Projet Analyse de Credit Scoring 2 avec Data mining

Itaf Hamrouni Nour El Islem Nbiba Sarra Madani

13 décembre 2017

Project: Analyse Credit Scoring basée sur data mining

Charger Library

```
library("MASS")
```

```
## Warning: package 'MASS' was built under R version 3.4.3
```

```
library("FactoMineR")
```

```
## Warning: package 'FactoMineR' was built under R version 3.4.3
```

```
library(ggplot2)
```

Importation de base apres faire nettoyage et renommage

```
data= read.csv("C:/Users/user/Desktop/3eme/datamining/projet data mining/tp/CleanCreditScoring.csv", hea
View(data)
dim(data)
```

```
## [1] 4446   27
```

```
str(data)
```

```
## 'data.frame':    4446 obs. of  27 variables:
##  $ Status    : Factor w/ 2 levels "bad","good": 2 2 1 2 2 2 2 2 2 1 ...
##  $ Seniority : int  9 17 10 0 0 1 29 9 0 0 ...
##  $ Home      : Factor w/ 6 levels "ignore","other",..: 6 6 3 6 6 3 3 4 3 4 ...
##  $ Time      : int  60 60 36 60 36 60 60 12 60 48 ...
##  $ Age       : int  30 58 46 24 26 36 44 27 32 41 ...
##  $ Marital   : Factor w/ 5 levels "divorced","married",..: 2 5 2 4 4 2 2 4 2 2 ...
##  $ Records   : Factor w/ 2 levels "no_rec","yes_rec": 1 1 2 1 1 1 1 1 1 1 ...
##  $ Job       : Factor w/ 4 levels "fixed","freelance",..: 2 1 2 1 1 1 1 1 2 4 ...
##  $ Expenses  : int  73 48 90 63 46 75 75 35 90 90 ...
##  $ Income    : int  129 131 200 182 107 214 125 80 107 80 ...
##  $ Assets    : int  0 0 3000 2500 0 3500 10000 0 15000 0 ...
##  $ Debt      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Amount    : int  800 1000 2000 900 310 650 1600 200 1200 1200 ...
##  $ Price     : int  846 1658 2985 1325 910 1645 1800 1093 1957 1468 ...
##  $ Finrat    : num  94.6 60.3 67 67.9 34.1 ...
##  $ Savings   : num  4.2 4.98 1.98 7.93 7.08 ...
##  $ seniorityR: Factor w/ 5 levels "sen (-1,1]","sen (1,3]",..: 5 3 5 1 1 1 3 5 1 1 ...
##  $ timeR     : Factor w/ 5 levels "time (0,12]",..: 5 5 3 5 3 5 5 1 5 4 ...
##  $ ageR      : Factor w/ 5 levels "age (0,25]","age (25,30]",..: 2 5 4 1 2 3 4 2 3 4 ...
##  $ expensesR : Factor w/ 5 levels "exp (0,40]","exp (40,50]",..: 4 2 5 4 2 4 4 1 5 5 ...
##  $ incomeR   : Factor w/ 5 levels "inc (0,80]","inc (110,140]",..: 2 2 4 3 5 4 2 1 5 1 ...
##  $ assetsR   : Factor w/ 5 levels "asset (-1,0]",..: 1 1 2 2 1 3 5 1 5 1 ...
##  $ debtR     : Factor w/ 5 levels "debt (-1,0]",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ amountR   : Factor w/ 5 levels "am (0,600]","am (1.1e+03,1.4e+03]",..: 4 5 3 4 1 4 3 1 2 2 ...
##  $ priceR    : Factor w/ 5 levels "priz (0,1e+03]",..: 1 3 4 2 1 3 3 5 4 2 ...
##  $ finratR   : Factor w/ 5 levels "finr (0,50]",..: 5 2 2 2 1 1 4 1 2 4 ...
```

```
##  $ savingsR  : Factor w/ 5 levels "sav (-99,0]",..: 4 4 2 5 5 5 2 3 2 1 ...
```

See what the data looks like : Summary

```
head(data)
```

```
##   Status Seniority  Home Time Age Marital Records       Job Expenses
## 1   good         9  rent   60  30 married  no_rec freelance       73
## 2   good        17  rent   60  58   widow  no_rec     fixed       48
## 3    bad        10 owner   36  46 married yes_rec freelance       90
## 4   good         0  rent   60  24  single  no_rec     fixed       63
## 5   good         0  rent   36  26  single  no_rec     fixed       46
## 6   good         1 owner   60  36 married  no_rec     fixed       75
##   Income Assets Debt Amount Price   Finrat   Savings  seniorityR
## 1    129      0    0    800   846 94.56265  4.200000  sen (8,14]
## 2    131      0    0   1000  1658 60.31363  4.980000 sen (14,99]
## 3    200   3000    0   2000  2985 67.00168  1.980000  sen (8,14]
## 4    182   2500    0    900  1325 67.92453  7.933333  sen (-1,1]
## 5    107      0    0    310   910 34.06593  7.083871  sen (-1,1]
## 6    214   3500    0    650  1645 39.51368 12.830769  sen (-1,1]
##          timeR        ageR      expensesR         incomeR
## 1 time (48,99] age (25,30]    exp (60,80]   inc (110,140]
## 2 time (48,99] age (50,99]    exp (40,50]   inc (110,140]
## 3 time (24,36] age (40,50] exp (80,1e+04] inc (190,1e+04]
## 4 time (48,99]  age (0,25]    exp (60,80]   inc (140,190]
## 5 time (24,36] age (25,30]    exp (40,50]    inc (80,110]
## 6 time (48,99] age (30,40]    exp (60,80] inc (190,1e+04]
##               assetsR       debtR            amountR
## 1        asset (-1,0] debt (-1,0]       am (600,900]
## 2        asset (-1,0] debt (-1,0]   am (900,1.1e+03]
## 3     asset (0,3e+03] debt (-1,0] am (1.4e+03,1e+05]
## 4     asset (0,3e+03] debt (-1,0]       am (600,900]
## 5        asset (-1,0] debt (-1,0]         am (0,600]
## 6 asset (3e+03,5e+03] debt (-1,0]       am (600,900]
##                   priceR       finratR   savingsR
## 1         priz (0,1e+03] finr (90,100]  sav (4,6]
## 2 priz (1.5e+03,1.8e+03]  finr (50,70]  sav (4,6]
## 3   priz (1.8e+03,1e+05]  finr (50,70]  sav (0,2]
## 4 priz (1.3e+03,1.5e+03]  finr (50,70] sav (6,99]
## 5         priz (0,1e+03]   finr (0,50] sav (6,99]
## 6 priz (1.5e+03,1.8e+03]   finr (0,50] sav (6,99]
```

```
summary(data)
```

```
##   Status       Seniority           Home           Time
##  bad :1249   Min.   : 0.000   ignore :  20   Min.   : 6.00
##  good:3197   1st Qu.: 2.000   other  : 319   1st Qu.:36.00
##              Median : 5.000   owner  :2106   Median :48.00
##              Mean   : 7.991   parents: 782   Mean   :46.45
##              3rd Qu.:12.000   priv   : 246   3rd Qu.:60.00
##              Max.   :48.000   rent   : 973   Max.   :72.00
##       Age             Marital        Records            Job
##  Min.   :18.00   divorced :  38   no_rec :3677   fixed    :2803
##  1st Qu.:28.00   married  :3238   yes_rec: 769   freelance:1021
##  Median :36.00   separated: 130                  others   : 171
##  Mean   :37.08   single   : 973                  partime  : 451
```

```
##  3rd Qu.:45.00   widow    :  67
##  Max.   :68.00
##     Expenses         Income          Assets            Debt
##  Min.   : 35.0   Min.   :  1.0   Min.   :     0   Min.   :    0.0
##  1st Qu.: 35.0   1st Qu.: 90.0   1st Qu.:     0   1st Qu.:    0.0
##  Median : 51.0   Median :124.0   Median :  3000   Median :    0.0
##  Mean   : 55.6   Mean   :140.6   Mean   :  5355   Mean   :  342.3
##  3rd Qu.: 72.0   3rd Qu.:170.0   3rd Qu.:  6000   3rd Qu.:    0.0
##  Max.   :180.0   Max.   :959.0   Max.   :300000   Max.   :30000.0
##      Amount          Price           Finrat           Savings
##  Min.   : 100   Min.   :  105   Min.   :  6.702   Min.   :-8.160
##  1st Qu.: 700   1st Qu.: 1116   1st Qu.: 60.030   1st Qu.: 1.615
##  Median :1000   Median : 1400   Median : 77.097   Median : 3.120
##  Mean   :1039   Mean   : 1462   Mean   : 72.616   Mean   : 3.860
##  3rd Qu.:1300   3rd Qu.: 1692   3rd Qu.: 88.460   3rd Qu.: 5.196
##  Max.   :5000   Max.   :11140   Max.   :100.000   Max.   :33.250
##        seniorityR            timeR               ageR
##  sen (-1,1] :1042   time (0,12] : 180   age (0,25] : 699
##  sen (1,3]  : 789   time (12,24]: 441   age (25,30]: 781
##  sen (14,99]: 880   time (24,36]: 991   age (30,40]:1415
##  sen (3,8]  : 978   time (36,48]: 885   age (40,50]: 900
##  sen (8,14] : 757   time (48,99]:1949   age (50,99]: 651
##
##           expensesR               incomeR                    assetsR
##  exp (0,40]    :1219   inc (0,80]     :886   asset (-1,0]       :1626
##  exp (40,50]   : 999   inc (110,140]  :866   asset (0,3e+03]    : 626
##  exp (50,60]   : 979   inc (140,190]  :915   asset (3e+03,5e+03]: 937
##  exp (60,80]   : 798   inc (190,1e+04]:825   asset (5e+03,8e+03]: 538
##  exp (80,1e+04]: 451   inc (80,110]   :954   asset (8e+03,1e+06]: 719
##
##                     debtR                       amountR
##  debt (-1,0]           :3667   am (0,600]           :895
##  debt (0,500]          : 193   am (1.1e+03,1.4e+03]:925
##  debt (1.5e+03,2.5e+03]: 159   am (1.4e+03,1e+05]  :770
##  debt (2.5e+03,1e+06]  : 197   am (600,900]        :911
##  debt (500,1.5e+03]    : 230   am (900,1.1e+03]    :945
##
##                     priceR              finratR           savingsR
##  priz (0,1e+03]        : 821   finr (0,50]  :716   sav (-99,0]: 298
##  priz (1.3e+03,1.5e+03]: 801   finr (50,70] :954   sav (0,2]  :1111
##  priz (1.5e+03,1.8e+03]:1028   finr (70,80] :821   sav (2,4]  :1396
##  priz (1.8e+03,1e+05]  : 811   finr (80,90] :995   sav (4,6]  : 814
##  priz (1e+03,1.3e+03]  : 985   finr (90,100]:960   sav (6,99] : 827
##
```

```
            #Partie Clustering#
```

select select categorized continuous variables

```
datacat = subset(data, select=c(seniorityR, timeR, ageR, expensesR, incomeR,
                            assetsR, debtR, amountR, priceR, finratR, savingsR, Status))
```

MCA

```
mca = MCA(datacat, ncp=40, quali.sup=12, graph=FALSE)
```

get the eigenvalues

```
eig = mca$eig
eigs=eig[,"eigenvalue"]
```

calculate significant dimension in MCA

```
dimmca = sum(eigs>1/length(eigs))
```

get factorial coordinates of individuals

```
coord = mca$ind$coord
```

K-means clustering on factorical coordinates from MCA results

The idea is to take the extracted dimensions
from the MCA in order to perform a k-means cluster analysis on them The first approach is to apply K-means on factorial coordinates Let's try k=5 groups

```
k1 = kmeans(coord, 5)
```

what does k1 contain?

```
attributes(k1)
```

```
## $names
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
##
## $class
## [1] "kmeans"
```

examine the following

```
# size of clusters
k1$size
```

```
## [1]  565 1476 1011  715  679
```

```
# within cluster variance
k1$withinss
```

```
## [1] 1940.754 5033.395 3468.125 2336.681 2519.399
```

```
# centers coordinates
k1$centers
```

```
##         Dim 1       Dim 2       Dim 3       Dim 4       Dim 5       Dim 6
## 1  0.41050363  0.29784120  0.14156136 -0.32930930 -0.31533970  0.22048379
## 2  0.08591084 -0.35271853 -0.18323668  0.08042069 -0.05139202 -0.04856248
## 3 -0.40773771  0.04992065 -0.05997183  0.00980667 -0.19704363  0.10201872
## 4  0.41832313 -0.10860319  0.44820479  0.04388507  0.31378054 -0.15408231
## 5 -0.36173369  0.55893044 -0.10214983  0.03838946  0.33708331 -0.06755109
##         Dim 7       Dim 8       Dim 9      Dim 10      Dim 11       Dim 12
## 1 -0.17648171  0.1328949 -0.08776180  0.17282382  0.01340513  0.173048338
## 2 -0.07068248  0.1123239 -0.03677617 -0.10274026 -0.04651671 -0.009813235
## 3  0.20117256 -0.2017906  0.12557979  0.07952246 -0.04085965  0.033854551
## 4  0.16195934 -0.2248877  0.03057247  0.03241203  0.11887595 -0.096742170
## 5 -0.16958304  0.1825175 -0.06620536 -0.07300830  0.02562234 -0.071199227
##         Dim 13       Dim 14       Dim 15        Dim 16       Dim 17
## 1  0.038531658 -0.008850204 -0.046170081  0.0065854002  0.020247642
```

```
## 2 -0.001054519 -0.043106649  0.061248172  0.0007085906  0.006936504
## 3 -0.110466987  0.038409500 -0.046179720  0.0217899489 -0.009469007
## 4  0.053954095 -0.002449406 -0.020981053 -0.0172216996 -0.040118774
## 5  0.077895478  0.046458175 -0.003869006 -0.0213295345  0.024418103
##        Dim 18        Dim 19      Dim 20      Dim 21       Dim 22
## 1 -0.05042055  6.692696e-05 -0.04260641 -0.02342813  0.015872454
## 2 -0.01707266  1.939957e-02 -0.01735737  0.01866709  0.026996294
## 3  0.01557968  8.443794e-04  0.02524231 -0.02389661 -0.026501091
## 4  0.02669524 -1.981124e-02 -0.01211020 -0.01277295 -0.035588418
## 5  0.02775952 -2.262182e-02  0.04835186  0.02794758  0.005042496
##        Dim 23       Dim 24       Dim 25       Dim 26       Dim 27
## 1 -0.03573093 -0.022993528 -0.005590173  0.004215240 -0.020057958
## 2 -0.03659032 -0.053301603  0.041428106  0.014322670 -0.053140395
## 3  0.02008987  0.076564741  0.007407082 -0.008155719  0.068801893
## 4  0.06284186  0.012720240 -0.017448018 -0.016032279  0.008566123
## 5  0.01318483  0.007603217 -0.078059887 -0.005616140  0.020742970
##          Dim 28      Dim 29       Dim 30      Dim 31        Dim 32
## 1 -0.0706677255 -0.01632696  0.001772637 -0.02852486  6.166826e-02
## 2  0.0042930331  0.01225277  0.015512766  0.04997626  2.229776e-02
## 3  0.0172578190  0.04956408  0.005778219 -0.05506712  6.562681e-05
## 4  0.0227482265 -0.04445816  0.026688042  0.01930024 -2.804362e-02
## 5 -0.0001795126 -0.04003250 -0.071902962 -0.02323303 -7.035230e-02
##        Dim 33      Dim 34      Dim 35      Dim 36      Dim 37       Dim 38
## 1 -0.03511275  0.01234266  0.03940228 -0.01476452  0.01782357 -0.051920704
## 2  0.01652172 -0.00330396  0.03590698 -0.03867718 -0.02708309  0.002428777
## 3 -0.02166970 -0.04663777  0.05111587 -0.01940633 -0.01650596  0.014619143
## 4 -0.03275655 -0.03508175 -0.05081201  0.04359383  0.05141450 -0.002054473
## 5  0.06006134  0.10329495 -0.13344411  0.07935152  0.01447790  0.018320056
##         Dim 39      Dim 40
## 1  0.057887164  0.03281676
## 2 -0.013502624  0.06057016
## 3 -0.003288366 -0.04693931
## 4  0.024355042 -0.14004791
## 5 -0.039566556  0.05839010
```

```
# between clusters sum of squares
sq = sum(rowSums(k1$centers^2) * k1$size)
sq
```

```
## [1] 2076.403
```

```
# within clusters sum of squares
Wss = sum(k1$withinss)
Wss
```

```
## [1] 15298.35
```

```
# total sum of squares
Tss = sum(rowSums(coord^2))
Tss
```

```
## [1] 17374.76
```

```
sq + Wss
```

```
## [1] 17374.76
```

```r
# let's calculate the decomposition of inertia
Ib1 = 100 * sq / (sq + Wss)
Ib1
```

```
## [1] 11.95069
```

```r
# let's repeat kmeans, again with k=5
k2 = kmeans(coord, 5)
# between clusters sum of squares
sq = sum(rowSums(k2$centers^2) * k2$size)
Wss = sum(k2$withinss)
# total sum of squares
Tss = sum(rowSums(coord^2))
sq + Wss    # Tss = sq + Wss
```

```
## [1] 17374.76
```

```r
# let's calculate the decomposition of inertia
Ib2 = 100 * sq / (sq + Wss)
Ib2
```

```
## [1] 11.12723
```

```r
# why are we obtaining different results?  (Ib1 != Ib2)
# you can keep playing with different values for k

# let's repeat kmeans, again with k=8
k3 = kmeans(coord, 8)
# between clusters sum of squares
sq = sum(rowSums(k3$centers^2) * k3$size)
Wss = sum(k3$withinss)
# total sum of squares
Tss = sum(rowSums(coord^2))
sq + Wss    # Tss = sq + Wss
```

```
## [1] 17374.76
```

```r
# let's calculate the decomposition of inertia
Ib3 = 100 * sq / (sq + Wss)
Ib3
```

