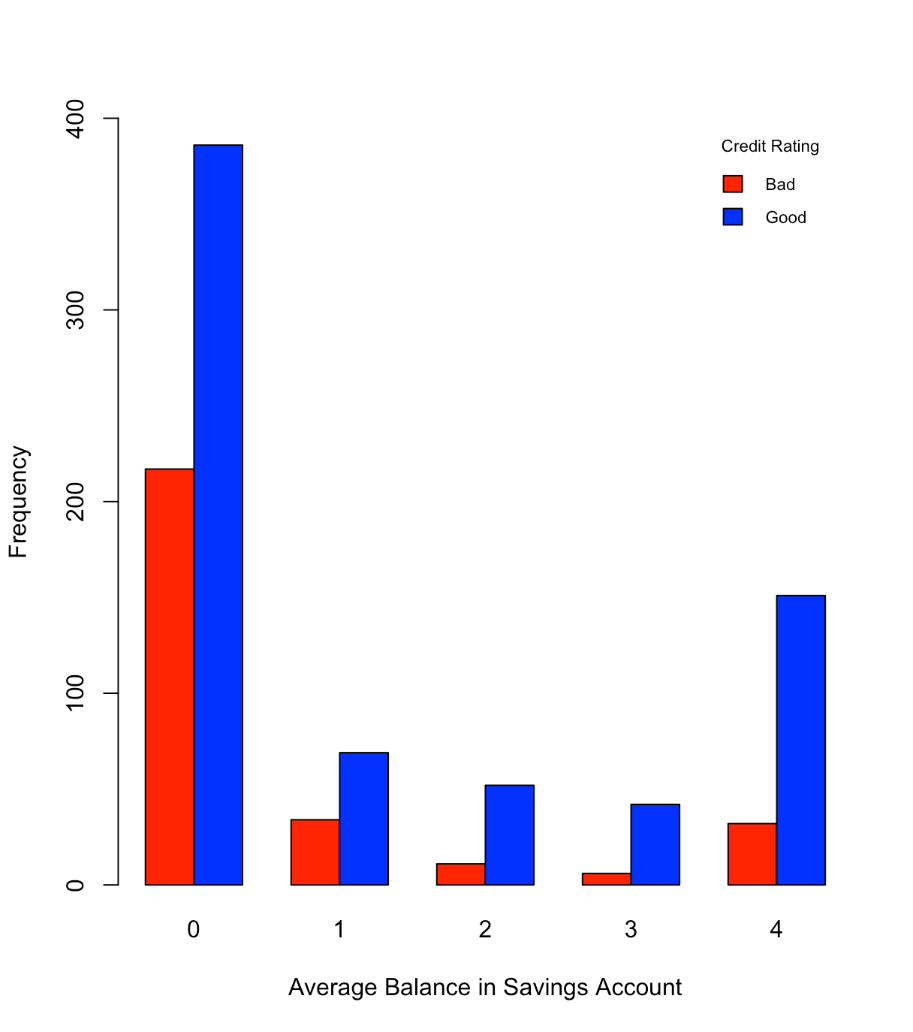
603 103 63 48 183

> table(German.Credit$SAV\_ACCT)/nrow(German.Credit)

0 1 2 3 4

0.603 0.103 0.063 0.048 0.183

> barplot(table(German.Credit$RESPONSE, German.Credit$SAV\_ACCT), beside = T, xlab = "Average Balance in Savings Account", ylab = "Frequency", col= c("red", "blue") ,legend = c("Bad", "Good"), args.legend = list(title = "Credit Rating", x = "topright", cex = 0.7, bty = "n"), ylim = c(0,400))



1. EMPLOYMENT

> table(German.Credit$EMPLOYMENT)

0 1 2 3 4

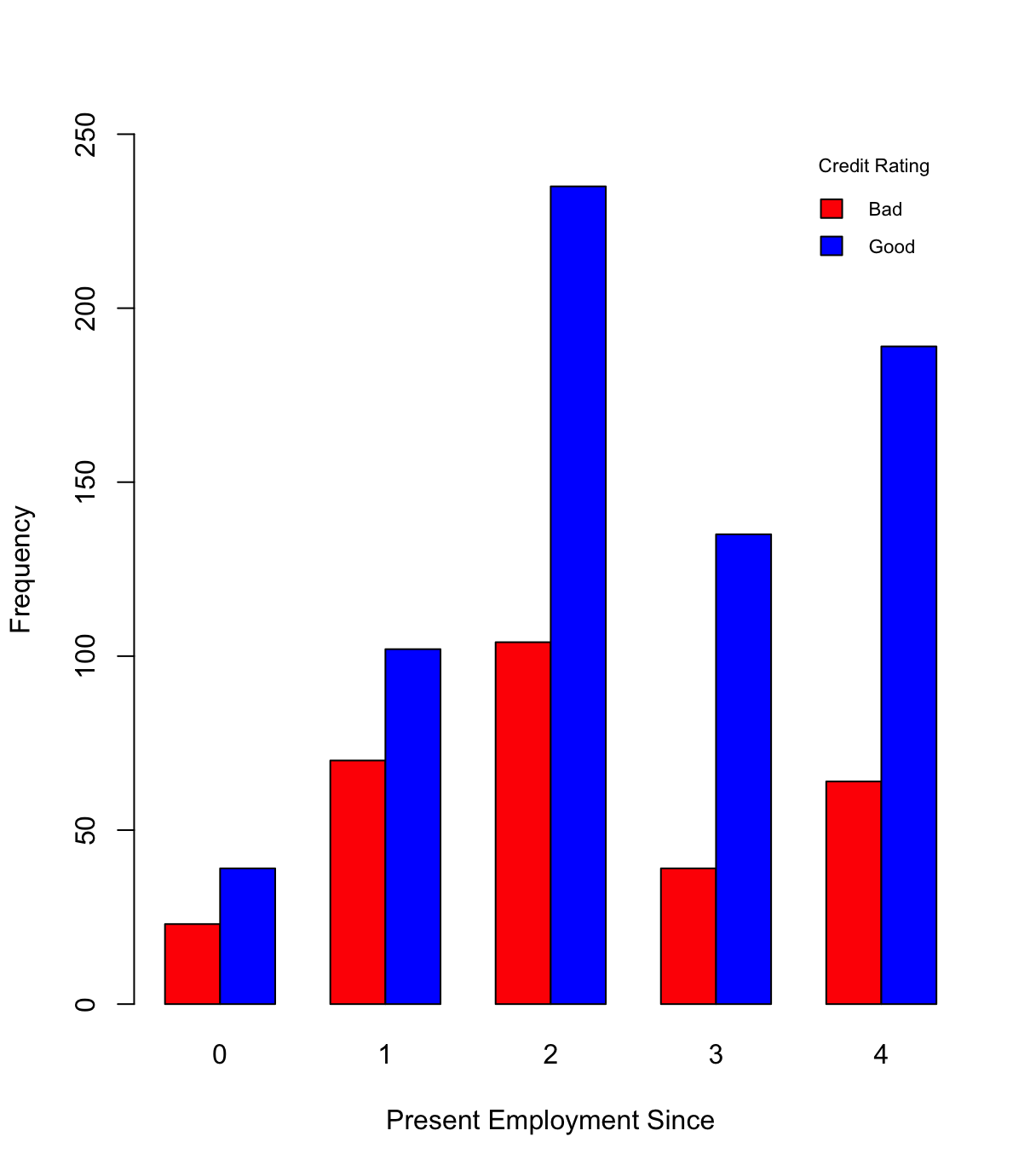
62 172 339 174 253

> table(German.Credit$EMPLOYMENT)/nrow(German.Credit)

0 1 2 3 4

0.062 0.172 0.339 0.174 0.253

> barplot(table(German.Credit$RESPONSE, German.Credit$EMPLOYMENT), beside = T, xlab = "Present Employment Since", ylab = "Frequency", col= c("red", "blue") ,legend = c("Bad", "Good"), args.legend = list(title = "Credit Rating", x = "topright", cex = 0.7, bty = "n"), ylim = c(0,250))



1. INSTALL\_RATE

> summary(German.Credit$INSTALL\_RATE)

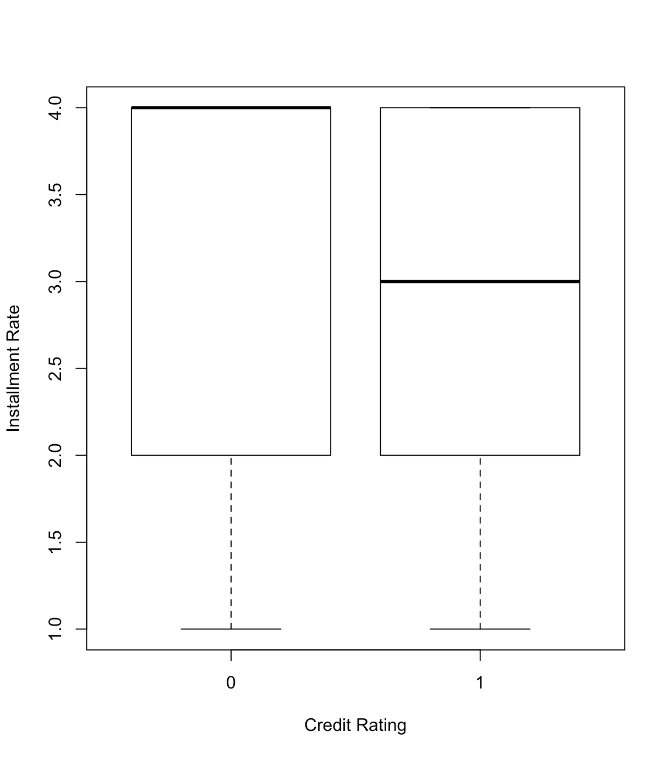
Min. 1st Qu. Median Mean 3rd Qu. Max.

1.000 2.000 3.000 2.973 4.000 4.000

> sd(German.Credit$INSTALL\_RATE)

[1] 1.118715

> boxplot(German.Credit$INSTALL\_RATE ~ German.Credit$RESPONSE, Data = German.Credit, xlab="Credit Rating", ylab="Installment Rate")



1. MALE\_DIV

> table(German.Credit$MALE\_DIV)

0 1

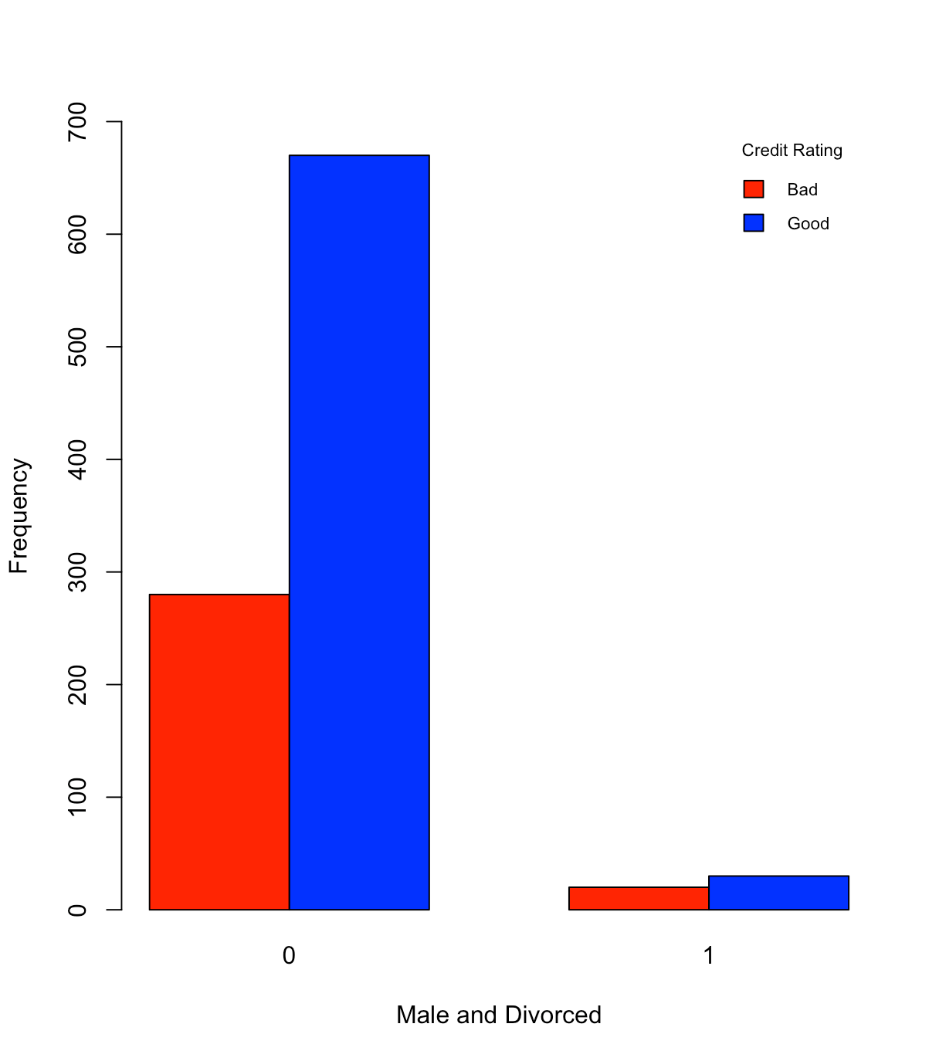
950 50

> table(German.Credit$MALE\_DIV)/nrow(German.Credit)

0 1

0.95 0.05

> barplot(table(German.Credit$RESPONSE, German.Credit$MALE\_DIV), beside = T, xlab = "Male and Divorced", ylab = "Frequency", col= c("red", "blue") ,legend = c("Bad", "Good"), args.legend = list(title = "Credit Rating", x = "topright", cex = 0.7, bty = "n"), ylim = c(0,700))



1. MALE\_SINGLE

> table(German.Credit$MALE\_SINGLE)

0 1

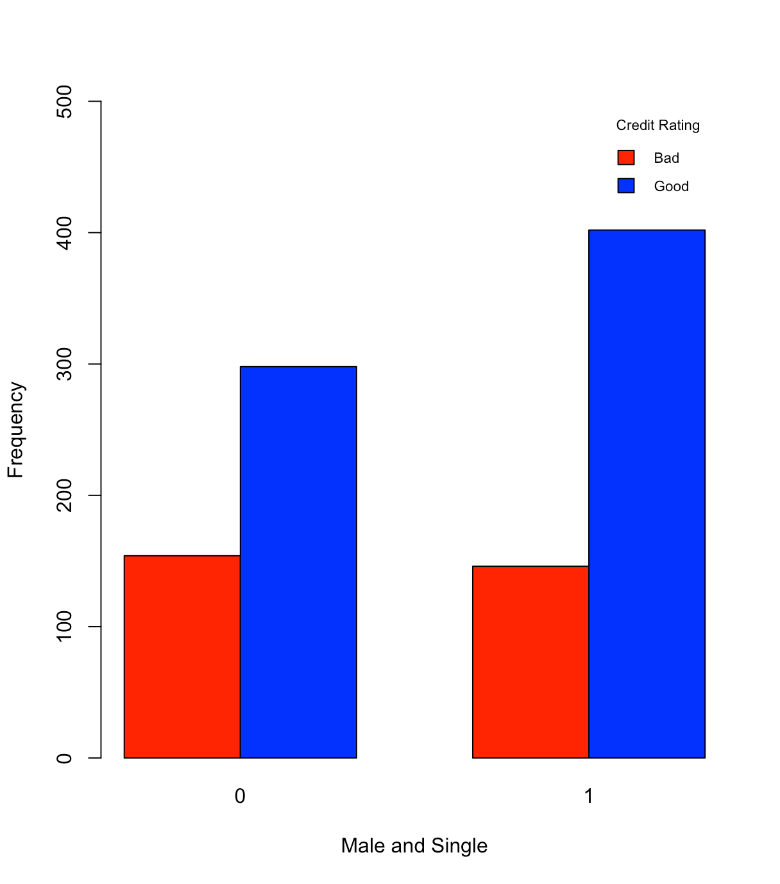
452 548

> table(German.Credit$MALE\_SINGLE)/nrow(German.Credit)

