Linear Model and Variable Selection

Benoit Mialet

02/12/2021

Working directory setting:

```
setwd("D:/Formations/DSTI/2021 07 - Advanced stats and ML/assignment")
```

This study uses two datasets : ukcomp1_r.dat (training set) and ukcomp2_r.dat (testing set). My objective is to explain the variable RETCAP by the others ans try to identify the variables really needed for the explanation.

To answer the question I will proceed as following:

1. Import data files into dataframes, and look at the data
2. First look at the correlation between explanatory variables
3. Construct a first linear model and check properties of the noise (distribution, variance)
4. Using the Learning sample, construct several models, one for each of the following variable selection methods:
   - Methods using correction of α:
     - Variable selection using Bonferroni correction
     - Variable selection using Benjamini & Hocheberg correction
   - Stepwise selection with different criteria:
     - F (Fisher criterion)
     - AIC (Akaike Information Criterion)
     - BIC (Bayesian Information Criterion)
   - Penalizing method: Lasso method combined with a cross validation method
   - Random Forest method
5. Compute and compare Testing error ($||\hat{Y} - Y||^2$) for each model, thanks to the testing sample, and then select the model with the minimum Testing error, as my best and final model.

I) Data handling and first look to the data

I first import each dataset into a dataframe and name them *ukcomp_train* and *ukcomp_test* for more clarity:

```
ukcomp_train <- read.table(
  "ukcomp1_r.dat", header=TRUE,
  sep="",dec=".",
  fileEncoding="latin1",
  check.names=FALSE)
```

```r
ukcomp_test <- read.table(
  "ukcomp2_r.dat",
  header=TRUE,
  sep="",dec=".",
  fileEncoding="latin1",
  check.names=FALSE
  )
```

I observe some basic information about the data: dimension of the dataframe, type (class) of variables, number of NA values, and first observations

```r
str(ukcomp_train)
```

```
## 'data.frame':    40 obs. of  13 variables:
##  $ RETCAP : num  0.26 0.57 0.09 0.32 0.17 0.24 0.53 0.26 0.13 0.16 ...
##  $ GEARRAT: num  0.46 0 0.24 0.45 0.91 0.26 0.52 0.24 0.19 0.29 ...
##  $ CAPINT : num  0.64 1.79 0.36 1.86 1.26 1.54 3.34 1.38 0.91 1.7 ...
##  $ WCFTDT : num  0.25 0.33 0.2 0.21 0.12 0.25 0.4 0.37 0.21 0.18 ...
##  $ LOGSALE: num  4.11 4.25 4.44 4.71 4.85 5.61 4.83 4.49 4.13 4.4 ...
##  $ LOGASST: num  4.3 4 4.88 4.44 4.75 5.42 4.3 4.35 4.17 4.17 ...
##  $ CURRAT : num  1.53 1.73 0.44 1.23 1.76 1.44 0.83 1.45 2.89 2.13 ...
##  $ QUIKRAT: num  0.18 1.26 0.39 0.69 0.9 1.23 0.83 0.58 1.95 0.56 ...
##  $ NFATAST: num  0.1 0.12 0.94 0.29 0.26 0.42 0.14 0.4 0.06 0.21 ...
##  $ INVTAST: num  0.74 0.27 0.01 0.29 0.33 0.06 0 0.36 0.29 0.58 ...
##  $ FATTOT : num  0.12 0.15 0.97 0.52 0.54 0.57 0.21 1.04 0.11 0.4 ...
##  $ PAYOUT : num  0.07 0.3 0.57 0 0.31 0.15 0.21 0.16 0.39 0.46 ...
##  $ WCFTCL : num  0.25 0.33 0.5 0.23 0.21 0.37 0.59 0.44 0.21 0.21 ...
```

```r
sum(is.na(ukcomp_train))
```

```
## [1] 0
```

```r
sum(is.na(ukcomp_test))
```

```
## [1] 0
```

```r
head(ukcomp_train,10)
```

```
##    RETCAP GEARRAT CAPINT WCFTDT LOGSALE LOGASST CURRAT QUIKRAT NFATAST INVTAST
## 1    0.26    0.46   0.64   0.25    4.11    4.30   1.53    0.18    0.10    0.74
## 2    0.57    0.00   1.79   0.33    4.25    4.00   1.73    1.26    0.12    0.27
## 3    0.09    0.24   0.36   0.20    4.44    4.88   0.44    0.39    0.94    0.01
## 4    0.32    0.45   1.86   0.21    4.71    4.44   1.23    0.69    0.29    0.29
## 5    0.17    0.91   1.26   0.12    4.85    4.75   1.76    0.90    0.26    0.33
## 6    0.24    0.26   1.54   0.25    5.61    5.42   1.44    1.23    0.42    0.06
## 7    0.53    0.52   3.34   0.40    4.83    4.30   0.83    0.83    0.14    0.00
## 8    0.26    0.24   1.38   0.37    4.49    4.35   1.45    0.58    0.40    0.36
## 9    0.13    0.19   0.91   0.21    4.13    4.17   2.89    1.95    0.06    0.29
## 10   0.16    0.29   1.70   0.18    4.40    4.17   2.13    0.56    0.21    0.58
##    FATTOT PAYOUT WCFTCL
```

```
## 1    0.12   0.07   0.25
## 2    0.15   0.30   0.33
## 3    0.97   0.57   0.50
## 4    0.52   0.00   0.23
## 5    0.54   0.31   0.21
## 6    0.57   0.15   0.37
## 7    0.21   0.21   0.59
## 8    1.04   0.16   0.44
## 9    0.11   0.39   0.21
## 10   0.40   0.46   0.21
```

```
head(ukcomp_test,10)
```

```
##    RETCAP WCFTCL WCFTDT GEARRAT LOGSALE LOGASST NFATAST CAPINT FATTOT INVTAST
## 1    0.19   0.16   0.16    0.15  5.2297  4.8375    0.28   2.47   0.36    0.42
## 2    0.22   0.26   0.16    0.54  4.1495  4.3402    0.13   0.64   0.16    0.04
## 3    0.17   0.26   0.20    0.49  5.3831  4.8811    0.43   3.18   0.74    0.13
## 4    0.12   0.08   0.08    0.39  4.1225  3.9333    0.23   1.55   0.50    0.37
## 5    0.21   0.34   0.34    0.11  4.7795  4.5877    0.30   1.56   0.50    0.20
## 6    0.12   0.25   0.25    0.19  4.1503  3.9086    0.34   1.74   0.38    0.31
## 7    0.15   0.25   0.16    0.35  5.6998  5.5577    0.48   1.39   0.62    0.22
## 8    0.10   0.12   0.09    0.39  4.4162  4.2128    0.26   1.60   0.42    0.30
## 9    0.08   0.04   0.04    0.50  4.7108  4.5126    0.25   1.58   0.33    0.31
## 10   0.31   0.12   0.11    0.41  4.4678  4.1928    0.17   1.88   0.25    0.31
##    PAYOUT QUIKRAT CURRAT
## 1    0.31    0.54   1.33
## 2    0.45    0.83   0.93
## 3    0.50    0.84   1.09
## 4    0.65    0.50   1.09
## 5    0.25    1.10   1.74
## 6    0.80    1.00   1.89
## 7    0.46    0.73   1.38
## 8    1.03    0.94   1.57
## 9    0.00    0.74   1.28
## 10   0.25    0.66   1.10
```

Here, all variables are numerical with no missing value.

In the following, I will use train sample for observation of correlations and for variable selection.

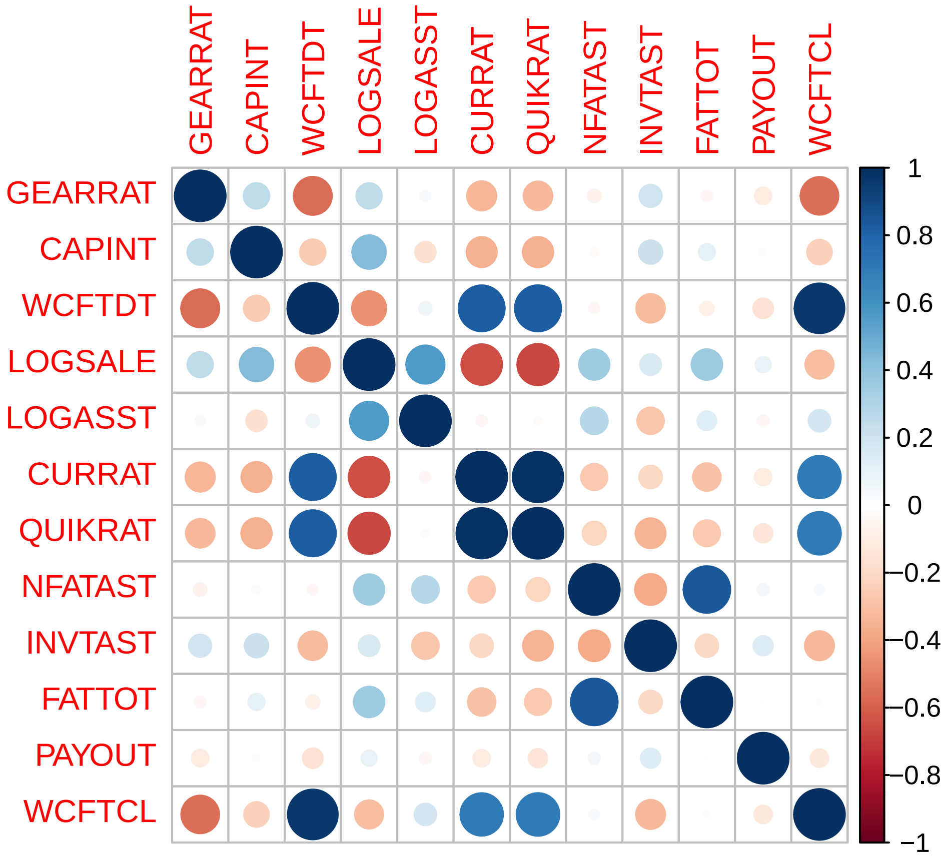
II) Correlation between explanatory variables

