Classification Analysis

Hazim Fitri

Contents

Logistic Regression

Assumption

- Dependent variable is binary
- Predictor variables must not have perfect multicollinearity
- Large sample size
- No influential outliers

ETL

```r
# import .csv file into R
gc = read.csv('./Data/german_credit.csv')

# take a look at the structure of the data
str(gc)
```

```
## 'data.frame':    1000 obs. of  21 variables:
##  $ Creditability                    : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Account.Balance                  : int  1 1 2 1 1 1 1 1 4 2 ...
##  $ Duration.of.Credit..month.       : int  18 9 12 12 12 10 8 6 18 24 ...
##  $ Payment.Status.of.Previous.Credit: int  4 4 2 4 4 4 4 4 4 2 ...
##  $ Purpose                          : int  2 0 9 0 0 0 0 0 3 3 ...
##  $ Credit.Amount                    : int  1049 2799 841 2122 2171 2241 3398 1361 1098 3758 ...
##  $ Value.Savings.Stocks             : int  1 1 2 1 1 1 1 1 1 3 ...
##  $ Length.of.current.employment     : int  2 3 4 3 3 2 4 2 1 1 ...
```

```
##  $ Instalment.per.cent              : int  4 2 2 3 4 1 1 2 4 1 ...
##  $ Sex...Marital.Status             : int  2 3 2 3 3 3 3 3 2 2 ...
##  $ Guarantors                       : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Duration.in.Current.address      : int  4 2 4 2 4 3 4 4 4 4 ...
##  $ Most.valuable.available.asset    : int  2 1 1 1 2 1 1 1 3 4 ...
##  $ Age..years.                      : int  21 36 23 39 38 48 39 40 65 23 ...
##  $ Concurrent.Credits               : int  3 3 3 3 1 3 3 3 3 3 ...
##  $ Type.of.apartment                : int  1 1 1 1 2 1 2 2 2 1 ...
##  $ No.of.Credits.at.this.Bank       : int  1 2 1 2 2 2 2 1 2 1 ...
##  $ Occupation                       : int  3 3 2 2 2 2 2 2 1 1 ...
##  $ No.of.dependents                 : int  1 2 1 2 1 2 1 2 1 1 ...
##  $ Telephone                        : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Foreign.Worker                   : int  1 1 1 2 2 2 2 2 1 1 ...
```

From the structure, we can see that a lot of categorical data being read as integer. Next, we will pre-process the data before proceed with fit the data into training logistic regression model.