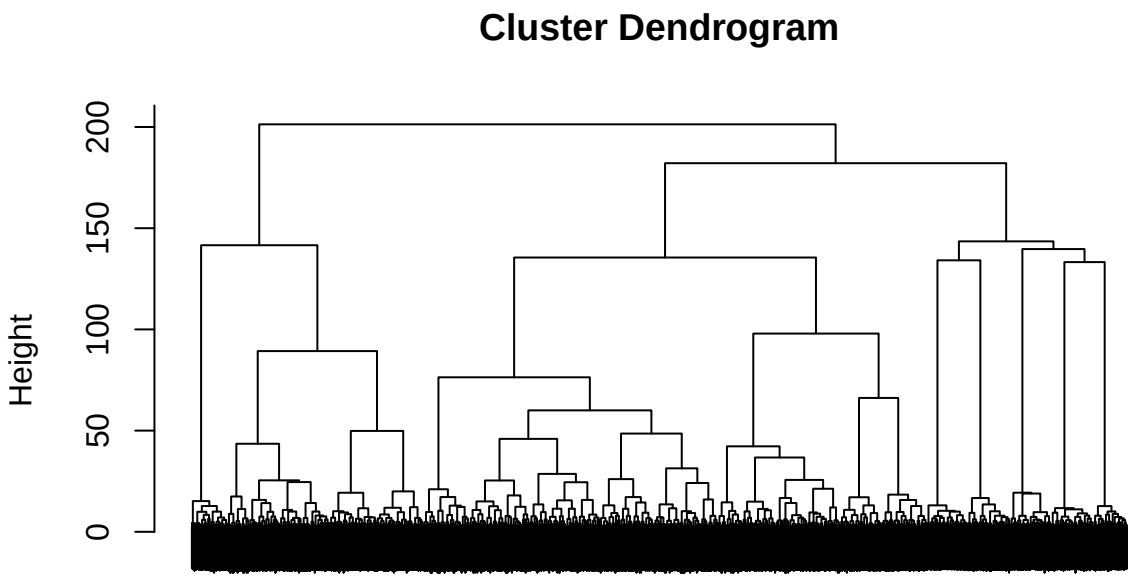
```
## [1] 17.80674
```

Hierarchical clustering on factorical coordinates from MCA results

```r
# Now, let's apply a hierarchical clustering
# first we calculate a distance matrix between individuals
idist = dist(coord)

# then we apply hclust with method="ward"
# notice the computation cost! (it takes a while to finish)
h1 = hclust(idist, method="ward")
```

```
## The "ward" method has been renamed to "ward.D"; note new "ward.D2"
```

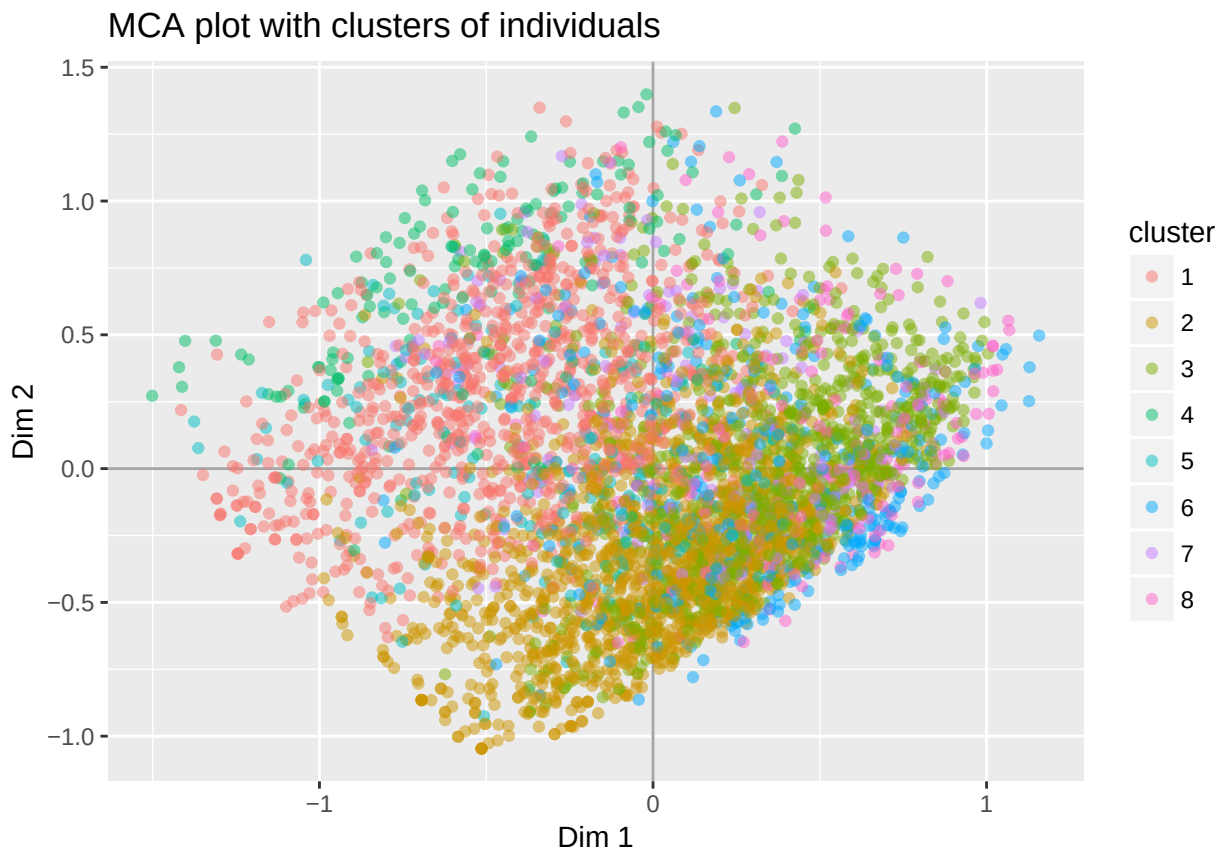
```r
# plot dendrogram
plot(h1, labels=FALSE)
```




```
# after checking the dendrogram, how many groups would you choose?
# where would you cut the dendrogram?
# let's try 8 clusters
nc = 8
# let's cut the tree and see the size of clusters
c1 = cutree(h1, nc)
table(c1)
```

```
## c1
##    1    2    3    4    5    6    7    8
##  930 1402  994  170  206  361  191  192
```

```
# prepare data frame for ggplot
df1 = data.frame(Status=data$Status, mca$ind$coord[,1:2], cluster=as.factor(c1))

# visualize clusters using the first two factorial coordinates
ggplot(data=df1, aes(x=Dim.1, y=Dim.2)) +
  geom_hline(yintercept=0, colour="gray65") +
  geom_vline(xintercept=0, colour="gray65") +
  geom_point(aes(colour=cluster), alpha=0.5) +
  labs(x="Dim 1", y="Dim 2") +
  ggtitle("MCA plot with clusters of individuals")
```




```
# centers of gravity of the clusters
cog = aggregate(as.data.frame(coord), list(c1), mean)[,-1]

# what's the quality of the hierarchical partition?
sq = sum(rowSums(cog^2) * table(c1))
Ib4 = 100 * sq / Tss
Ib4
```

```
## [1] 16.37535
```

Combining k-means and hierarchical clustering with MCA results

```
# let's consolidate the partition
# we'll apply  k-means using the cog's from the hierarchical clustering
k5 = kmeans(coord, center=cog)
k5$size
```

```
## [1] 1037 1705  569  178  224  350  190  193
```

```
sq = sum(rowSums(k5$centers^2) * k5$size)
Wss = sum(k5$withinss)
Ib5 = 100 * sq / (sq + Wss)
Ib5
```

```
## [1] 17.68192
```

```
# clustering of large data sets
# first 2 kmeans with k=14
n1 = 14
```

```
km1 = kmeans(coord, n1)
km2 = kmeans(coord, n1)

# what's the overlapping between clusters?
table(km2$cluster, km1$cluster)
```

```
##
##        1   2   3   4   5   6   7   8   9  10  11  12  13  14
##   1    0   0   0   2   1   4   0   5 141 192   1   0   0   0
##   2  152   0   0   0   0   0   0   0   0   0   0   0   0   0
##   3    0   0  46  45  22  14   0  29  17  48  36   0   4  31
##   4    0   0 139   0   0  38   0  90  11  20  63   0   0  59
##   5    0   0   8 193   0  63   0  46  25  28  17   0   0   0
##   6    0   1  21  58  14   5   0  21  12  10   6   0   6  39
##   7    0 178   0   0   0   0   0   0   0   0   0   0   0   0
##   8    0   0  92   3   0  46   0 115  13  43  52   0   0  21
##   9    0   0   0  42   1  32   0   0   0   0   1   0   0 300
##   10   0   0   0   1   0  18   0   0   0   0   5   0 329   0
##   11   0   0   0   0   0   0 190   0   0   0   0   0   0   0
##   12   0   0   3   0 341  13   0   9  12  63  17   0   4   7
##   13   0   0   0   0   0   0   0   0   0   0   0 221   0   0
##   14   0   0 128   0   0  46   0 178  27  34  46   0   1  31
```

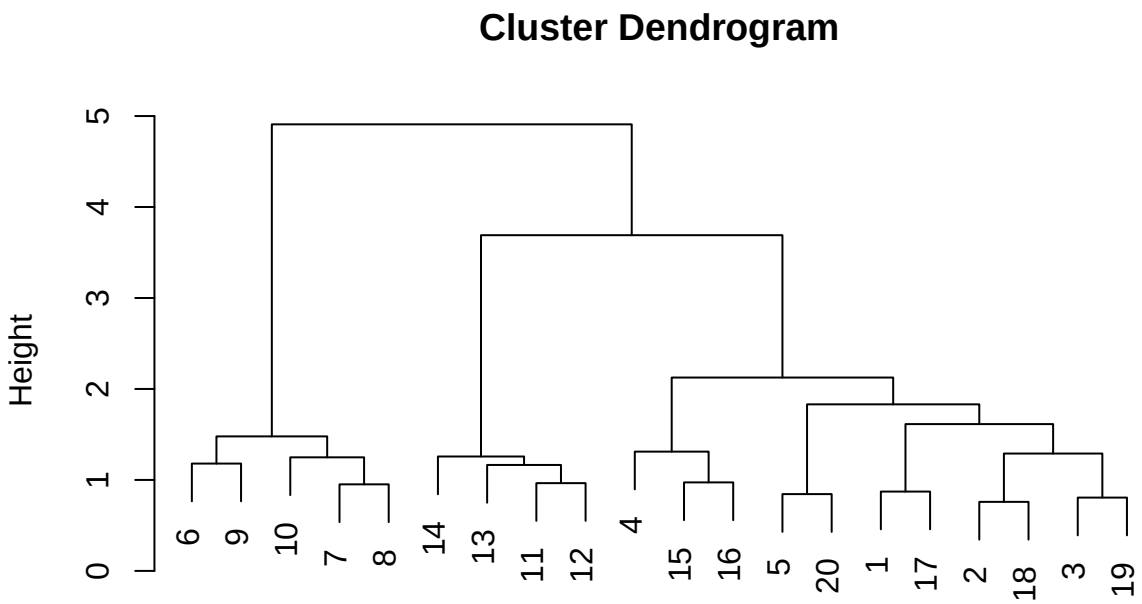
```
clas = (k2$cluster - 1)*n1 + k1$cluster
freq = table(clas)
freq[1:10]
```

```
## clas
##   1   2   3   4   5  15  16  17  18  19
##  41 564 462  22 162  59  53  22  42  21
```

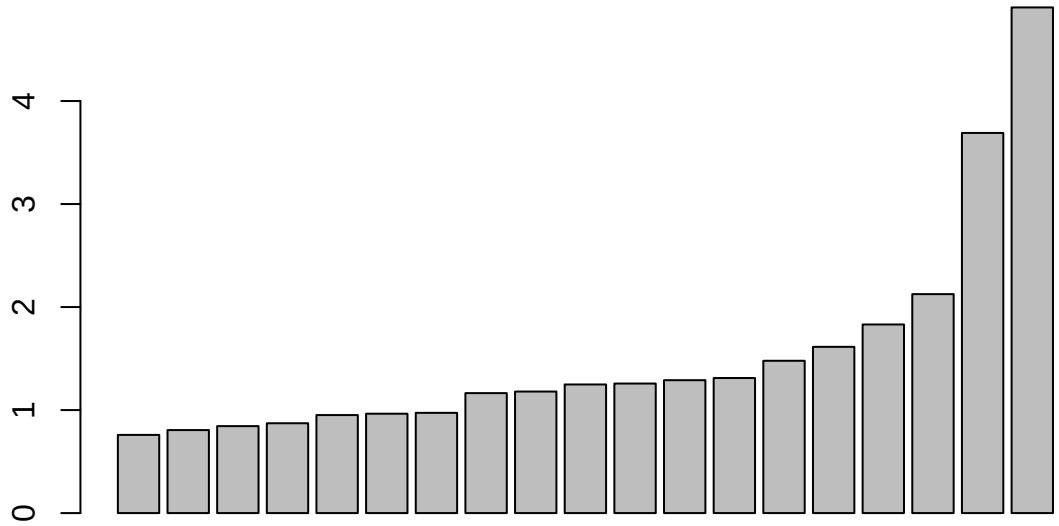
```
# what do we have in freq?
cogclas <- aggregate(as.data.frame(coord), list(clas), mean)[,2:(ncol(coord)+1)]

# perform a hierarchical clustering using cogclas
# compare the computational cost (this is way much faster!)
d2 = dist(cogclas)
h2 = hclust(d2, method="ward", members=freq)
```

```
## The "ward" method has been renamed to "ward.D"; note new "ward.D2"
```

```
# dendrogram
plot(h2)
```




```
# barplot
barplot(h2$height)
```




```
# cut tree in nc=8 groups
c2 <- cutree(h2, nc)
```

Probabilistic clustering on MCA results

```
# load package mclust
library(mclust)
```

```
## Warning: package 'mclust' was built under R version 3.4.3
```

```
## Package 'mclust' version 5.4
## Type 'citation("mclust")' for citing this R package in publications.
```

```
# check the computational cost
emc <- Mclust(coord, G=7:9)
print(emc)
```

```
## 'Mclust' model object:
##  best model: ellipsoidal, equal shape (VEV) with 8 components
```

```
attributes(emc)
```

```
## $names
##  [1] "call"           "data"           "modelName"      "n"
##  [5] "d"              "G"              "BIC"            "bic"
##  [9] "loglik"         "df"             "hypvol"         "parameters"
## [13] "z"              "classification" "uncertainty"
##
## $class
```

```
## [1] "Mclust"
```

```
# In this case, we have a probability for each individual
emc$z[1:10,]
```

```
##             [,1]          [,2]         [,3] [,4] [,5] [,6] [,7] [,8]
## 1  0.000000e+00  1.000000e+00 0.000000e+00    0    0    0    0    0
## 2  0.000000e+00  3.998893e-15 1.000000e+00    0    0    0    0    0
## 3  1.000000e+00  3.264870e-31 4.522799e-36    0    0    0    0    0
## 4  0.000000e+00  1.000000e+00 6.092650e-19    0    0    0    0    0
## 5  0.000000e+00  1.000000e+00 0.000000e+00    0    0    0    0    0
## 6  0.000000e+00  1.000000e+00 0.000000e+00    0    0    0    0    0
## 7  1.000000e+00  1.231278e-38 2.655344e-35    0    0    0    0    0
## 8  0.000000e+00 2.366643e-142 0.000000e+00    0    1    0    0    0
## 9  1.933187e-13  3.748554e-34 1.000000e+00    0    0    0    0    0
## 10 0.000000e+00  1.000000e+00 0.000000e+00    0    0    0    0    0
```

```
# let's see the membership for every individual
emc$classification[1:10]
```

```
##  1  2  3  4  5  6  7  8  9 10
##  2  3  1  2  2  2  1  5  3  2
```