0 1

0.452 0.548

> barplot(table(German.Credit$RESPONSE, German.Credit$MALE\_SINGLE), beside = T, xlab = "Male and Single", ylab = "Frequency", col= c("red", "blue") ,legend = c("Bad", "Good"), args.legend = list(title = "Credit Rating", x = "topright", cex = 0.7, bty = "n"), ylim = c(0,500))



1. MALE\_MAR\_WID

> table(German.Credit$MALE\_MAR\_or\_WID)

0 1

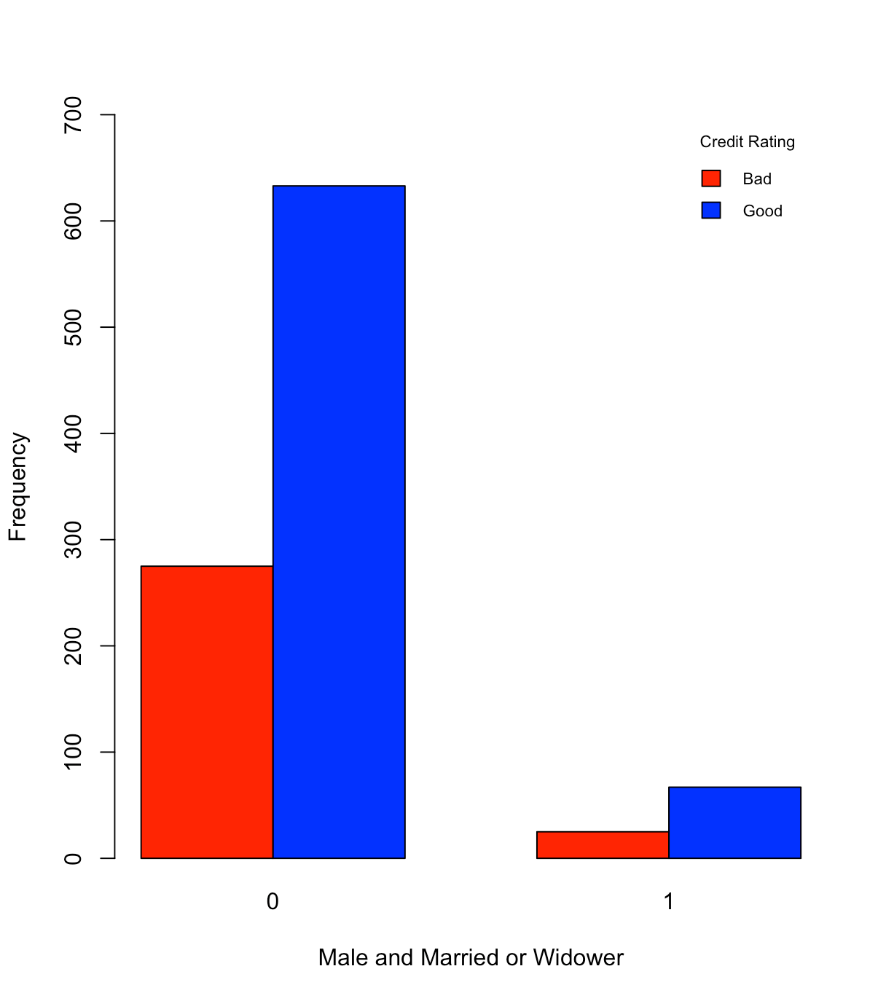
908 92

> table(German.Credit$MALE\_MAR\_or\_WID)/nrow(German.Credit)

0 1

0.908 0.092

> barplot(table(German.Credit$RESPONSE, German.Credit$MALE\_MAR\_or\_WID), beside = T, xlab = "Male and Married or Widower", ylab = "Frequency", col= c("red", "blue") ,legend = c("Bad", "Good"), args.legend = list(title = "Credit Rating", x = "topright", cex = 0.7, bty = "n"), ylim = c(0,700))



1. CO\_APPLICANT

> table(German.Credit$CO.APPLICANT)

0 1

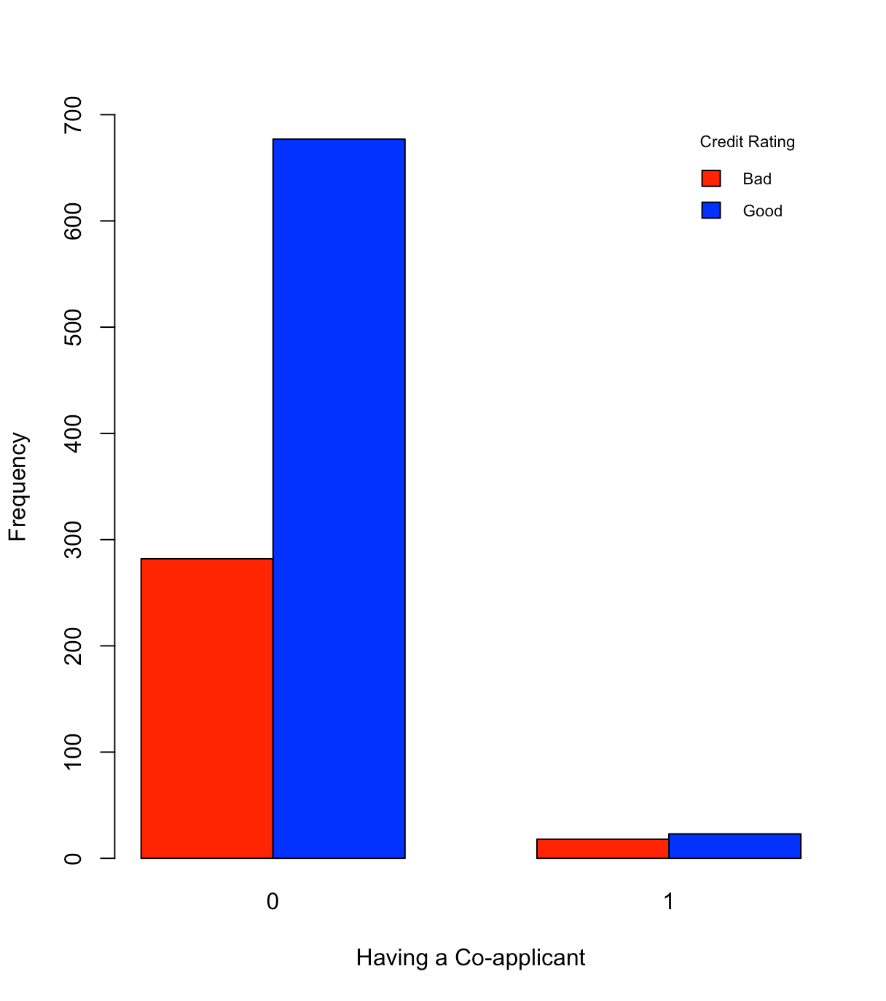
959 41

> table(German.Credit$CO.APPLICANT)/nrow(German.Credit)

0 1

0.959 0.041

> barplot(table(German.Credit$RESPONSE, German.Credit$CO.APPLICANT), beside = T, xlab = "Having a Co-applicant", ylab = "Frequency", col= c("red", "blue") ,legend = c("Bad", "Good"), args.legend = list(title = "Credit Rating", x = "topright", cex = 0.7, bty = "n"), ylim = c(0,700))



1. GUARANTOR

> table(German.Credit$GUARANTOR)

0 1

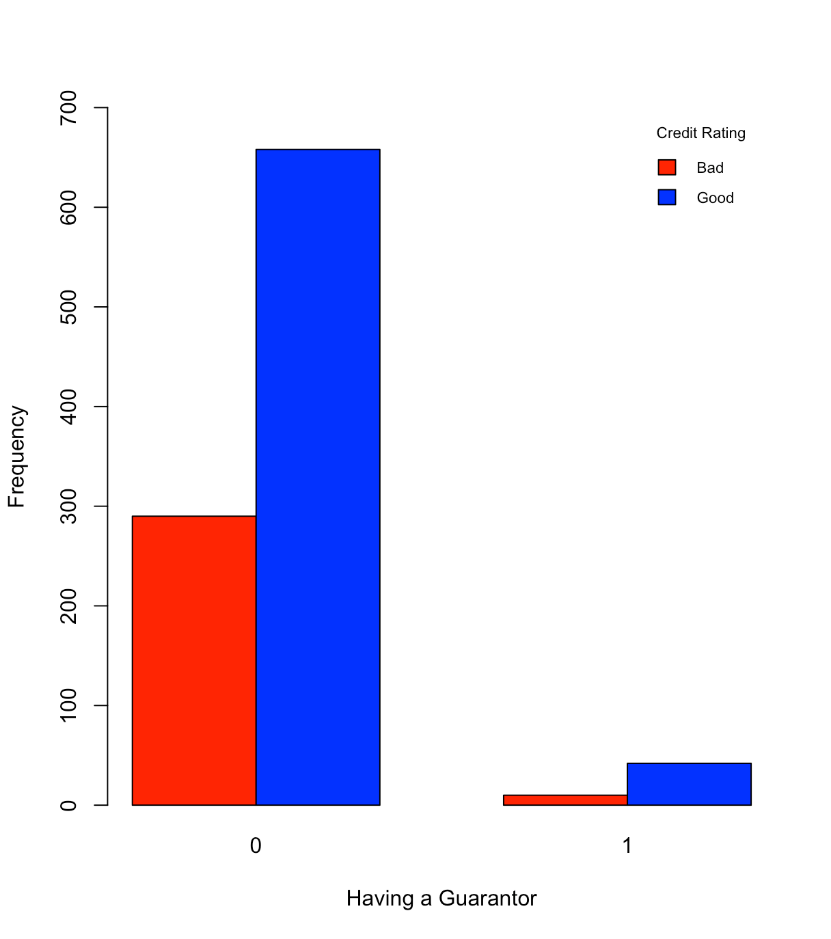
948 52

> table(German.Credit$GUARANTOR)/nrow(German.Credit)

0 1

0.948 0.052

> barplot(table(German.Credit$RESPONSE, German.Credit$GUARANTOR), beside = T, xlab = "Having a Guarantor", ylab = "Frequency", col= c("red", "blue") ,legend = c("Bad", "Good"), args.legend = list(title = "Credit Rating", x = "topright", cex = 0.7, bty = "n"), ylim = c(0,700))



1. PRESENT\_RESIDENT

> table(German.Credit$PRESENT\_RESIDENT)

1 2 3 4

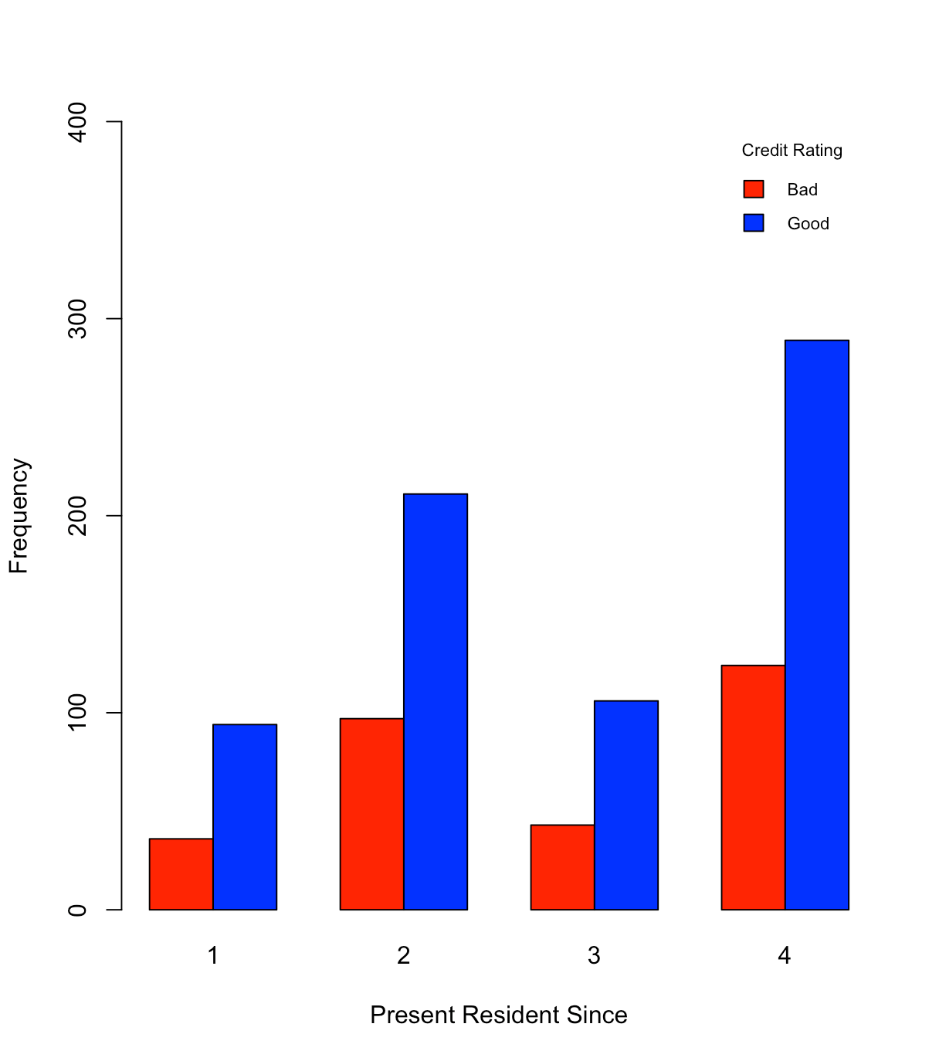
130 308 149 413

> table(German.Credit$PRESENT\_RESIDENT)/nrow(German.Credit)

1 2 3 4

0.130 0.308 0.149 0.413

> barplot(table(German.Credit$RESPONSE, German.Credit$PRESENT\_RESIDENT), beside = T, xlab = "Present Resident Since", ylab = "Frequency", col= c("red", "blue") ,legend = c("Bad", "Good"), args.legend = list(title = "Credit Rating", x = "topright", cex = 0.7, bty = "n"), ylim = c(0,400))



1. REAL\_ESTATE

> table(German.Credit$REAL\_ESTATE)