Data Pre-processing

Convert data type

```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
names(gc)[1] = 'cr'
table(gc$cr)/sum(table(gc$cr))
```

```
##
##   0   1
## 0.3 0.7
```

```r
gc$cr = as.factor(gc$cr)

names(gc)[2] = 'acc.bal'
table(gc$acc.bal)/sum(table(gc$acc.bal))
```

```
##
##     1     2     3     4
## 0.274 0.269 0.063 0.394
```

```r
# we wish to combine category 4 into 3
gc$acc.bal = replace(gc$acc.bal, gc$acc.bal == 4, 3)
gc$acc.bal = factor(gc$acc.bal, levels = seq(1:3),
                       labels = c('No Acc', 'No Bal', 'Has Bal'))
table(gc$acc.bal)/sum(table(gc$acc.bal))
```

```
##
##  No Acc  No Bal Has Bal
##   0.274   0.269   0.457
```

```r
names(gc)[4] = 'pmt.stat.prev.cr'
table(gc$pmt.stat.prev.cr)/sum(table(gc$pmt.stat.prev.cr))
```

```
##
##     0     1     2     3     4
## 0.040 0.049 0.530 0.088 0.293
```

```r
# we wish to combine category 1 into 0, and 4 into 3. Then shift downward
gc$pmt.stat.prev.cr[gc$pmt.stat.prev.cr <= 1] = 1
gc$pmt.stat.prev.cr[gc$pmt.stat.prev.cr == 2] = 2
gc$pmt.stat.prev.cr[gc$pmt.stat.prev.cr >= 3] = 3
table(gc$pmt.stat.prev.cr)/sum(table(gc$pmt.stat.prev.cr))
```

```
##
##     1     2     3
## 0.089 0.530 0.381
```

```r
gc$pmt.stat.prev.cr = factor(gc$pmt.stat.prev.cr, levels = seq(1:3),
                             labels = c('Some probs', 'paid up', 'no prob'))

names(gc)[7] = 'savings.stocks'
table(gc$savings.stocks)/sum(table(gc$savings.stocks))
```

```
##
##     1     2     3     4     5
## 0.603 0.103 0.063 0.048 0.183
```

```r
gc$savings.stocks[gc$savings.stocks == 4] = 3
gc$savings.stocks[gc$savings.stocks == 5] = 4
gc$savings.stocks = factor(gc$savings.stocks, levels = seq(1:4),
                           labels = c('none', '<100', '100-1000', '>1000'))
table(gc$savings.stocks)/sum(table(gc$savings.stocks))
```

```
##
##     none     <100 100-1000    >1000
##    0.603    0.103    0.111    0.183
```

```r
names(gc)[8] = 'len.emp'
table(gc$len.emp)/sum(table(gc$len.emp))
```

```
##
##     1     2     3     4     5
## 0.062 0.172 0.339 0.174 0.253
```

```r
gc$len.emp[gc$len.emp == 2] = 1
gc$len.emp[gc$len.emp == 3] = 2
gc$len.emp[gc$len.emp == 4] = 3
gc$len.emp[gc$len.emp == 5] = 4
gc$len.emp = factor(gc$len.emp, levels = seq(1:4),
                    labels = c('< 1', '1-4', '4-7', '>7'))

table(gc$Occupation)/sum(table(gc$Occupation))
```

```
##
##     1     2     3     4
## 0.022 0.200 0.630 0.148
```

```r
gc$Occupation[gc$Occupation == 2] = 1
gc$Occupation[gc$Occupation == 3] = 2
gc$Occupation[gc$Occupation == 4] = 3
gc$Occupation = factor(gc$Occupation, levels = seq(1:3),
                       labels = c('unemp', 'skilled', 'exec'))

names(gc)[10] = 'sex'
table(gc$sex)/sum(table(gc$sex))
```

```
##
##     1     2     3     4
## 0.050 0.310 0.548 0.092
```

```r
gc$sex[gc$sex == 2] = 1
gc$sex[gc$sex == 3] = 2
gc$sex[gc$sex == 4] = 3
gc$sex = factor(gc$sex, levels = seq(1:3),
                labels = c('single male', 'married male', 'female'))

names(gc)[17] = 'cr.at.bank'
table(gc$cr.at.bank)/sum(table(gc$cr.at.bank))
```

```
##
##     1     2     3     4
## 0.633 0.333 0.028 0.006
```

```r
gc$cr.at.bank[gc$cr.at.bank >= 2] = 2
gc$cr.at.bank = factor(gc$cr.at.bank, levels = seq(1:2),
                       labels = c('1', '>1'))

table(gc$Guarantors)/sum(table(gc$Guarantors))
```

```
##
##     1     2     3
## 0.907 0.041 0.052
```

```r
gc$Guarantors[gc$Guarantors >= 2] = 2
gc$Guarantors = factor(gc$Guarantors, levels = seq(1:2),
                       labels = c('no', 'yes'))

table(gc$Concurrent.Credits)/sum(table(gc$Concurrent.Credits))
```

```
##
##     1     2     3
## 0.139 0.047 0.814
```

```r
gc$Concurrent.Credits[gc$Concurrent.Credits <= 2] = 1
gc$Concurrent.Credits[gc$Concurrent.Credits <= 3] = 2
gc$Concurrent.Credits = factor(gc$Concurrent.Credits, levels = seq(1:2),
                               labels = c('yes', 'no'))

table(gc$Purpose)/sum(table(gc$Purpose))*100
```

```
##
##    0    1    2    3    4    5    6    8    9   10
## 23.4 10.3 18.1 28.0  1.2  2.2  5.0  0.9  9.7  1.2
```

```r
gc$Purpose[gc$Purpose %in% c(2, 3, 4, 5)] = 3
gc$Purpose[gc$Purpose %in% c(6, 7, 8, 9)] = 4
gc$Purpose[gc$Purpose == 1] = 2
gc$Purpose[gc$Purpose == 0] = 1
gc$Purpose = factor(gc$Purpose, levels = seq(1:4),
                    labels = c('new car', 'used car', 'house', 'other'))
```

```r
str(gc)
```

```
## 'data.frame':    1000 obs. of  21 variables:
##  $ cr                           : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
##  $ acc.bal                      : Factor w/ 3 levels "No Acc","No Bal",..: 1 1 2 1 1 1 1 1 3 2 ...
##  $ Duration.of.Credit..month.   : int  18 9 12 12 12 10 8 6 18 24 ...
##  $ pmt.stat.prev.cr             : Factor w/ 3 levels "Some probs","paid up",..: 3 3 2 3 3 3 3 3 3 2 ...
##  $ Purpose                      : Factor w/ 4 levels "new car","used car",..: 3 1 4 1 1 1 1 1 3 3 ...
##  $ Credit.Amount                : int  1049 2799 841 2122 2171 2241 3398 1361 1098 3758 ...
##  $ savings.stocks               : Factor w/ 4 levels "none","<100",..: 1 1 2 1 1 1 1 1 1 3 ...
##  $ len.emp                      : Factor w/ 4 levels "< 1","1-4","4-7",..: 1 2 3 2 2 1 3 1 1 1 ...
##  $ Instalment.per.cent          : int  4 2 2 3 4 1 1 2 4 1 ...
##  $ sex                          : Factor w/ 3 levels "single male",..: 1 2 1 2 2 2 2 2 1 1 ...
##  $ Guarantors                   : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Duration.in.Current.address  : int  4 2 4 2 4 3 4 4 4 4 ...
##  $ Most.valuable.available.asset: int  2 1 1 1 2 1 1 1 3 4 ...
##  $ Age..years.                  : int  21 36 23 39 38 48 39 40 65 23 ...
##  $ Concurrent.Credits           : Factor w/ 2 levels "yes","no": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Type.of.apartment            : int  1 1 1 1 2 1 2 2 2 1 ...
##  $ cr.at.bank                   : Factor w/ 2 levels "1",">1": 1 2 1 2 2 2 2 1 2 1 ...
##  $ Occupation                   : Factor w/ 3 levels "unemp","skilled",..: 2 2 1 1 1 1 1 1 1 1 ...
##  $ No.of.dependents             : int  1 2 1 2 1 2 1 2 1 1 ...
##  $ Telephone                    : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Foreign.Worker               : int  1 1 1 2 2 2 2 2 1 1 ...
```