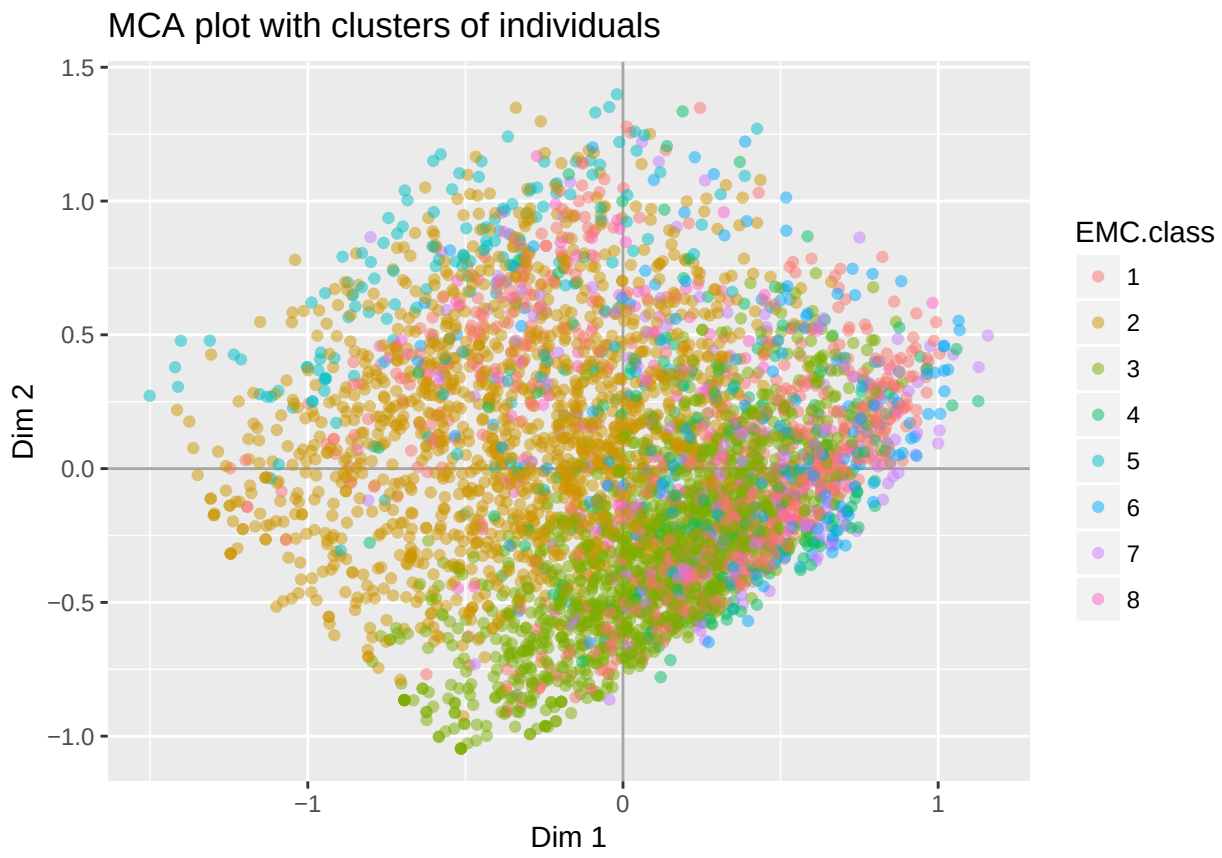
```
table(emc$classification)
```

```
##
##    1    2    3    4    5    6    7    8
##  732 1417 1370  221  157  197  159  193
```

```
# what's the quality of the partition?
cog <- aggregate(as.data.frame(coord), list(emc$classification), mean)[,2:(ncol(coord)+1)]
sq <- sum(rowSums(cog^2)*as.numeric(table(c1)))
Ib7 <- 100*sq/Tss
Ib7
```

```
## [1] 19.75878
```

```
# adata emc classification to data frame
df1$EMC.class = as.factor(emc$classification)

# visualize clusters using the first two factorial coordinates
ggplot(data=df1, aes(x=Dim.1, y=Dim.2)) +
  geom_hline(yintercept=0, colour="gray65") +
  geom_vline(xintercept=0, colour="gray65") +
  geom_point(aes(colour=EMC.class), alpha=0.5) +
  labs(x="Dim 1", y="Dim 2") +
  ggtitle("MCA plot with clusters of individuals")
```



Apply Decision Trees Analysis

```
# load package FactoMineR and ggplot2
require(ggplot2)
require(rpart)
```

```
## Loading required package: rpart
```

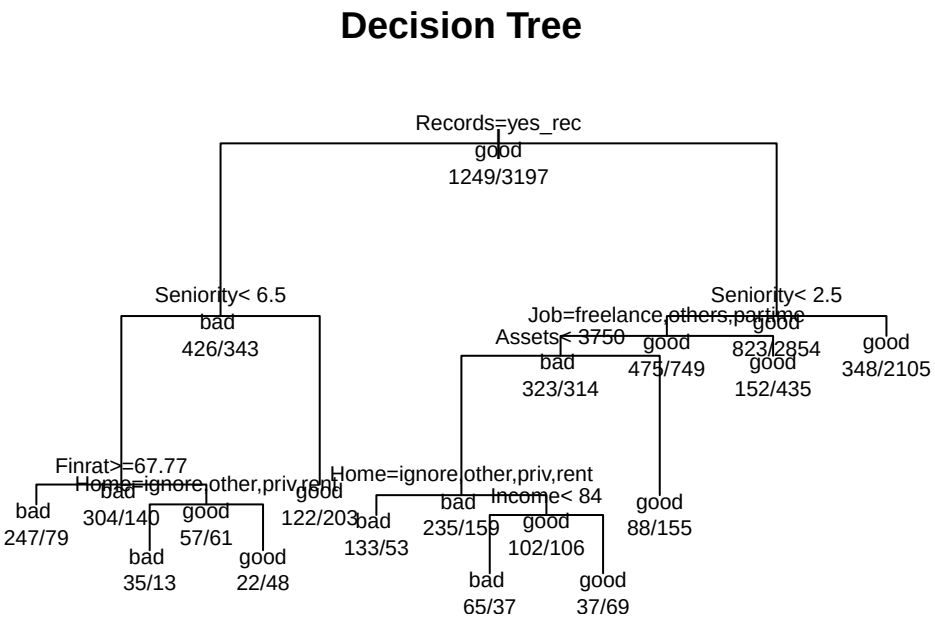
```
# Let's obtain a decision tree with the function 'rpart'
# using all the variables (both continuous and categorical)
ct = rpart(Status ~ ., data=data)
# let's see how the output looks like
ct
```

```
## n= 4446
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 4446 1249 good (0.2809267 0.7190733)
##    2) Records=yes_rec 769  343 bad (0.5539662 0.4460338)
##      4) Seniority< 6.5 444  140 bad (0.6846847 0.3153153)
##        8) Finrat>=67.77427 326   79 bad (0.7576687 0.2423313) *
##        9) Finrat< 67.77427 118   57 good (0.4830508 0.5169492)
##         18) Home=ignore,other,priv,rent 48   13 bad (0.7291667 0.2708333) *
##         19) Home=owner,parents 70   22 good (0.3142857 0.6857143) *
##      5) Seniority>=6.5 325  122 good (0.3753846 0.6246154) *
##    3) Records=no_rec 3677  823 good (0.2238238 0.7761762)
```

```
##      6) Seniority< 2.5 1224  475 good (0.3880719 0.6119281)
##       12) Job=freelance,others,partime 637  314 bad (0.5070644 0.4929356)
##         24) Assets< 3750 394  159 bad (0.5964467 0.4035533)
##           48) Home=ignore,other,priv,rent 186   53 bad (0.7150538 0.2849462) *
##           49) Home=owner,parents 208  102 good (0.4903846 0.5096154)
##             98) Income< 84 102   37 bad (0.6372549 0.3627451) *
##             99) Income>=84 106   37 good (0.3490566 0.6509434) *
##         25) Assets>=3750 243   88 good (0.3621399 0.6378601) *
##       13) Job=fixed 587  152 good (0.2589438 0.7410562) *
##      7) Seniority>=2.5 2453  348 good (0.1418671 0.8581329) *
```

how to read the output? "node), split, n, loss, yval, (yprob)" node): indicates the node number split: indicates the split criterion n: indicates the number of individuals in the groupe loss: indicates the the number of individuals misclassified yval: indicates the predicted value (yprob): indicates the probability of belonging to each class

```
# it's much easier to read a tree with a graphic
plot(ct, margin=0.05, compress=TRUE, main="Decision Tree")
text(ct, use.n=TRUE, pretty=1, all=TRUE, cex=0.7)
```



one of the goals is to obtain a tree in which the nodes are as much homogenous as possible, but also a tree with good prediction ability In order to improve our decision tree, we need to have 1) a train (aka learning) dataset 2) a test dataset let's keep 2/3 of the data for learning, and 1/3 for testing

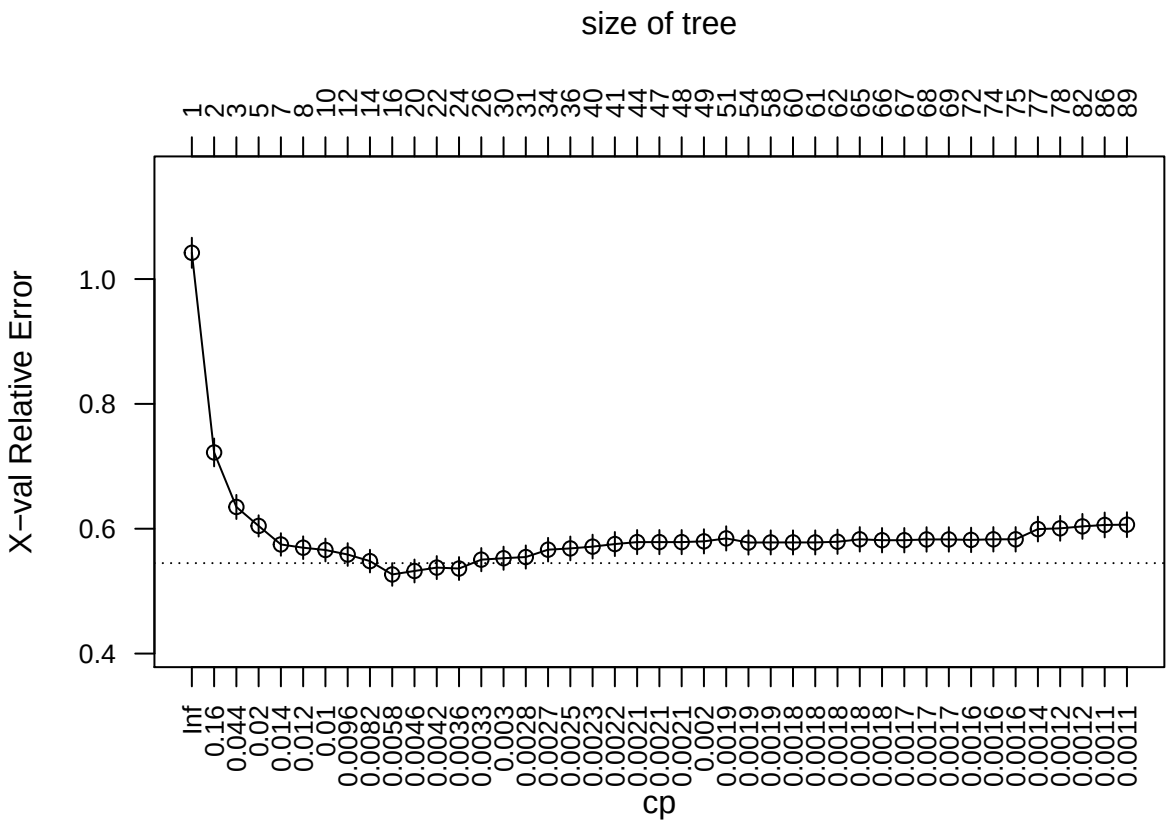
```
n = nrow(data)
learn = sample(1:n, size=round(0.67 * n))
nlearn = length(learn)
ntest = n - nlearn
```

```r
# selection of model by crossvalidation
# first we need a maximal tree with low value of cp
# and quiprobability of classes
ct1 = rpart(Status ~ ., data = data[learn,], method="class",
            parms = list(prior=c(0.50, 0.50), split='gini'),
            control = rpart.control(cp=0.001, xval=10, maxdepth=15))

# check results of the complexity parameter table
ct1$cptable
```

```
##             CP nsplit rel error    xerror       xstd
## 1  0.277819456      0 1.0000000 1.0420286 0.02407896
## 2  0.087297774      1 0.7221805 0.7221805 0.02228811
## 3  0.022583574      2 0.6348828 0.6348828 0.01926025
## 4  0.017029437      4 0.5897156 0.6044997 0.01701986
## 5  0.012035729      6 0.5556567 0.5748011 0.01804697
## 6  0.011135586      7 0.5436210 0.5696093 0.01820245
## 7  0.009732978      9 0.5213498 0.5659240 0.01824133
## 8  0.009493868     11 0.5018839 0.5586587 0.01819969
## 9  0.007138446     13 0.4828962 0.5481240 0.01814815
## 10 0.004670715     15 0.4686193 0.5267412 0.01818103
## 11 0.004489850     19 0.4488969 0.5322920 0.01836398
## 12 0.003968714     21 0.4399172 0.5377403 0.01845057
## 13 0.003346417     23 0.4319798 0.5362365 0.01835202
## 14 0.003163229     25 0.4252870 0.5504538 0.01857913
## 15 0.002904984     29 0.4112319 0.5527378 0.01853839
## 16 0.002749293     30 0.4083269 0.5546574 0.01859000
## 17 0.002567500     33 0.4000790 0.5664339 0.01891885
## 18 0.002385241     35 0.3949440 0.5684561 0.01906035
## 19 0.002284080     39 0.3821308 0.5713098 0.01911207
## 20 0.002151706     40 0.3798467 0.5754594 0.01927897
## 21 0.002101822     43 0.3728947 0.5785722 0.01935269
## 22 0.002076183     46 0.3653162 0.5785722 0.01935269
## 23 0.002024905     47 0.3632400 0.5785722 0.01935269
## 24 0.001945202     48 0.3612151 0.5797656 0.01937748
## 25 0.001868286     50 0.3573247 0.5844875 0.01943142
## 26 0.001868286     53 0.3517199 0.5778460 0.01932880
## 27 0.001841254     57 0.3433126 0.5780539 0.01930680
## 28 0.001817008     59 0.3396301 0.5780539 0.01930680
## 29 0.001817008     60 0.3378131 0.5780539 0.01930680
## 30 0.001791370     61 0.3359960 0.5790906 0.01939847
## 31 0.001762944     64 0.3299241 0.5830323 0.01958410
## 32 0.001762944     65 0.3281612 0.5815798 0.01953697
## 33 0.001711666     66 0.3263982 0.5815798 0.01953697
## 34 0.001711666     67 0.3246866 0.5828244 0.01960581
## 35 0.001658995     68 0.3229749 0.5828244 0.01960581
## 36 0.001634750     71 0.3169570 0.5820981 0.01958229
## 37 0.001609111     73 0.3136875 0.5830323 0.01958410
## 38 0.001505163     74 0.3120784 0.5830323 0.01958410
## 39 0.001244595     76 0.3090681 0.5994283 0.01985498
## 40 0.001219653     77 0.3078235 0.6006216 0.01987872
## 41 0.001158203     81 0.3029449 0.6037345 0.01994890
## 42 0.001141111     85 0.2976533 0.6060185 0.01990891
```

```
## 43 0.001000000      88 0.2942299 0.6064856 0.01990962
```

```
# we can use the function 'plotcp' to see the results
# the 'best' tree is the one with the lowest xerror
plotcp(ct1, las=2, cex.axis=0.8)
```




```
# what is the minimum XERROR?
min(ct1$cptable[,4])
```