0 1

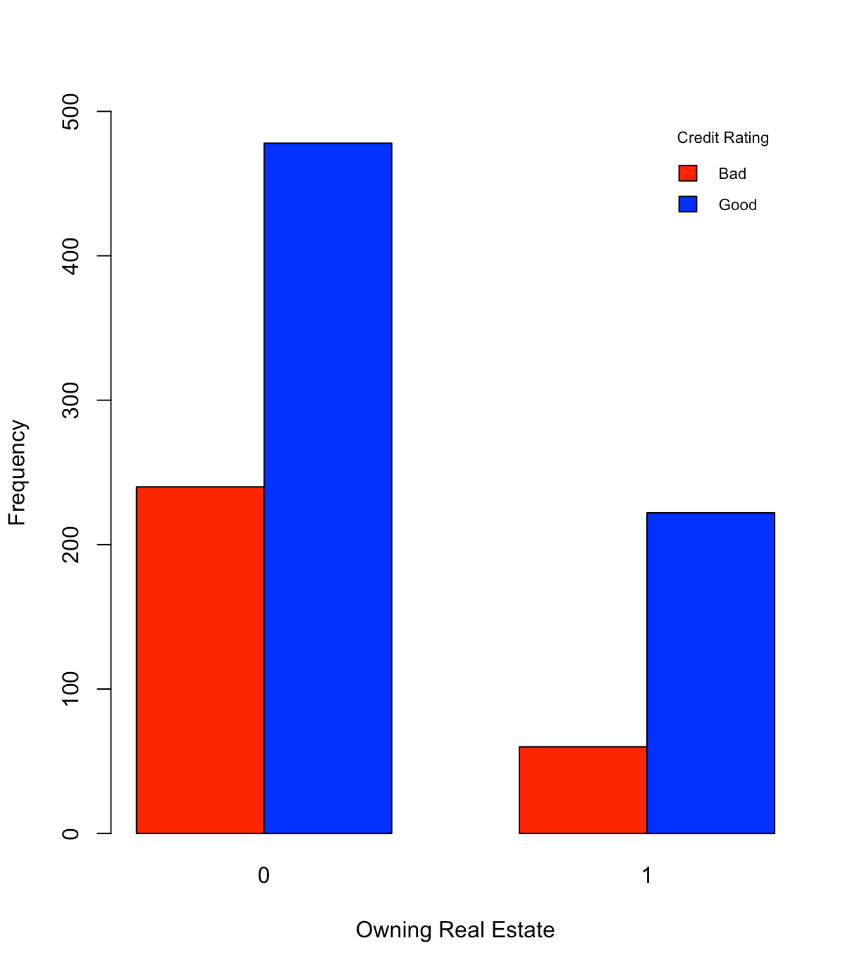
718 282

> table(German.Credit$REAL\_ESTATE)/nrow(German.Credit)

0 1

0.718 0.282

> barplot(table(German.Credit$RESPONSE, German.Credit$REAL\_ESTATE), beside = T, xlab = "Owning Real Estate", ylab = "Frequency", col= c("red", "blue") ,legend = c("Bad", "Good"), args.legend = list(title = "Credit Rating", x = "topright", cex = 0.7, bty = "n"), ylim = c(0,500))



1. PROP\_UNKN\_NONE

> table(German.Credit$PROP\_UNKN\_NONE)

0 1

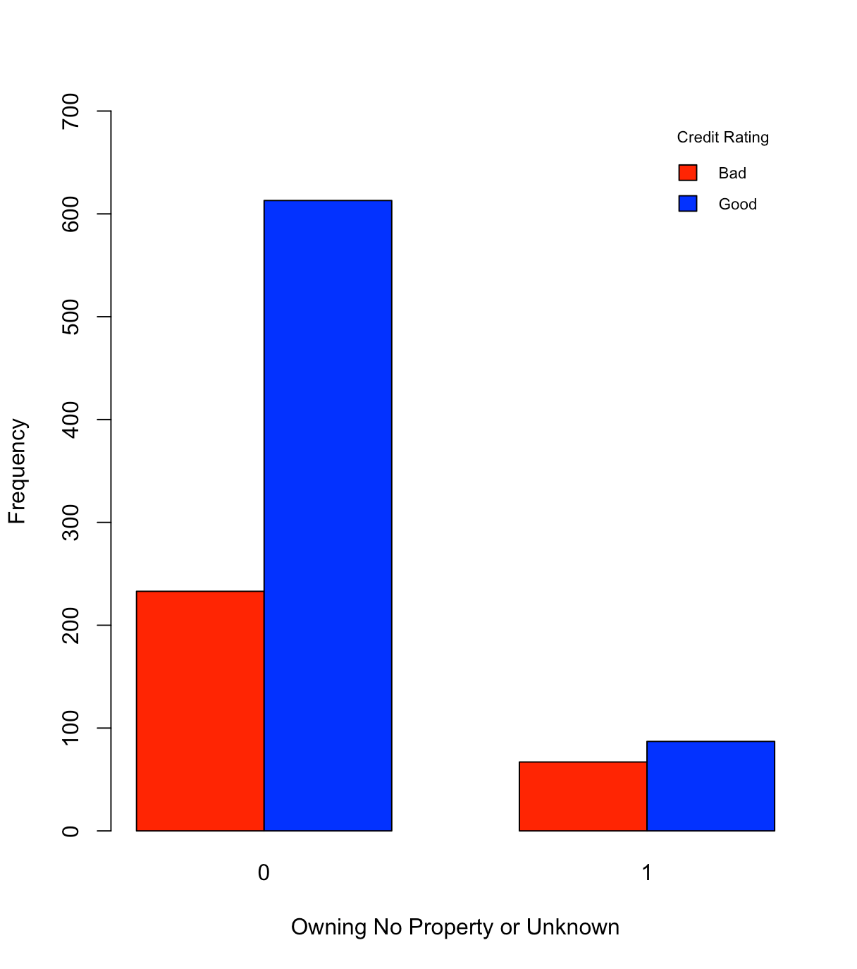
846 154

> table(German.Credit$PROP\_UNKN\_NONE)/nrow(German.Credit)

0 1

0.846 0.154

> barplot(table(German.Credit$RESPONSE, German.Credit$PROP\_UNKN\_NONE), beside = T, xlab = "Owning No Property or Unknown", ylab = "Frequency", col= c("red", "blue") ,legend = c("Bad", "Good"), args.legend = list(title = "Credit Rating", x = "topright", cex = 0.7, bty = "n"), ylim = c(0,700))



1. AGE

> summary(German.Credit$AGE)

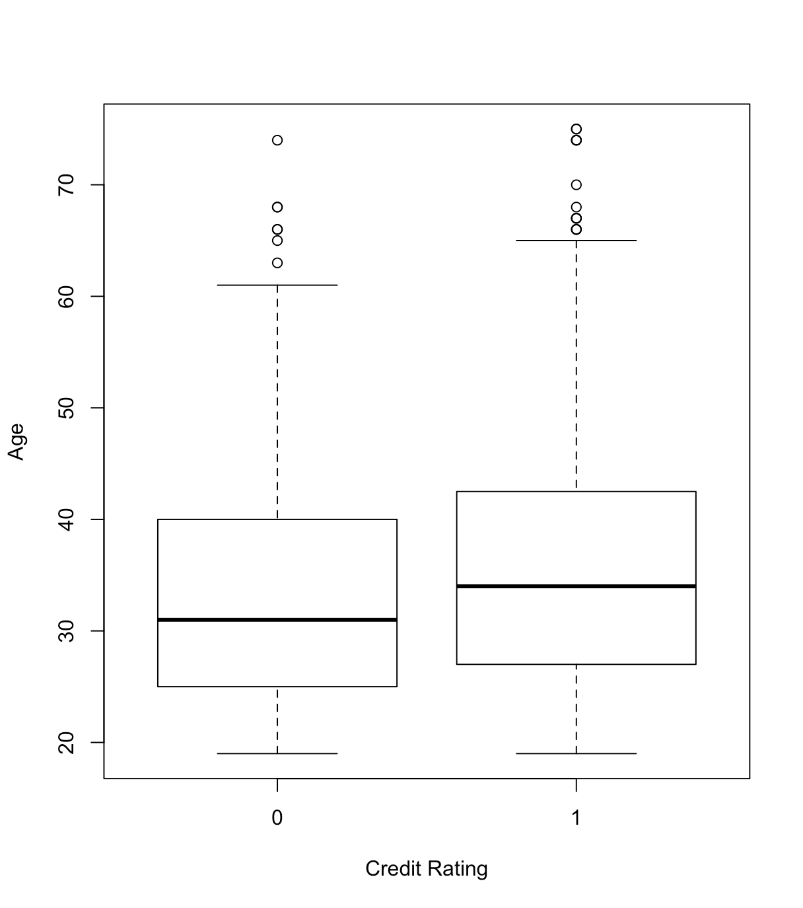
Min. 1st Qu. Median Mean 3rd Qu. Max.

19.00 27.00 33.00 35.55 42.00 75.00

> sd(German.Credit$AGE)

[1] 11.37547

> boxplot(German.Credit$AGE ~ German.Credit$RESPONSE, Data = German.Credit, xlab="Credit Rating", ylab="Age")



1. OTHER\_INSTALL

> table(German.Credit$OTHER\_INSTALL)

0 1

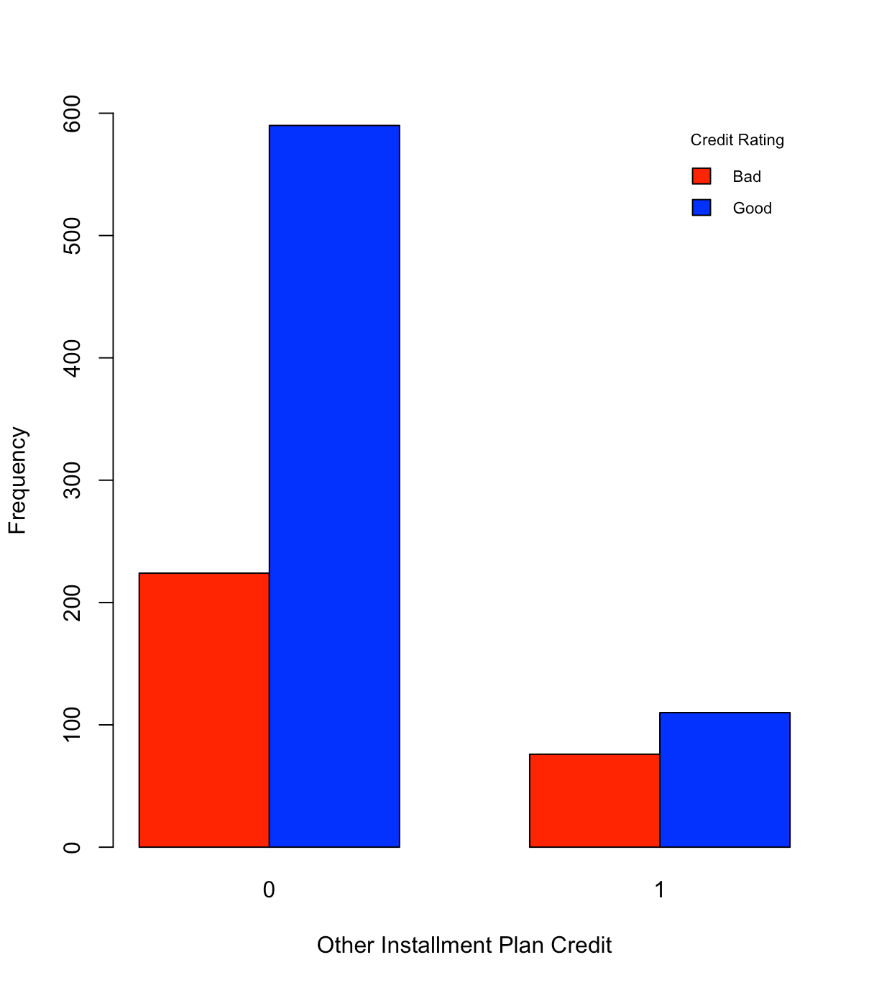
814 186

> table(German.Credit$OTHER\_INSTALL)/nrow(German.Credit)

0 1

0.814 0.186

> barplot(table(German.Credit$RESPONSE, German.Credit$OTHER\_INSTALL), beside = T, xlab = "Other Installment Plan Credit", ylab = "Frequency", col= c("red", "blue") ,legend = c("Bad", "Good"), args.legend = list(title = "Credit Rating", x = "topright", cex = 0.7, bty = "n"), ylim = c(0,600))



1. RENT

> table(German.Credit$RENT)

0 1

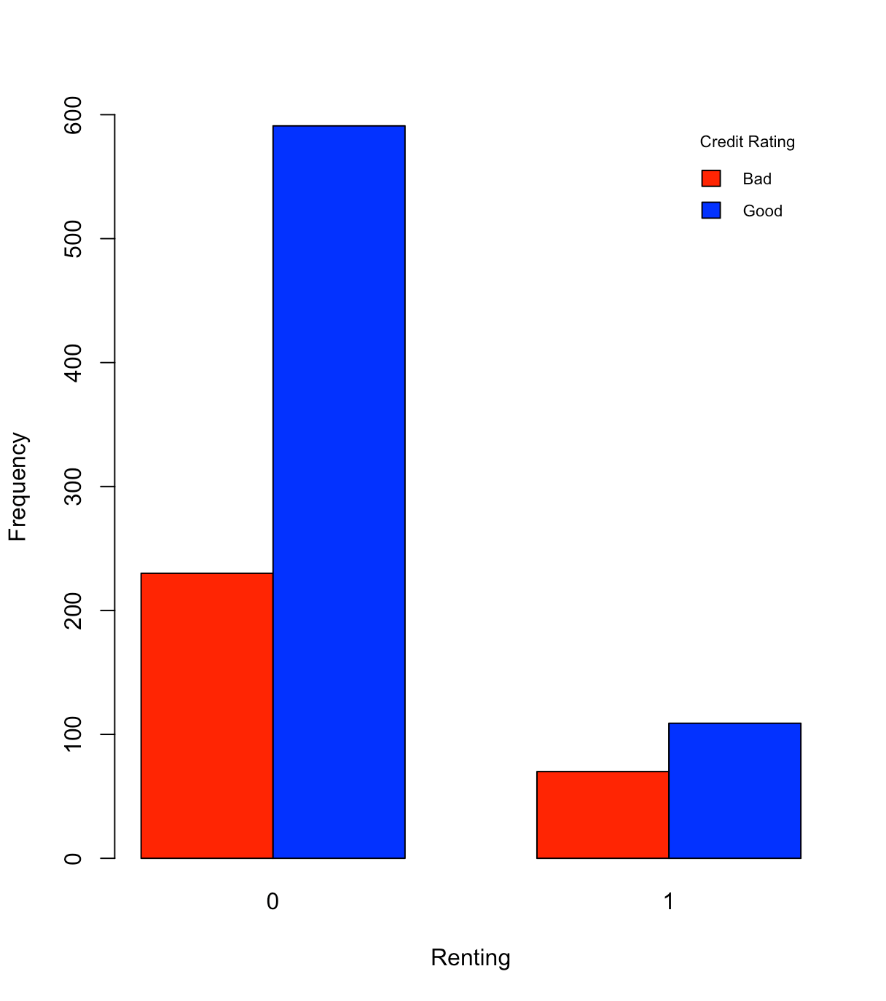
821 179

> table(German.Credit$RENT)/nrow(German.Credit)

0 1

0.821 0.179

> barplot(table(German.Credit$RESPONSE, German.Credit$RENT), beside = T, xlab = "Renting", ylab = "Frequency", col= c("red", "blue") ,legend = c("Bad", "Good"), args.legend = list(title = "Credit Rating", x = "topright", cex = 0.7, bty = "n"), ylim = c(0,600))



1. OWN\_RES

> table(German.Credit$OWN\_RES)