Statistical testing

```r
cat.table = data.frame(var = character(), p.value = numeric())

for (i in colnames(gc))

str(chisq.test(gc$cr, gc$acc.bal))$p.value
```

```
## List of 9
##  $ statistic: Named num 121
##   ..- attr(*, "names")= chr "X-squared"
##  $ parameter: Named int 2
##   ..- attr(*, "names")= chr "df"
##  $ p.value  : num 5.74e-27
##  $ method   : chr "Pearson's Chi-squared test"
##  $ data.name: chr "gc$cr and gc$acc.bal"
##  $ observed : 'table' int [1:2, 1:3] 135 139 105 164 60 397
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ expected : num [1:2, 1:3] 82.2 191.8 80.7 188.3 137.1 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ residuals: 'table' num [1:2, 1:3] 5.82 -3.81 2.71 -1.77 -6.58 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
```

```
##  $ stdres   : 'table' num [1:2, 1:3] 8.17 -8.17 3.78 -3.78 -10.68 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  - attr(*, "class")= chr "htest"
## List of 9
##  $ statistic: Named num 121
##   ..- attr(*, "names")= chr "X-squared"
##  $ parameter: Named int 2
##   ..- attr(*, "names")= chr "df"
##  $ p.value  : num 5.74e-27
##  $ method   : chr "Pearson's Chi-squared test"
##  $ data.name: chr "gc$cr and gc$acc.bal"
##  $ observed : 'table' int [1:2, 1:3] 135 139 105 164 60 397
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ expected : num [1:2, 1:3] 82.2 191.8 80.7 188.3 137.1 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ residuals: 'table' num [1:2, 1:3] 5.82 -3.81 2.71 -1.77 -6.58 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ stdres   : 'table' num [1:2, 1:3] 8.17 -8.17 3.78 -3.78 -10.68 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  - attr(*, "class")= chr "htest"
## List of 9
##  $ statistic: Named num 121
##   ..- attr(*, "names")= chr "X-squared"
##  $ parameter: Named int 2
##   ..- attr(*, "names")= chr "df"
##  $ p.value  : num 5.74e-27
##  $ method   : chr "Pearson's Chi-squared test"
##  $ data.name: chr "gc$cr and gc$acc.bal"
##  $ observed : 'table' int [1:2, 1:3] 135 139 105 164 60 397
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ expected : num [1:2, 1:3] 82.2 191.8 80.7 188.3 137.1 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ residuals: 'table' num [1:2, 1:3] 5.82 -3.81 2.71 -1.77 -6.58 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ stdres   : 'table' num [1:2, 1:3] 8.17 -8.17 3.78 -3.78 -10.68 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  - attr(*, "class")= chr "htest"
## List of 9
##  $ statistic: Named num 121
##   ..- attr(*, "names")= chr "X-squared"
##  $ parameter: Named int 2
##   ..- attr(*, "names")= chr "df"
##  $ p.value  : num 5.74e-27
##  $ method   : chr "Pearson's Chi-squared test"
##  $ data.name: chr "gc$cr and gc$acc.bal"
##  $ observed : 'table' int [1:2, 1:3] 135 139 105 164 60 397
```

```
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ expected : num [1:2, 1:3] 82.2 191.8 80.7 188.3 137.1 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ residuals: 'table' num [1:2, 1:3] 5.82 -3.81 2.71 -1.77 -6.58 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ stdres   : 'table' num [1:2, 1:3] 8.17 -8.17 3.78 -3.78 -10.68 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  - attr(*, "class")= chr "htest"
## List of 9
##  $ statistic: Named num 121
##   ..- attr(*, "names")= chr "X-squared"
##  $ parameter: Named int 2
##   ..- attr(*, "names")= chr "df"
##  $ p.value  : num 5.74e-27
##  $ method   : chr "Pearson's Chi-squared test"
##  $ data.name: chr "gc$cr and gc$acc.bal"
##  $ observed : 'table' int [1:2, 1:3] 135 139 105 164 60 397
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ expected : num [1:2, 1:3] 82.2 191.8 80.7 188.3 137.1 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ residuals: 'table' num [1:2, 1:3] 5.82 -3.81 2.71 -1.77 -6.58 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ stdres   : 'table' num [1:2, 1:3] 8.17 -8.17 3.78 -3.78 -10.68 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  - attr(*, "class")= chr "htest"
## List of 9
##  $ statistic: Named num 121
##   ..- attr(*, "names")= chr "X-squared"
##  $ parameter: Named int 2
##   ..- attr(*, "names")= chr "df"
##  $ p.value  : num 5.74e-27
##  $ method   : chr "Pearson's Chi-squared test"
##  $ data.name: chr "gc$cr and gc$acc.bal"
##  $ observed : 'table' int [1:2, 1:3] 135 139 105 164 60 397
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ expected : num [1:2, 1:3] 82.2 191.8 80.7 188.3 137.1 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ residuals: 'table' num [1:2, 1:3] 5.82 -3.81 2.71 -1.77 -6.58 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ stdres   : 'table' num [1:2, 1:3] 8.17 -8.17 3.78 -3.78 -10.68 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