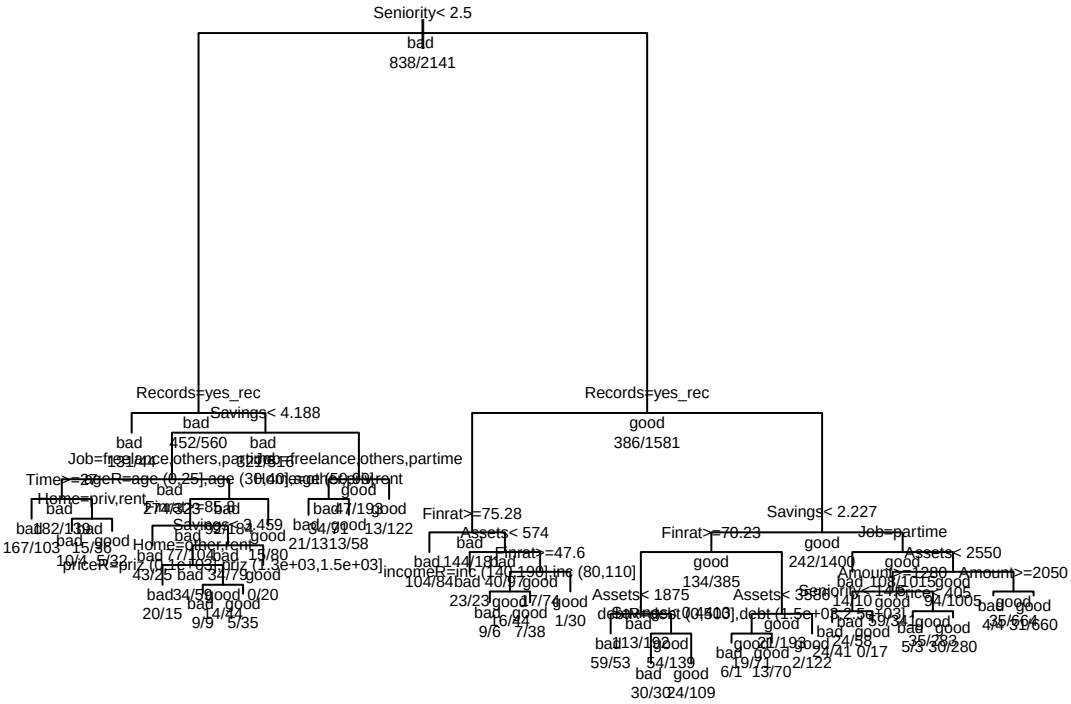
```
## [1] 0.5267412
```

```
min.xe = which(ct1$cptable[,4] == min(ct1$cptable[,4]))
# the optimal tree corresponds to a cp=0.003
ct1$cptable[min.xe,]
```

```
##           CP       nsplit    rel error       xerror         xstd
##  0.004670715 15.000000000  0.468619263  0.526741235  0.018181034
```

```
# Now that we know that the 'best' tree has cp=0.03
# we can plugin that information in the parameters
ct2 = rpart(Status ~ .,
            data = data[learn,],
            parms = list(prior=c(0.50, 0.50), split='gini'),
            control = rpart.control(cp=0.00285, xval=0, maxdepth=15))
# plot
par(mar = c(1,1,2,0.5))
plot(ct2, margin=0.05, compress=TRUE, main="Decision Tree")
text(ct2, use.n=TRUE, pretty=1, all=TRUE, cex=0.5)
```