0 1

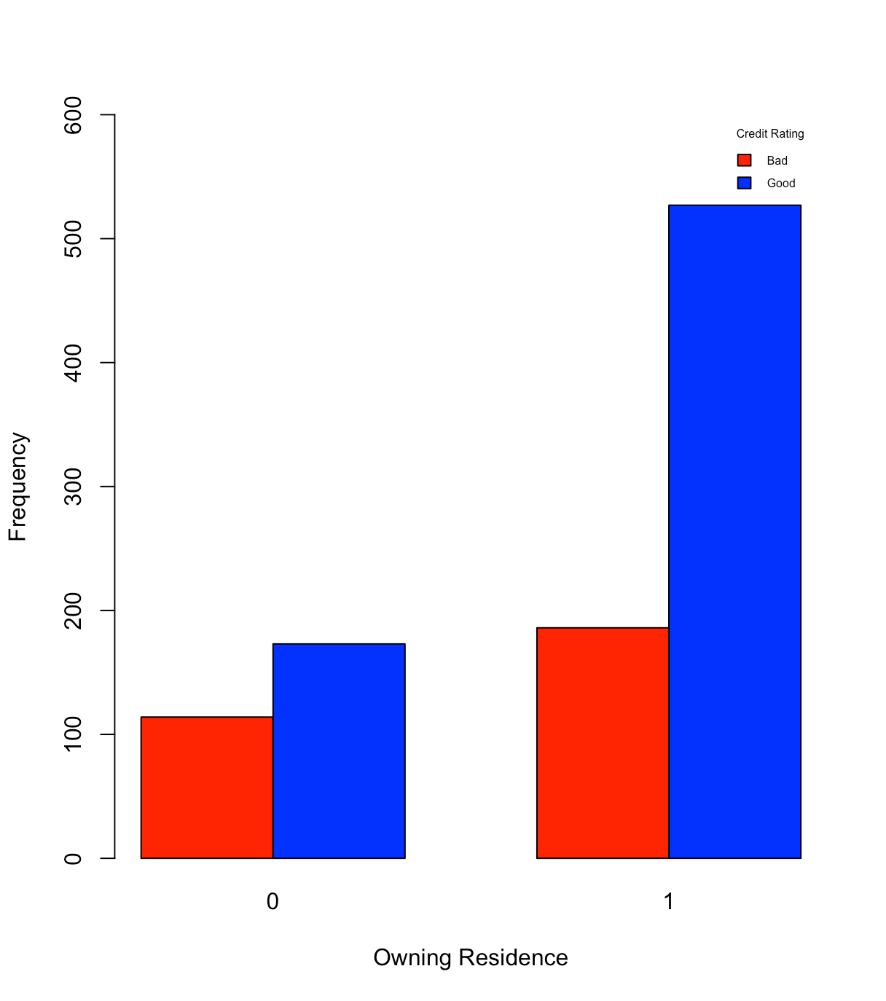
287 713

> table(German.Credit$OWN\_RES)/nrow(German.Credit)

0 1

0.287 0.713

> barplot(table(German.Credit$RESPONSE, German.Credit$OWN\_RES), beside = T, xlab = "Owning Residence", ylab = "Frequency", col= c("red", "blue") ,legend = c("Bad", "Good"), args.legend = list(title = "Credit Rating", x = "topright", cex = 0.5, bty = "n"), ylim = c(0,600))



1. NUM\_CREDITS

> summary(German.Credit$NUM\_CREDITS)

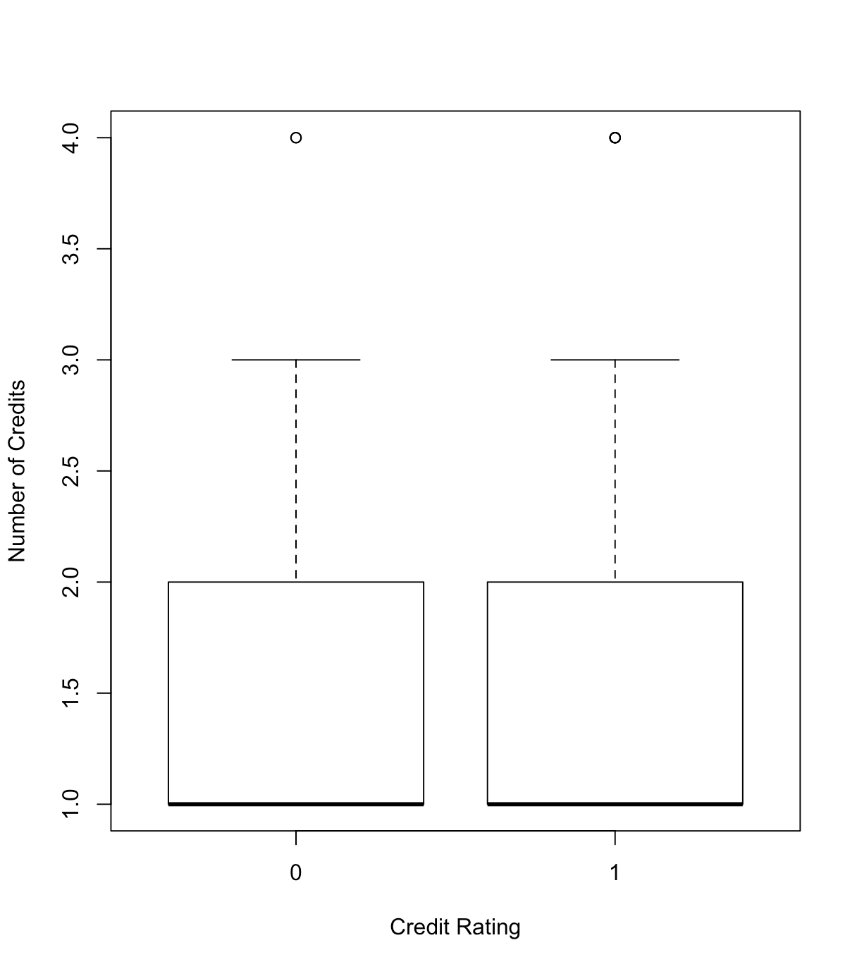
Min. 1st Qu. Median Mean 3rd Qu. Max.

1.000 1.000 1.000 1.407 2.000 4.000

> sd(German.Credit$NUM\_CREDITS)

[1] 0.5776545

> boxplot(German.Credit$NUM\_CREDITS ~ German.Credit$RESPONSE, Data = German.Credit, xlab="Credit Rating", ylab="Number of Credits")



1. JOB

> table(German.Credit$JOB)

0 1 2 3

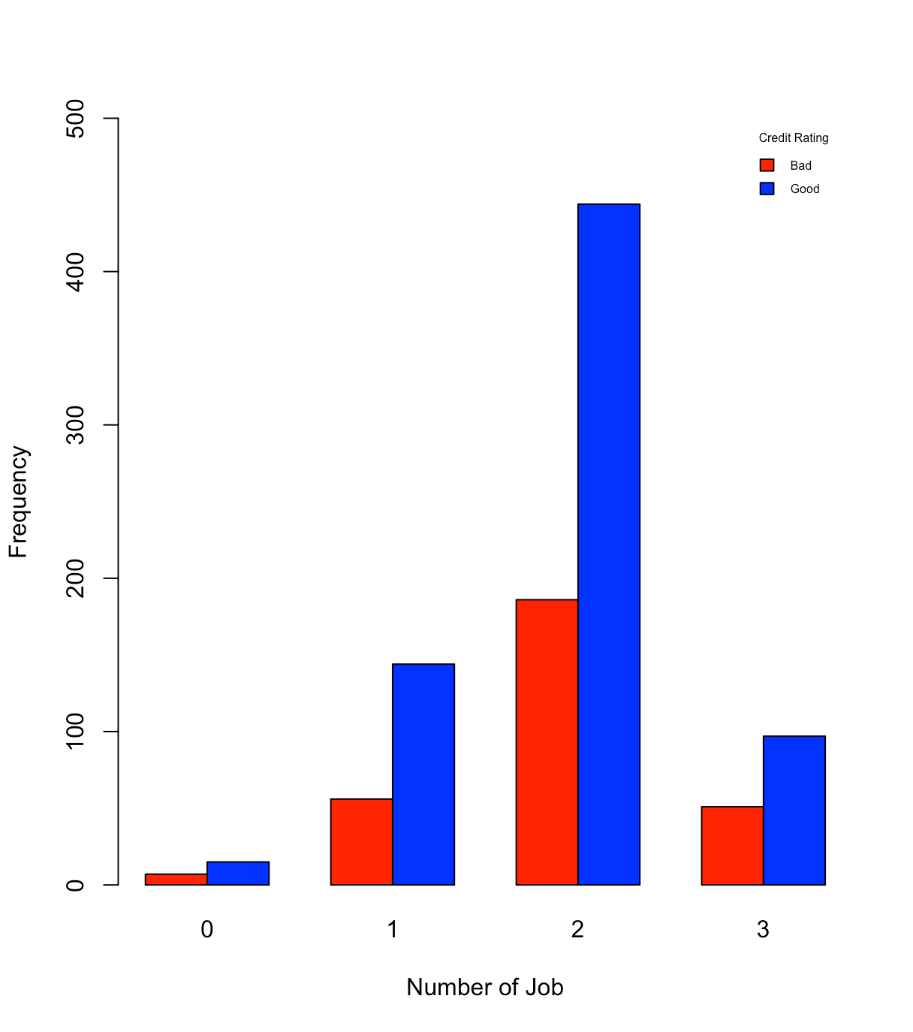
22 200 630 148

> table(German.Credit$JOB)/nrow(German.Credit)

0 1 2 3

0.022 0.200 0.630 0.148

> barplot(table(German.Credit$RESPONSE, German.Credit$JOB), beside = T, xlab = "Number of Job", ylab = "Frequency", col= c("red", "blue") ,legend = c("Bad", "Good"), args.legend = list(title = "Credit Rating", x = "topright", cex = 0.5, bty = "n"), ylim = c(0,500))



1. NUM\_DEPENDENTS

> summary(German.Credit$NUM\_DEPENDENTS)

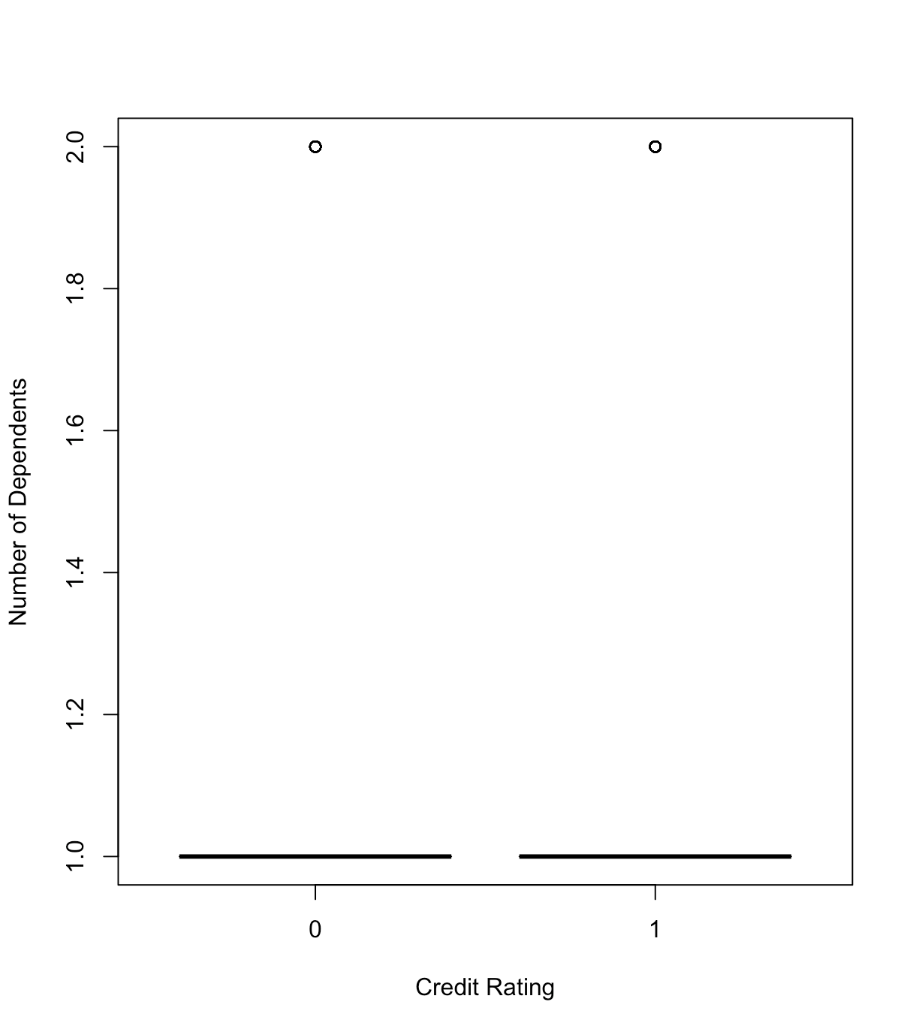
Min. 1st Qu. Median Mean 3rd Qu. Max.

1.000 1.000 1.000 1.155 1.000 2.000

> sd(German.Credit$NUM\_DEPENDENTS)

[1] 0.3620858

> boxplot(German.Credit$NUM\_DEPENDENTS ~ German.Credit$RESPONSE, Data = German.Credit, xlab="Credit Rating", ylab="Number of Dependents")



1. TELEPHONE

> table(German.Credit$TELEPHONE)

0 1

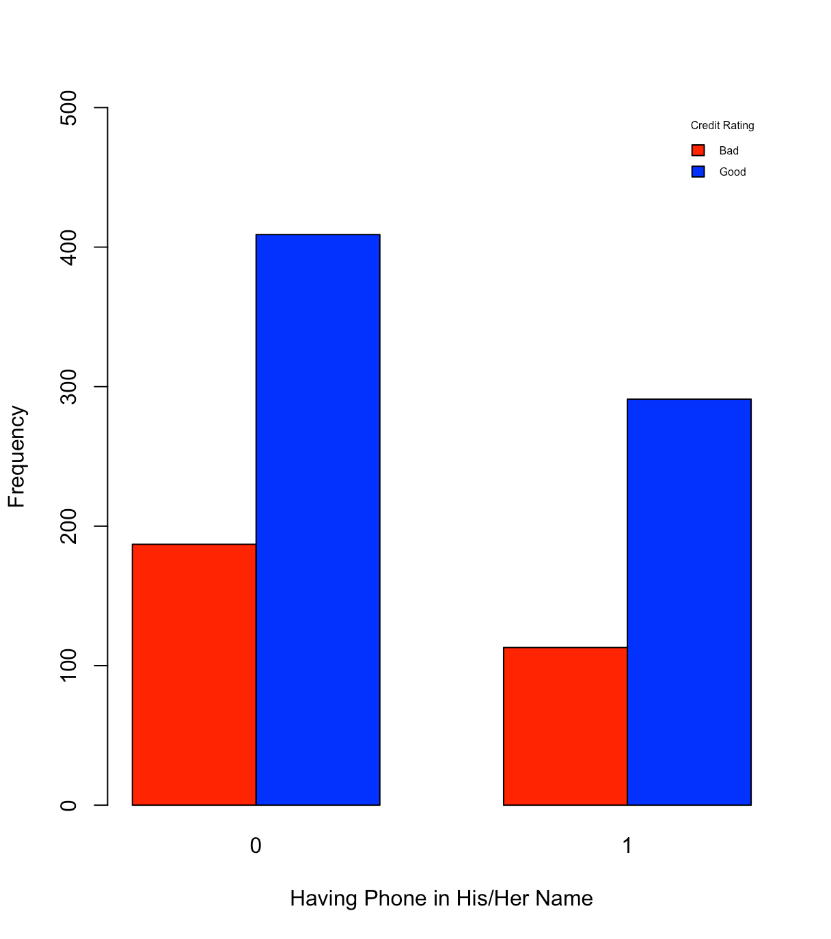
596 404

> table(German.Credit$TELEPHONE)/nrow(German.Credit)

0 1

0.596 0.404

> barplot(table(German.Credit$RESPONSE, German.Credit$TELEPHONE), beside = T, xlab = "Having Phone in His/Her Name", ylab = "Frequency", col= c("red", "blue") ,legend = c("Bad", "Good"), args.legend = list(title = "Credit Rating", x = "topright", cex = 0.5, bty = "n"), ylim = c(0,500))