Visualization of correlations between explanatory variables can give an idea of which one are strongly correlated. It doesn't allow any variable selection but can help to explain why one variable will be selected over another correlated one, during variable selection step.

```
library(corrplot)
```

```
## corrplot 0.90 loaded
```

```r
corrplot(cor(ukcomp_train[,-1],use="complete.obs"),method="circle") #I exclude response variable #1
```




```r
cor(ukcomp_train[,-1])
```

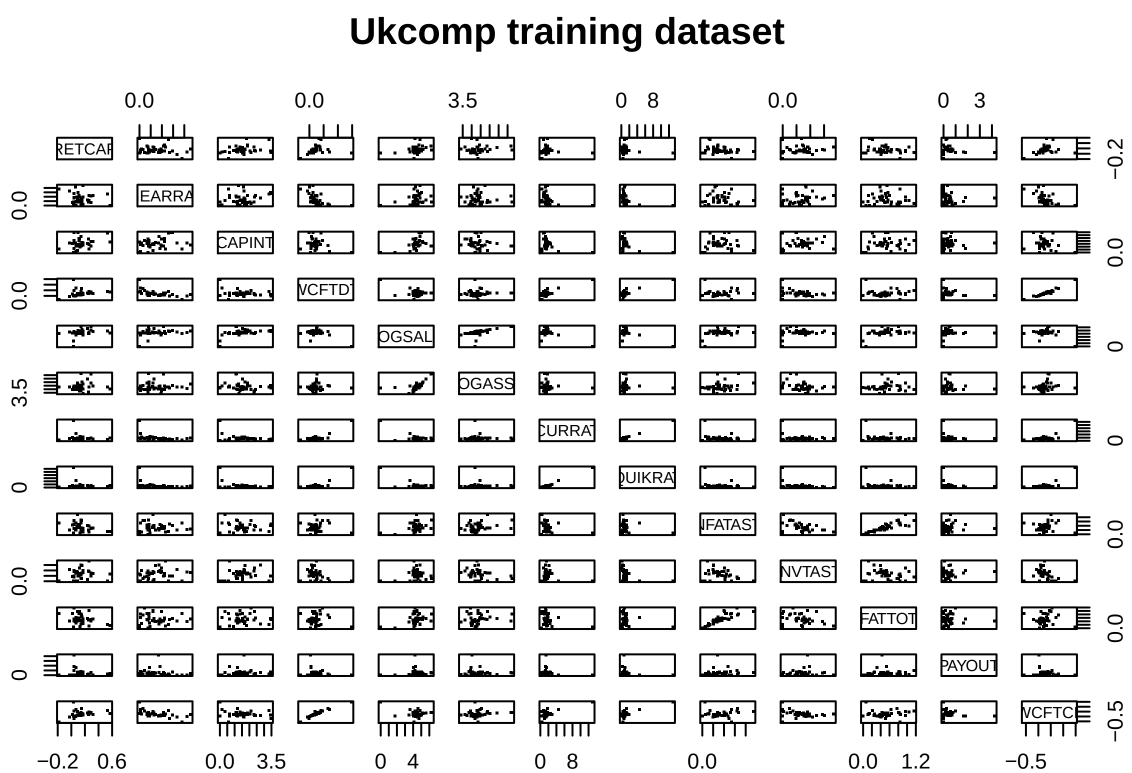
```
##             GEARRAT      CAPINT      WCFTDT     LOGSALE     LOGASST      CURRAT
## GEARRAT  1.00000000  0.25323859 -0.56112672  0.25018002  0.03871926 -0.33093568
## CAPINT   0.25323859  1.00000000 -0.25158456  0.43747369 -0.16483426 -0.35298327
## WCFTDT  -0.56112672 -0.25158456  1.00000000 -0.45334754  0.06388027  0.82052636
## LOGSALE  0.25018002  0.43747369 -0.45334754  1.00000000  0.56770391 -0.64055168
## LOGASST  0.03871926 -0.16483426  0.06388027  0.56770391  1.00000000 -0.04598247
## CURRAT  -0.33093568 -0.35298327  0.82052636 -0.64055168 -0.04598247  1.00000000
## QUIKRAT -0.32015975 -0.35821195  0.82503436 -0.66237926 -0.02466453  0.98476629
## NFATAST -0.06681103 -0.02861684 -0.04178674  0.35873663  0.28121345 -0.26983474
## INVTAST  0.19325449  0.21418908 -0.31870760  0.16944751 -0.27515621 -0.20228585
## FATTOT  -0.04815797  0.10288097 -0.07290425  0.36299165  0.13889076 -0.29596553
## PAYOUT  -0.10929295  0.01840112 -0.15240228  0.09203103 -0.04951406 -0.10833134
## WCFTCL  -0.55195163 -0.23757748  0.96199607 -0.30997547  0.18291323  0.70114844
##             QUIKRAT     NFATAST    INVTAST       FATTOT       PAYOUT
## GEARRAT -0.32015975 -0.06681103  0.1932545 -0.048157973 -0.109292954
## CAPINT  -0.35821195 -0.02861684  0.2141891  0.102880974  0.018401120
## WCFTDT   0.82503436 -0.04178674 -0.3187076 -0.072904246 -0.152402281
## LOGSALE -0.66237926  0.35873663  0.1694475  0.362991650  0.092031034
## LOGASST -0.02466453  0.28121345 -0.2751562  0.138890759 -0.049514055
## CURRAT   0.98476629 -0.26983474 -0.2022858 -0.295965528 -0.108331343
## QUIKRAT  1.00000000 -0.21161742 -0.3490480 -0.266606389 -0.133357725
```

```
## NFATAST -0.21161742  1.00000000 -0.3745099  0.844412710  0.051839364
## INVTAST -0.34904800 -0.37450992  1.0000000 -0.200048001  0.145645058
## FATTOT  -0.26660639  0.84441271 -0.2000480  1.000000000  0.004029383
## PAYOUT  -0.13335772  0.05183936  0.1456451  0.004029383  1.000000000
## WCFTCL   0.70702569  0.03827180 -0.3279368 -0.012536149 -0.121727324
##              WCFTCL
## GEARRAT -0.55195163
## CAPINT  -0.23757748
## WCFTDT   0.96199607
## LOGSALE -0.30997547
## LOGASST  0.18291323
## CURRAT   0.70114844
## QUIKRAT  0.70702569
## NFATAST  0.03827180
## INVTAST -0.32793678
## FATTOT  -0.01253615
## PAYOUT  -0.12172732
## WCFTCL   1.00000000
```

I observe strong correlations between some variable. I thus expect that the variable selection methods will potentially discard some variables.

I also use another graphical representation that can indicate type of relation (linear, not linear) between pairs of variables.

```
pairs(ukcomp_train, main = "Ukcomp training dataset", pch = ".")
```



III) Multiple linear regression model and noise properties checking

III.1) linear model construction

I build a first linear model to be used in further computations:

```
model = lm(RETCAP~.,data = ukcomp_train)
summary(model)
```

```
##
## Call:
## lm(formula = RETCAP ~ ., data = ukcomp_train)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.126501 -0.043091 -0.002002  0.036908  0.201047
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.18807    0.13392   1.404  0.17160
## GEARRAT     -0.04044    0.07677  -0.527  0.60270
## CAPINT      -0.01414    0.02338  -0.605  0.55048
## WCFTDT       0.30556    0.29737   1.028  0.31328
## LOGSALE      0.11844    0.03612   3.279  0.00287 **
## LOGASST     -0.07696    0.04517  -1.704  0.09994 .
## CURRAT      -0.22328    0.08773  -2.545  0.01696 *
## QUIKRAT      0.17671    0.09163   1.929  0.06437 .
## NFATAST     -0.36998    0.13740  -2.693  0.01202 *
## INVTAST      0.25056    0.18587   1.348  0.18884
## FATTOT      -0.10099    0.08764  -1.152  0.25932
## PAYOUT      -0.01884    0.01769  -1.065  0.29645
## WCFTCL       0.21513    0.19788   1.087  0.28658
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07441 on 27 degrees of freedom
## Multiple R-squared:  0.7889, Adjusted R-squared:  0.6951
## F-statistic: 8.408 on 12 and 27 DF,  p-value: 2.555e-06
```

The p-value of the Fisher test, associated to the null hypothesis $\mathcal{H}_0$ stating that all coefficients are not different from 0, is very low. I thus should consider that one or more variables have an influence on the response variable RETCAP and that their respective coefficient are different from 0. However, **this p-value is to be used only in a gaussian setting**, that's why I have to check first the gaussianity of the distribution of the noise.