Decision Tree




```
summary(ct2)
```

```
## Call:
## rpart(formula = Status ~ ., data = data[learn, ], parms = list(prior = c(0.5,
##     0.5), split = "gini"), control = rpart.control(cp = 0.00285,
##     xval = 0, maxdepth = 15))
##   n= 2979
##
##             CP nsplit rel error
## 1  0.277819456      0 1.0000000
## 2  0.087297774      1 0.7221805
## 3  0.022583574      2 0.6348828
## 4  0.017029437      4 0.5897156
## 5  0.012035729      6 0.5556567
## 6  0.011135586      7 0.5436210
## 7  0.009732978      9 0.5213498
## 8  0.009493868     11 0.5018839
## 9  0.007138446     13 0.4828962
## 10 0.004670715     15 0.4686193
## 11 0.004489850     19 0.4488969
## 12 0.003968714     21 0.4399172
## 13 0.003346417     23 0.4319798
## 14 0.003163229     25 0.4252870
## 15 0.002904984     29 0.4112319
## 16 0.002850000     30 0.4083269
##
```

```
## Variable importance
##  Seniority    Records seniorityR    Savings        Job   savingsR
##         11          9          9          8          7          6
##     Finrat    finratR       Home     Assets     Income    assetsR
##          5          5          5          4          4          4
##     Amount    incomeR       ageR        Age       Time    amountR
##          3          3          3          2          2          2
##      Price    Marital      timeR      debtR     priceR
##          2          1          1          1          1
##
## Node number 1: 2979 observations,    complexity param=0.2778195
##   predicted class=bad   expected loss=0.5  P(node) =1
##     class counts:   838  2141
##    probabilities: 0.500 0.500
##   left son=2 (1012 obs) right son=3 (1967 obs)
##   Primary splits:
##       Seniority  < 2.5       to the left,  improve=119.70850, (0 missing)
##       seniorityR splits as  LLRRR, improve=110.00920, (0 missing)
##       Records    splits as  RL, improve=109.19100, (0 missing)
##       Assets     < 2350      to the left,  improve= 88.06506, (0 missing)
##       Finrat     < 70.56687  to the right, improve= 87.37720, (0 missing)
##   Surrogate splits:
##       seniorityR splits as  LLRRR, agree=0.926, adj=0.783, (0 split)
##       Job        splits as  RRLL, agree=0.740, adj=0.235, (0 split)
##       Age        < 24.5      to the left,  agree=0.703, adj=0.125, (0 split)
##       ageR       splits as  LRRRR, agree=0.702, adj=0.123, (0 split)
##       Marital    splits as  RRRLR, agree=0.685, adj=0.074, (0 split)
##
## Node number 2: 1012 observations,    complexity param=0.01702944
##   predicted class=bad   expected loss=0.3265665  P(node) =0.4004697
##     class counts:   452   560
##    probabilities: 0.673 0.327
##   left son=4 (175 obs) right son=5 (837 obs)
##   Primary splits:
##       Records  splits as  RL, improve=29.92989, (0 missing)
##       Savings  < 4.204615  to the left,  improve=27.13412, (0 missing)
##       Job      splits as  RLLL, improve=25.71396, (0 missing)
##       Finrat   < 70.38018  to the right, improve=24.73849, (0 missing)
##       savingsR splits as  LLLRR, improve=24.54408, (0 missing)
##   Surrogate splits:
##       Price    < 2628      to the right, agree=0.834, adj=0.040, (0 split)
##       Expenses < 144       to the right, agree=0.829, adj=0.011, (0 split)
##       Amount   < 2175      to the right, agree=0.829, adj=0.011, (0 split)
##
## Node number 3: 1967 observations,    complexity param=0.08729777
##   predicted class=good  expected loss=0.3841512  P(node) =0.5995303
##     class counts:   386  1581
##    probabilities: 0.384 0.616
##   left son=6 (325 obs) right son=7 (1642 obs)
##   Primary splits:
##       Records splits as  RL, improve=79.51797, (0 missing)
##       Assets  < 1775      to the left,  improve=52.50765, (0 missing)
##       Finrat  < 68.82601  to the right, improve=50.42371, (0 missing)
##       finratR splits as  RRLLL, improve=48.22299, (0 missing)
```

```
##       assetsR splits as  LLRRR, improve=43.70741, (0 missing)
##   Surrogate splits:
##       Amount < 3125      to the right, agree=0.835, adj=0.003, (0 split)
##
## Node number 4: 175 observations
##   predicted class=bad   expected loss=0.1161897  P(node) =0.08843786
##     class counts:   131    44
##    probabilities: 0.884 0.116
##
## Node number 5: 837 observations,    complexity param=0.01702944
##   predicted class=bad   expected loss=0.3861927  P(node) =0.3120319
##     class counts:   321   516
##    probabilities: 0.614 0.386
##   left son=10 (597 obs) right son=11 (240 obs)
##   Primary splits:
##       Savings  < 4.188     to the left,  improve=30.16554, (0 missing)
##       Job      splits as  RLLL, improve=30.14267, (0 missing)
##       savingsR splits as  LLLRR, improve=28.72795, (0 missing)
##       Finrat   < 71.64577  to the right, improve=22.12857, (0 missing)
##       finratR  splits as  RRLLL, improve=20.19856, (0 missing)
##   Surrogate splits:
##       savingsR splits as  LLLRR, agree=0.977, adj=0.921, (0 split)
##       Income   < 145.5     to the left,  agree=0.849, adj=0.475, (0 split)
##       incomeR  splits as  LLRRL, agree=0.843, adj=0.454, (0 split)
##       Assets   < 1e+05     to the left,  agree=0.716, adj=0.008, (0 split)
##       Expenses < 111.5     to the left,  agree=0.714, adj=0.004, (0 split)
##
## Node number 6: 325 observations,    complexity param=0.007138446
##   predicted class=bad   expected loss=0.3297477  P(node) =0.1281888
##     class counts:   144   181
##    probabilities: 0.670 0.330
##   left son=12 (188 obs) right son=13 (137 obs)
##   Primary splits:
##       Finrat  < 75.27964  to the right, improve=10.751960, (0 missing)
##       finratR splits as  RLLLL, improve= 8.639520, (0 missing)
##       Home    splits as  RLRLLL, improve= 8.159015, (0 missing)
##       Assets  < 574       to the left,  improve= 6.954906, (0 missing)
##       Income  < 109.5     to the left,  improve= 6.481317, (0 missing)
##   Surrogate splits:
##       finratR splits as  RRLLL, agree=0.923, adj=0.818, (0 split)
##       Amount  < 837.5     to the right, agree=0.751, adj=0.409, (0 split)
##       amountR splits as  RLLRL, agree=0.738, adj=0.380, (0 split)
##       timeR   splits as  RRLRL, agree=0.686, adj=0.255, (0 split)
##       Time    < 33        to the right, agree=0.677, adj=0.234, (0 split)
##
## Node number 7: 1642 observations,    complexity param=0.02258357
##   predicted class=good  expected loss=0.3063414  P(node) =0.4713414
##     class counts:   242  1400
##    probabilities: 0.306 0.694
##   left son=14 (519 obs) right son=15 (1123 obs)
##   Primary splits:
##       Savings  < 2.227444  to the left,  improve=42.73541, (0 missing)
##       Finrat   < 70.23458  to the right, improve=38.59446, (0 missing)
##       finratR  splits as  RRLLL, improve=37.49342, (0 missing)
```

```
##       savingsR splits as  LLRRR, improve=37.28172, (0 missing)
##       Assets   < 3586      to the left,  improve=37.05155, (0 missing)
##   Surrogate splits:
##       savingsR splits as  LLRRR, agree=0.964, adj=0.886, (0 split)
##       Income   < 97.5      to the left,  agree=0.831, adj=0.466, (0 split)
##       incomeR  splits as  LRRRL, agree=0.816, adj=0.418, (0 split)
##       Time     < 15        to the left,  agree=0.692, adj=0.027, (0 split)
##       timeR    splits as  LRRRR, agree=0.692, adj=0.027, (0 split)
##
## Node number 10: 597 observations,    complexity param=0.009732978
##   predicted class=bad   expected loss=0.3157255  P(node) =0.2389165
##     class counts:   274   323
##    probabilities: 0.684 0.316
##   left son=20 (321 obs) right son=21 (276 obs)
##   Primary splits:
##       Job     splits as  RLLL, improve=15.029720, (0 missing)
##       Finrat  < 71.64577  to the right, improve=14.406310, (0 missing)
##       finratR splits as  RRLLL, improve=11.887890, (0 missing)
##       timeR   splits as  RRLLL, improve= 9.870686, (0 missing)
##       Time    < 27        to the right, improve= 9.870686, (0 missing)
##   Surrogate splits:
##       Seniority  < 0.5       to the left,  agree=0.645, adj=0.232, (0 split)
##       seniorityR splits as  LR---, agree=0.643, adj=0.228, (0 split)
##       Income     < 85.5      to the left,  agree=0.603, adj=0.141, (0 split)
##       Savings    < 1.647958  to the left,  agree=0.600, adj=0.134, (0 split)
##       savingsR   splits as  LLRL-, agree=0.598, adj=0.130, (0 split)
##
## Node number 11: 240 observations,    complexity param=0.009493868
##   predicted class=good  expected loss=0.383544  P(node) =0.07311536
##     class counts:    47   193
##    probabilities: 0.384 0.616
##   left son=22 (105 obs) right son=23 (135 obs)
##   Primary splits:
##       Job     splits as  RLLL, improve=12.313570, (0 missing)
##       Home    splits as  -LRRLL, improve= 8.991849, (0 missing)
##       amountR splits as  LLLRR, improve= 6.803730, (0 missing)
##       Assets  < 2250      to the left,  improve= 5.963506, (0 missing)
##       assetsR splits as  LRRRR, improve= 5.827538, (0 missing)
##   Surrogate splits:
##       Seniority  < 0.5       to the left,  agree=0.617, adj=0.124, (0 split)
##       seniorityR splits as  LR---, agree=0.608, adj=0.105, (0 split)
##       Age        < 21.5      to the left,  agree=0.604, adj=0.095, (0 split)
##       Assets     < 4250      to the right, agree=0.604, adj=0.095, (0 split)
##       Time       < 33        to the left,  agree=0.600, adj=0.086, (0 split)
##
## Node number 12: 188 observations
##   predicted class=bad   expected loss=0.2401998  P(node) =0.08166951
##     class counts:   104    84
##    probabilities: 0.760 0.240
##
## Node number 13: 137 observations,    complexity param=0.007138446
##   predicted class=bad   expected loss=0.4869583  P(node) =0.04651931
##     class counts:    40    97
##    probabilities: 0.513 0.487
```

```
##   left son=26 (46 obs) right son=27 (91 obs)
##   Primary splits:
##       Assets    < 574       to the left,  improve=8.161677, (0 missing)
##       Home      splits as  -LRLRL, improve=7.875192, (0 missing)
##       assetsR   splits as  LRRRR, improve=7.577441, (0 missing)
##       Seniority < 11.5      to the left,  improve=5.852244, (0 missing)
##       Time      < 21        to the right, improve=4.819471, (0 missing)
##   Surrogate splits:
##       assetsR splits as  LRRRR, agree=0.985, adj=0.957, (0 split)
##       Home    splits as  -LRLRL, agree=0.920, adj=0.761, (0 split)
##       Marital splits as  -RLLR, agree=0.730, adj=0.196, (0 split)
##       Age     < 27.5      to the left,  agree=0.693, adj=0.087, (0 split)
##       Finrat  < 72.02305  to the right, agree=0.693, adj=0.087, (0 split)
##
## Node number 14: 519 observations,    complexity param=0.02258357
##   predicted class=good  expected loss=0.4706853  P(node) =0.1698635
##     class counts:   134   385
##    probabilities: 0.471 0.529
##   left son=28 (305 obs) right son=29 (214 obs)
##   Primary splits:
##       Finrat   < 70.23458  to the right, improve=33.28179, (0 missing)
##       finratR  splits as  RRLLL, improve=31.98542, (0 missing)
##       Assets   < 3586      to the left,  improve=20.80176, (0 missing)
##       assetsR  splits as  LLLRR, improve=18.70308, (0 missing)
##       Expenses < 81.5      to the right, improve=13.74221, (0 missing)
##   Surrogate splits:
##       finratR splits as  RRLLL, agree=0.994, adj=0.986, (0 split)
##       Time    < 27        to the right, agree=0.730, adj=0.346, (0 split)
##       timeR   splits as  RRLLL, agree=0.730, adj=0.346, (0 split)
##       Amount  < 815       to the right, agree=0.728, adj=0.341, (0 split)
##       amountR splits as  RLLRL, agree=0.709, adj=0.294, (0 split)
##
## Node number 15: 1123 observations,    complexity param=0.01203573
##   predicted class=good  expected loss=0.2137442  P(node) =0.3014779
##     class counts:   108  1015
##    probabilities: 0.214 0.786
##   left son=30 (24 obs) right son=31 (1099 obs)
##   Primary splits:
##       Job     splits as  RRRL, improve=21.28309, (0 missing)
##       Assets  < 2550      to the left,  improve=17.58550, (0 missing)
##       Finrat  < 82.37497  to the right, improve=15.02562, (0 missing)
##       assetsR splits as  LLRRR, improve=13.87911, (0 missing)
##       finratR splits as  RRRLL, improve=13.66296, (0 missing)
##
## Node number 20: 321 observations,    complexity param=0.00448985
##   predicted class=bad   expected loss=0.2301361  P(node) =0.1410534
##     class counts:   182   139
##    probabilities: 0.770 0.230
##   left son=40 (270 obs) right son=41 (51 obs)
##   Primary splits:
##       Time    < 27        to the right, improve=7.622062, (0 missing)
##       timeR   splits as  RRLLL, improve=7.622062, (0 missing)
##       Finrat  < 69.74764  to the right, improve=6.931061, (0 missing)
##       finratR splits as  RRLLL, improve=6.562940, (0 missing)
```

```
##       Assets  < 7250      to the left,  improve=6.380556, (0 missing)
##   Surrogate splits:
##       Amount  < 367.5     to the right, agree=0.875, adj=0.216, (0 split)
##       Finrat  < 35.32617  to the right, agree=0.875, adj=0.216, (0 split)
##       Price   < 387.5     to the right, agree=0.850, adj=0.059, (0 split)
##       finratR splits as  RLLLL, agree=0.850, adj=0.059, (0 split)
##
## Node number 21: 276 observations,    complexity param=0.009732978
##   predicted class=bad   expected loss=0.4390883  P(node) =0.09786318
##     class counts:    92   184
##    probabilities: 0.561 0.439
##   left son=42 (181 obs) right son=43 (95 obs)
##   Primary splits:
##       ageR    splits as  LRLRL, improve=12.88861, (0 missing)
##       Finrat  < 79.83308  to the right, improve=11.53539, (0 missing)
##       Home    splits as  LLRRRL, improve=11.40421, (0 missing)
##       amountR splits as  RLLLR, improve=10.83089, (0 missing)
##       finratR splits as  RRRLL, improve= 9.32726, (0 missing)
##   Surrogate splits:
##       Age      < 40.5      to the left,  agree=0.746, adj=0.263, (0 split)
##       Savings  < 3.905158  to the left,  agree=0.670, adj=0.042, (0 split)
##       Expenses < 89        to the left,  agree=0.667, adj=0.032, (0 split)
##       Marital  splits as  LLRLL, agree=0.663, adj=0.021, (0 split)
##       Income   < 177.5     to the left,  agree=0.663, adj=0.021, (0 split)
##
## Node number 22: 105 observations,    complexity param=0.009493868
##   predicted class=bad   expected loss=0.4497475  P(node) =0.03686743
##     class counts:    34    71
##    probabilities: 0.550 0.450
##   left son=44 (34 obs) right son=45 (71 obs)
##   Primary splits:
##       Home     splits as  -LRRLL, improve=10.413170, (0 missing)
##       Assets   < 2250      to the left,  improve= 9.691406, (0 missing)
##       assetsR  splits as  LRRRR, improve= 9.039795, (0 missing)
##       priceR   splits as  LLRLR, improve= 5.459168, (0 missing)
##       Expenses < 46        to the right, improve= 3.438065, (0 missing)
##   Surrogate splits:
##       Expenses  < 46        to the right, agree=0.743, adj=0.206, (0 split)
##       ageR      splits as  RRLRR, agree=0.743, adj=0.206, (0 split)
##       expensesR splits as  RRRLL, agree=0.743, adj=0.206, (0 split)
##       Assets    < 1750      to the left,  agree=0.724, adj=0.147, (0 split)
##       assetsR   splits as  LRRRR, agree=0.714, adj=0.118, (0 split)
##
## Node number 23: 135 observations
##   predicted class=good  expected loss=0.2139864  P(node) =0.03624792
##     class counts:    13   122
##    probabilities: 0.214 0.786
##
## Node number 26: 46 observations
##   predicted class=bad   expected loss=0.2813025  P(node) =0.01909447
##     class counts:    23    23
##    probabilities: 0.719 0.281
##
## Node number 27: 91 observations,    complexity param=0.003968714
```

```
##   predicted class=good  expected loss=0.3698544  P(node) =0.02742484
##     class counts:    17    74
##    probabilities: 0.370 0.630
##   left son=54 (60 obs) right son=55 (31 obs)
##   Primary splits:
##       Finrat    < 47.60035  to the right, improve=5.320696, (0 missing)
##       Time      < 33        to the right, improve=4.413863, (0 missing)
##       timeR     splits as  RRLLL, improve=4.126937, (0 missing)
##       Seniority < 11.5      to the left,  improve=2.586744, (0 missing)
##       expensesR splits as  LRRRL, improve=2.370883, (0 missing)
##   Surrogate splits:
##       finratR splits as  RLL--, agree=0.967, adj=0.903, (0 split)
##       Amount  < 560       to the right, agree=0.802, adj=0.419, (0 split)
##       amountR splits as  RLLLL, agree=0.769, adj=0.323, (0 split)
##       Time    < 33        to the right, agree=0.703, adj=0.129, (0 split)
##       Age     < 27.5      to the right, agree=0.692, adj=0.097, (0 split)
##
## Node number 28: 305 observations,    complexity param=0.01113559
##   predicted class=bad   expected loss=0.3994151  P(node) =0.1122613
##     class counts:   113   192
##    probabilities: 0.601 0.399
##   left son=56 (112 obs) right son=57 (193 obs)
##   Primary splits:
##       Assets    < 1875      to the left,  improve=9.544324, (0 missing)
##       Expenses  < 88.5      to the right, improve=8.504675, (0 missing)
##       Savings   < 0.8863636 to the left,  improve=7.668591, (0 missing)
##       expensesR splits as  RRRRL, improve=7.036196, (0 missing)
##       assetsR   splits as  LLLRR, improve=6.963943, (0 missing)
##   Surrogate splits:
##       assetsR splits as  LRRRR, agree=0.967, adj=0.911, (0 split)
##       Home    splits as  -LRLRL, agree=0.839, adj=0.562, (0 split)
##       Marital splits as  LRRLR, agree=0.682, adj=0.134, (0 split)
##       Age     < 24.5      to the left,  agree=0.662, adj=0.080, (0 split)
##       Amount  < 462.5     to the left,  agree=0.652, adj=0.054, (0 split)
##
## Node number 29: 214 observations,    complexity param=0.003346417
##   predicted class=good  expected loss=0.2175234  P(node) =0.05760223
##     class counts:    21   193
##    probabilities: 0.218 0.782
##   left son=58 (90 obs) right son=59 (124 obs)
##   Primary splits:
##       Assets  < 3586      to the left,  improve=11.474350, (0 missing)
##       assetsR splits as  LLRRR, improve= 9.019293, (0 missing)
##       debtR   splits as  RLRRR, improve= 8.873948, (0 missing)
##       Age     < 36.5      to the left,  improve= 7.795581, (0 missing)
##       Home    splits as  -LRRLR, improve= 7.430395, (0 missing)
##   Surrogate splits:
##       assetsR splits as  LLRRR, agree=0.963, adj=0.911, (0 split)
##       Home    splits as  -LRLLL, agree=0.794, adj=0.511, (0 split)
##       Age     < 28.5      to the left,  agree=0.692, adj=0.267, (0 split)
##       Marital splits as  RRRLR, agree=0.664, adj=0.200, (0 split)
##       ageR    splits as  LLRRR, agree=0.664, adj=0.200, (0 split)
##
## Node number 30: 24 observations
```

```
##   predicted class=bad   expected loss=0.2184909  P(node) =0.01068858
##     class counts:    14    10
##    probabilities: 0.782 0.218
##
## Node number 31: 1099 observations,    complexity param=0.003163229
##   predicted class=good  expected loss=0.1928747  P(node) =0.2907893
##     class counts:    94  1005
##    probabilities: 0.193 0.807
##   left son=62 (400 obs) right son=63 (699 obs)
##   Primary splits:
##       Assets  < 2550      to the left,  improve=14.60992, (0 missing)
##       Finrat  < 79.47655  to the right, improve=12.80257, (0 missing)
##       assetsR splits as  LRRRR, improve=12.14084, (0 missing)
##       Home    splits as  LLRLRL, improve=12.13259, (0 missing)
##       finratR splits as  RRRLL, improve=11.33896, (0 missing)
##   Surrogate splits:
##       assetsR splits as  LLRRR, agree=0.938, adj=0.830, (0 split)
##       Home    splits as  RLRLRL, agree=0.878, adj=0.665, (0 split)
##       Marital splits as  RRLLR, agree=0.668, adj=0.087, (0 split)
##       Age     < 26.5      to the left,  agree=0.656, adj=0.055, (0 split)
##       ageR    splits as  LRRRR, agree=0.653, adj=0.047, (0 split)
##
## Node number 40: 270 observations
##   predicted class=bad   expected loss=0.1944618  P(node) =0.1236962
##     class counts:   167   103
##    probabilities: 0.806 0.194
##
## Node number 41: 51 observations,    complexity param=0.00448985
##   predicted class=bad   expected loss=0.4843697  P(node) =0.01735717
##     class counts:    15    36
##    probabilities: 0.516 0.484
##   left son=82 (14 obs) right son=83 (37 obs)
##   Primary splits:
##       Home    splits as  RRRRLL, improve=8.312737, (0 missing)
##       finratR splits as  RLLRR, improve=6.401945, (0 missing)
##       Finrat  < 50.95748  to the right, improve=4.000770, (0 missing)
##       Assets  < 2250      to the left,  improve=4.000770, (0 missing)
##       Savings < 2.416667  to the left,  improve=3.316496, (0 missing)
##   Surrogate splits:
##       Price     < 428       to the left,  agree=0.784, adj=0.214, (0 split)
##       finratR   splits as  RRLRR, agree=0.765, adj=0.143, (0 split)
##       Seniority < 1.5       to the right, agree=0.745, adj=0.071, (0 split)
##       Marital   splits as  -RLR-, agree=0.745, adj=0.071, (0 split)
##       Assets    < 19000     to the right, agree=0.745, adj=0.071, (0 split)
##
## Node number 42: 181 observations,    complexity param=0.004670715
##   predicted class=bad   expected loss=0.3458289  P(node) =0.07023044
##     class counts:    77   104
##    probabilities: 0.654 0.346
##   left son=84 (68 obs) right son=85 (113 obs)
##   Primary splits:
##       Finrat  < 85.79927  to the right, improve=8.758754, (0 missing)
##       finratR splits as  RRRLL, improve=7.637574, (0 missing)
##       Home    splits as  LLRRRL, improve=6.878251, (0 missing)
```

```
##       amountR splits as  RLLLR, improve=5.189681, (0 missing)
##       Assets  < 1650      to the left,  improve=4.853888, (0 missing)
##   Surrogate splits:
##       finratR splits as  RRRRL, agree=0.884, adj=0.691, (0 split)
##       Amount  < 1330      to the right, agree=0.696, adj=0.191, (0 split)
##       amountR splits as  RRLRR, agree=0.674, adj=0.132, (0 split)
##       Price   < 1007      to the left,  agree=0.652, adj=0.074, (0 split)
##       priceR  splits as  LRRRR, agree=0.652, adj=0.074, (0 split)
##
## Node number 43: 95 observations
##   predicted class=good  expected loss=0.3238868  P(node) =0.02763274
##     class counts:    15    80
##    probabilities: 0.324 0.676
##
## Node number 44: 34 observations
##   predicted class=bad   expected loss=0.1950407  P(node) =0.0155658
##     class counts:    21    13
##    probabilities: 0.805 0.195
##
## Node number 45: 71 observations
##   predicted class=good  expected loss=0.3641299  P(node) =0.02130164
##     class counts:    13    58
##    probabilities: 0.364 0.636
##
## Node number 54: 60 observations,    complexity param=0.003968714
##   predicted class=good  expected loss=0.4816106  P(node) =0.01982211
##     class counts:    16    44
##    probabilities: 0.482 0.518
##   left son=108 (15 obs) right son=109 (45 obs)
##   Primary splits:
##       incomeR   splits as  RRLRL, improve=5.943565, (0 missing)
##       timeR     splits as  LRLLL, improve=3.249133, (0 missing)
##       Income    < 110       to the left,  improve=2.753481, (0 missing)
##       Savings   < 8.404286  to the left,  improve=2.289904, (0 missing)
##       expensesR splits as  LRRRL, improve=2.270408, (0 missing)
##   Surrogate splits:
##       Finrat  < 48.48595  to the left,  agree=0.783, adj=0.133, (0 split)
##       Home    splits as  -RRLRR, agree=0.767, adj=0.067, (0 split)
##       Age     < 30.5      to the left,  agree=0.767, adj=0.067, (0 split)
##       ageR    splits as  -LRRR, agree=0.767, adj=0.067, (0 split)
##       finratR splits as  LRR--, agree=0.767, adj=0.067, (0 split)
##
## Node number 55: 31 observations
##   predicted class=good  expected loss=0.07847953  P(node) =0.007602731
##     class counts:     1    30
##    probabilities: 0.078 0.922
##
## Node number 56: 112 observations
##   predicted class=bad   expected loss=0.2601372  P(node) =0.04758026
##     class counts:    59    53
##    probabilities: 0.740 0.260
##
## Node number 57: 193 observations,    complexity param=0.01113559
##   predicted class=good  expected loss=0.4981301  P(node) =0.06468104
```

```
##     class counts:    54   139
##    probabilities: 0.498 0.502
##   left son=114 (60 obs) right son=115 (133 obs)
##   Primary splits:
##       Savings   < 0.4413333 to the left,  improve=11.739480, (0 missing)
##       Expenses  < 80        to the right, improve= 9.522977, (0 missing)
##       expensesR splits as  RRRRL, improve= 9.522977, (0 missing)
##       Marital   splits as  RLLRL, improve= 7.042992, (0 missing)
##       Age       < 26.5      to the right, improve= 6.601569, (0 missing)
##   Surrogate splits:
##       savingsR  splits as  LRR--, agree=0.845, adj=0.500, (0 split)
##       Income    < 68        to the left,  agree=0.777, adj=0.283, (0 split)
##       Expenses  < 80        to the right, agree=0.725, adj=0.117, (0 split)
##       expensesR splits as  RRRRL, agree=0.725, adj=0.117, (0 split)
##       Amount    < 462.5     to the left,  agree=0.705, adj=0.050, (0 split)
##
## Node number 58: 90 observations,    complexity param=0.003346417
##   predicted class=good  expected loss=0.4060713  P(node) =0.02791755
##     class counts:    19    71
##    probabilities: 0.406 0.594
##   left son=116 (7 obs) right son=117 (83 obs)
##   Primary splits:
##       debtR     splits as  RLL-R, improve=7.467203, (0 missing)
##       Finrat    < 45.42485  to the right, improve=5.938332, (0 missing)
##       Debt      < 0.5       to the right, improve=4.804710, (0 missing)
##       expensesR splits as  LLRRL, improve=4.059766, (0 missing)
##       Home      splits as  -LRRLR, improve=4.002488, (0 missing)
##   Surrogate splits:
##       Debt < 0.5       to the right, agree=0.967, adj=0.571, (0 split)
##       Home splits as  -RRRLR, agree=0.933, adj=0.143, (0 split)
##
## Node number 59: 124 observations
##   predicted class=good  expected loss=0.04019978  P(node) =0.02968468
##     class counts:     2   122
##    probabilities: 0.040 0.960
##
## Node number 62: 400 observations,    complexity param=0.003163229
##   predicted class=good  expected loss=0.3065422  P(node) =0.1148385
##     class counts:    59   341
##    probabilities: 0.307 0.693
##   left son=124 (82 obs) right son=125 (318 obs)
##   Primary splits:
##       Amount  < 1280      to the right, improve=9.425519, (0 missing)
##       Job     splits as  RLR-, improve=9.111476, (0 missing)
##       Finrat  < 73.75541  to the right, improve=8.027891, (0 missing)
##       finratR splits as  RRRLL, improve=6.924853, (0 missing)
##       Price   < 405       to the left,  improve=5.743579, (0 missing)
##   Surrogate splits:
##       amountR splits as  RRLRR, agree=0.908, adj=0.549, (0 split)
##       Price   < 1997.5    to the right, agree=0.828, adj=0.159, (0 split)
##       priceR  splits as  RRRLR, agree=0.810, adj=0.073, (0 split)
##       Debt    < 380       to the right, agree=0.802, adj=0.037, (0 split)
##       debtR   splits as  RRR-L, agree=0.802, adj=0.037, (0 split)
##
```

```
## Node number 63: 699 observations,    complexity param=0.002904984
##   predicted class=good  expected loss=0.1186869  P(node) =0.1759508
##     class counts:    35   664
##    probabilities: 0.119 0.881
##   left son=126 (8 obs) right son=127 (691 obs)
##   Primary splits:
##       Amount  < 2050      to the right, improve=7.259942, (0 missing)
##       Price   < 2744      to the right, improve=5.080536, (0 missing)
##       Finrat  < 76.08052  to the right, improve=4.823086, (0 missing)
##       Assets  < 57500     to the right, improve=4.242301, (0 missing)
##       finratR splits as  RRRLL, improve=3.880929, (0 missing)
##   Surrogate splits:
##       Price < 4462      to the right, agree=0.991, adj=0.25, (0 split)
##
## Node number 82: 14 observations
##   predicted class=bad   expected loss=0.1353687  P(node) =0.00690073
##     class counts:    10     4
##    probabilities: 0.865 0.135
##
## Node number 83: 37 observations
##   predicted class=good  expected loss=0.2853069  P(node) =0.01045644
##     class counts:     5    32
##    probabilities: 0.285 0.715
##
## Node number 84: 68 observations
##   predicted class=bad   expected loss=0.1853769  P(node) =0.03149472
##     class counts:    43    25
##    probabilities: 0.815 0.185
##
## Node number 85: 113 observations,    complexity param=0.004670715
##   predicted class=bad   expected loss=0.4762871  P(node) =0.03873572
##     class counts:    34    79
##    probabilities: 0.524 0.476
##   left son=170 (93 obs) right son=171 (20 obs)
##   Primary splits:
##       Savings < 3.458769  to the left,  improve=8.679078, (0 missing)
##       Assets  < 1650      to the left,  improve=6.605593, (0 missing)
##       Income  < 108.5     to the left,  improve=6.589169, (0 missing)
##       incomeR splits as  LRR-L, improve=5.692712, (0 missing)
##       Home    splits as  -LRRRL, improve=5.589621, (0 missing)
##   Surrogate splits:
##       Income   < 141.5     to the left,  agree=0.850, adj=0.15, (0 split)
##       Finrat   < 85.35308  to the left,  agree=0.841, adj=0.10, (0 split)
##       incomeR  splits as  LLR-L, agree=0.841, adj=0.10, (0 split)
##       savingsR splits as  LLLR-, agree=0.841, adj=0.10, (0 split)
##       Marital  splits as  LLLLR, agree=0.832, adj=0.05, (0 split)
##
## Node number 108: 15 observations
##   predicted class=bad   expected loss=0.2069391  P(node) =0.006771143
##     class counts:     9     6
##    probabilities: 0.793 0.207
##
## Node number 109: 45 observations
##   predicted class=good  expected loss=0.3200231  P(node) =0.01305097
```

```
##     class counts:     7    38
##    probabilities: 0.320 0.680
##
## Node number 114: 60 observations
##   predicted class=bad   expected loss=0.2813025  P(node) =0.02490583
##     class counts:    30    30
##    probabilities: 0.719 0.281
##
## Node number 115: 133 observations
##   predicted class=good  expected loss=0.3600185  P(node) =0.0397752
##     class counts:    24   109
##    probabilities: 0.360 0.640
##
## Node number 116: 7 observations
##   predicted class=bad   expected loss=0.0612394  P(node) =0.003813488
##     class counts:     6     1
##    probabilities: 0.939 0.061
##
## Node number 117: 83 observations
##   predicted class=good  expected loss=0.3217948  P(node) =0.02410406
##     class counts:    13    70
##    probabilities: 0.322 0.678
##
## Node number 124: 82 observations,    complexity param=0.003163229
##   predicted class=bad   expected loss=0.4860983  P(node) =0.02786488
##     class counts:    24    58
##    probabilities: 0.514 0.486
##   left son=248 (65 obs) right son=249 (17 obs)
##   Primary splits:
##       Seniority  < 14.5      to the left,  improve=7.284791, (0 missing)
##       seniorityR splits as  -LRLL, improve=7.284791, (0 missing)
##       Savings    < 6.883761  to the right, improve=4.934247, (0 missing)
##       Income     < 245.5     to the right, improve=4.130740, (0 missing)
##       savingsR   splits as  --RLL, improve=3.118667, (0 missing)
##   Surrogate splits:
##       Price < 1382.5    to the right, agree=0.841, adj=0.235, (0 split)
##       debtR splits as  LR--L, agree=0.805, adj=0.059, (0 split)
##
## Node number 125: 318 observations,    complexity param=0.003163229
##   predicted class=good  expected loss=0.2401078  P(node) =0.08697367
##     class counts:    35   283
##    probabilities: 0.240 0.760
##   left son=250 (8 obs) right son=251 (310 obs)
##   Primary splits:
##       Price  < 405       to the left,  improve=7.438997, (0 missing)
##       Job    splits as  RLR-, improve=6.508769, (0 missing)
##       Income < 113.5     to the left,  improve=4.424561, (0 missing)
##       Finrat < 61.81591  to the right, improve=3.724642, (0 missing)
##       Age    < 23.5      to the left,  improve=3.473651, (0 missing)
##
## Node number 126: 8 observations
##   predicted class=bad   expected loss=0.2813025  P(node) =0.003320778
##     class counts:     4     4
##    probabilities: 0.719 0.281
```

```
##
## Node number 127: 691 observations
##   predicted class=good  expected loss=0.1071449  P(node) =0.17263
##     class counts:    31   660
##    probabilities: 0.107 0.893
##
## Node number 170: 93 observations,    complexity param=0.004670715
##   predicted class=bad   expected loss=0.4044799  P(node) =0.034065
##     class counts:    34    59
##    probabilities: 0.596 0.404
##   left son=340 (35 obs) right son=341 (58 obs)
##   Primary splits:
##       Home      splits as  -LRRRL, improve=5.301208, (0 missing)
##       Assets    < 1650      to the left,  improve=4.889697, (0 missing)
##       Finrat    < 55.36398  to the right, improve=4.817600, (0 missing)
##       Savings   < 1.155     to the left,  improve=4.476099, (0 missing)
##       expensesR splits as  RLRRL, improve=3.638053, (0 missing)
##   Surrogate splits:
##       Assets    < 750       to the left,  agree=0.796, adj=0.457, (0 split)
##       assetsR   splits as  LRRRR, agree=0.796, adj=0.457, (0 split)
##       expensesR splits as  RLRLL, agree=0.677, adj=0.143, (0 split)
##       timeR     splits as  RRRLR, agree=0.667, adj=0.114, (0 split)
##       Seniority < 0.5       to the left,  agree=0.656, adj=0.086, (0 split)
##
## Node number 171: 20 observations
##   predicted class=good  expected loss=0  P(node) =0.004670715
##     class counts:     0    20
##    probabilities: 0.000 1.000
##
## Node number 248: 65 observations
##   predicted class=bad   expected loss=0.4007138  P(node) =0.02389477
##     class counts:    24    41
##    probabilities: 0.599 0.401
##
## Node number 249: 17 observations
##   predicted class=good  expected loss=0  P(node) =0.003970107
##     class counts:     0    17
##    probabilities: 0.000 1.000
##
## Node number 250: 8 observations
##   predicted class=bad   expected loss=0.1901808  P(node) =0.003683901
##     class counts:     5     3
##    probabilities: 0.810 0.190
##
## Node number 251: 310 observations
##   predicted class=good  expected loss=0.2149095  P(node) =0.08328977
##     class counts:    30   280
##    probabilities: 0.215 0.785
##
## Node number 340: 35 observations
##   predicted class=bad   expected loss=0.2269363  P(node) =0.01543621
##     class counts:    20    15
##    probabilities: 0.773 0.227
##
```

```
## Node number 341: 58 observations,    complexity param=0.004670715
##   predicted class=good  expected loss=0.4484038  P(node) =0.01862879
##     class counts:    14    44
##    probabilities: 0.448 0.552
##   left son=682 (18 obs) right son=683 (40 obs)
##   Primary splits:
##       priceR    splits as  LLRRR, improve=5.430383, (0 missing)
##       Savings   < 1.125     to the left,  improve=4.280187, (0 missing)
##       Income    < 63.5      to the left,  improve=3.676496, (0 missing)
##       timeR     splits as  RLLRL, improve=3.569661, (0 missing)
##       Seniority < 1.5       to the right, improve=2.937761, (0 missing)
##   Surrogate splits:
##       Price   < 982       to the left,  agree=0.793, adj=0.333, (0 split)
##       Income  < 67.5      to the left,  agree=0.776, adj=0.278, (0 split)
##       incomeR splits as  LRR-R, agree=0.741, adj=0.167, (0 split)
##       Finrat  < 80.08016  to the right, agree=0.724, adj=0.111, (0 split)
##       Savings < 3.210801  to the right, agree=0.724, adj=0.111, (0 split)
##
## Node number 682: 18 observations
##   predicted class=bad   expected loss=0.2813025  P(node) =0.00747175
##     class counts:     9     9
##    probabilities: 0.719 0.281
##
## Node number 683: 40 observations
##   predicted class=good  expected loss=0.267391  P(node) =0.01115704
##     class counts:     5    35
##    probabilities: 0.267 0.733
```