1. FOREIGN

> table(German.Credit$FOREIGN)

0 1

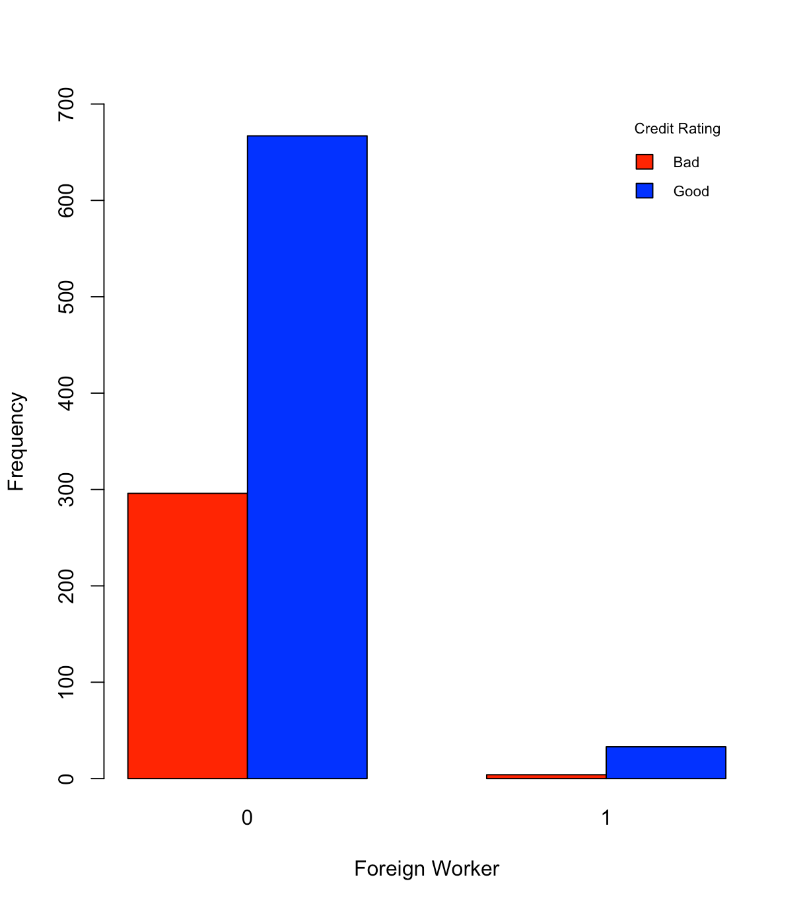
963 37

> table(German.Credit$FOREIGN)/nrow(German.Credit)

0 1

0.963 0.037

> barplot(table(German.Credit$RESPONSE, German.Credit$FOREIGN), beside = T, xlab = "Foreign Worker", ylab = "Frequency", col= c("red", "blue") ,legend = c("Bad", "Good"), args.legend = list(title = "Credit Rating", x = "topright", cex = 0.7, bty = "n"), ylim = c(0,700))



* Noteworthy things in the data **(Please add more)**
* Majority of clients with good credit rating have checking account status =3 (no checking account).
* Majority of clients with bad credit rating have checking account status = 0 (< 0 DM) or 1 (0 < … < 200DM).
* Majority of clients with good credit rating are applicants having no installment plan credit.
* Majority of clients with bad credit rating has average balance in savings account < 100DM.

(b)

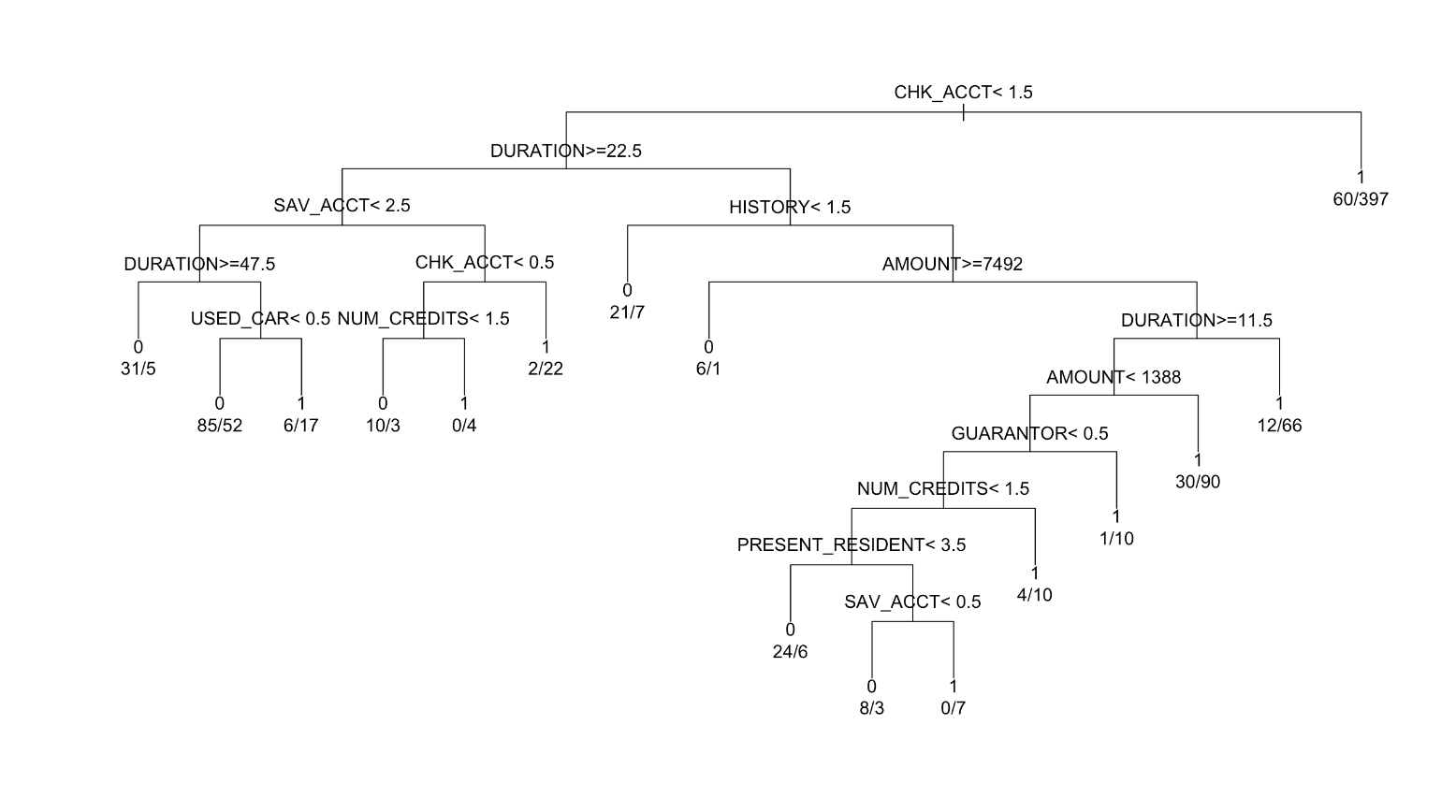
* Develop a decision tree

> library(rpart)

> GC\_rpart = rpart(RESPONSE~.,data=German.Credit[,-1], control = rpart.control(minsplit = 10), parms=list(split="gini"), method = "class")

> plot(GC\_rpart, uniform=TRUE)

> text(GC\_rpart, use.n=T, xpd=T)



* Accuracy and error

> table(predict(GC\_rpart, type="class"), German.Credit$RESPONSE, dnn=c("Predictions", "Actual Values"))

Actual Values

Predictions 0 1

0 185 77

1 115 623

* The number of (Predictions=0 AND Actual values=0): 185
* The number of (Predictions=0 AND Actual values=1): 77
* The number of (Predictions=1 AND Actual values=0): 115
* The number of (Predictions=1 AND Actual values=1): 623
* Accuracy: (185+623)/1000 = 0.808 or 80.8%
* Error: (77+115)/1000 = 0.192 or 19.2%
* Accuracy for the good cases (sensitivity): 623/(77+623) = 0.89 or 89%
* Accuracy for the bad cases (specificity): 185/(185+115) = 0.6167 or 61.67%
* This is not a robust model. The accuracy is good, but there is a big difference between sensitivity and specificity. The accuracy for the good cases is 89%, whereas the accuracy for the bad cases is only 61.67%.

(c)

* 50% for Training and 50% for Test

> set.seed(1234)

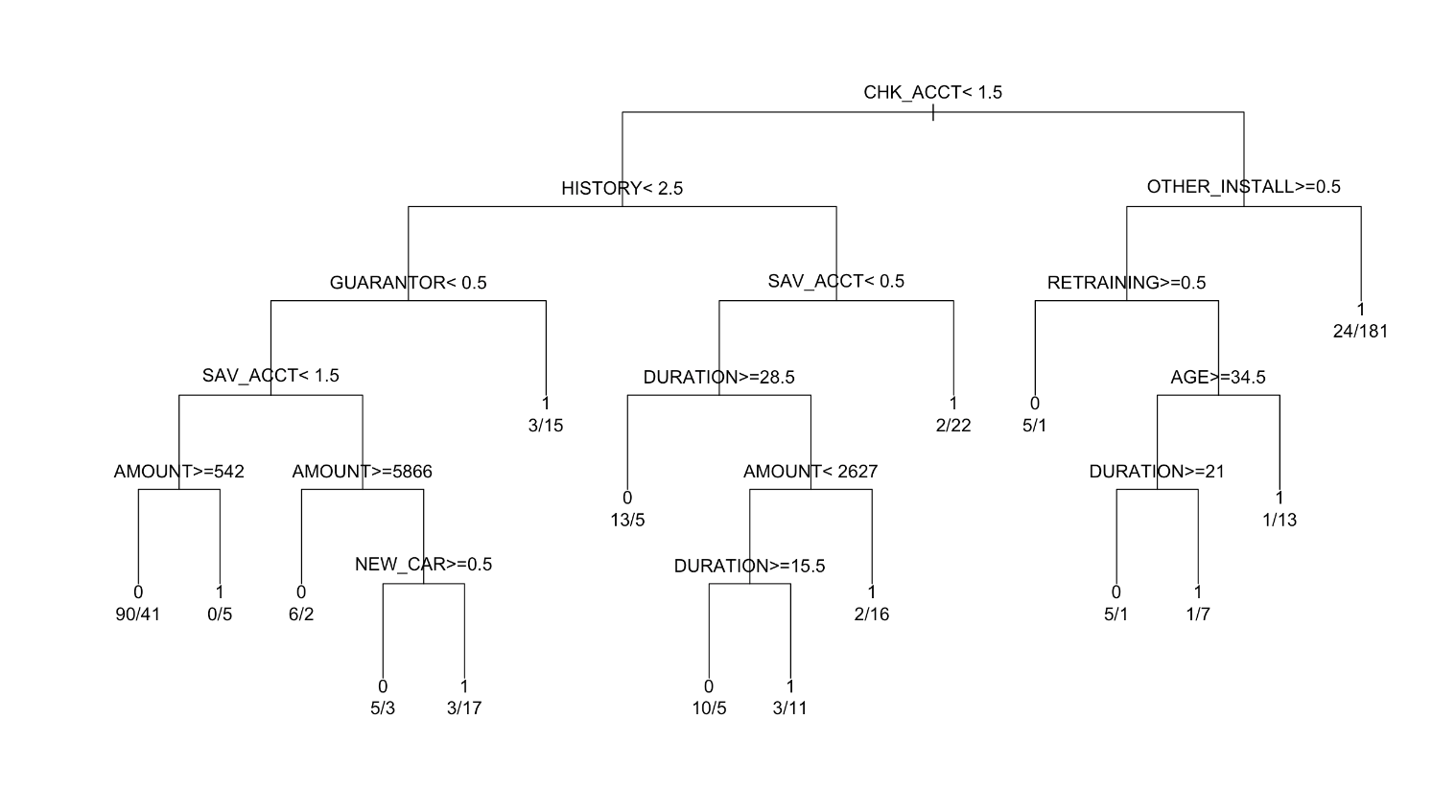
> ind1 = sample(2, nrow(German.Credit), replace=TRUE, prob=c(0.5, 0.5))

> trainData1 = German.Credit[ind1==1,]

> testData1 = German.Credit[ind1==2,]

> GC\_rpart1 = rpart(RESPONSE~.,data=trainData1[,-1], control = rpart.control(minsplit = 10), parms=list(split="gini"), method = "class")

> plot(GC\_rpart1, uniform=TRUE)

> text(GC\_rpart1, use.n=T, xpd=T)

> table(predict(GC\_rpart1, testData1, type="class"), testData1$RESPONSE, dnn=c("Predictions", "Actual Values"))

Actual Values

Predictions 0 1

0 81 85

1 46 270

* The number of (Predictions=0 AND Actual values=0): 81
* The number of (Predictions=0 AND Actual values=1): 85
* The number of (Predictions=1 AND Actual values=0): 46
* The number of (Predictions=1 AND Actual values=1): 270
* Accuracy: (81+270)/482 = 0.7282or 72.82%
* Error: (85+46)/482 = 0.2718 or 27.18%
* Accuracy for the good cases (sensitivity): 270/(85+270) = 0.7606or 76.06%
* Accuracy for the bad cases (specificity): 81/(81+46) = 0.6378 or 63.78%
* This model is more reliable than other models in that the difference between sensitivity and specificity is smaller than others. Also, it has the second lowest error rate.
* 70% for Training and 30% for Test

> set.seed(1234)

> ind2 = sample(2, nrow(German.Credit), replace=TRUE, prob=c(0.7, 0.3))

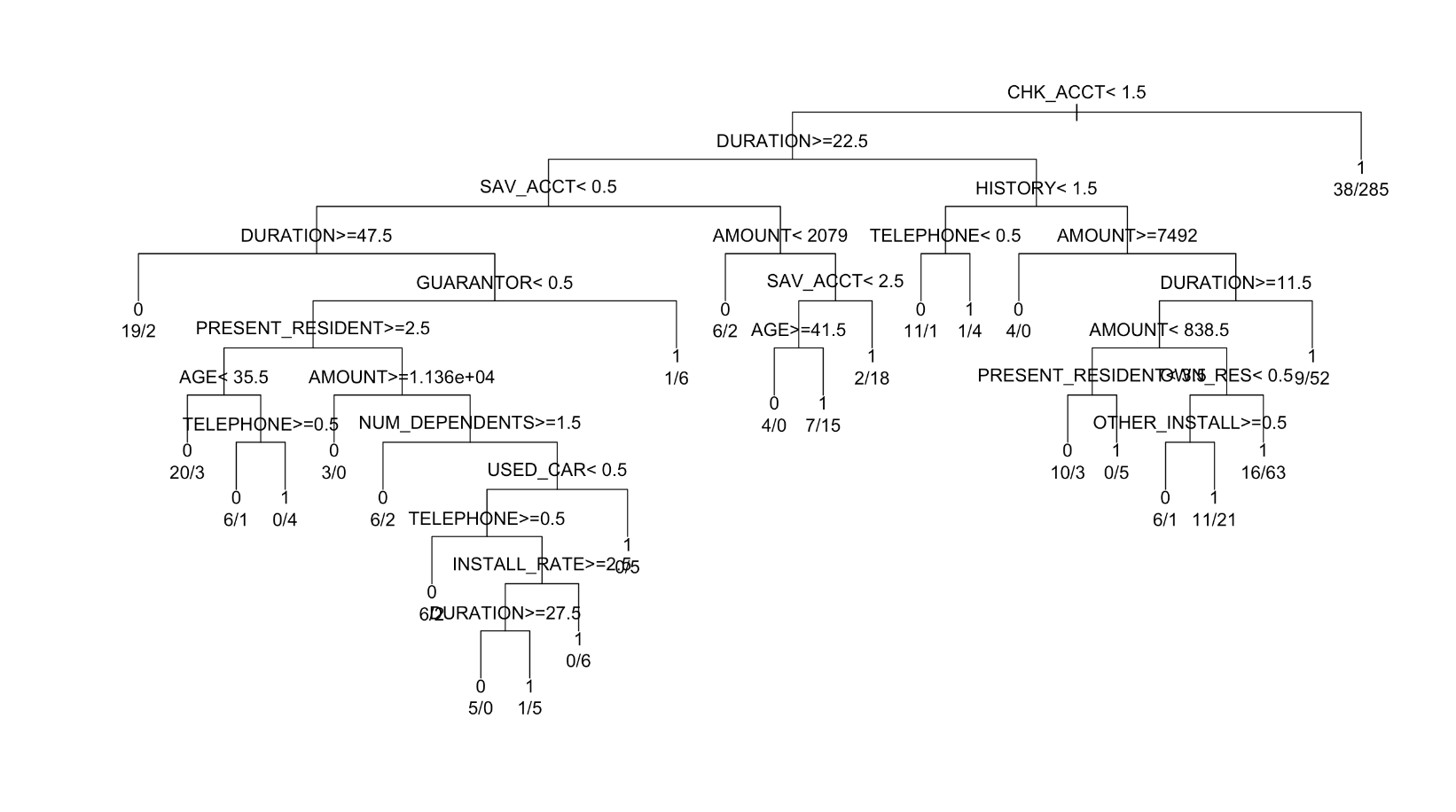
> trainData2 = German.Credit[ind2==1,]