III.2) Gaussianity of the distribution of the noise

Before selecting variables, I have to check several conditions in order to validate the linear model.

I first check gaussianity of the distribution of the noise. To do this, I use standardized residuals, because residuals can have different distributions. I will then perform a goodness of fit test (Kolmogorov-Smirnov test) to compare standardized residuals distribution with a standard normal distribution:

```r
st_residuals=rstandard(model)
ks.test(st_residuals, pnorm)
```

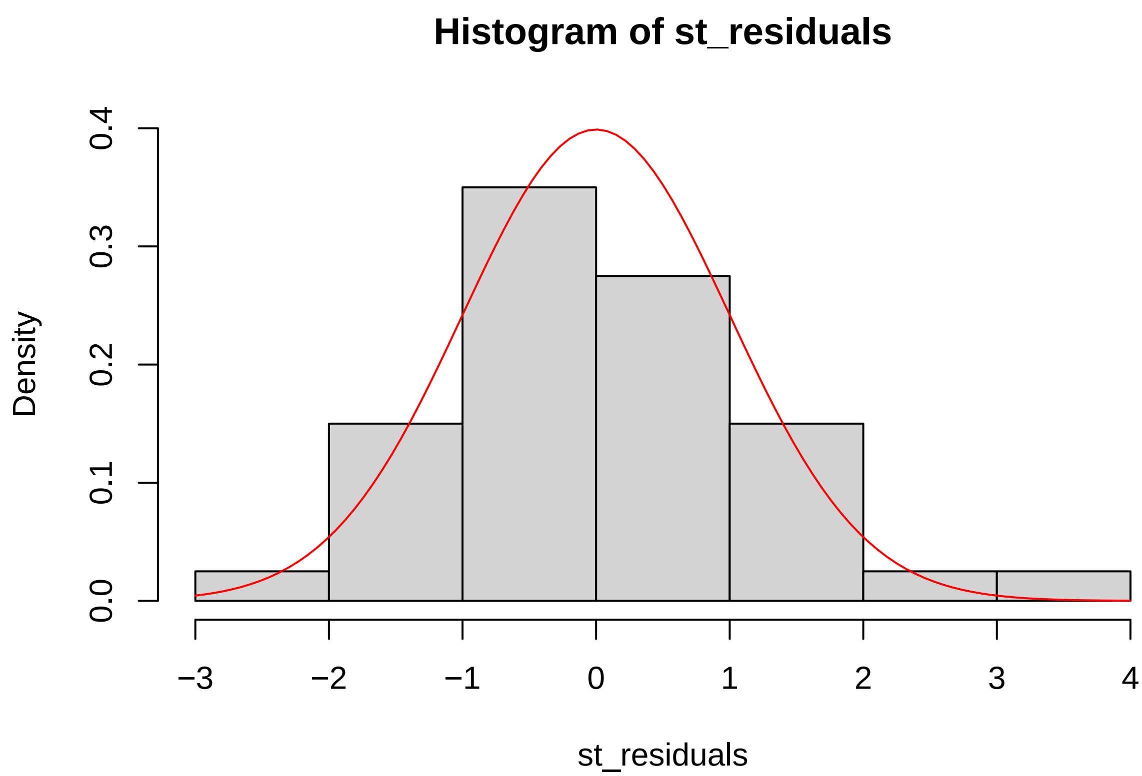
```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  st_residuals
## D = 0.08897, p-value = 0.8817
## alternative hypothesis: two-sided
```

I accept Null Hypothesis as p-value obtained is very high: the standardized residuals distribution is assumed to be same as a standard normal distribution.

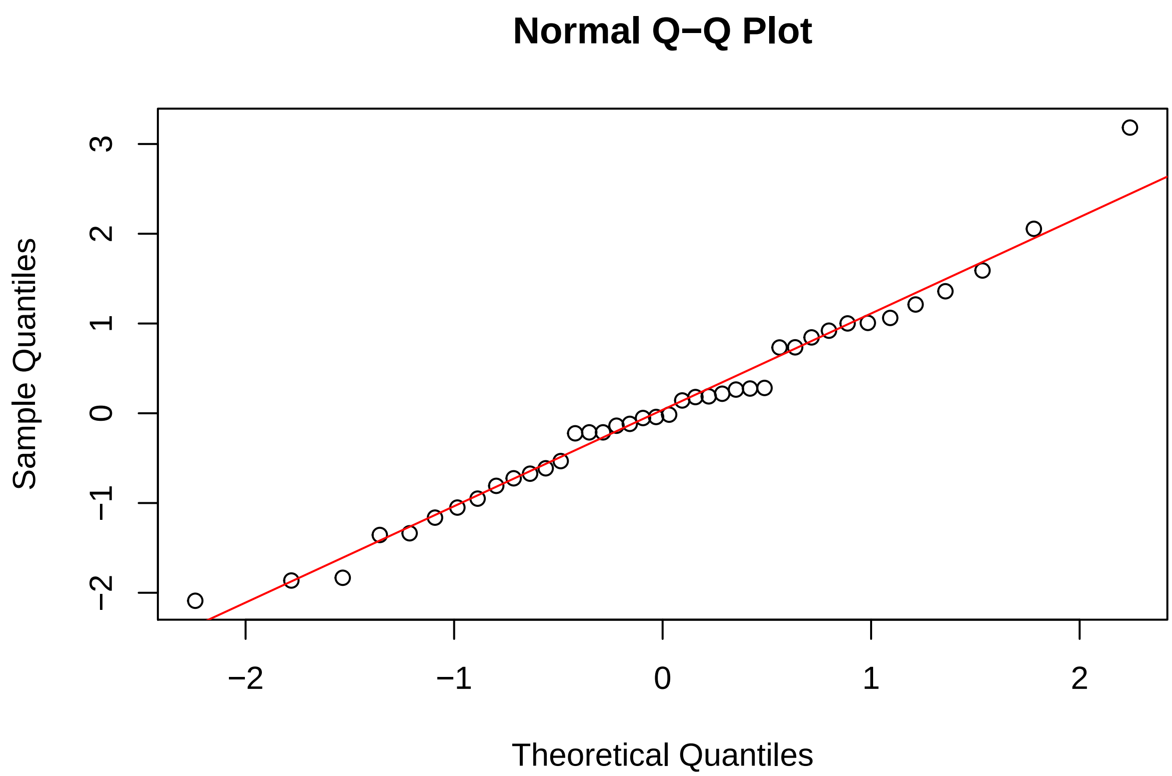
In order to visualize and confirm the result:

- I plot the histogram of the distribution of standardized residuals, I compare this distribution with the probability density function of a standard normal distribution (in red)
- I also plot a QQ-plot comparing quantiles of both distributions

```r
# Density histogram
hist(st_residuals, freq=FALSE, ylim =c(0,0.4))
curve(dnorm(x, mean = 0, sd = 1), from = -3, col = "red", add = TRUE)
```



```
# QQ-PLOT
qqnorm(st_residuals, main = "Normal Q-Q Plot",
       xlab = "Theoretical Quantiles", ylab = "Sample Quantiles",
       plot.it = TRUE, datax = FALSE)
qqline(st_residuals, datax = FALSE, distribution = qnorm,
       probs = c(0.25, 0.75), qtype = 7, col='red')
```



On both graphs, distribution of standardized residuals looks normal

III.3) Variance of the noise

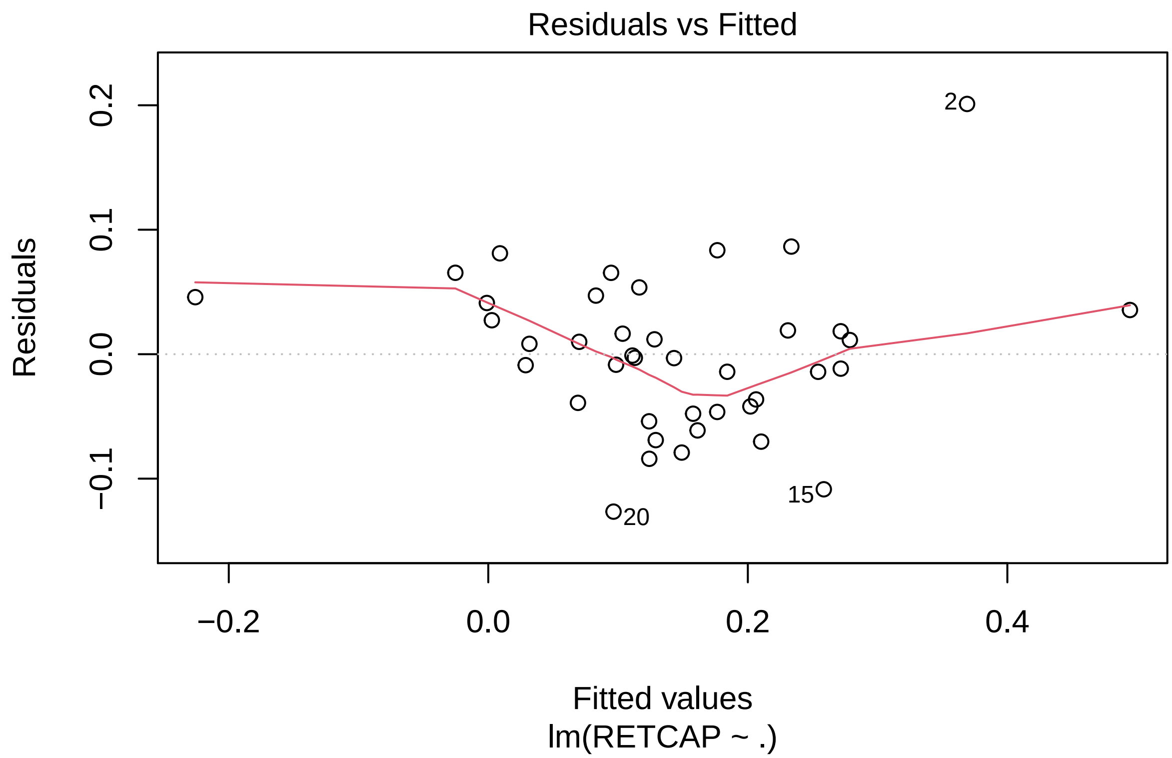
I then check if variance of the noise is constant, by visualization. I will check following conditions:

- Distribution of the noise must be Centered
- Distribution of the noise must be Symmetric
- Variance of the noise must stay Constant (no pattern should be visible)

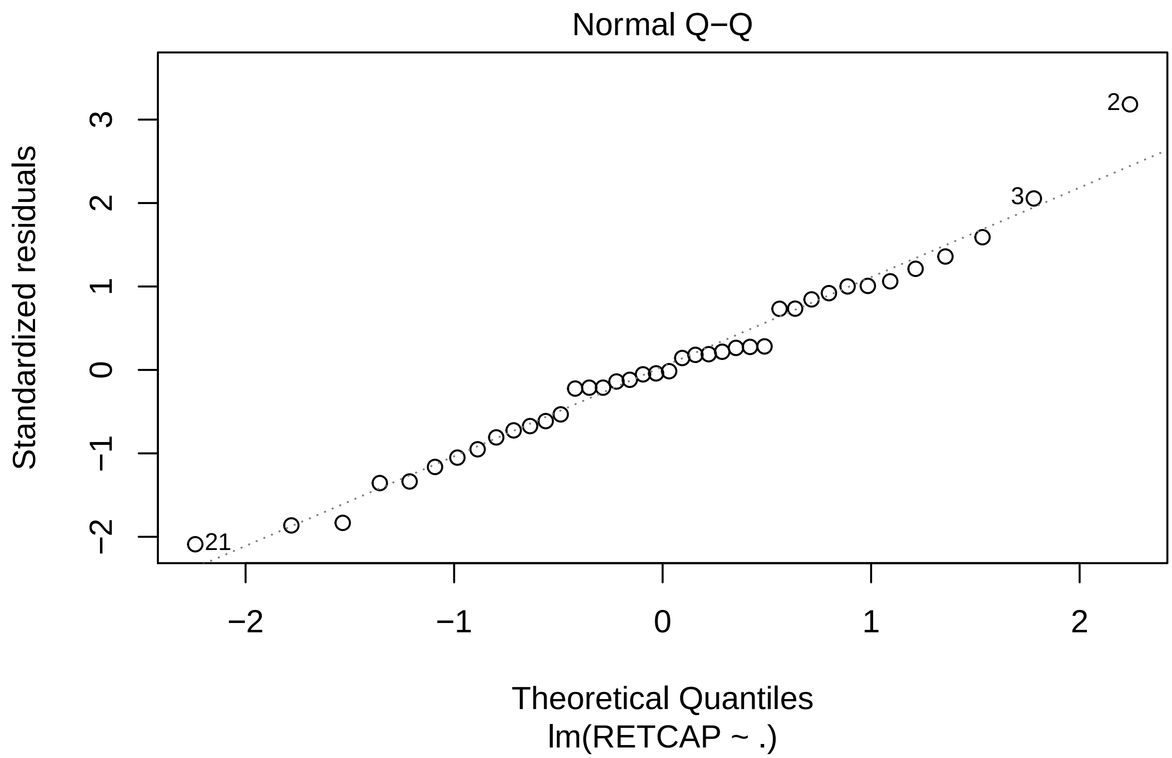
Two options are available for this, leading to the same observations:

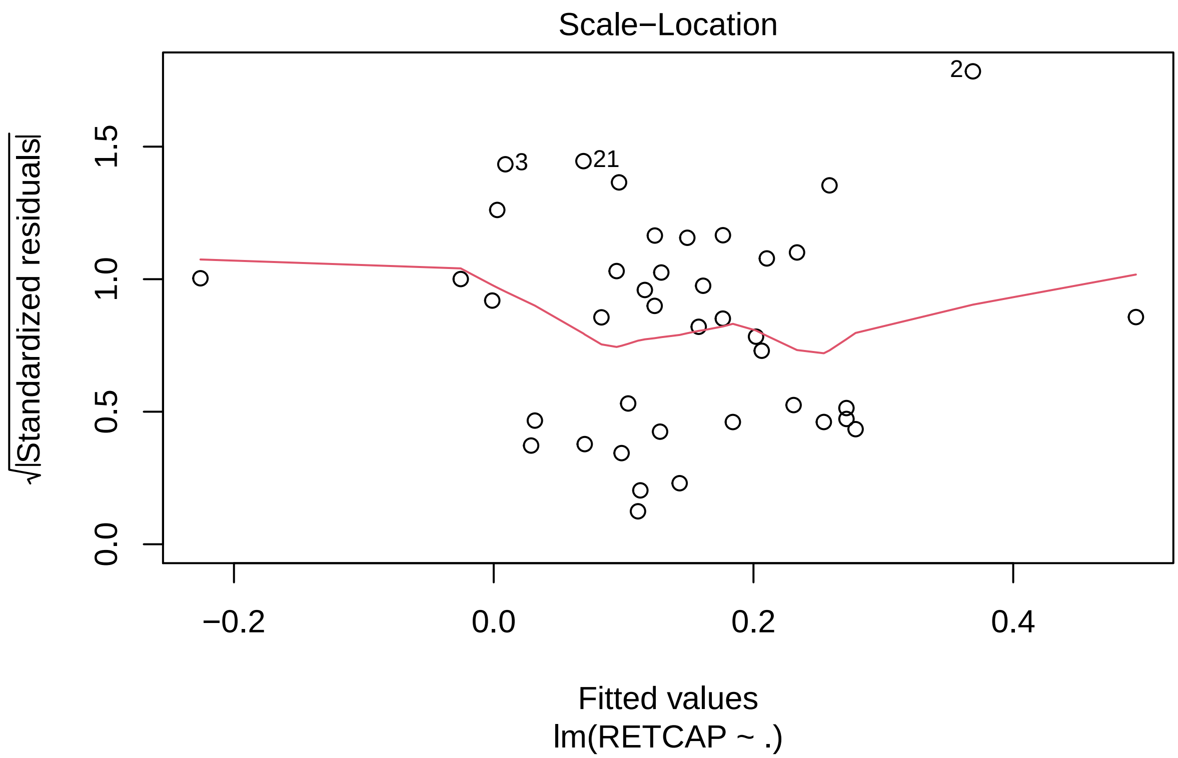
OPTION 1: using plot with lm() model

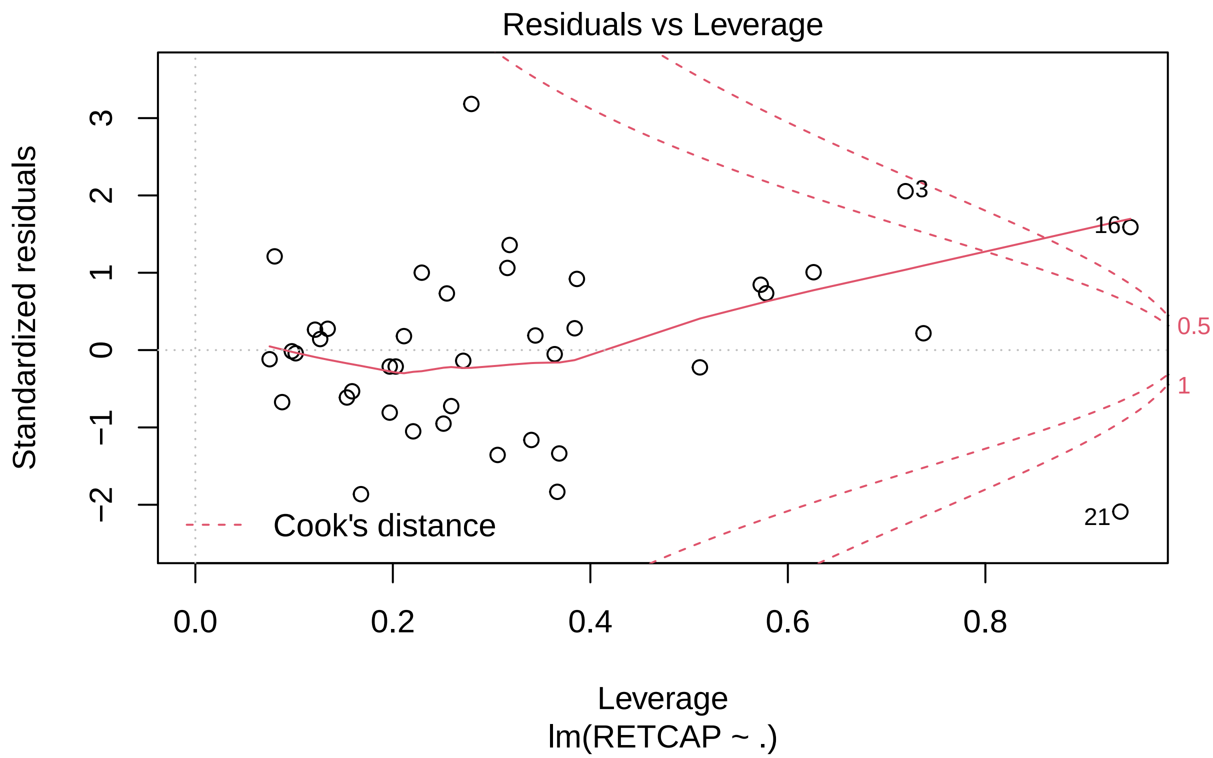
```
plot(model)
```



Residuals vs Fitted
Residuals
0.2
0.1
0.0
−0.1
2
15
20
−0.2
0.0
0.2
0.4
Fitted values
lm(RETCAP ~ .)