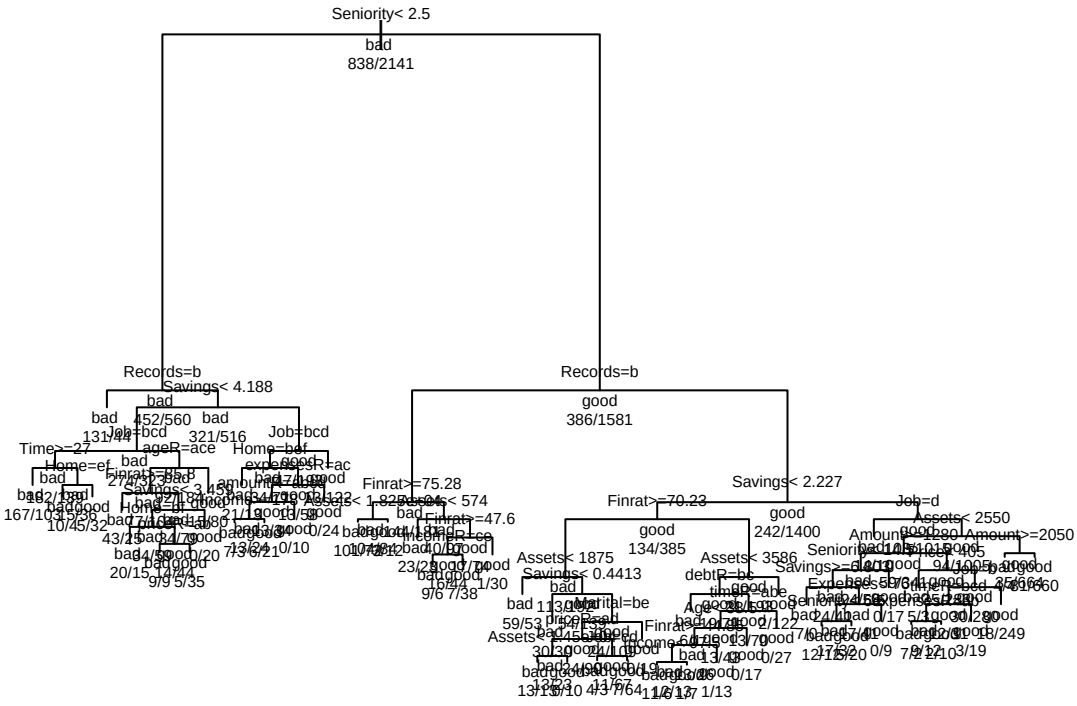
```
# calculate error rate in the learning sample
# (this will give a matrix)
ct2.learn = predict(ct2, data=data[learn,])
# create a vector with predicted status
ct2.learnp = rep("", nlearn)
ct2.learnp[ct2.learn[,1] < 0.5] = "pred_neg"
ct2.learnp[ct2.learn[,1] >= 0.5] = "pred_pos"
# let's make a table
status_learn = table(data$Status[learn], ct2.learnp)
# classification error
100 * sum(diag(status_learn)) / nlearn
```

```
## [1] 21.04733
```

```
# calculate error rate in the testing sample
# (this will give a matrix)
ct2.test = predict(ct2, newdata=data[-learn,])
# create a vector with predicted status
ct2.testp = rep("", ntest)
ct2.testp[ct2.test[,1] < 0.5] = "pred_neg"
ct2.testp[ct2.test[,1] >= 0.5] = "pred_pos"
# let's make a table
status_test = table(data$Status[-learn], ct2.testp)
# classification error
100 * sum(diag(status_test)) / ntest
```

```
## [1] 27.47103
```

```
# we'll repeat the same but changing the cp=0.002
ct3 = rpart(Status ~ .,
            data = data[learn,],
            parms = list(prior=c(0.50, 0.50), split='gini'),
            control = rpart.control(cp=0.002, xval=0, maxdepth=15))

par(mar = c(1,1,2,0.5))
plot(ct3, margin=0.05, compress=TRUE, main="Decision Tree")
text(ct3, use.n=TRUE, all=TRUE, cex=0.5)
```




```
# calculate error rate in the learning sample
# (this will give a matrix)
ct3.learn = predict(ct3, data=data[learn,])
# create a vector with predicted status
ct3.learnp = rep("", nlearn)
ct3.learnp[ct3.learn[,1] < 0.5] = "pred_neg"
ct3.learnp[ct3.learn[,1] >= 0.5] = "pred_pos"
# let's make a table
table(data$Status[learn], ct3.learnp)
```

```
##       ct3.learnp
##        pred_neg pred_pos
##   bad       125      713
##   good     1687      454
```

```
# classification error
100 * sum(diag(table(data$Status[learn], ct3.learnp))) / nlearn
```

```
## [1] 19.43605
```

```
# calculate error rate in the testing sample
# (this will give a matrix)
ct3.test = predict(ct3, newdata=data[-learn,])
# create a vector with predicted status
ct3.testp = rep("", ntest)
ct3.testp[ct3.test[,1] < 0.5] = "pred_neg"
ct3.testp[ct3.test[,1] >= 0.5] = "pred_pos"
# let's make a table
table(data$Status[-learn], ct3.testp)
```

```
##       ct3.testp
##        pred_neg pred_pos
##   bad       110      301
##   good      760      296
```

```
# classification error
100 * sum(diag(table(data$Status[-learn], ct3.testp))) / ntest
```

```
## [1] 27.67553
```

```
# concentration curve
# the positive predictions on the test sample
pred.test = ct2.test[,1]
# the number of individuals in each value
totn = table(-pred.test) / ntest
ac_totn = 100 * cumsum(as.numeric(totn))
# ranking the predictions
rank_pred.test = rank(pred.test)
# how many positive are in each leave?
Status.test = data$Status[-learn]
table(Status.test)
```

```
## Status.test
##  bad good
##  411 1056
```

```
npos = table(Status.test)[1]

tapply(Status.test == "good", rank_pred.test, sum)
```

```
##      9   45.5     80  253.5    444  548.5  642.5  666.5    688    720
##     15     54     12    316     38    144     21     17     12     32
##  760.5    820    879    913    952  992.5 1030.5   1099   1151 1160.5
##     22     69     23     23     31     18     27     45      6      8
## 1170.5   1189   1263   1325   1345 1365.5   1416   1466
##      8     14     47      2     24      3     22      3
```

```
ac_true.pos = 100 * cumsum(rev(as.numeric(totn))) / npos
```

Description: Logistic Regression

```
# load package FactoMineR and ggplot2
require(FactoMineR)
require(ggplot2)
```