> testData2 = German.Credit[ind2==2,]

> GC\_rpart2 = rpart(RESPONSE~.,data=trainData2[,-1], control = rpart.control(minsplit = 10), parms=list(split="gini"), method = "class")

> plot(GC\_rpart2, uniform=TRUE)

> text(GC\_rpart2, use.n=T, xpd=T)



> table(predict(GC\_rpart2, testData2, type="class"), testData2$RESPONSE, dnn=c("Predictions", "Actual Values"))

Actual Values

Predictions 0 1

0 39 20

1 69 174

* The number of (Predictions=0 AND Actual values=0): 39
* The number of (Predictions=0 AND Actual values=1): 20
* The number of (Predictions=1 AND Actual values=0): 69
* The number of (Predictions=1 AND Actual values=1): 174
* Accuracy: (39+174)/302 = 0.7053 or 70.53%
* Error: (20+69)/302 = 0.2947 or 29.47%
* Accuracy for the good cases (sensitivity): 174/(20+174) = 0.8969 or 89.69%
* Accuracy for the bad cases (specificity): 39/(39+69) = 0.3611 or 36.11%
* This model is not reliable due to the high sensitivity and the low specificity.
* 80% for Training and 20% for Test

> set.seed(1234)

> ind3 = sample(2, nrow(German.Credit), replace=TRUE, prob=c(0.8, 0.2))

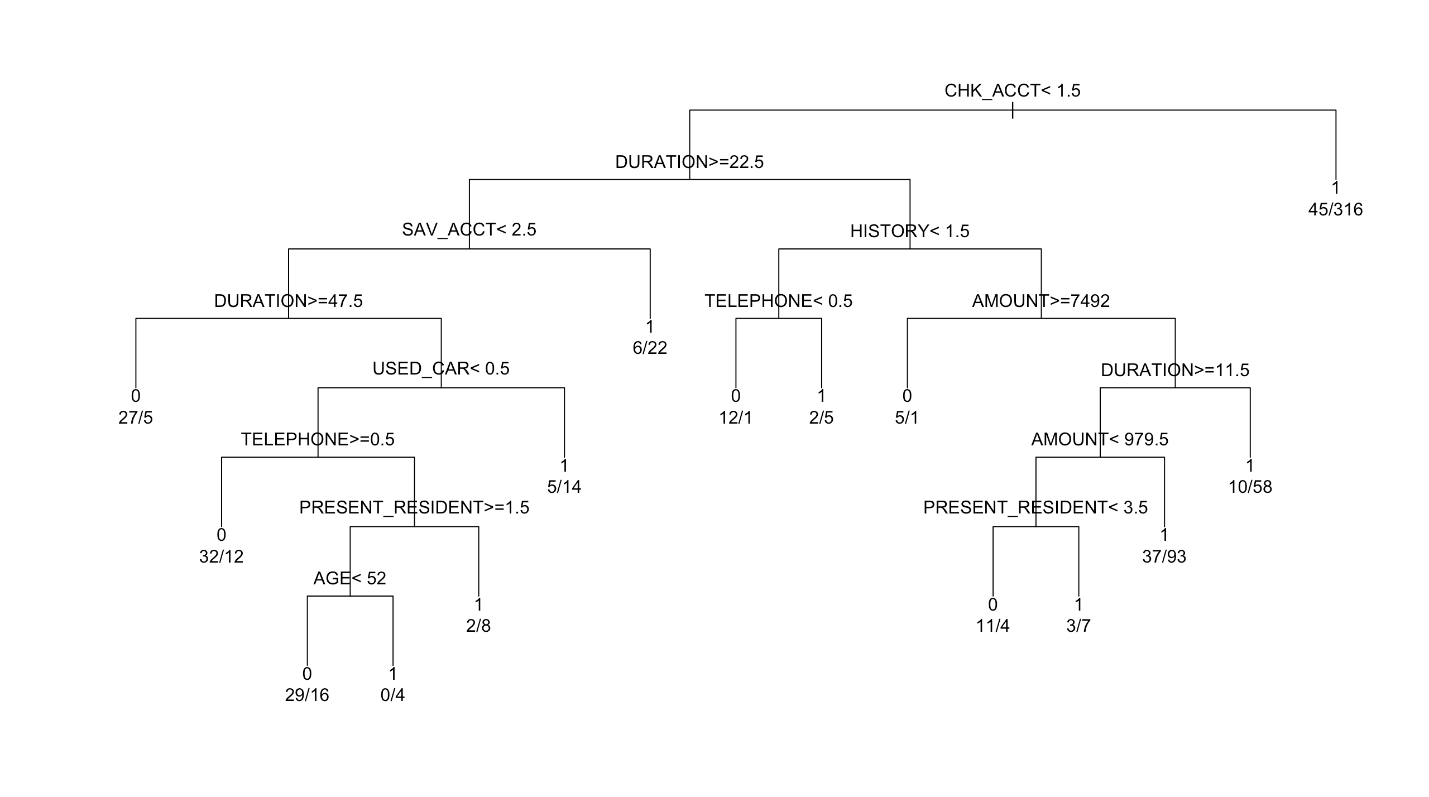
> trainData3 = German.Credit[ind3==1,]

> testData3 = German.Credit[ind3==2,]

> GC\_rpart3 = rpart(RESPONSE~.,data=trainData3[,-1], control = rpart.control(minsplit = 10), parms=list(split="gini"), method = "class")

> plot(GC\_rpart3, uniform=TRUE)

> text(GC\_rpart3, use.n=T, xpd=T)



> table(predict(GC\_rpart3, testData3, type="class"), testData3$RESPONSE, dnn=c("Predictions", "Actual Values"))

Actual Values

Predictions 0 1

0 30 16

1 44 118

* The number of (Predictions=0 AND Actual values=0): 30
* The number of (Predictions=0 AND Actual values=1): 16
* The number of (Predictions=1 AND Actual values=0): 44
* The number of (Predictions=1 AND Actual values=1): 118
* Accuracy: (30+118)/208 = 0.7115 or 71.15%
* Error: (16+44)/208 = 0.2885 or 28.85%
* Accuracy for the good cases (sensitivity): 118/(16+118) = 0.8806 or 88.06%
* Accuracy for the bad cases (specificity): 30/(30+44) = 0.4054 or 40.54%
* This model is not reliable due to the high sensitivity and the low specificity.
* 90% for Training and 10% for Test

> set.seed(1234)

> ind4 = sample(2, nrow(German.Credit), replace=TRUE, prob=c(0.9, 0.1))

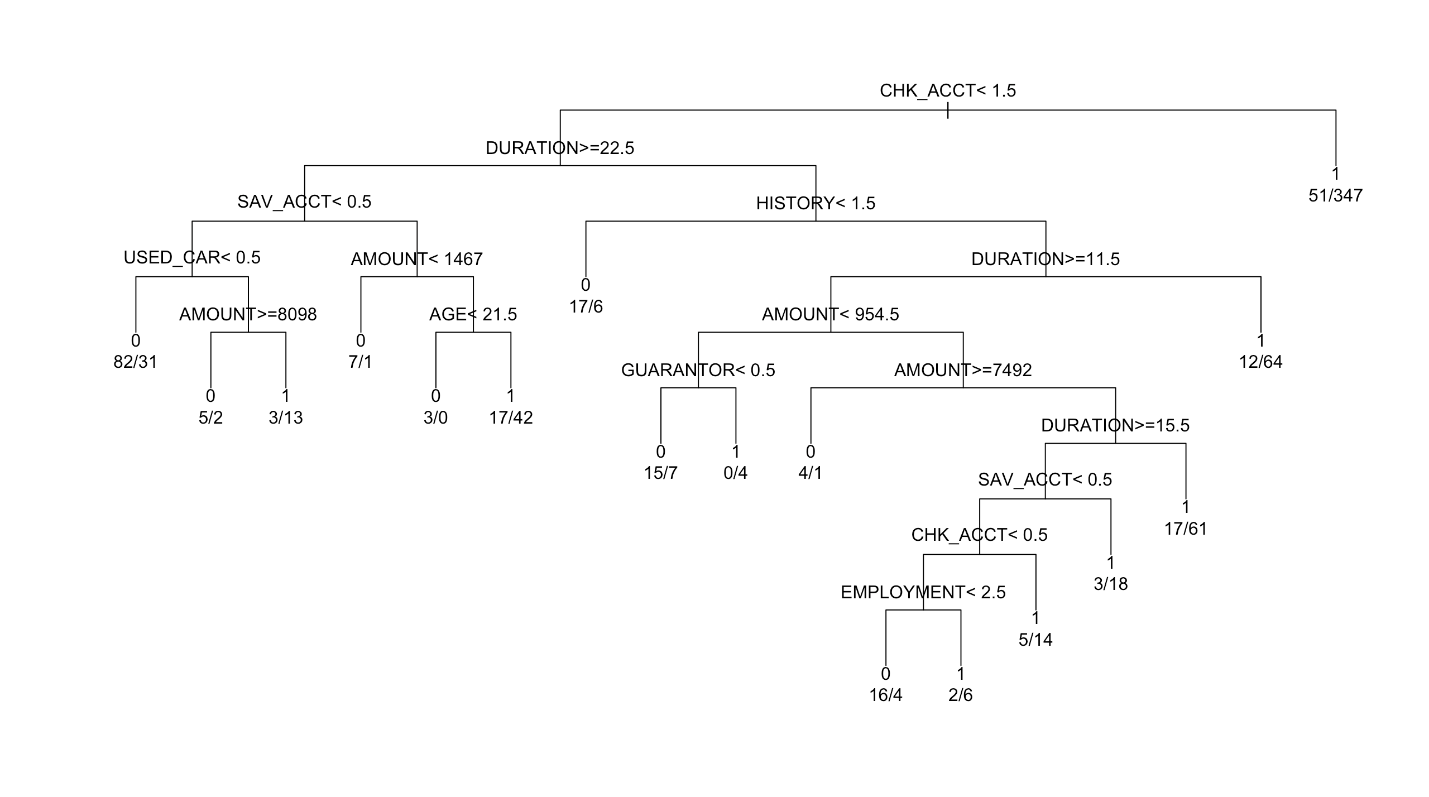
> trainData4 = German.Credit[ind4==1,]

> testData4 = German.Credit[ind4==2,]

> GC\_rpart4 = rpart(RESPONSE~.,data=trainData4[,-1], control = rpart.control(minsplit = 10), parms=list(split="gini"), method = "class")

> plot(GC\_rpart4, uniform=TRUE)

> text(GC\_rpart4, use.n=T, xpd=T)



> table(predict(GC\_rpart4, testData4, type="class"), testData4$RESPONSE, dnn=c("Predictions", "Actual Values"))

Actual Values

Predictions 0 1

0 20 10

1 21 69

* The number of (Predictions=0 AND Actual values=0): 20
* The number of (Predictions=0 AND Actual values=1): 10
* The number of (Predictions=1 AND Actual values=0): 21
* The number of (Predictions=1 AND Actual values=1): 69
* Accuracy: (20+69)/120 = 0.7417 or 74.17%
* Error: (10+21)/120 = 0.2583 or 25.83%
* Accuracy for the good cases (sensitivity): 69/(10+69) = 0.8734 or 87.34%
* Accuracy for the bad cases (specificity): 20/(20+21) = 0.4878 or 48.78%
* This model is not reliable due to the high sensitivity and the low specificity.
* Preferred model

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Accuracy | Error | Sensitivity | Specificity |
| **50/50** | **72.82%** | **27.18%** | **76.06%** | **63.78%** |
| 70/30 | 70.53% | 29.47% | 89.69% | 36.11% |
| 80/20 | 71.15% | 28.85% | 88.06% | 40.54% |
| 90/10 | 74.17% | 25.83% | 87.34% | 48.78% |

As mentioned above, when the data is partitioned into 50% for Training and 50% for Test, the model has smaller difference between sensitivity and specificity than other models. Although the sensitivity is lower than others, specificity is far better than others. In addition, its error rate is the second lowest; the model from the data partitioned into 90% for Training and 10% for Test has the lowest error rate, but it has a huge difference between sensitivity and specificity. All things considered, the model from the data partitioned into 50% for Training and 50% for Test is preferred. However, the preferred model can be changed depending on the amount of importance of sensitivity and specificity.

(d)

If certain applications carrying unreliable data such as outliers and missing data on the different attributes are used in the model, It can mislead the model. Therefore, the decision tree model cannot be reliable and will not be usable. In such cases, we propose to use random forest to create a model since random forest can be used to handle outliers and missing values using proximity matrix.