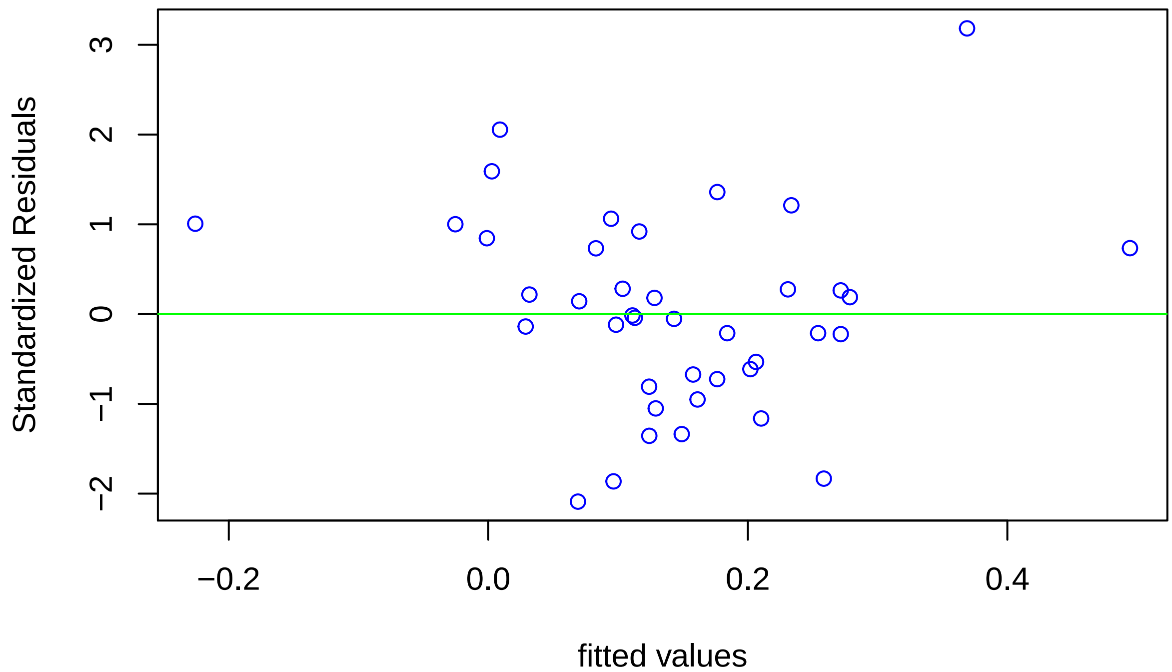

Normal Q−Q
Standardized residuals
3
2
1
0
−1
−2
2
3
21
−2
−1
0
1
2
Theoretical Quantiles
lm(RETCAP ~ .)


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
−0.2
0.0
0.2
0.4
3
21
2
Fitted values
lm(RETCAP ~ .)



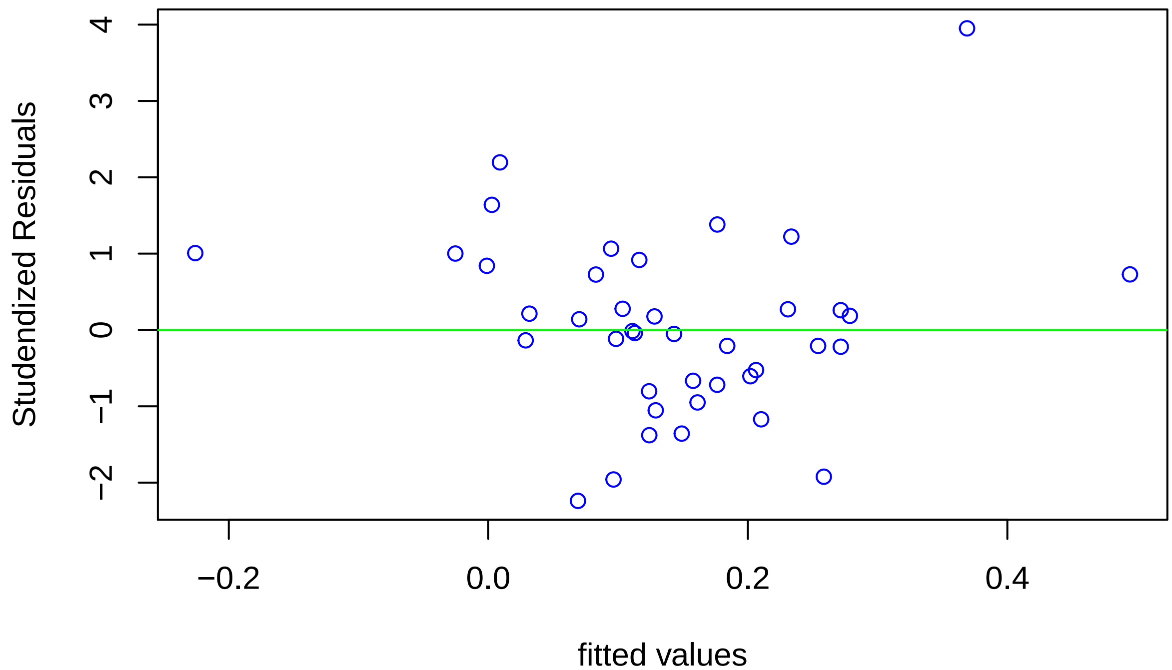
OPTION 2: using a custom function that plots residuals on Y axis and fitted values on X axis:

```
plot.res=function(x, y, title = "", label_x = "", label_y = "")
{
  plot(x,y,col='blue',main=title, xlab = label_x, ylab = label_y)
  abline(h=0,col='green')
}
plot.res(predict(model),st_residuals,"",'fitted values','Standardized Residuals')
```




```
student_residuals=rstudent(model)

plot.res(predict(model),student_residuals,"",'fitted values','Studendized Residuals')
```



Considering that:

- Most of the standardized residual values are between -2 and +2, except for one outlier
- There are approximately as many negative and positive values
- The shape of the noise remains constant with increasing fitted values

I consider that distribution of the noise is centered, symmetric and constant.

CONCLUSION of III) : I assume the distribution of noise to be **gaussian**, **centered**, **symmetric** and **constant**. All the conditions are thus valid to build a linear regression model.

IV) VARIABLE SELECTION

IV.1) Bonferroni correction (non penalizing)

I arbitrary chose a risk level $\alpha = 0.05$.

I first compute the rank of the explanatory variable matrix and then I compute a value for a Bonferroni adjusted risk level alpha:

```
library(Matrix)
rank <- rankMatrix(as.matrix(ukcomp_train[,-1]))[1]
bonferroni_adjusted_alpha <- 0.05 / (rank-1)
bonferroni_adjusted_alpha
```

```
## [1] 0.004545455
```

I observe which variable to select by keeping only ones for which p-value is < bonferroni_adjusted_alpha:

```
summary(model)
```

```
##
## Call:
## lm(formula = RETCAP ~ ., data = ukcomp_train)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.126501 -0.043091 -0.002002  0.036908  0.201047
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.18807    0.13392   1.404  0.17160
## GEARRAT     -0.04044    0.07677  -0.527  0.60270
## CAPINT      -0.01414    0.02338  -0.605  0.55048
## WCFTDT       0.30556    0.29737   1.028  0.31328
## LOGSALE      0.11844    0.03612   3.279  0.00287 **
## LOGASST     -0.07696    0.04517  -1.704  0.09994 .
## CURRAT      -0.22328    0.08773  -2.545  0.01696 *
## QUIKRAT      0.17671    0.09163   1.929  0.06437 .
## NFATAST     -0.36998    0.13740  -2.693  0.01202 *
## INVTAST      0.25056    0.18587   1.348  0.18884
## FATTOT      -0.10099    0.08764  -1.152  0.25932
## PAYOUT      -0.01884    0.01769  -1.065  0.29645
## WCFTCL       0.21513    0.19788   1.087  0.28658
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07441 on 27 degrees of freedom
## Multiple R-squared:  0.7889, Adjusted R-squared:  0.6951
## F-statistic: 8.408 on 12 and 27 DF,  p-value: 2.555e-06
```

only variable *LOGSALE* should be selected with this method.