```

```
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  - attr(*, "class")= chr "htest"
## List of 9
##  $ statistic: Named num 121
##   ..- attr(*, "names")= chr "X-squared"
##  $ parameter: Named int 2
##   ..- attr(*, "names")= chr "df"
##  $ p.value  : num 5.74e-27
##  $ method   : chr "Pearson's Chi-squared test"
##  $ data.name: chr "gc$cr and gc$acc.bal"
##  $ observed : 'table' int [1:2, 1:3] 135 139 105 164 60 397
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ expected : num [1:2, 1:3] 82.2 191.8 80.7 188.3 137.1 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ residuals: 'table' num [1:2, 1:3] 5.82 -3.81 2.71 -1.77 -6.58 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ stdres   : 'table' num [1:2, 1:3] 8.17 -8.17 3.78 -3.78 -10.68 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  - attr(*, "class")= chr "htest"
## List of 9
##  $ statistic: Named num 121
##   ..- attr(*, "names")= chr "X-squared"
##  $ parameter: Named int 2
##   ..- attr(*, "names")= chr "df"
##  $ p.value  : num 5.74e-27
##  $ method   : chr "Pearson's Chi-squared test"
##  $ data.name: chr "gc$cr and gc$acc.bal"
##  $ observed : 'table' int [1:2, 1:3] 135 139 105 164 60 397
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ expected : num [1:2, 1:3] 82.2 191.8 80.7 188.3 137.1 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ residuals: 'table' num [1:2, 1:3] 5.82 -3.81 2.71 -1.77 -6.58 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ stdres   : 'table' num [1:2, 1:3] 8.17 -8.17 3.78 -3.78 -10.68 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  - attr(*, "class")= chr "htest"
## List of 9
##  $ statistic: Named num 121
##   ..- attr(*, "names")= chr "X-squared"
##  $ parameter: Named int 2
##   ..- attr(*, "names")= chr "df"
##  $ p.value  : num 5.74e-27
##  $ method   : chr "Pearson's Chi-squared test"
##  $ data.name: chr "gc$cr and gc$acc.bal"
##  $ observed : 'table' int [1:2, 1:3] 135 139 105 164 60 397
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
```

```
##  $ expected : num [1:2, 1:3] 82.2 191.8 80.7 188.3 137.1 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ residuals: 'table' num [1:2, 1:3] 5.82 -3.81 2.71 -1.77 -6.58 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ stdres   : 'table' num [1:2, 1:3] 8.17 -8.17 3.78 -3.78 -10.68 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  - attr(*, "class")= chr "htest"
## List of 9
##  $ statistic: Named num 121
##   ..- attr(*, "names")= chr "X-squared"
##  $ parameter: Named int 2
##   ..- attr(*, "names")= chr "df"
##  $ p.value  : num 5.74e-27
##  $ method   : chr "Pearson's Chi-squared test"
##  $ data.name: chr "gc$cr and gc$acc.bal"
##  $ observed : 'table' int [1:2, 1:3] 135 139 105 164 60 397
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ expected : num [1:2, 1:3] 82.2 191.8 80.7 188.3 137.1 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ residuals: 'table' num [1:2, 1:3] 5.82 -3.81 2.71 -1.77 -6.58 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ stdres   : 'table' num [1:2, 1:3] 8.17 -8.17 3.78 -3.78 -10.68 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  - attr(*, "class")= chr "htest"
## List of 9
##  $ statistic: Named num 121
##   ..- attr(*, "names")= chr "X-squared"
##  $ parameter: Named int 2
##   ..- attr(*, "names")= chr "df"
##  $ p.value  : num 5.74e-27
##  $ method   : chr "Pearson's Chi-squared test"
##  $ data.name: chr "gc$cr and gc$acc.bal"
##  $ observed : 'table' int [1:2, 1:3] 135 139 105 164 60 397
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ expected : num [1:2, 1:3] 82.2 191.8 80.7 188.3 137.1 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ residuals: 'table' num [1:2, 1:3] 5.82 -3.81 2.71 -1.77 -6.58 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ stdres   : 'table' num [1:2, 1:3] 8.17 -8.17 3.78 -3.78 -10.68 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  - attr(*, "class")= chr "htest"
## List of 9
```

```