(e) Use the misclassification costs in obtaining a model

* 50% for Training and 50% for Test

> loss <- matrix(c(0, 500, 100, 0), byrow=TRUE, ncol=2)

> GC\_fit1 = rpart(RESPONSE~.,data=trainData1[,-1], control = rpart.control(minsplit = 10), parms=list(split="gini", loss=loss), method = "class")

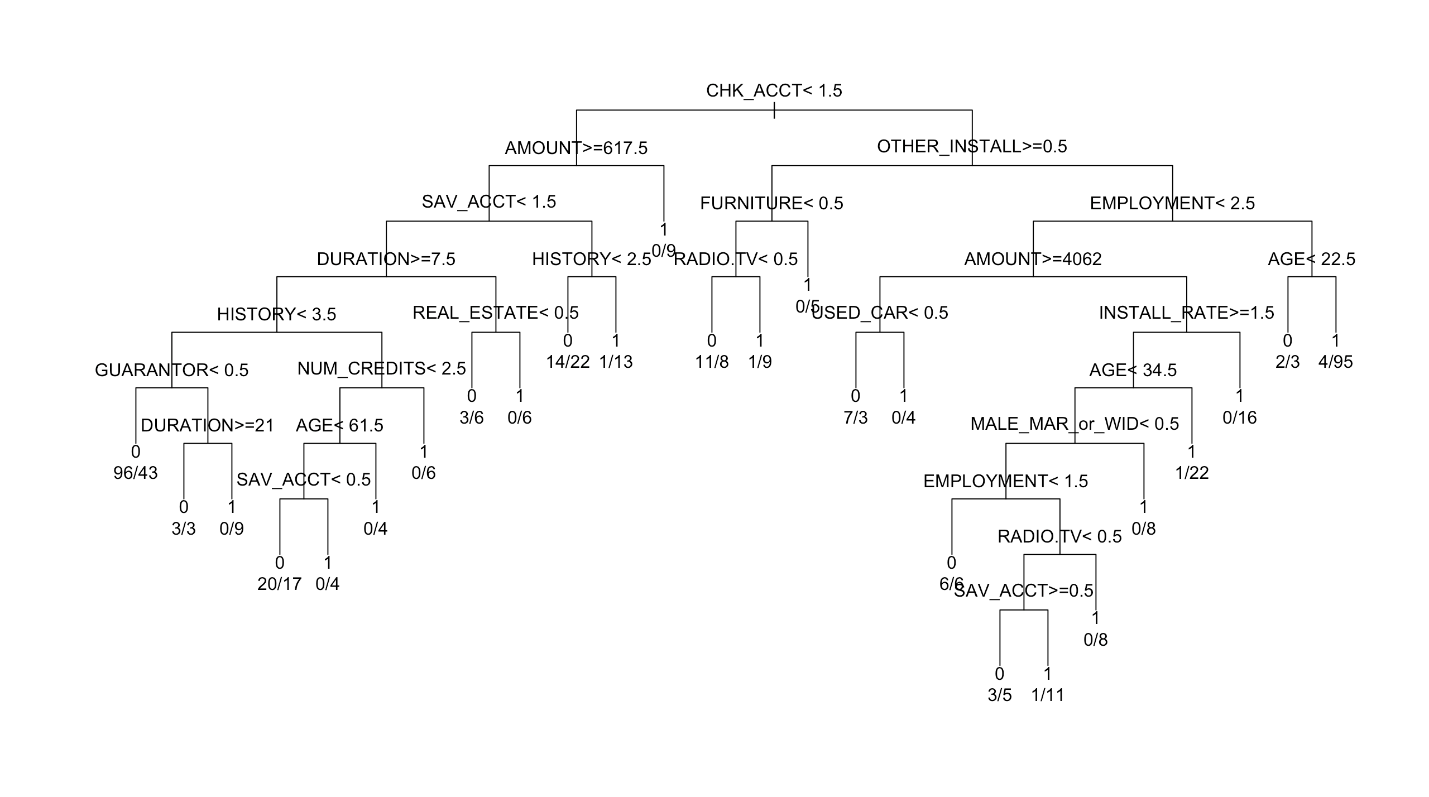
> plot(GC\_fit1, uniform=TRUE)

> text(GC\_fit1, use.n=T, xpd=T)

Dat1 = sample(2, nrow(German.Credit), replace=TRUE, prob=c(0.5, 0.5))

trainData = German.Credit[Dat1==1,]

testData = German.Credit[Dat1==2,]



> table(predict(GC\_fit1, testData1, type="class"), testData1$RESPONSE, dnn=c("Predictions", "Actual Values"))

Actual Values

Predictions 0 1

0 107 172

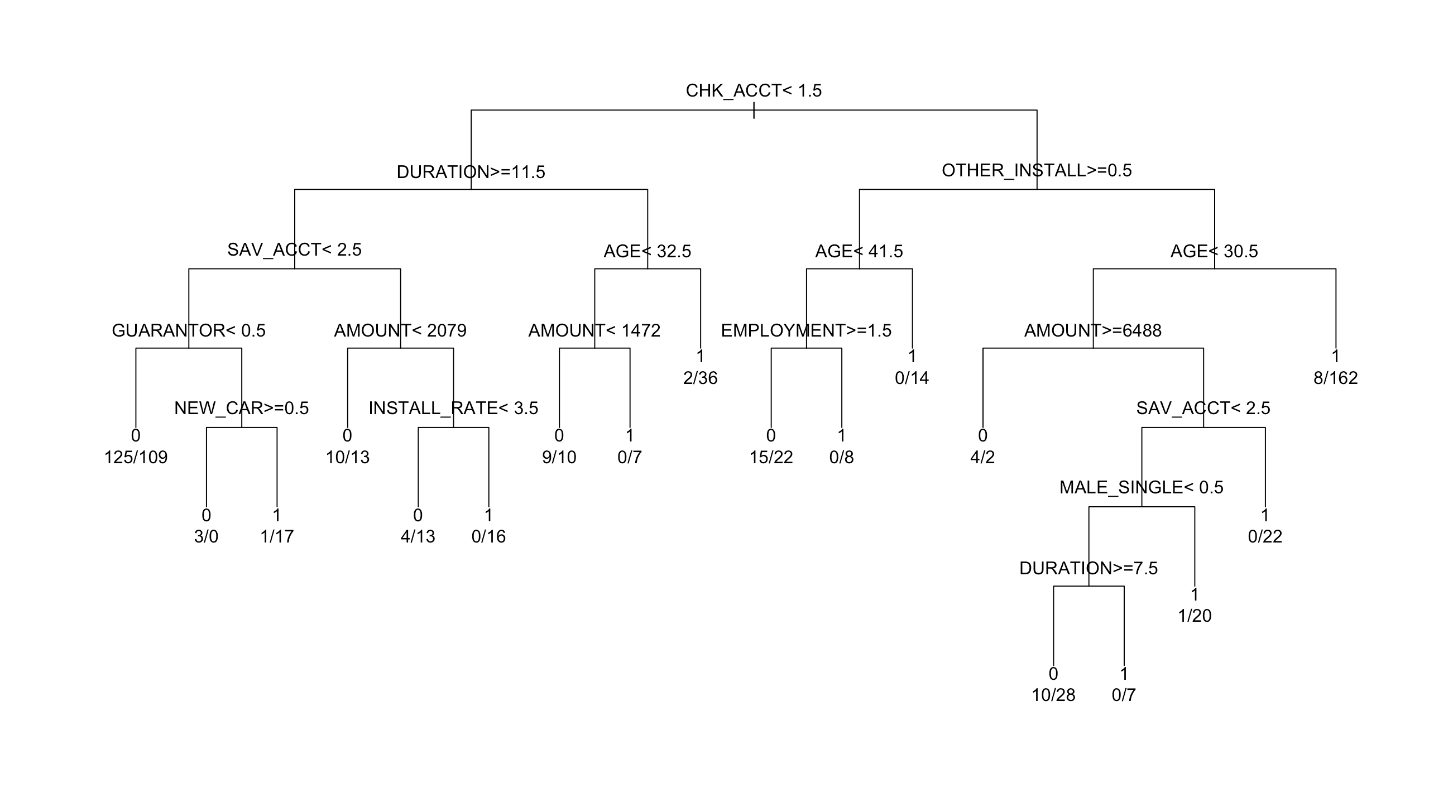
1 20 183

* The number of (Predictions=0 AND Actual values=0): 107
* The number of (Predictions=0 AND Actual values=1): 172
* The number of (Predictions=1 AND Actual values=0): 20
* The number of (Predictions=1 AND Actual values=1): 183
* Accuracy: (107+183)/482 = 0.6017 or 60.17%
* Error: (172+20)/482 = 0.3983 or 39.83%
* Accuracy for the good cases (sensitivity): 183/(172+183) = 0.5155 or 51.55%
* Accuracy for the bad cases (specificity): 107/(107+20) = 0.8425 or 84.25%
* 70% for Training and 30% for Test

> GC\_fit2 = rpart(RESPONSE~.,data=trainData2[ ,-1], control = rpart.control(minsplit = 10), parms=list(split="gini", loss=loss), method = "class")

> plot(GC\_fit2, uniform=TRUE)

> text(GC\_fit2, use.n=T, xpd=T)



> table(predict(GC\_fit2, testData2, type="class"), testData2$RESPONSE, dnn=c("Predictions", "Actual Values"))

Actual Values

Predictions 0 1

0 80 89

1 28 105

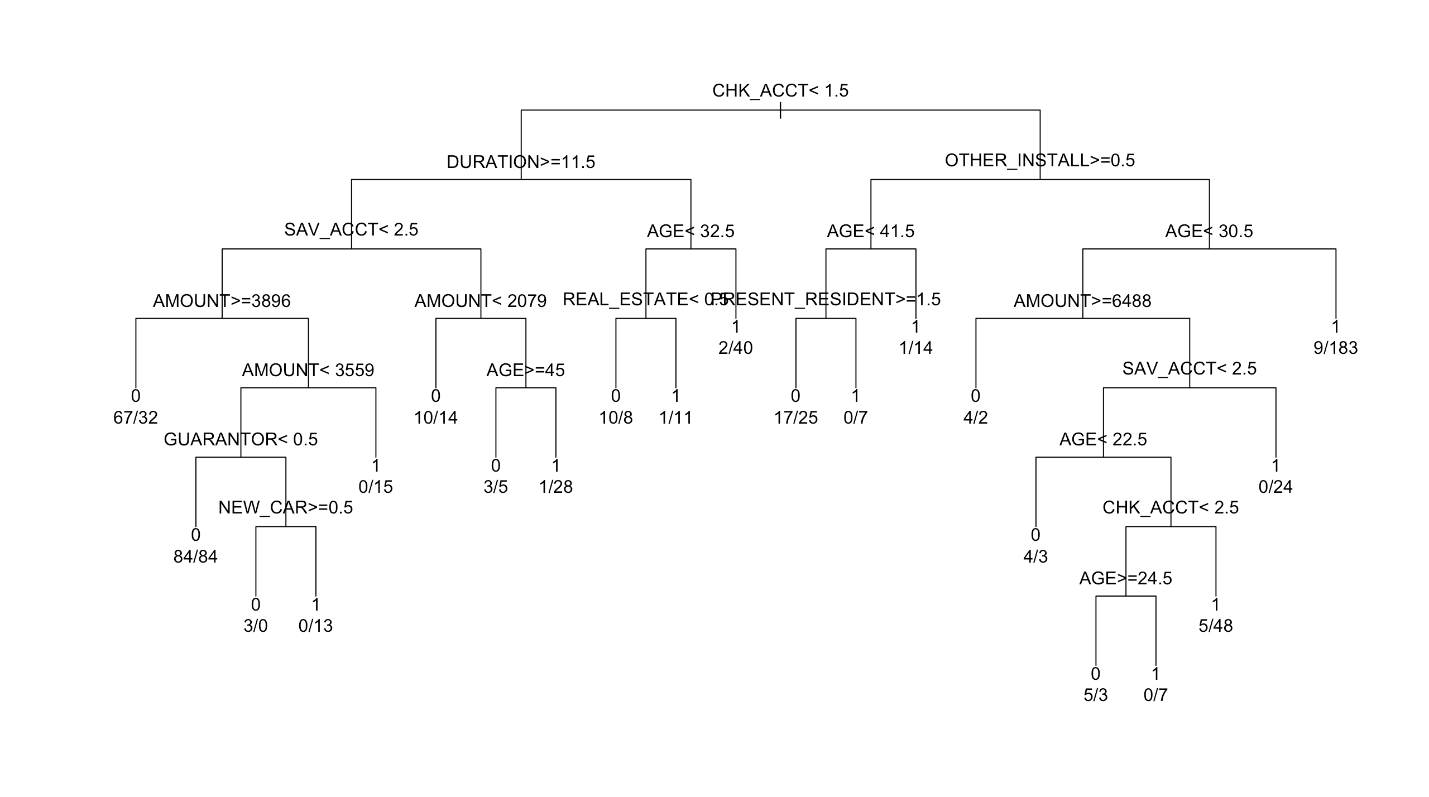
* The number of (Predictions=0 AND Actual values=0): 80
* The number of (Predictions=0 AND Actual values=1): 89
* The number of (Predictions=1 AND Actual values=0): 28
* The number of (Predictions=1 AND Actual values=1): 105
* Accuracy: (80+105)/302 = 0.6126 or 61.26%
* Error: (89+28)/302 = 0.3874 or 38.74%
* Accuracy for the good cases (sensitivity): 105/(89+105) = 0.5412 or 54.12%
* Accuracy for the bad cases (specificity): 80/(80+28) = 0.7407 or 74.07%
* 80% for Training and 20% for Test

> loss <- matrix(c(0, 500, 100, 0), byrow=TRUE, ncol=2)

> GC\_fit3 = rpart(RESPONSE~.,data=trainData3[ ,-1], control = rpart.control(minsplit = 10), parms=list(split="gini", loss=loss), method = "class")

> plot(GC\_fit3, uniform=TRUE)

> text(GC\_fit3, use.n=T, xpd=T)



> table(predict(GC\_fit3, testData3, type="class"), testData3$RESPONSE, dnn=c("Predictions", "Actual Values"))

Actual Values

Predictions 0 1

0 50 41

1 24 93

* The number of (Predictions=0 AND Actual values=0): 50
* The number of (Predictions=0 AND Actual values=1): 41
* The number of (Predictions=1 AND Actual values=0): 24
* The number of (Predictions=1 AND Actual values=1): 93
* Accuracy: (50+93)/208 = 0.6875 or 68.75%
* Error: (41+24)/208 = 0.3125 or 31.25%
* Accuracy for the good cases (sensitivity): 93/(41+93) = 0.6940 or 69.40%
* Accuracy for the bad cases (specificity): 50/(50+24) = 0.6757 or 67.57%
* Changes in the model and performance
* Differences in the decision tree

Each decision tree is changed when using the misclassification costs.

1. Different internal nodes are used.
2. As shown in the table below, the number of leaves increase or decrease.

|  |  |  |
| --- | --- | --- |
|  | before | after |
| 50/50 | 16 | 26 |
| 70/30 | 26 | 18 |
| 80/20 | 15 | 21 |
| 90/10 | 17 | 8 |

1. Whether or not using the misclassification costs, the root node is same as Chk\_Acct < 1.5 except for the model from the data partitioned into 90% for Training and 10% for Test which root node is changed to Chk\_Acct < 2.5 from Chk\_Acct < 1.5.

* Preferred model without using the misclassification costs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Accuracy | Error | Sensitivity | Specificity |
| **50/50** | **72.82%** | **27.18%** | **76.06%** | **63.78%** |
| 70/30 | 70.53% | 29.47% | 89.69% | 36.11% |
| 80/20 | 71.15% | 28.85% | 88.06% | 40.54% |
| 90/10 | 74.17% | 25.83% | 87.34% | 48.78% |

* Preferred model with using the misclassification costs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Accuracy | Error | Sensitivity | Specificity |
| 50/50 | 60.17% | 39.83% | 51.55% | 84.25% |
| 70/30 | 61.26% | 38.74% | 54.12% | 74.07% |
| **80/20** | **68.75%** | **31.25%** | **69.4%** | **67.57%** |
| 90/10 | 61.67% | 38.33% | 48.1% | 87.8% |

For each model, using the misclassification costs makes the accuracy and sensitivity lower and makes the specificity higher. The model from the data partitioned into 50% for Training and 50% for Test is preferred when not using the misclassification costs, whereas the model from the data partitioned into 80% for Training and 20% for Test is preferred when using the misclassification costs (because it has the highest accuracy with better balanced accuracy meaning small difference between sensitivity and specificity).

* Benefits from specifying misclassification costs

Using the misclassification costs enable the model to take into account the opportunity costs by applying weights to specific outcomes. It can change the prediction and prevent costly mistakes.