The linear model would be thus:

```
model_final_BONFERRONI <- lm(RETCAP ~ LOGSALE, data = ukcomp_train)
```

It is actually a simple linear model.

IV.2) Benjamini & Hocheberg correction (non penalizing)

I first get the set of p-values from model summary, then I use the function p.adjust with Benjamini & Hocheberg method ("fdr"):

```
pvalues_model_raw <- summary(model)$coeff[,4]
pvalues_model_adjusted <- p.adjust(pvalues_model_raw, "fdr")
pvalues_model_adjusted
```

```
## (Intercept)     GEARRAT      CAPINT      WCFTDT     LOGSALE     LOGASST
##  0.35070562  0.60269881  0.59635301  0.37024093  0.03725797  0.25983343
##      CURRAT     QUIKRAT     NFATAST     INVTAST      FATTOT      PAYOUT
##  0.07347979  0.20918686  0.07347979  0.35070562  0.37024093  0.37024093
##      WCFTCL
##  0.37024093
```

Here we again select only variable *LOGSALE* with an adjusted p-value < alpha. The model will the the same as the one obtained with Bonferroni correction.

IV.3) Stepwise selection with Fisher criterion

```
library(MASS)
model_final_STEPWISE_F <- stepAIC(model,~.,direction=c("both"),test="F")
```

```
## Start:  AIC=-197.57
## RETCAP ~ GEARRAT + CAPINT + WCFTDT + LOGSALE + LOGASST + CURRAT +
##     QUIKRAT + NFATAST + INVTAST + FATTOT + PAYOUT + WCFTCL
##
##           Df Sum of Sq     RSS     AIC F Value    Pr(F)
## - GEARRAT  1  0.001536 0.15105 -199.16  0.2774 0.602699
## - CAPINT   1  0.002024 0.15154 -199.03  0.3656 0.550480
## - WCFTDT   1  0.005847 0.15536 -198.03  1.0559 0.313281
## - PAYOUT   1  0.006277 0.15579 -197.93  1.1335 0.296452
## - WCFTCL   1  0.006545 0.15606 -197.86  1.1819 0.286582
## - FATTOT   1  0.007352 0.15687 -197.65  1.3277 0.259319
## <none>                 0.14951 -197.57
## - INVTAST  1  0.010063 0.15958 -196.96  1.8173 0.188841
## - LOGASST  1  0.016072 0.16559 -195.49  2.9023 0.099936 .
## - QUIKRAT  1  0.020595 0.17011 -194.41  3.7192 0.064365 .
## - CURRAT   1  0.035865 0.18538 -190.97  6.4768 0.016957 *
## - NFATAST  1  0.040152 0.18967 -190.06  7.2509 0.012025 *
## - LOGSALE  1  0.059554 0.20907 -186.16 10.7545 0.002866 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=-199.16
## RETCAP ~ CAPINT + WCFTDT + LOGSALE + LOGASST + CURRAT + QUIKRAT +
##     NFATAST + INVTAST + FATTOT + PAYOUT + WCFTCL
##
##           Df Sum of Sq     RSS     AIC F Value    Pr(F)
## - CAPINT   1  0.002588 0.15364 -200.48  0.4797 0.494255
## - PAYOUT   1  0.005092 0.15614 -199.84  0.9438 0.339617
## - WCFTCL   1  0.005567 0.15662 -199.71  1.0319 0.318414
## - FATTOT   1  0.007599 0.15865 -199.20  1.4086 0.245267
## <none>                 0.15105 -199.16
## - INVTAST  1  0.008574 0.15962 -198.95  1.5894 0.217812
## - WCFTDT   1  0.009823 0.16087 -198.64  1.8209 0.188015
## + GEARRAT  1  0.001536 0.14951 -197.57  0.2774 0.602699
## - LOGASST  1  0.016709 0.16776 -196.96  3.0974 0.089339 .
## - QUIKRAT  1  0.019187 0.17024 -196.38  3.5566 0.069722 .
## - CURRAT   1  0.034549 0.18560 -192.92  6.4043 0.017286 *
```

```
## - NFATAST  1  0.040106 0.19116 -191.74  7.4344 0.010915 *
## - LOGSALE  1  0.058563 0.20961 -188.06 10.8557 0.002676 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=-200.48
## RETCAP ~ WCFTDT + LOGSALE + LOGASST + CURRAT + QUIKRAT + NFATAST +
##     INVTAST + FATTOT + PAYOUT + WCFTCL
##
##           Df Sum of Sq     RSS     AIC F Value     Pr(F)
## - PAYOUT   1  0.004966 0.15860 -201.21  0.9373 0.3409808
## - WCFTCL   1  0.006079 0.15972 -200.93  1.1474 0.2929192
## - INVTAST  1  0.007156 0.16079 -200.66  1.3508 0.2546140
## <none>                 0.15364 -200.48
## - FATTOT   1  0.008379 0.16202 -200.36  1.5815 0.2185724
## - WCFTDT   1  0.008907 0.16254 -200.23  1.6812 0.2049853
## + CAPINT   1  0.002588 0.15105 -199.16  0.4797 0.4942550
## + GEARRAT  1  0.002100 0.15154 -199.03  0.3880 0.5384020
## - LOGASST  1  0.015302 0.16894 -198.68  2.8884 0.0999246 .
## - QUIKRAT  1  0.016604 0.17024 -198.38  3.1341 0.0871840 .
## - CURRAT   1  0.032326 0.18596 -194.84  6.1018 0.0196291 *
## - NFATAST  1  0.037518 0.19116 -193.74  7.0818 0.0125606 *
## - LOGSALE  1  0.085310 0.23895 -184.82 16.1028 0.0003864 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=-201.21
## RETCAP ~ WCFTDT + LOGSALE + LOGASST + CURRAT + QUIKRAT + NFATAST +
##     INVTAST + FATTOT + WCFTCL
##
##           Df Sum of Sq     RSS     AIC F Value     Pr(F)
## - WCFTCL   1  0.004794 0.16340 -202.02  0.9068 0.3485761
## - FATTOT   1  0.006933 0.16554 -201.50  1.3114 0.2611946
## - INVTAST  1  0.007420 0.16602 -201.38  1.4036 0.2454283
## <none>                 0.15860 -201.21
## + PAYOUT   1  0.004966 0.15364 -200.48  0.9373 0.3409808
## - WCFTDT   1  0.011401 0.17000 -200.43  2.1565 0.1523737
## - LOGASST  1  0.013470 0.17207 -199.95  2.5478 0.1209307
## + CAPINT   1  0.002462 0.15614 -199.84  0.4573 0.5042635
## + GEARRAT  1  0.000643 0.15796 -199.37  0.1181 0.7335495
## - QUIKRAT  1  0.018358 0.17696 -198.83  3.4724 0.0722159 .
## - CURRAT   1  0.035686 0.19429 -195.09  6.7500 0.0143929 *
## - NFATAST  1  0.043176 0.20178 -193.58  8.1668 0.0076823 **
## - LOGSALE  1  0.083618 0.24222 -186.27 15.8163 0.0004068 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=-202.02
## RETCAP ~ WCFTDT + LOGSALE + LOGASST + CURRAT + QUIKRAT + NFATAST +
##     INVTAST + FATTOT
##
##           Df Sum of Sq     RSS     AIC F Value     Pr(F)
## - INVTAST  1  0.006559 0.16996 -202.44   1.244 0.2732138
## <none>                 0.16340 -202.02
```

```
## - FATTOT   1  0.008397 0.17179 -202.01   1.593 0.2163090
## - LOGASST  1  0.010104 0.17350 -201.62   1.917 0.1760716
## + WCFTCL   1  0.004794 0.15860 -201.21   0.907 0.3485761
## + PAYOUT   1  0.003681 0.15972 -200.93   0.691 0.4122752
## + CAPINT   1  0.002924 0.16047 -200.74   0.547 0.4654694
## - QUIKRAT  1  0.015986 0.17938 -200.28   3.033 0.0915055 .
## + GEARRAT  1  0.000217 0.16318 -200.07   0.040 0.8429593
## - CURRAT   1  0.034396 0.19779 -196.38   6.526 0.0157646 *
## - NFATAST  1  0.040969 0.20437 -195.07   7.773 0.0089802 **
## - LOGSALE  1  0.082492 0.24589 -187.67  15.650 0.0004127 ***
## - WCFTDT   1  0.310122 0.47352 -161.46  58.837  1.19e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=-202.44
## RETCAP ~ WCFTDT + LOGSALE + LOGASST + CURRAT + QUIKRAT + NFATAST +
##     FATTOT
##
##           Df Sum of Sq     RSS     AIC F Value     Pr(F)
## <none>                 0.16996 -202.44
## - FATTOT   1  0.009059 0.17902 -202.37   1.706  0.200866
## + INVTAST  1  0.006559 0.16340 -202.02   1.244  0.273214
## - QUIKRAT  1  0.011787 0.18174 -201.76   2.219  0.146087
## + PAYOUT   1  0.003997 0.16596 -201.40   0.747  0.394197
## + WCFTCL   1  0.003932 0.16602 -201.38   0.734  0.398087
## - LOGASST  1  0.015575 0.18553 -200.94   2.933  0.096484 .
## + CAPINT   1  0.001408 0.16855 -200.78   0.259  0.614423
## + GEARRAT  1  0.000167 0.16979 -200.48   0.031  0.862465
## - NFATAST  1  0.045986 0.21594 -194.87   8.658  0.006012 **
## - CURRAT   1  0.048877 0.21883 -194.33   9.203  0.004767 **
## - LOGSALE  1  0.084438 0.25439 -188.31  15.898  0.000363 ***
## - WCFTDT   1  0.303564 0.47352 -163.46  57.156 1.305e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(model_final_STEPWISE_F)
```

```
##
## Call:
## lm(formula = RETCAP ~ WCFTDT + LOGSALE + LOGASST + CURRAT + QUIKRAT +
##     NFATAST + FATTOT, data = ukcomp_train)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.130281 -0.044026  0.002847  0.029266  0.216228
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.14802    0.10075   1.469 0.151548
## WCFTDT       0.61237    0.08100   7.560  1.3e-08 ***
## LOGSALE      0.09899    0.02483   3.987 0.000363 ***
## LOGASST     -0.05548    0.03240  -1.712 0.096484 .
## CURRAT      -0.12169    0.04011  -3.034 0.004767 **
## QUIKRAT      0.06089    0.04087   1.490 0.146087
```

```
## NFATAST     -0.37365    0.12698  -2.943 0.006012 **
## FATTOT      -0.10996    0.08420  -1.306 0.200866
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07288 on 32 degrees of freedom
## Multiple R-squared:   0.76,  Adjusted R-squared:  0.7075
## F-statistic: 14.48 on 7 and 32 DF,  p-value: 2.546e-08
```