##  $ statistic: Named num 121
##   ..- attr(*, "names")= chr "X-squared"
##  $ parameter: Named int 2
##   ..- attr(*, "names")= chr "df"
##  $ p.value  : num 5.74e-27
##  $ method   : chr "Pearson's Chi-squared test"
##  $ data.name: chr "gc$cr and gc$acc.bal"
##  $ observed : 'table' int [1:2, 1:3] 135 139 105 164 60 397
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ expected : num [1:2, 1:3] 82.2 191.8 80.7 188.3 137.1 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ residuals: 'table' num [1:2, 1:3] 5.82 -3.81 2.71 -1.77 -6.58 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ stdres   : 'table' num [1:2, 1:3] 8.17 -8.17 3.78 -3.78 -10.68 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  - attr(*, "class")= chr "htest"
## List of 9
##  $ statistic: Named num 121
##   ..- attr(*, "names")= chr "X-squared"
##  $ parameter: Named int 2
##   ..- attr(*, "names")= chr "df"
##  $ p.value  : num 5.74e-27
##  $ method   : chr "Pearson's Chi-squared test"
##  $ data.name: chr "gc$cr and gc$acc.bal"
##  $ observed : 'table' int [1:2, 1:3] 135 139 105 164 60 397
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ expected : num [1:2, 1:3] 82.2 191.8 80.7 188.3 137.1 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ residuals: 'table' num [1:2, 1:3] 5.82 -3.81 2.71 -1.77 -6.58 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ stdres   : 'table' num [1:2, 1:3] 8.17 -8.17 3.78 -3.78 -10.68 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  - attr(*, "class")= chr "htest"
## List of 9
##  $ statistic: Named num 121
##   ..- attr(*, "names")= chr "X-squared"
##  $ parameter: Named int 2
##   ..- attr(*, "names")= chr "df"
##  $ p.value  : num 5.74e-27
##  $ method   : chr "Pearson's Chi-squared test"
##  $ data.name: chr "gc$cr and gc$acc.bal"
##  $ observed : 'table' int [1:2, 1:3] 135 139 105 164 60 397
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ expected : num [1:2, 1:3] 82.2 191.8 80.7 188.3 137.1 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
```

```
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ residuals: 'table' num [1:2, 1:3] 5.82 -3.81 2.71 -1.77 -6.58 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ stdres   : 'table' num [1:2, 1:3] 8.17 -8.17 3.78 -3.78 -10.68 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  - attr(*, "class")= chr "htest"
## List of 9
##  $ statistic: Named num 121
##   ..- attr(*, "names")= chr "X-squared"
##  $ parameter: Named int 2
##   ..- attr(*, "names")= chr "df"
##  $ p.value  : num 5.74e-27
##  $ method   : chr "Pearson's Chi-squared test"
##  $ data.name: chr "gc$cr and gc$acc.bal"
##  $ observed : 'table' int [1:2, 1:3] 135 139 105 164 60 397
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ expected : num [1:2, 1:3] 82.2 191.8 80.7 188.3 137.1 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ residuals: 'table' num [1:2, 1:3] 5.82 -3.81 2.71 -1.77 -6.58 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ stdres   : 'table' num [1:2, 1:3] 8.17 -8.17 3.78 -3.78 -10.68 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  - attr(*, "class")= chr "htest"
## List of 9
##  $ statistic: Named num 121
##   ..- attr(*, "names")= chr "X-squared"
##  $ parameter: Named int 2
##   ..- attr(*, "names")= chr "df"
##  $ p.value  : num 5.74e-27
##  $ method   : chr "Pearson's Chi-squared test"
##  $ data.name: chr "gc$cr and gc$acc.bal"
##  $ observed : 'table' int [1:2, 1:3] 135 139 105 164 60 397
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ expected : num [1:2, 1:3] 82.2 191.8 80.7 188.3 137.1 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ residuals: 'table' num [1:2, 1:3] 5.82 -3.81 2.71 -1.77 -6.58 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ stdres   : 'table' num [1:2, 1:3] 8.17 -8.17 3.78 -3.78 -10.68 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  - attr(*, "class")= chr "htest"
## List of 9
##  $ statistic: Named num 121
##   ..- attr(*, "names")= chr "X-squared"
##  $ parameter: Named int 2
```

```
##   ..- attr(*, "names")= chr "df"
##  $ p.value  : num 5.74e-27
##  $ method   : chr "Pearson's Chi-squared test"
##  $ data.name: chr "gc$cr and gc$acc.bal"
##  $ observed : 'table' int [1:2, 1:3] 135 139 105 164 60 397
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ expected : num [1:2, 1:3] 82.2 191.8 80.7 188.3 137.1 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ residuals: 'table' num [1:2, 1:3] 5.82 -3.81 2.71 -1.77 -6.58 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ stdres   : 'table' num [1:2, 1:3] 8.17 -8.17 3.78 -3.78 -10.68 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  - attr(*, "class")= chr "htest"