(f) Pruning the decision tree with the minimum prediction error

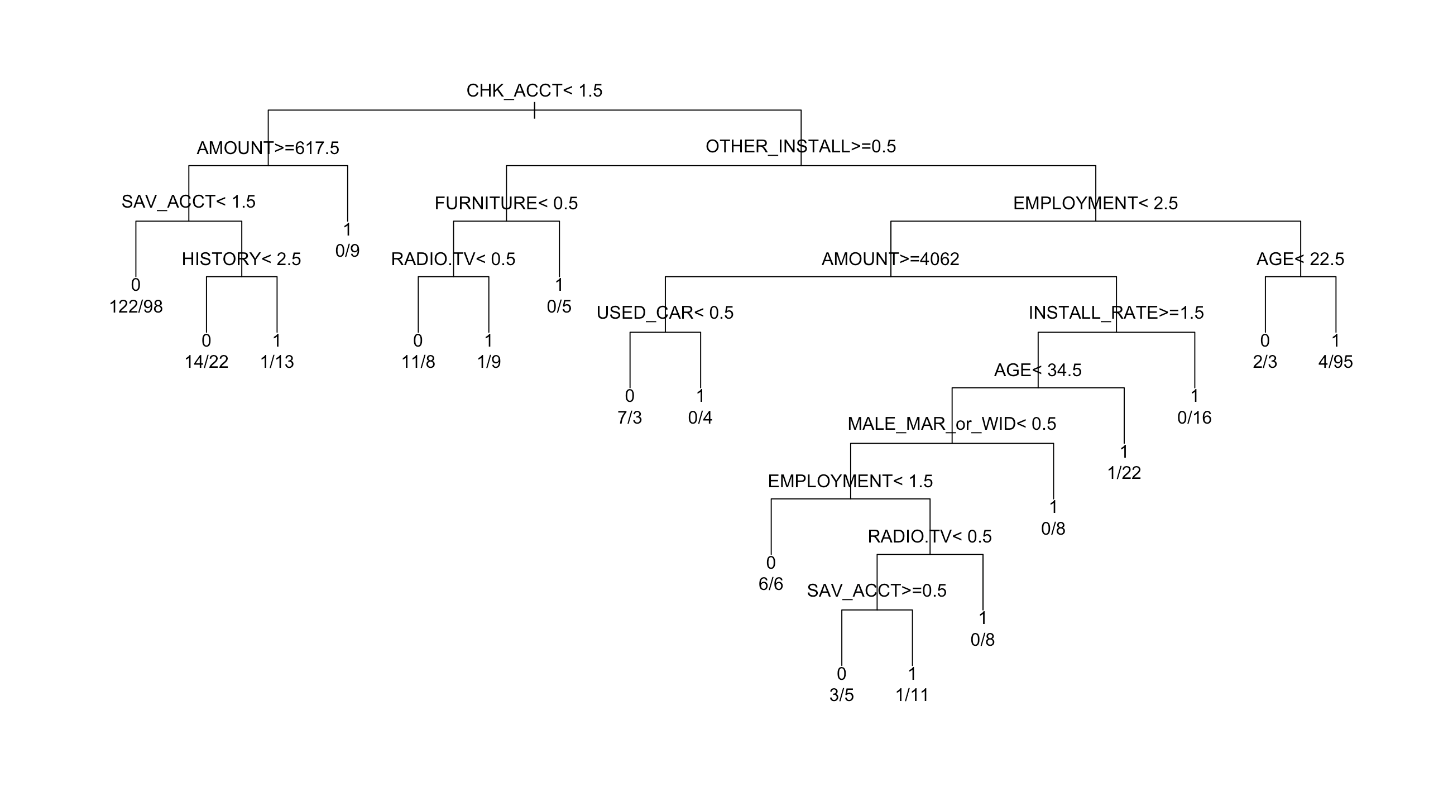
* 50% for Training and 50% for Test

> opt = which.min(GC\_fit1$cptable[,"xerror"])

> cp = GC\_fit1$cptable[opt, "CP"]

> GC\_fit1\_prune = prune(GC\_fit1, cp = cp)

> plot(GC\_fit1\_prune, uniform=TRUE)

> text(GC\_fit1\_prune, use.n=T, xpd=T)

> confusionMatrix(predict(GC\_fit1\_prune, testData1, type="class"), testData1$RESPONSE, dnn=c("Predictions", "Actual Values"), positive = "1")

Confusion Matrix and Statistics

Actual Values

Predictions 0 1

0 111 185

1 16 170

Balanced Accuracy : 0.6764

'Positive' Class : 1

* 70% for Training and 30% for Test

(70/30: after pruning is same as before pruning… I’m doubtful.)

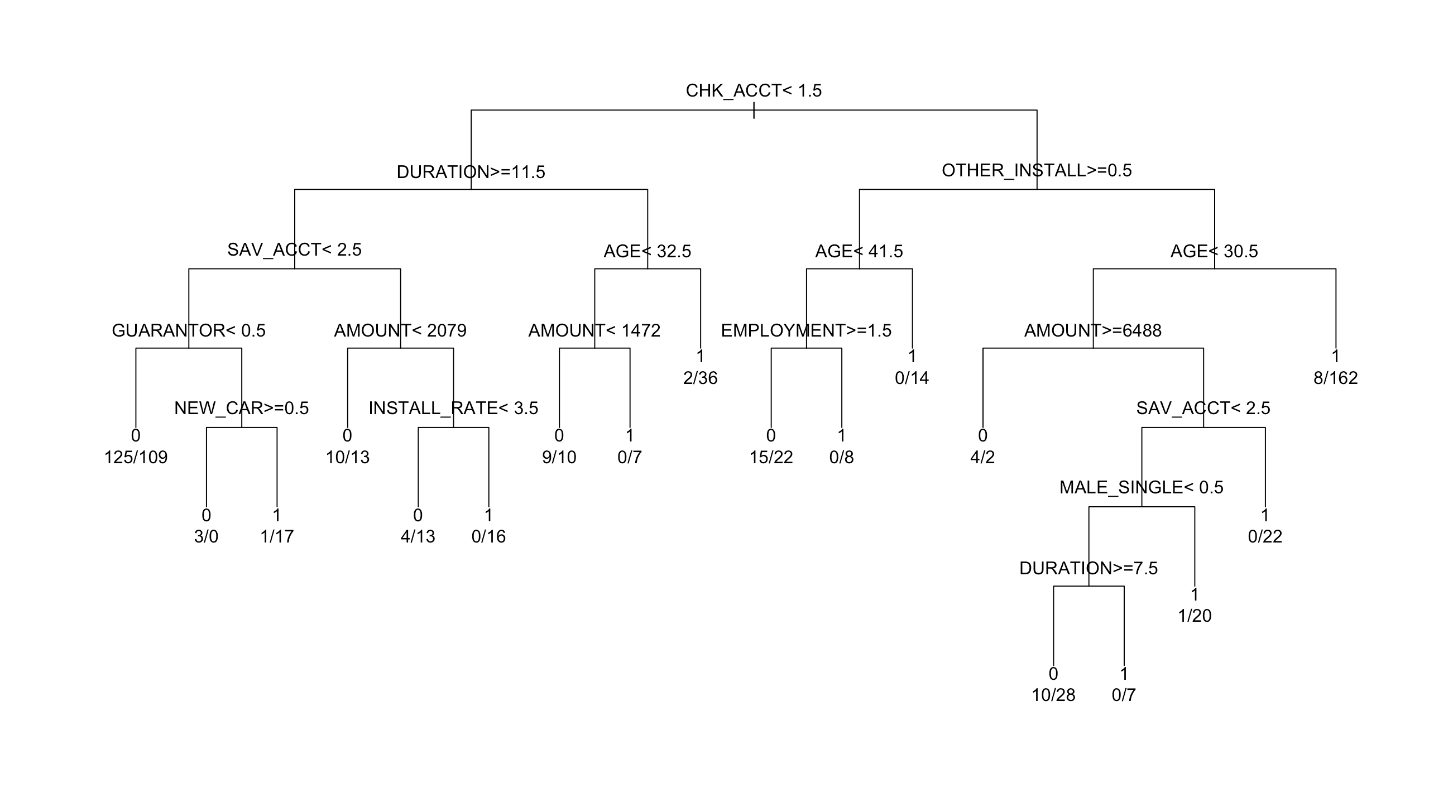
> opt = which.min(GC\_fit2$cptable[,"xerror"])

> cp = GC\_fit2$cptable[opt, "CP"]

> GC\_fit2\_prune = prune(GC\_fit2, cp = cp)

> plot(GC\_fit2\_prune, uniform=TRUE)

> text(GC\_fit2\_prune, use.n=T, xpd=T)



> confusionMatrix(predict(GC\_fit2\_prune, testData2, type="class"), testData2$RESPONSE, dnn=c("Predictions", "Actual Values"), positive = "1")

Confusion Matrix and Statistics

Actual Values

Predictions 0 1

0 80 89

1 28 105

* 80% for Training and 20% for Test

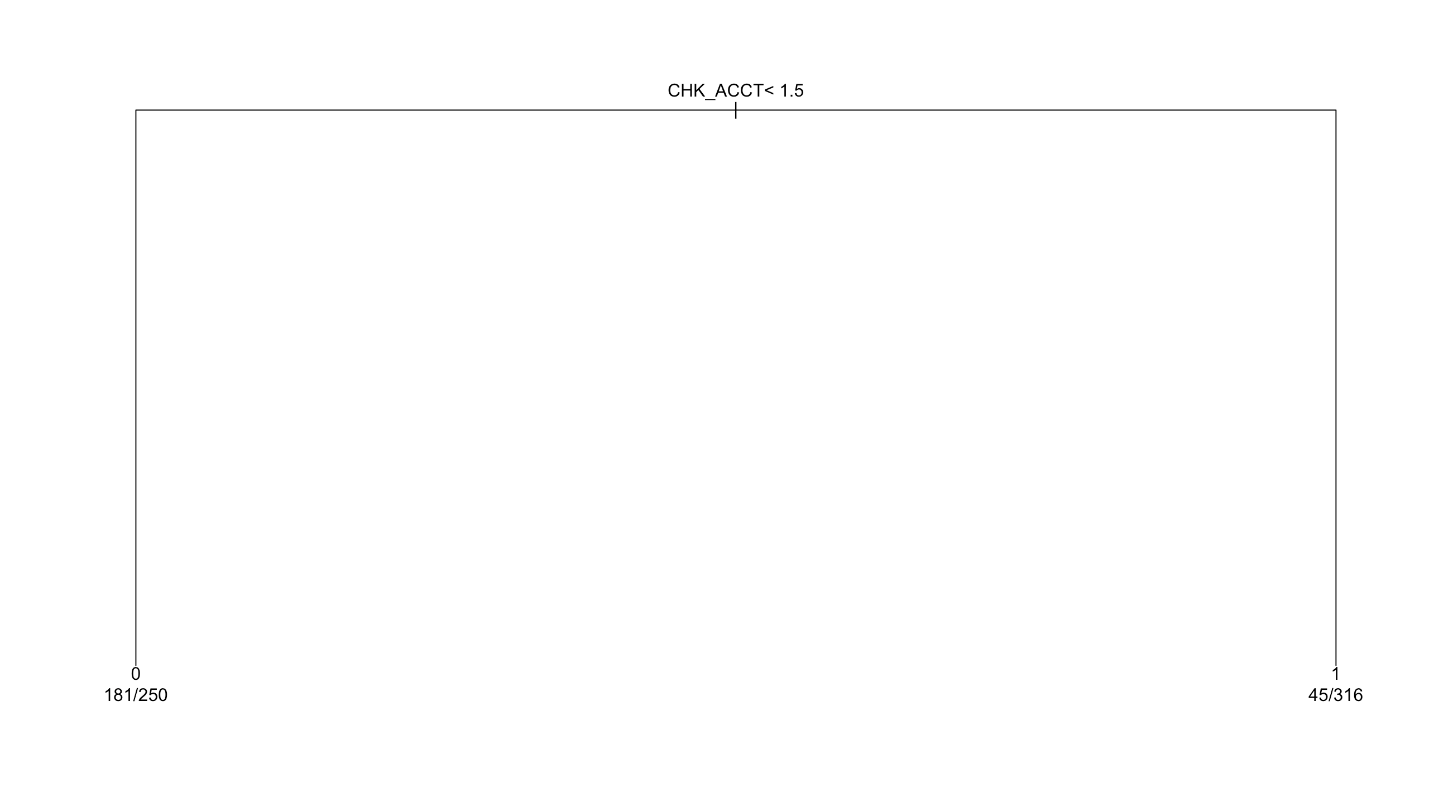
> opt = which.min(GC\_fit3$cptable[,"xerror"])

> cp = GC\_fit3$cptable[opt, "CP"]

> GC\_fit3\_prune = prune(GC\_fit3, cp = cp)

> plot(GC\_fit3\_prune, uniform=TRUE)

> text(GC\_fit3\_prune, use.n=T, xpd=T)



> confusionMatrix(predict(GC\_fit3\_prune, testData3, type="class"), testData3$RESPONSE, dnn=c("Predictions", "Actual Values"), positive = "1")

Confusion Matrix and Statistics

Actual Values

Predictions 0 1

0 59 53

1 15 81

* Changes in the model and performance
* Differences in the decision tree

Each decision tree is changed after pruning.

1. Different internal nodes are used.
2. As shown in the table below, the number of leaves decreases or stays same.

|  |  |  |
| --- | --- | --- |
|  | before | after |
| 50/50 | 26 | 18 |
| 70/30 | 18 | 18 |
| 80/20 | 21 | 2 |
| 90/10 | 8 | 2 |

1. After the pruning, the root node of each tree is same as before.

* Performance before pruning

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Accuracy | Error | Sensitivity | Specificity |
| 50/50 | 60.17% | 39.83% | 51.55% | 84.25% |
| 70/30 | 61.26% | 38.74% | 54.12% | 74.07% |
| **80/20** | **68.75%** | **31.25%** | **69.4%** | **67.57%** |
| 90/10 | 61.67% | 38.33% | 48.1% | 87.8% |

* Performance after pruning

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Accuracy | Error | Sensitivity | Specificity |
| 50/50 | 58.3% | 41.7% | 47.89% | 87.4% |
| 70/30 | 61.26% | 38.74% | 54.12% | 74.07% |
| **80/20** | **67.31%** | **32.69%** | **60.45%** | **79.73%** |
| 90/10 | 62.5% | 37.5% | 51.9% | 82.93% |

Pruning the tree does not improve the accuracy of the model. When the pruning decreases the sensitivity of the model, it decreases the specificity of the model, and vice versa. (70/30: after pruning is same as before pruning… I’m doubtful.)

1. Output rule with the best nodes for classifying “Good” applicants

* We chose the decision tree with the data partitioned into 80% for Training and 20% for Test using the classification costs without pruning. The output rules are based on this model.
* If CHK\_ACCT >= 1.5 and OTHER\_INSTALL < 0.5 and AGE >= 30.5, then RESPONSE = 1 meaning good clients: error = 9/(9+183) = 0.0469
* If CHK\_ACCT < 1.5 and DURATION < 11.5 and AGE >= 32.5, then RESPONSE = 1 meaning good clients: error = 2/(2+40) = 0.0476
* Although other combinations of nodes have an error rate at 0, the instances in the leaves are too small such as 0/16, 0/7, ect.

1. Summary **(Please add more esp for pruning)**

When choosing the best model, we need to consider sensitivity and specificity in addition to general accuracy. The model with a good accuracy rate can have a big difference between sensitivity and specificity, making the model unreliable.

The amount of importance of sensitivity and specificity can help decide the best model. Therefore, when building a model, the opportunity cost should be taken into account. Using the misclassification costs changes the prediction to avoid costly mistakes by applying weights to specific outcomes.