Apply logistic regression using all the variables

```r
# let's keep 2/3 of the data for learning, and 1/3 for testing
n = nrow(data)
learn = sample(1:n, size=round(0.67 * n))
nlearn = length(learn)
ntest = n - nlearn

# "whole enchilada" logistic regression model
gl1 = glm(Status ~ ., data=data[learn,], family=binomial)

# use the 'summary' function to obtain more details on the logistic model
# (pay attention to the signicance of the coefficients)
summary(gl1)
```

```
##
## Call:
## glm(formula = Status ~ ., family = binomial, data = data[learn,
##     ])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8385  -0.5349   0.3528   0.6427   2.5235
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 1.223e+00  1.340e+00   0.913 0.361463
## Seniority                   5.019e-02  2.614e-02   1.920 0.054905 .
## Homeother                   6.303e-01  6.367e-01   0.990 0.322230
## Homeowner                   1.083e+00  6.054e-01   1.789 0.073684 .
## Homeparents                 1.400e+00  6.284e-01   2.228 0.025858 *
## Homepriv                    5.232e-01  6.310e-01   0.829 0.406984
## Homerent                    9.664e-01  6.240e-01   1.549 0.121424
## Time                       -9.522e-02  4.925e-02  -1.933 0.053181 .
## Age                        -1.302e-02  1.964e-02  -0.663 0.507316
## Maritalmarried              8.007e-01  5.182e-01   1.545 0.122303
## Maritalseparated           -2.488e-01  5.787e-01  -0.430 0.667222
## Maritalsingle               5.318e-01  5.283e-01   1.007 0.314078
## Maritalwidow                8.052e-02  6.536e-01   0.123 0.901959
## Recordsyes_rec             -1.644e+00  1.291e-01 -12.739  < 2e-16 ***
## Jobfreelance               -7.790e-01  1.325e-01  -5.880 4.10e-09 ***
## Jobothers                  -4.124e-01  2.736e-01  -1.507 0.131715
## Jobpartime                 -1.244e+00  1.637e-01  -7.601 2.93e-14 ***
## Expenses                   -1.255e-02  1.060e-02  -1.183 0.236616
## Income                      2.432e-03  1.730e-03   1.406 0.159801
## Assets                      7.172e-06  8.514e-06   0.842 0.399599
## Debt                        3.094e-05  8.019e-05   0.386 0.699624
## Amount                     -1.132e-03  5.521e-04  -2.051 0.040284 *
## Price                       3.228e-04  3.498e-04   0.923 0.356043
## Finrat                     -9.197e-03  1.354e-02  -0.680 0.496813
## Savings                    -5.636e-02  3.871e-02  -1.456 0.145441
## seniorityRsen (1,3]         3.180e-01  1.540e-01   2.065 0.038877 *
## seniorityRsen (14,99]       7.157e-01  5.432e-01   1.318 0.187657
## seniorityRsen (3,8]         5.289e-01  2.014e-01   2.626 0.008629 **
## seniorityRsen (8,14]        6.737e-01  3.308e-01   2.036 0.041713 *
## timeRtime (12,24]           6.124e-01  7.061e-01   0.867 0.385768
```

```
## timeRtime (24,36]             1.519e+00  1.272e+00   1.195 0.232185
## timeRtime (36,48]             2.625e+00  1.840e+00   1.426 0.153763
## timeRtime (48,99]             3.990e+00  2.423e+00   1.647 0.099641 .
## ageRage (25,30]               1.771e-01  2.003e-01   0.884 0.376669
## ageRage (30,40]               2.093e-01  2.901e-01   0.721 0.470690
## ageRage (40,50]               4.261e-01  4.644e-01   0.918 0.358863
## ageRage (50,99]               1.403e-01  6.659e-01   0.211 0.833075
## expensesRexp (40,50]          5.272e-02  2.086e-01   0.253 0.800455
## expensesRexp (50,60]          2.245e-02  3.205e-01   0.070 0.944153
## expensesRexp (60,80]          1.650e-01  4.647e-01   0.355 0.722462
## expensesRexp (80,1e+04]      -1.453e-01  6.795e-01  -0.214 0.830644
## incomeRinc (110,140]          6.594e-01  2.293e-01   2.876 0.004028 **
## incomeRinc (140,190]          6.550e-01  2.860e-01   2.290 0.022019 *
## incomeRinc (190,1e+04]        7.488e-01  3.913e-01   1.913 0.055686 .
## incomeRinc (80,110]           1.546e-01  1.785e-01   0.866 0.386403
## assetsRasset (0,3e+03]        3.495e-01  2.043e-01   1.710 0.087184 .
## assetsRasset (3e+03,5e+03]    7.916e-01  2.098e-01   3.773 0.000162 ***
## assetsRasset (5e+03,8e+03]    1.033e+00  2.418e-01   4.273 1.93e-05 ***
## assetsRasset (8e+03,1e+06]    1.144e+00  2.816e-01   4.063 4.85e-05 ***
## debtRdebt (0,500]            -3.119e-01  2.653e-01  -1.176 0.239723
## debtRdebt (1.5e+03,2.5e+03]  -8.940e-01  3.176e-01  -2.815 0.004877 **
## debtRdebt (2.5e+03,1e+06]    -1.071e+00  4.198e-01  -2.551 0.010733 *
## debtRdebt (500,1.5e+03]      -3.431e-01  2.601e-01  -1.319 0.187131
## amountRam (1.1e+03,1.4e+03]   6.555e-02  3.771e-01   0.174 0.862012
## amountRam (1.4e+03,1e+05]     3.245e-02  4.906e-01   0.066 0.947260
## amountRam (600,900]           2.647e-01  2.298e-01   1.152 0.249416
## amountRam (900,1.1e+03]       2.128e-01  3.054e-01   0.697 0.485975
## priceRpriz (1.3e+03,1.5e+03]  5.617e-01  2.668e-01   2.105 0.035259 *
## priceRpriz (1.5e+03,1.8e+03]  5.331e-01  2.984e-01   1.786 0.074043 .
## priceRpriz (1.8e+03,1e+05]    1.726e-01  3.954e-01   0.437 0.662430
## priceRpriz (1e+03,1.3e+03]    6.679e-01  2.151e-01   3.105 0.001903 **
## finratRfinr (50,70]           2.678e-01  3.384e-01   0.791 0.428741
## finratRfinr (70,80]           5.593e-02  4.665e-01   0.120 0.904569
## finratRfinr (80,90]           2.482e-02  5.626e-01   0.044 0.964805
## finratRfinr (90,100]         -1.792e-01  6.785e-01  -0.264 0.791739
## savingsRsav (0,2]             4.538e-01  2.345e-01   1.935 0.053003 .
## savingsRsav (2,4]             7.983e-01  2.984e-01   2.675 0.007476 **
## savingsRsav (4,6]             6.888e-01  3.666e-01   1.879 0.060223 .
## savingsRsav (6,99]            1.115e+00  4.757e-01   2.344 0.019056 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3528.8  on 2978  degrees of freedom
## Residual deviance: 2481.1  on 2910  degrees of freedom
## AIC: 2619.1
##
## Number of Fisher Scoring iterations: 5
```

```
# let's use the 'anova' function to get a sequential analysis of
# variance of the model fit (ie importance of variables in the model)
anova(gl1)
```

```
## Analysis of Deviance Table
```

```
##
## Model: binomial, link: logit
##
## Response: Status
##
## Terms added sequentially (first to last)
##
##
##            Df Deviance Resid. Df Resid. Dev
## NULL                        2978     3528.8
## Seniority   1  270.943      2977     3257.9
## Home        5   89.105      2972     3168.8
## Time        1   21.795      2971     3147.0
## Age         1    5.094      2970     3141.9
## Marital     4   11.576      2966     3130.3
## Records     1  185.783      2965     2944.5
## Job         3  138.118      2962     2806.4
## Expenses    1   10.920      2961     2795.5
## Income      1   55.361      2960     2740.1
## Assets      1    6.971      2959     2733.1
## Debt        1    8.643      2958     2724.5
## Amount      1   56.274      2957     2668.2
## Price       1   45.543      2956     2622.7
## Finrat      1   10.241      2955     2612.4
## Savings     1    0.497      2954     2611.9
## seniorityR  4    9.179      2950     2602.8
## timeR       4    9.418      2946     2593.3
## ageR        4    3.507      2942     2589.8
## expensesR   4    3.140      2938     2586.7
## incomeR     4   30.258      2934     2556.4
## assetsR     4   20.713      2930     2535.7
## debtR       4   12.499      2926     2523.2
## amountR     4   15.003      2922     2508.2
## priceR      4   13.724      2918     2494.5
## finratR     4    4.281      2914     2490.2
## savingsR    4    9.136      2910     2481.1
```

```
# we can also use the 'step' function to perform model selection by
# applying a backward elimination method based on AIC (Akaike Info. Criterion)
step(gl1)
```

```
## Start:  AIC=2619.08
## Status ~ Seniority + Home + Time + Age + Marital + Records +
##     Job + Expenses + Income + Assets + Debt + Amount + Price +
##     Finrat + Savings + seniorityR + timeR + ageR + expensesR +
##     incomeR + assetsR + debtR + amountR + priceR + finratR +
##     savingsR
##
##              Df Deviance    AIC
## - expensesR   4   2483.6 2613.6
## - amountR     4   2483.9 2613.9
## - ageR        4   2485.1 2615.1
## - finratR     4   2485.8 2615.8
## - Debt        1   2481.2 2617.2
## - Age         1   2481.5 2617.5
```

```
## - Finrat      1   2481.5 2617.5
## - Assets      1   2481.9 2617.9
## - Price       1   2482.1 2618.1
## - Expenses    1   2482.4 2618.4
## - Income      1   2483.1 2619.1
## - Savings     1   2483.1 2619.1
## <none>            2481.1 2619.1
## - seniorityR  4   2490.2 2620.2
## - savingsR    4   2490.2 2620.2
## - Seniority   1   2485.1 2621.1
## - Time        1   2485.1 2621.1
## - debtR       4   2491.6 2621.6
## - incomeR     4   2491.8 2621.8
## - Amount      1   2486.2 2622.2
## - timeR       4   2493.8 2623.8
## - Marital     4   2496.9 2626.9
## - priceR      4   2497.2 2627.2
## - Home        5   2504.0 2632.0
## - assetsR     4   2506.9 2636.9
## - Job         3   2559.0 2691.0
## - Records     1   2650.0 2786.0
##
## Step:  AIC=2613.6
## Status ~ Seniority + Home + Time + Age + Marital + Records +
##     Job + Expenses + Income + Assets + Debt + Amount + Price +
##     Finrat + Savings + seniorityR + timeR + ageR + incomeR +
##     assetsR + debtR + amountR + priceR + finratR + savingsR
##
##              Df Deviance    AIC
## - amountR     4   2486.4 2608.4
## - ageR        4   2487.9 2609.9
## - finratR     4   2488.4 2610.4
## - Debt        1   2483.7 2611.7
## - Finrat      1   2484.1 2612.1
## - Age         1   2484.2 2612.2
## - Assets      1   2484.5 2612.5
## - Price       1   2484.5 2612.5
## - Savings     1   2485.6 2613.6
## <none>            2483.6 2613.6
## - Income      1   2485.6 2613.6
## - savingsR    4   2492.6 2614.6
## - seniorityR  4   2492.7 2614.7
## - Time        1   2487.5 2615.5
## - Seniority   1   2487.7 2615.7
## - debtR       4   2493.8 2615.8
## - Amount      1   2488.4 2616.4
## - incomeR     4   2494.6 2616.6
## - timeR       4   2496.2 2618.2
## - priceR      4   2499.9 2621.9
## - Expenses    1   2494.8 2622.8
## - Marital     4   2502.0 2624.0
## - Home        5   2506.0 2626.0
## - assetsR     4   2509.4 2631.4
## - Job         3   2562.1 2686.1
```

```
## - Records     1   2653.8 2781.8
##
## Step:  AIC=2608.45
## Status ~ Seniority + Home + Time + Age + Marital + Records +
##     Job + Expenses + Income + Assets + Debt + Amount + Price +
##     Finrat + Savings + seniorityR + timeR + ageR + incomeR +
##     assetsR + debtR + priceR + finratR + savingsR
##
##              Df Deviance    AIC
## - ageR        4   2490.7 2604.7
## - Debt        1   2486.5 2606.5
## - Finrat      1   2486.6 2606.6
## - Age         1   2486.9 2606.9
## - Assets      1   2487.3 2607.3
## - finratR     4   2493.4 2607.4
## <none>            2486.4 2608.4
## - Price       1   2488.6 2608.6
## - Savings     1   2488.9 2608.9
## - savingsR    4   2495.0 2609.0
## - Income      1   2489.0 2609.0
## - seniorityR  4   2495.3 2609.3
## - Time        1   2490.0 2610.0
## - Seniority   1   2490.5 2610.5
## - debtR       4   2497.0 2611.0
## - timeR       4   2498.2 2612.2
## - incomeR     4   2498.2 2612.2
## - Amount      1   2496.1 2616.1
## - Marital     4   2504.5 2618.5
## - Expenses    1   2498.7 2618.7
## - Home        5   2509.2 2621.2
## - assetsR     4   2511.8 2625.8
## - priceR      4   2512.6 2626.6
## - Job         3   2565.4 2681.4
## - Records     1   2657.7 2777.7
##
## Step:  AIC=2604.7
## Status ~ Seniority + Home + Time + Age + Marital + Records +
##     Job + Expenses + Income + Assets + Debt + Amount + Price +
##     Finrat + Savings + seniorityR + timeR + incomeR + assetsR +
##     debtR + priceR + finratR + savingsR
##
##              Df Deviance    AIC
## - Debt        1   2490.8 2602.8
## - Finrat      1   2491.0 2603.0
## - finratR     4   2497.5 2603.5
## - Assets      1   2491.6 2603.6
## - Age         1   2492.0 2604.0
## - Price       1   2492.6 2604.6
## <none>            2490.7 2604.7
## - savingsR    4   2499.0 2605.0
## - Savings     1   2493.2 2605.2
## - Income      1   2493.4 2605.4
## - Seniority   1   2493.8 2605.8
## - Time        1   2494.0 2606.0
```

```
## - seniorityR  4   2501.1 2607.1
## - debtR       4   2501.5 2607.5
## - timeR       4   2502.1 2608.1
## - incomeR     4   2502.3 2608.3
## - Amount      1   2499.8 2611.8
## - Expenses    1   2501.2 2613.2
## - Marital     4   2507.8 2613.8
## - Home        5   2512.9 2616.9
## - priceR      4   2516.0 2622.0
## - assetsR     4   2516.1 2622.1
## - Job         3   2569.2 2677.2
## - Records     1   2660.1 2772.1
##
## Step:  AIC=2602.84
## Status ~ Seniority + Home + Time + Age + Marital + Records +
##     Job + Expenses + Income + Assets + Amount + Price + Finrat +
##     Savings + seniorityR + timeR + incomeR + assetsR + debtR +
##     priceR + finratR + savingsR
##
##              Df Deviance    AIC
## - Finrat      1   2491.1 2601.1
## - finratR     4   2497.6 2601.6
## - Assets      1   2491.7 2601.7
## - Age         1   2492.2 2602.2
## - Price       1   2492.7 2602.7
## <none>            2490.8 2602.8
## - savingsR    4   2499.0 2603.0
## - Savings     1   2493.4 2603.4
## - Income      1   2493.9 2603.9
## - Seniority   1   2493.9 2603.9
## - Time        1   2494.2 2604.2
## - seniorityR  4   2501.2 2605.2
## - timeR       4   2502.2 2606.2
## - incomeR     4   2502.6 2606.6
## - Amount      1   2500.0 2610.0
## - Expenses    1   2501.6 2611.6
## - Marital     4   2507.9 2611.9
## - debtR       4   2510.3 2614.3
## - Home        5   2513.0 2615.0
## - priceR      4   2516.1 2620.1
## - assetsR     4   2516.2 2620.2
## - Job         3   2569.6 2675.6
## - Records     1   2660.8 2770.8
##
## Step:  AIC=2601.11
## Status ~ Seniority + Home + Time + Age + Marital + Records +
##     Job + Expenses + Income + Assets + Amount + Price + Savings +
##     seniorityR + timeR + incomeR + assetsR + debtR + priceR +
##     finratR + savingsR
##
##              Df Deviance    AIC
## - Assets      1   2492.0 2600.0
## - Age         1   2492.5 2600.5
## <none>            2491.1 2601.1
```

```
## - savingsR    4   2499.4 2601.4
## - Savings     1   2493.5 2601.5
## - Income      1   2494.0 2602.0
## - Seniority   1   2494.2 2602.2
## - Price       1   2494.3 2602.3
## - Time        1   2494.6 2602.6
## - seniorityR  4   2501.5 2603.5
## - finratR     4   2501.7 2603.7
## - incomeR     4   2502.6 2604.6
## - timeR       4   2503.4 2605.4
## - Expenses    1   2501.7 2609.7
## - Marital     4   2508.2 2610.2
## - Amount      1   2504.4 2612.4
## - debtR       4   2510.5 2612.5
## - Home        5   2513.4 2613.4
## - assetsR     4   2516.4 2618.4
## - priceR      4   2516.8 2618.8
## - Job         3   2569.7 2673.7
## - Records     1   2660.8 2768.8
##
## Step:  AIC=2599.98
## Status ~ Seniority + Home + Time + Age + Marital + Records +
##     Job + Expenses + Income + Amount + Price + Savings + seniorityR +
##     timeR + incomeR + assetsR + debtR + priceR + finratR + savingsR
##
##              Df Deviance    AIC
## - Age         1   2493.4 2599.4
## <none>            2492.0 2600.0
## - savingsR    4   2500.3 2600.3
## - Savings     1   2494.5 2600.5
## - Income      1   2495.1 2601.1
## - Seniority   1   2495.1 2601.1
## - Price       1   2495.4 2601.4
## - Time        1   2495.5 2601.5
## - seniorityR  4   2502.4 2602.4
## - finratR     4   2502.5 2602.5
## - incomeR     4   2503.6 2603.6
## - timeR       4   2504.6 2604.6
## - Expenses    1   2502.7 2608.7
## - Marital     4   2508.9 2608.9
## - debtR       4   2511.4 2611.4
## - Amount      1   2505.4 2611.4
## - Home        5   2514.3 2612.3
## - priceR      4   2517.6 2617.6
## - assetsR     4   2531.9 2631.9
## - Job         3   2570.1 2672.1
## - Records     1   2661.8 2767.8
##
## Step:  AIC=2599.38
## Status ~ Seniority + Home + Time + Marital + Records + Job +
##     Expenses + Income + Amount + Price + Savings + seniorityR +
##     timeR + incomeR + assetsR + debtR + priceR + finratR + savingsR
##
##              Df Deviance    AIC
```

```
## <none>            2493.4 2599.4
## - savingsR    4   2501.6 2599.6
## - Savings     1   2495.9 2599.9
## - Seniority   1   2496.0 2600.0
## - Income      1   2496.4 2600.4
## - Price       1   2496.8 2600.8
## - Time        1   2497.0 2601.0
## - finratR     4   2503.7 2601.7
## - seniorityR  4   2504.5 2602.5
## - incomeR     4   2504.9 2602.9
## - timeR       4   2506.4 2604.4
## - Marital     4   2510.4 2608.4
## - Expenses    1   2504.6 2608.6
## - debtR       4   2511.6 2609.6
## - Amount      1   2507.1 2611.1
## - Home        5   2516.7 2612.7
## - priceR      4   2519.0 2617.0
## - assetsR     4   2532.1 2630.1
## - Job         3   2571.9 2671.9
## - Records     1   2665.3 2769.3
##
## Call:  glm(formula = Status ~ Seniority + Home + Time + Marital + Records +
##     Job + Expenses + Income + Amount + Price + Savings + seniorityR +
##     timeR + incomeR + assetsR + debtR + priceR + finratR + savingsR,
##     family = binomial, data = data[learn, ])
##
## Coefficients:
##                   (Intercept)                     Seniority
##                     0.5462050                     0.0405282
##                     Homeother                     Homeowner
##                     0.6098392                     1.0768895
##                    Homeparents                      Homepriv
##                     1.3718592                     0.5160695
##                      Homerent                          Time
##                     0.9292852                    -0.0892810
##                Maritalmarried              Maritalseparated
##                     0.8337042                    -0.1738908
##                 Maritalsingle                 Maritalwidow
##                     0.5579600                     0.0614361
##                Recordsyes_rec                Jobfreelance
##                    -1.6368653                    -0.7776485
##                     Jobothers                     Jobpartime
##                    -0.5630251                    -1.2354969
##                      Expenses                        Income
##                    -0.0134819                     0.0029197
##                         Amount                         Price
##                    -0.0014604                     0.0005319
##                       Savings           seniorityRsen (1,3]
##                    -0.0613866                     0.3413326
##         seniorityRsen (14,99]           seniorityRsen (3,8]
##                     0.8457011                     0.5915305
##          seniorityRsen (8,14]             timeRtime (12,24]
##                     0.7639826                     0.5383464
```

```
##            timeRtime (24,36]              timeRtime (36,48]
##                    1.3873974                      2.4629163
##            timeRtime (48,99]           incomeRinc (110,140]
##                    3.7601028                      0.6805777
##         incomeRinc (140,190]         incomeRinc (190,1e+04]
##                    0.7017612                      0.7699443
##          incomeRinc (80,110]         assetsRasset (0,3e+03]
##                    0.1767094                      0.3531707
##   assetsRasset (3e+03,5e+03]    assetsRasset (5e+03,8e+03]
##                    0.8108260                      1.0407828
##   assetsRasset (8e+03,1e+06]             debtRdebt (0,500]
##                    1.2305642                     -0.2916967
##  debtRdebt (1.5e+03,2.5e+03]     debtRdebt (2.5e+03,1e+06]
##                   -0.8142196                     -0.9177960
##      debtRdebt (500,1.5e+03]  priceRpriz (1.3e+03,1.5e+03]
##                   -0.2898358                      0.5669796
## priceRpriz (1.5e+03,1.8e+03]    priceRpriz (1.8e+03,1e+05]
##                    0.5198064                      0.1412146
##   priceRpriz (1e+03,1.3e+03]           finratRfinr (50,70]
##                    0.7176613                      0.2159518
##          finratRfinr (70,80]           finratRfinr (80,90]
##                   -0.0828739                     -0.1885193
##         finratRfinr (90,100]             savingsRsav (0,2]
##                   -0.4588176                      0.4392379
##            savingsRsav (2,4]             savingsRsav (4,6]
##                    0.7427290                      0.6278230
##           savingsRsav (6,99]
##                    1.0060084
##
## Degrees of Freedom: 2978 Total (i.e. Null);  2926 Residual
## Null Deviance:       3529
## Residual Deviance: 2493  AIC: 2599
```