## List of 9
##  $ statistic: Named num 121
##   ..- attr(*, "names")= chr "X-squared"
##  $ parameter: Named int 2
##   ..- attr(*, "names")= chr "df"
##  $ p.value  : num 5.74e-27
##  $ method   : chr "Pearson's Chi-squared test"
##  $ data.name: chr "gc$cr and gc$acc.bal"
##  $ observed : 'table' int [1:2, 1:3] 135 139 105 164 60 397
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ expected : num [1:2, 1:3] 82.2 191.8 80.7 188.3 137.1 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ residuals: 'table' num [1:2, 1:3] 5.82 -3.81 2.71 -1.77 -6.58 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ stdres   : 'table' num [1:2, 1:3] 8.17 -8.17 3.78 -3.78 -10.68 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  - attr(*, "class")= chr "htest"
## List of 9
##  $ statistic: Named num 121
##   ..- attr(*, "names")= chr "X-squared"
##  $ parameter: Named int 2
##   ..- attr(*, "names")= chr "df"
##  $ p.value  : num 5.74e-27
##  $ method   : chr "Pearson's Chi-squared test"
##  $ data.name: chr "gc$cr and gc$acc.bal"
##  $ observed : 'table' int [1:2, 1:3] 135 139 105 164 60 397
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ expected : num [1:2, 1:3] 82.2 191.8 80.7 188.3 137.1 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ residuals: 'table' num [1:2, 1:3] 5.82 -3.81 2.71 -1.77 -6.58 ...
##   ..- attr(*, "dimnames")=List of 2
```

```
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ stdres   : 'table' num [1:2, 1:3] 8.17 -8.17 3.78 -3.78 -10.68 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  - attr(*, "class")= chr "htest"
## List of 9
##  $ statistic: Named num 121
##   ..- attr(*, "names")= chr "X-squared"
##  $ parameter: Named int 2
##   ..- attr(*, "names")= chr "df"
##  $ p.value  : num 5.74e-27
##  $ method   : chr "Pearson's Chi-squared test"
##  $ data.name: chr "gc$cr and gc$acc.bal"
##  $ observed : 'table' int [1:2, 1:3] 135 139 105 164 60 397
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ expected : num [1:2, 1:3] 82.2 191.8 80.7 188.3 137.1 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ residuals: 'table' num [1:2, 1:3] 5.82 -3.81 2.71 -1.77 -6.58 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ stdres   : 'table' num [1:2, 1:3] 8.17 -8.17 3.78 -3.78 -10.68 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  - attr(*, "class")= chr "htest"
## List of 9
##  $ statistic: Named num 121
##   ..- attr(*, "names")= chr "X-squared"
##  $ parameter: Named int 2
##   ..- attr(*, "names")= chr "df"
##  $ p.value  : num 5.74e-27
##  $ method   : chr "Pearson's Chi-squared test"
##  $ data.name: chr "gc$cr and gc$acc.bal"
##  $ observed : 'table' int [1:2, 1:3] 135 139 105 164 60 397
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ expected : num [1:2, 1:3] 82.2 191.8 80.7 188.3 137.1 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ residuals: 'table' num [1:2, 1:3] 5.82 -3.81 2.71 -1.77 -6.58 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  $ stdres   : 'table' num [1:2, 1:3] 8.17 -8.17 3.78 -3.78 -10.68 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ gc$cr     : chr [1:2] "0" "1"
##   .. ..$ gc$acc.bal: chr [1:3] "No Acc" "No Bal" "Has Bal"
##  - attr(*, "class")= chr "htest"
```

```r
chisq.test(gc$cr, gc$acc.bal)$statistic
```

```
## X-squared
##  120.8438
```

```r
library(gmodels)

CrossTable(gc$cr,
           gc$acc.bal, digits = 1, prop.r = F, prop.t = F,
           prop.chisq = F, chisq = T)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Col Total |
## |-------------------------|
##
##
## Total Observations in Table:  1000
##
##
##              | gc$acc.bal
##        gc$cr |    No Acc |    No Bal |   Has Bal | Row Total |
## -------------|-----------|-----------|-----------|-----------|
##            0 |       135 |       105 |        60 |       300 |
##              |       0.5 |       0.4 |       0.1 |           |
## -------------|-----------|-----------|-----------|-----------|
##            1 |       139 |       164 |       397 |       700 |
##              |       0.5 |       0.6 |       0.9 |           |
## -------------|-----------|-----------|-----------|-----------|
## Column Total |       274 |       269 |       457 |      1000 |
##              |       0.3 |       0.3 |       0.5 |           |
## -------------|-----------|-----------|-----------|-----------|
##
##
## Statistics for All Table Factors
##
##
## Pearson's Chi-squared test
## ------------------------------------------------------------
## Chi^2 =  120.8438     d.f. =  2     p =  5.742621e-27
##
##
##
```

```r
# digits : how many number after a decimal point
# prop.r : percentage (proportion) of row total

margin.table(prop.table(table(gc$cr, gc$acc.bal)),1)
```

```
##
##   0   1
## 0.3 0.7
```