7 explanatory variables are selected.

IV.4) Stepwise selection with AIC criterion

I proceed to the variable selection:

```
model_final_STEPWISE_AIC <- stepAIC(model,~.,direction=c("both"))
```

```
## Start:  AIC=-197.57
## RETCAP ~ GEARRAT + CAPINT + WCFTDT + LOGSALE + LOGASST + CURRAT +
##     QUIKRAT + NFATAST + INVTAST + FATTOT + PAYOUT + WCFTCL
##
##           Df Sum of Sq     RSS     AIC
## - GEARRAT  1  0.001536 0.15105 -199.16
## - CAPINT   1  0.002024 0.15154 -199.03
## - WCFTDT   1  0.005847 0.15536 -198.03
## - PAYOUT   1  0.006277 0.15579 -197.93
## - WCFTCL   1  0.006545 0.15606 -197.86
## - FATTOT   1  0.007352 0.15687 -197.65
## <none>                 0.14951 -197.57
## - INVTAST  1  0.010063 0.15958 -196.96
## - LOGASST  1  0.016072 0.16559 -195.49
## - QUIKRAT  1  0.020595 0.17011 -194.41
## - CURRAT   1  0.035865 0.18538 -190.97
## - NFATAST  1  0.040152 0.18967 -190.06
## - LOGSALE  1  0.059554 0.20907 -186.16
##
## Step:  AIC=-199.16
## RETCAP ~ CAPINT + WCFTDT + LOGSALE + LOGASST + CURRAT + QUIKRAT +
##     NFATAST + INVTAST + FATTOT + PAYOUT + WCFTCL
##
##           Df Sum of Sq     RSS     AIC
## - CAPINT   1  0.002588 0.15364 -200.48
## - PAYOUT   1  0.005092 0.15614 -199.84
## - WCFTCL   1  0.005567 0.15662 -199.71
## - FATTOT   1  0.007599 0.15865 -199.20
## <none>                 0.15105 -199.16
## - INVTAST  1  0.008574 0.15962 -198.95
## - WCFTDT   1  0.009823 0.16087 -198.64
## + GEARRAT  1  0.001536 0.14951 -197.57
## - LOGASST  1  0.016709 0.16776 -196.96
## - QUIKRAT  1  0.019187 0.17024 -196.38
## - CURRAT   1  0.034549 0.18560 -192.92
## - NFATAST  1  0.040106 0.19116 -191.74
```

```
## - LOGSALE  1  0.058563 0.20961 -188.06
##
## Step:  AIC=-200.48
## RETCAP ~ WCFTDT + LOGSALE + LOGASST + CURRAT + QUIKRAT + NFATAST +
##     INVTAST + FATTOT + PAYOUT + WCFTCL
##
##           Df Sum of Sq     RSS     AIC
## - PAYOUT   1  0.004966 0.15860 -201.21
## - WCFTCL   1  0.006079 0.15972 -200.93
## - INVTAST  1  0.007156 0.16079 -200.66
## <none>                 0.15364 -200.48
## - FATTOT   1  0.008379 0.16202 -200.36
## - WCFTDT   1  0.008907 0.16254 -200.23
## + CAPINT   1  0.002588 0.15105 -199.16
## + GEARRAT  1  0.002100 0.15154 -199.03
## - LOGASST  1  0.015302 0.16894 -198.68
## - QUIKRAT  1  0.016604 0.17024 -198.38
## - CURRAT   1  0.032326 0.18596 -194.84
## - NFATAST  1  0.037518 0.19116 -193.74
## - LOGSALE  1  0.085310 0.23895 -184.82
##
## Step:  AIC=-201.21
## RETCAP ~ WCFTDT + LOGSALE + LOGASST + CURRAT + QUIKRAT + NFATAST +
##     INVTAST + FATTOT + WCFTCL
##
##           Df Sum of Sq     RSS     AIC
## - WCFTCL   1  0.004794 0.16340 -202.02
## - FATTOT   1  0.006933 0.16554 -201.50
## - INVTAST  1  0.007420 0.16602 -201.38
## <none>                 0.15860 -201.21
## + PAYOUT   1  0.004966 0.15364 -200.48
## - WCFTDT   1  0.011401 0.17000 -200.43
## - LOGASST  1  0.013470 0.17207 -199.95
## + CAPINT   1  0.002462 0.15614 -199.84
## + GEARRAT  1  0.000643 0.15796 -199.37
## - QUIKRAT  1  0.018358 0.17696 -198.83
## - CURRAT   1  0.035686 0.19429 -195.09
## - NFATAST  1  0.043176 0.20178 -193.58
## - LOGSALE  1  0.083618 0.24222 -186.27
##
## Step:  AIC=-202.02
## RETCAP ~ WCFTDT + LOGSALE + LOGASST + CURRAT + QUIKRAT + NFATAST +
##     INVTAST + FATTOT
##
##           Df Sum of Sq     RSS     AIC
## - INVTAST  1  0.006559 0.16996 -202.44
## <none>                 0.16340 -202.02
## - FATTOT   1  0.008397 0.17179 -202.01
## - LOGASST  1  0.010104 0.17350 -201.62
## + WCFTCL   1  0.004794 0.15860 -201.21
## + PAYOUT   1  0.003681 0.15972 -200.93
## + CAPINT   1  0.002924 0.16047 -200.74
## - QUIKRAT  1  0.015986 0.17938 -200.28
## + GEARRAT  1  0.000217 0.16318 -200.07
```

```
## - CURRAT   1  0.034396 0.19779 -196.38
## - NFATAST  1  0.040969 0.20437 -195.07
## - LOGSALE  1  0.082492 0.24589 -187.67
## - WCFTDT   1  0.310122 0.47352 -161.46
##
## Step:  AIC=-202.44
## RETCAP ~ WCFTDT + LOGSALE + LOGASST + CURRAT + QUIKRAT + NFATAST +
##     FATTOT
##
##           Df Sum of Sq     RSS     AIC
## <none>                 0.16996 -202.44
## - FATTOT   1  0.009059 0.17902 -202.37
## + INVTAST  1  0.006559 0.16340 -202.02
## - QUIKRAT  1  0.011787 0.18174 -201.76
## + PAYOUT   1  0.003997 0.16596 -201.40
## + WCFTCL   1  0.003932 0.16602 -201.38
## - LOGASST  1  0.015575 0.18553 -200.94
## + CAPINT   1  0.001408 0.16855 -200.78
## + GEARRAT  1  0.000167 0.16979 -200.48
## - NFATAST  1  0.045986 0.21594 -194.87
## - CURRAT   1  0.048877 0.21883 -194.33
## - LOGSALE  1  0.084438 0.25439 -188.31
## - WCFTDT   1  0.303564 0.47352 -163.46
```

12 explanatory variables are selected.