Apply logistic regression after removing some variables

```
# new model
glf <- glm(formula = Status ~ Seniority + Age + Income + Debt + Amount + Finrat +
             seniorityR + expensesR + assetsR + priceR + savingsR + Home + Marital + Records +
             Job, family = binomial, data = data[learn, ])
# check summary
summary(glf)
```

```
##
## Call:
## glm(formula = Status ~ Seniority + Age + Income + Debt + Amount +
##     Finrat + seniorityR + expensesR + assetsR + priceR + savingsR +
##     Home + Marital + Records + Job, family = binomial, data = data[learn,
##     ])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.0934  -0.5440   0.3667   0.6636   2.6426
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept)                  -1.420e-01  8.817e-01  -0.161 0.872022
## Seniority                     4.802e-02  2.603e-02   1.845 0.065096 .
## Age                          -5.793e-03  6.294e-03  -0.920 0.357342
## Income                        3.129e-03  1.241e-03   2.521 0.011717 *
## Debt                         -1.245e-04  4.647e-05  -2.680 0.007361 **
## Amount                       -6.345e-04  2.448e-04  -2.592 0.009534 **
## Finrat                       -2.168e-02  4.346e-03  -4.989 6.07e-07 ***
## seniorityRsen (1,3]           3.338e-01  1.517e-01   2.201 0.027755 *
## seniorityRsen (14,99]         7.839e-01  5.400e-01   1.452 0.146566
## seniorityRsen (3,8]           5.488e-01  1.976e-01   2.777 0.005482 **
## seniorityRsen (8,14]          6.692e-01  3.256e-01   2.055 0.039851 *
## expensesRexp (40,50]         -2.188e-02  1.727e-01  -0.127 0.899200
## expensesRexp (50,60]         -1.206e-01  1.914e-01  -0.630 0.528531
## expensesRexp (60,80]         -8.100e-02  2.130e-01  -0.380 0.703729
## expensesRexp (80,1e+04]      -5.493e-01  2.487e-01  -2.209 0.027202 *
## assetsRasset (0,3e+03]        3.309e-01  1.972e-01   1.678 0.093399 .
## assetsRasset (3e+03,5e+03]    7.344e-01  2.004e-01   3.664 0.000248 ***
## assetsRasset (5e+03,8e+03]    1.004e+00  2.291e-01   4.383 1.17e-05 ***
## assetsRasset (8e+03,1e+06]    1.217e+00  2.299e-01   5.294 1.20e-07 ***
## priceRpriz (1.3e+03,1.5e+03]  5.625e-01  2.027e-01   2.775 0.005521 **
## priceRpriz (1.5e+03,1.8e+03]  5.244e-01  2.235e-01   2.346 0.018966 *
## priceRpriz (1.8e+03,1e+05]    2.189e-01  3.185e-01   0.687 0.491873
## priceRpriz (1e+03,1.3e+03]    7.179e-01  1.746e-01   4.111 3.94e-05 ***
## savingsRsav (0,2]             5.427e-01  2.130e-01   2.548 0.010832 *
## savingsRsav (2,4]             9.849e-01  2.339e-01   4.211 2.55e-05 ***
## savingsRsav (4,6]             8.902e-01  2.709e-01   3.287 0.001014 **
## savingsRsav (6,99]            1.041e+00  3.338e-01   3.118 0.001824 **
## Homeother                     7.020e-01  6.440e-01   1.090 0.275674
## Homeowner                     1.100e+00  6.129e-01   1.795 0.072659 .
## Homeparents                   1.374e+00  6.358e-01   2.160 0.030741 *
## Homepriv                      5.455e-01  6.372e-01   0.856 0.391944
## Homerent                      9.665e-01  6.318e-01   1.530 0.126075
## Maritalmarried                7.138e-01  5.020e-01   1.422 0.155046
## Maritalseparated             -2.594e-01  5.610e-01  -0.462 0.643762
## Maritalsingle                 4.881e-01  5.098e-01   0.957 0.338401
## Maritalwidow                  9.029e-02  6.351e-01   0.142 0.886957
## Recordsyes_rec               -1.616e+00  1.262e-01 -12.808  < 2e-16 ***
## Jobfreelance                 -7.512e-01  1.284e-01  -5.849 4.94e-09 ***
## Jobothers                    -4.194e-01  2.646e-01  -1.585 0.113034
## Jobpartime                   -1.254e+00  1.605e-01  -7.814 5.56e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3528.8  on 2978  degrees of freedom
## Residual deviance: 2534.5  on 2939  degrees of freedom
## AIC: 2614.5
##
## Number of Fisher Scoring iterations: 5
```

```
# check coefficients
exp(glf$coefficients)
```

```
##                  (Intercept)                    Seniority
```

```
##                     0.8675886                     1.0491900
##                           Age                        Income
##                     0.9942236                     1.0031339
##                          Debt                        Amount
##                     0.9998755                     0.9993657
##                        Finrat           seniorityRsen (1,3]
##                     0.9785512                     1.3962026
##         seniorityRsen (14,99]           seniorityRsen (3,8]
##                     2.1900178                     1.7312302
##          seniorityRsen (8,14]          expensesRexp (40,50]
##                     1.9525967                     0.9783588
##          expensesRexp (50,60]          expensesRexp (60,80]
##                     0.8863639                     0.9221950
##       expensesRexp (80,1e+04]        assetsRasset (0,3e+03]
##                     0.5773446                     1.3922042
##    assetsRasset (3e+03,5e+03]    assetsRasset (5e+03,8e+03]
##                     2.0842009                     2.7290361
##    assetsRasset (8e+03,1e+06] priceRpriz (1.3e+03,1.5e+03]
##                     3.3779624                     1.7551040
## priceRpriz (1.5e+03,1.8e+03]    priceRpriz (1.8e+03,1e+05]
##                     1.6894056                     1.2447248
##    priceRpriz (1e+03,1.3e+03]             savingsRsav (0,2]
##                     2.0501524                     1.7206446
##             savingsRsav (2,4]             savingsRsav (4,6]
##                     2.6775392                     2.4356302
##            savingsRsav (6,99]                      Homeother
##                     2.8307267                     2.0177890
##                      Homeowner                   Homeparents
##                     3.0047064                     3.9493960
##                       Homepriv                      Homerent
##                     1.7254245                     2.6288074
##                Maritalmarried               Maritalseparated
##                     2.0417095                     0.7714918
##                 Maritalsingle                   Maritalwidow
##                     1.6291861                     1.0944900
##                Recordsyes_rec                  Jobfreelance
##                     0.1986371                     0.4718103
##                      Jobothers                    Jobpartime
##                     0.6574702                     0.2853673
```

```r
# re-expressed fitted values
glf$fitted.values = 1 - glf$fitted.values

# create vector for predictions
glfpred = rep(NA, length(glf$fitted.values))
glfpred[glf$fitted.values < 0.5] = 0
glfpred[glf$fitted.values >= 0.5] = 1

# how is the prediction? (confusion matrix)
table(data$Status[learn], glfpred)
```

```
##       glfpred
##          0    1
##   bad   400  432
##   good 1958  189
```

```r
# error rate
error_rate.learn = 100*sum(diag(table(data$Status[learn], glfpred))) / nlearn
error_rate.learn
```

```
## [1] 19.77174
```

```r
# let's use the test data to get predictions
glft = predict(glf, newdata=data[-learn,])
pt = 1 / (1 + exp(-glft))
pt = 1 - pt

# vector of predicted values
glftpred = rep(NA, length(pt))
glftpred[pt < 0.5] = 0
glftpred[pt >= 0.5] = 1

# confusion matrix
table(data$Status[-learn], glftpred)
```

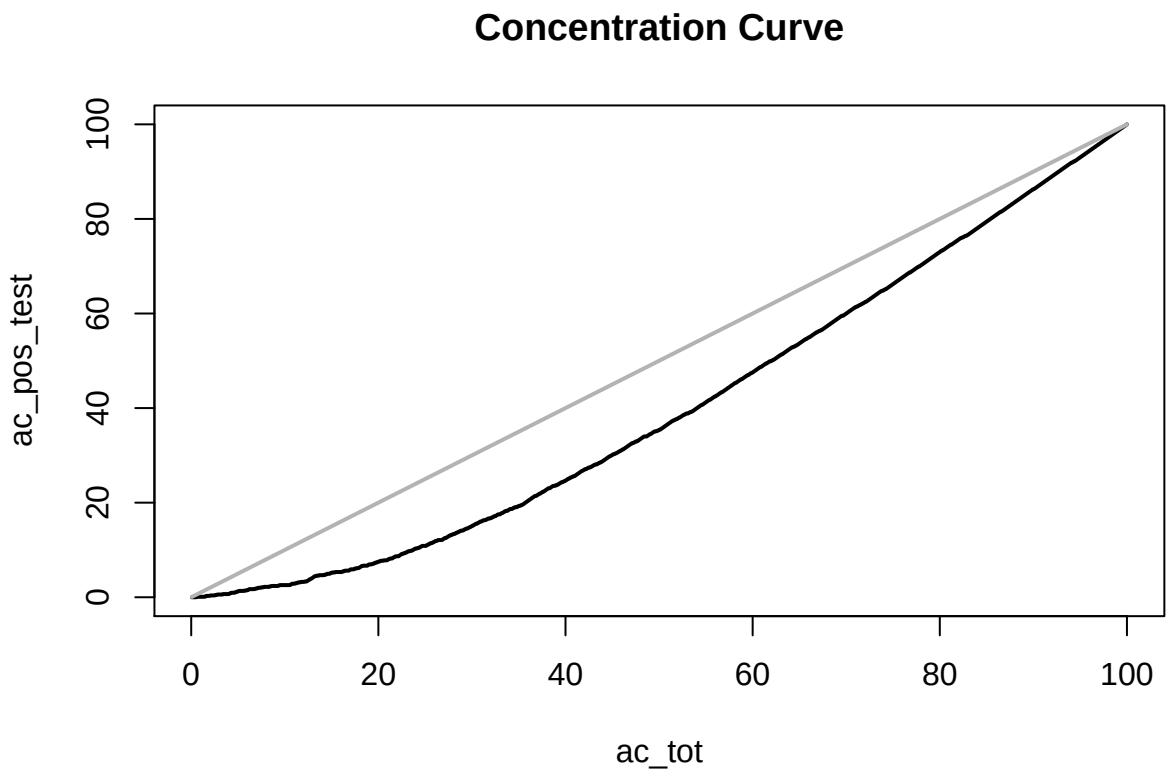
```
##       glftpred
##          0   1
##   bad  201 216
##   good 971  79
```

```r
error_rate.test = 100*sum(diag(table(data$Status[-learn], glftpred))) / ntest
error_rate.test
```

```
## [1] 19.08657
```

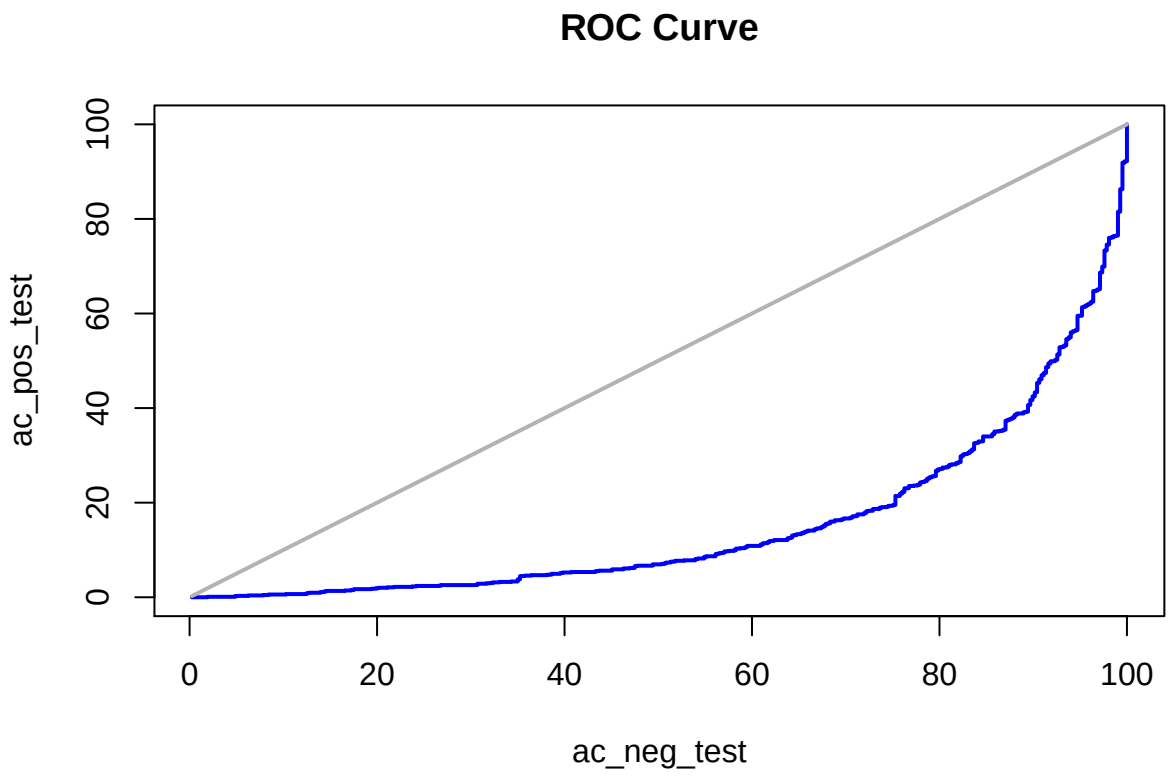
Concentration curve

```r
ac_tot = 100*(1:ntest) / ntest

pt.ord = order(pt, decreasing=T)

Status_test = data$Status[-learn]
npos = table(Status_test)[2]

ac_pos_test = 100*cumsum(Status_test[pt.ord] == "good") / npos

plot(ac_tot, ac_pos_test, type="l", lwd=2,
     main="Concentration Curve")
lines(ac_tot, ac_tot, col="gray70", lwd=2)
```



ROC Curve

```
nneg = ntest - npos

ac_neg_test = 100*cumsum(Status_test[pt.ord]=="bad") / nneg

plot(ac_neg_test, ac_pos_test, type="l", lwd=2, main="ROC Curve", col="blue")
lines(ac_neg_test, ac_neg_test, col="grey70", lwd=2)
```


ROC Curve
ac_pos_test
0
20
40
60
80
100
ac_neg_test
0
20
40
60
80
100