Data Pre-processing

- Feature Engineering: Create new relevant features from existing data (e.g., extracting time-based features from a time stamp)
- Feature scaling: Standardize or normalize to ensure numerical stability
- Feature selection: Remove irrelevant feature or correlated feature

```r
# look for the dependency of the response (Creditability) on each of the predictor variable
```

```
# for both are numeric variable, we can use correlation, cor()
# In case of binary logistic regression where the response is binary categorical variable,
# if both are categorical variable, we use chi square test


# chi square test
chisq.test(gc$cr, gc$acc.bal)
```

```
##
##  Pearson's Chi-squared test
##
## data:  gc$cr and gc$acc.bal
## X-squared = 120.84, df = 2, p-value < 2.2e-16
```

```
chisq.test(gc$cr, gc$Instalment.per.cent)
```

```
##
##  Pearson's Chi-squared test
##
## data:  gc$cr and gc$Instalment.per.cent
## X-squared = 5.4768, df = 3, p-value = 0.14
```

Model Training

- Select appropriate ML algorithm
- Split data into test/train
- Train the model

```
# split data into train and test
indexes = sample(1:nrow(gc), size = 0.5 * nrow(gc))
# sample from dataset of a seq of number 1 to 1000 and take 500 sample dataset from it

train = gc[indexes, ]
test = gc[-indexes, ]
# this step wont carry the data type of the previous data, you'll need to re-convert tha data type into factor whe
```

generalized linear model = glm()

- when y is binary

$H_0 : B_j = 0$

$H_1 : B_j$ a b

```
model = glm(cr~acc.bal, family=binomial, data = gc)

summary(model)
```

```
##
## Call:
## glm(formula = cr ~ acc.bal, family = binomial, data = gc)
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)       0.0292     0.1208   0.242   0.8091
## acc.balNo Bal     0.4167     0.1739   2.397   0.0165 *
## acc.balHas Bal    1.8604     0.1838  10.121   <2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1221.7  on 999  degrees of freedom
## Residual deviance: 1095.0  on 997  degrees of freedom
## AIC: 1101
##
## Number of Fisher Scoring iterations: 4
```

Remove non-significant variable, repeat this step until all variables in the model are significant

Model Evaluation

```r
fit = fitted.values(model)

# setting threshold
t = rep(0,500)
for (i in 1:500) {
  if (fit[i] >= 0.5) {
    t[i] = 1
  }
}

# create cross table
conf.mat = table(t, train$cr)
```

Tree Based Method

Model Training

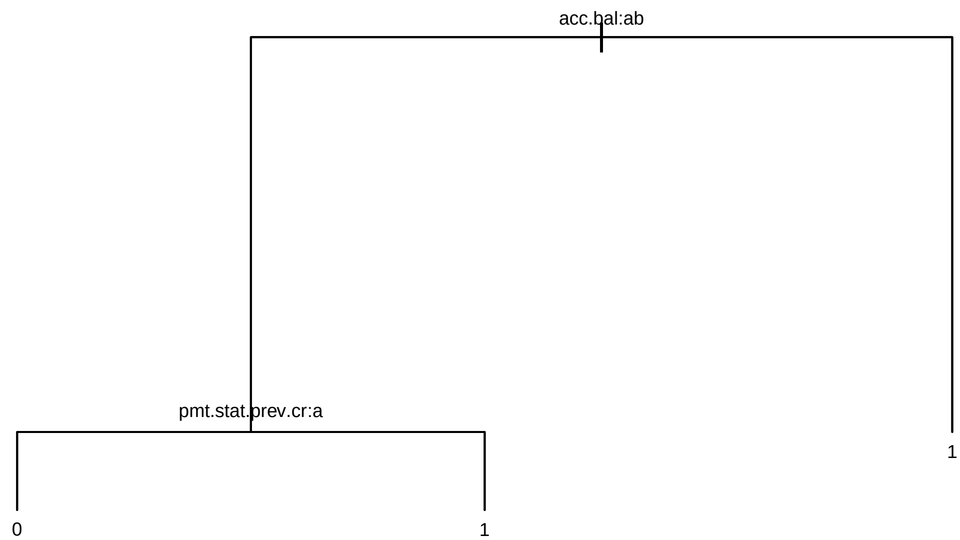
```r
library(tree)

model_tree = tree(cr~acc.bal+pmt.stat.prev.cr, data = train,
                    method = 'class')

summary
```

```
## function (object, ...)
## UseMethod("summary")
## <bytecode: 0x000001df7d719370>
## <environment: namespace:base>
```

```r
plot(model_tree)
text(model_tree, cex = 0.6)
```



Model Evaluation

Evaluate train set

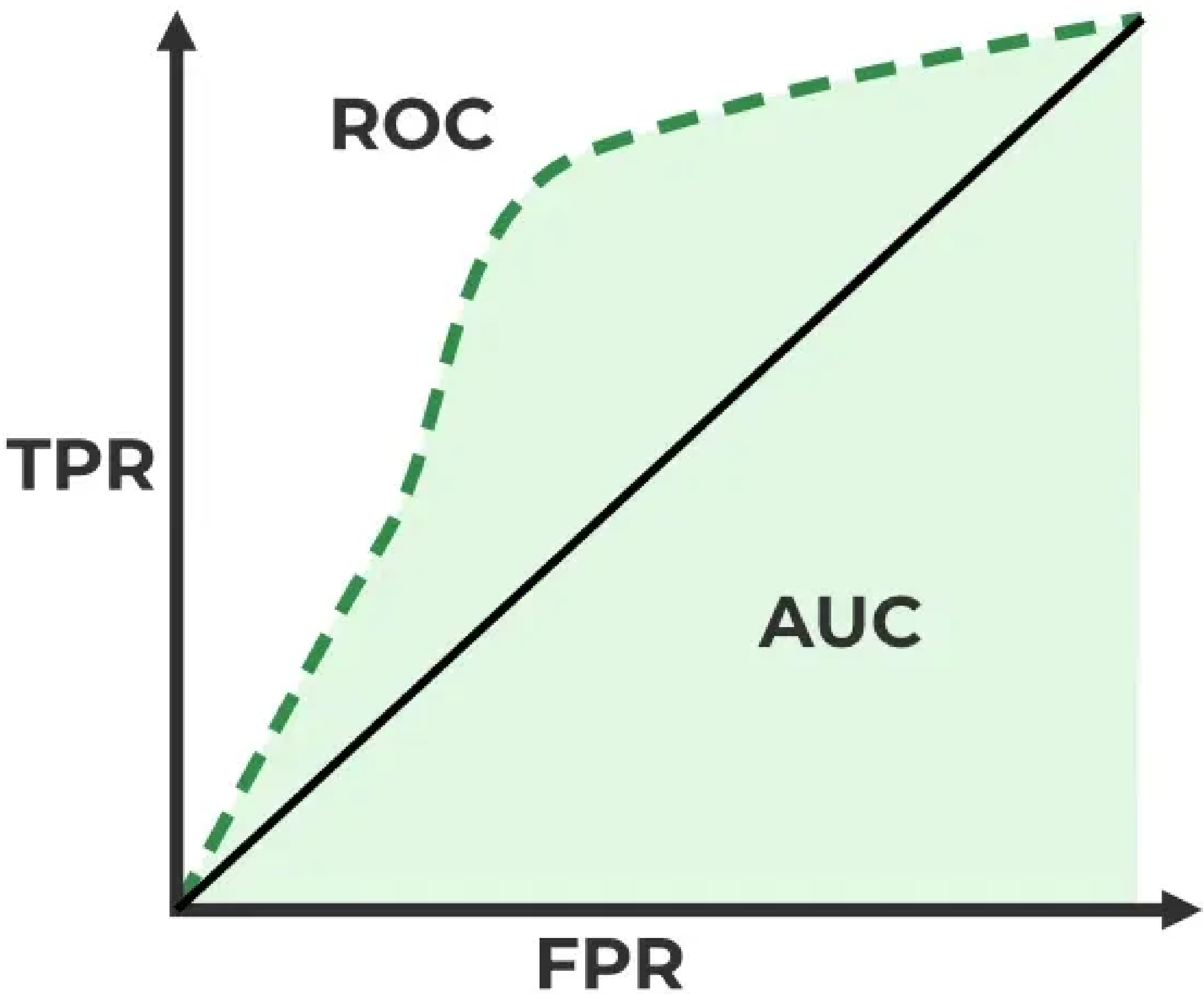
```
train_pred = predict(model_tree, train, type='class')
ct1 = table(train$cr, train_pred)
t_train_acc = sum(diag(ct1))/500*100
```

Evaluate test set

AUC-ROC curve

Used for evaluating binary classification model. Plots **True Positive Rate (TPR) VS False Positive Rate (FPR)** at different threshold

- TPR : ratio of correctly predicted positive instances
- FPR : ratio of wrongly predicted positive instances
- AUC : Area Under the Curve
- ROC : Receiver Operating Characteristic Curve



Pruning

```
model_tree_prune = prune.misclass(model_tree, best = 8)
```

```
## Warning in prune.tree(tree = model_tree, best = 8, method = "misclass"): best
## is bigger than tree size
```