IV.5) Stepwise selection with BIC criterion

BIC criterion uses the number of observations. I compute them first and then proceed to the variable selection:

```
nb_obs <- length(ukcomp_train$RETCAP)
model_final_STEPWISE_BIC <- stepAIC(model,~.,direction=c("both"),k=log(nb_obs))
```

```
## Start:  AIC=-175.61
## RETCAP ~ GEARRAT + CAPINT + WCFTDT + LOGSALE + LOGASST + CURRAT +
##     QUIKRAT + NFATAST + INVTAST + FATTOT + PAYOUT + WCFTCL
##
##           Df Sum of Sq     RSS     AIC
## - GEARRAT  1  0.001536 0.15105 -178.89
## - CAPINT   1  0.002024 0.15154 -178.76
## - WCFTDT   1  0.005847 0.15536 -177.77
## - PAYOUT   1  0.006277 0.15579 -177.66
## - WCFTCL   1  0.006545 0.15606 -177.59
## - FATTOT   1  0.007352 0.15687 -177.38
## - INVTAST  1  0.010063 0.15958 -176.70
## <none>                 0.14951 -175.61
## - LOGASST  1  0.016072 0.16559 -175.22
## - QUIKRAT  1  0.020595 0.17011 -174.14
## - CURRAT   1  0.035865 0.18538 -170.70
## - NFATAST  1  0.040152 0.18967 -169.79
## - LOGSALE  1  0.059554 0.20907 -165.89
##
```

```
## Step:  AIC=-178.89
## RETCAP ~ CAPINT + WCFTDT + LOGSALE + LOGASST + CURRAT + QUIKRAT +
##     NFATAST + INVTAST + FATTOT + PAYOUT + WCFTCL
##
##           Df Sum of Sq     RSS     AIC
## - CAPINT   1  0.002588 0.15364 -181.90
## - PAYOUT   1  0.005092 0.15614 -181.26
## - WCFTCL   1  0.005567 0.15662 -181.14
## - FATTOT   1  0.007599 0.15865 -180.62
## - INVTAST  1  0.008574 0.15962 -180.38
## - WCFTDT   1  0.009823 0.16087 -180.06
## <none>                 0.15105 -178.89
## - LOGASST  1  0.016709 0.16776 -178.39
## - QUIKRAT  1  0.019187 0.17024 -177.80
## + GEARRAT  1  0.001536 0.14951 -175.61
## - CURRAT   1  0.034549 0.18560 -174.34
## - NFATAST  1  0.040106 0.19116 -173.16
## - LOGSALE  1  0.058563 0.20961 -169.48
##
## Step:  AIC=-181.9
## RETCAP ~ WCFTDT + LOGSALE + LOGASST + CURRAT + QUIKRAT + NFATAST +
##     INVTAST + FATTOT + PAYOUT + WCFTCL
##
##           Df Sum of Sq     RSS     AIC
## - PAYOUT   1  0.004966 0.15860 -184.32
## - WCFTCL   1  0.006079 0.15972 -184.04
## - INVTAST  1  0.007156 0.16079 -183.77
## - FATTOT   1  0.008379 0.16202 -183.47
## - WCFTDT   1  0.008907 0.16254 -183.34
## <none>                 0.15364 -181.90
## - LOGASST  1  0.015302 0.16894 -181.79
## - QUIKRAT  1  0.016604 0.17024 -181.49
## + CAPINT   1  0.002588 0.15105 -178.89
## + GEARRAT  1  0.002100 0.15154 -178.76
## - CURRAT   1  0.032326 0.18596 -177.95
## - NFATAST  1  0.037518 0.19116 -176.85
## - LOGSALE  1  0.085310 0.23895 -167.93
##
## Step:  AIC=-184.32
## RETCAP ~ WCFTDT + LOGSALE + LOGASST + CURRAT + QUIKRAT + NFATAST +
##     INVTAST + FATTOT + WCFTCL
##
##           Df Sum of Sq     RSS     AIC
## - WCFTCL   1  0.004794 0.16340 -186.82
## - FATTOT   1  0.006933 0.16554 -186.30
## - INVTAST  1  0.007420 0.16602 -186.18
## - WCFTDT   1  0.011401 0.17000 -185.23
## - LOGASST  1  0.013470 0.17207 -184.75
## <none>                 0.15860 -184.32
## - QUIKRAT  1  0.018358 0.17696 -183.63
## + PAYOUT   1  0.004966 0.15364 -181.90
## + CAPINT   1  0.002462 0.15614 -181.26
## + GEARRAT  1  0.000643 0.15796 -180.79
## - CURRAT   1  0.035686 0.19429 -179.89
```

```
## - NFATAST  1  0.043176 0.20178 -178.38
## - LOGSALE  1  0.083618 0.24222 -171.07
##
## Step:  AIC=-186.82
## RETCAP ~ WCFTDT + LOGSALE + LOGASST + CURRAT + QUIKRAT + NFATAST +
##     INVTAST + FATTOT
##
##           Df Sum of Sq     RSS     AIC
## - INVTAST  1  0.006559 0.16996 -188.93
## - FATTOT   1  0.008397 0.17179 -188.50
## - LOGASST  1  0.010104 0.17350 -188.11
## <none>                 0.16340 -186.82
## - QUIKRAT  1  0.015986 0.17938 -186.77
## + WCFTCL   1  0.004794 0.15860 -184.32
## + PAYOUT   1  0.003681 0.15972 -184.04
## + CAPINT   1  0.002924 0.16047 -183.85
## + GEARRAT  1  0.000217 0.16318 -183.18
## - CURRAT   1  0.034396 0.19779 -182.87
## - NFATAST  1  0.040969 0.20437 -181.56
## - LOGSALE  1  0.082492 0.24589 -174.16
## - WCFTDT   1  0.310122 0.47352 -147.95
##
## Step:  AIC=-188.93
## RETCAP ~ WCFTDT + LOGSALE + LOGASST + CURRAT + QUIKRAT + NFATAST +
##     FATTOT
##
##           Df Sum of Sq     RSS     AIC
## - FATTOT   1  0.009059 0.17902 -190.54
## - QUIKRAT  1  0.011787 0.18174 -189.94
## - LOGASST  1  0.015575 0.18553 -189.11
## <none>                 0.16996 -188.93
## + INVTAST  1  0.006559 0.16340 -186.82
## + PAYOUT   1  0.003997 0.16596 -186.20
## + WCFTCL   1  0.003932 0.16602 -186.18
## + CAPINT   1  0.001408 0.16855 -185.58
## + GEARRAT  1  0.000167 0.16979 -185.28
## - NFATAST  1  0.045986 0.21594 -183.04
## - CURRAT   1  0.048877 0.21883 -182.51
## - LOGSALE  1  0.084438 0.25439 -176.49
## - WCFTDT   1  0.303564 0.47352 -151.63
##
## Step:  AIC=-190.54
## RETCAP ~ WCFTDT + LOGSALE + LOGASST + CURRAT + QUIKRAT + NFATAST
##
##           Df Sum of Sq     RSS     AIC
## - LOGASST  1  0.010949 0.18996 -191.86
## - QUIKRAT  1  0.014852 0.19387 -191.04
## <none>                 0.17902 -190.54
## + FATTOT   1  0.009059 0.16996 -188.93
## + INVTAST  1  0.007221 0.17179 -188.50
## + WCFTCL   1  0.005286 0.17373 -188.05
## + PAYOUT   1  0.002428 0.17659 -187.40
## + CAPINT   1  0.002034 0.17698 -187.31
## + GEARRAT  1  0.000077 0.17894 -186.87
```

```
## - CURRAT   1  0.054900 0.23392 -183.53
## - LOGSALE  1  0.077979 0.25699 -179.77
## - NFATAST  1  0.257575 0.43659 -158.57
## - WCFTDT   1  0.295955 0.47497 -155.20
##
## Step:  AIC=-191.86
## RETCAP ~ WCFTDT + LOGSALE + CURRAT + QUIKRAT + NFATAST
##
##           Df Sum of Sq     RSS     AIC
## - QUIKRAT  1  0.007807 0.19777 -193.94
## <none>                 0.18996 -191.86
## + INVTAST  1  0.011723 0.17824 -190.72
## + LOGASST  1  0.010949 0.17902 -190.54
## + FATTOT   1  0.004433 0.18553 -189.11
## + PAYOUT   1  0.001964 0.18800 -188.59
## + CAPINT   1  0.001419 0.18855 -188.47
## + WCFTCL   1  0.001076 0.18889 -188.40
## + GEARRAT  1  0.000207 0.18976 -188.21
## - CURRAT   1  0.045879 0.23584 -186.89
## - LOGSALE  1  0.088777 0.27874 -180.21
## - NFATAST  1  0.255392 0.44536 -161.47
## - WCFTDT   1  0.295532 0.48550 -158.01
##
## Step:  AIC=-193.94
## RETCAP ~ WCFTDT + LOGSALE + CURRAT + NFATAST
##
##           Df Sum of Sq     RSS     AIC
## <none>                 0.19777 -193.94
## + QUIKRAT  1  0.007807 0.18996 -191.86
## + FATTOT   1  0.007278 0.19049 -191.75
## + LOGASST  1  0.003904 0.19387 -191.04
## + PAYOUT   1  0.003363 0.19441 -190.93
## + CAPINT   1  0.002144 0.19563 -190.68
## + GEARRAT  1  0.001227 0.19654 -190.50
## + WCFTCL   1  0.001015 0.19676 -190.45
## + INVTAST  1  0.000566 0.19721 -190.36
## - LOGSALE  1  0.081329 0.27910 -183.85
## - CURRAT   1  0.153916 0.35169 -174.60
## - NFATAST  1  0.258443 0.45621 -164.19
## - WCFTDT   1  0.310297 0.50807 -159.88
```

```
summary(model_final_STEPWISE_BIC)
```

```
##
## Call:
## lm(formula = RETCAP ~ WCFTDT + LOGSALE + CURRAT + NFATAST, data = ukcomp_train)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.130709 -0.034368 -0.005593  0.029094  0.261002
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.02420    0.07971   0.304 0.763187
```

```
## WCFTDT       0.61188    0.08257   7.410 1.14e-08 ***
## LOGSALE      0.06096    0.01607   3.794 0.000564 ***
## CURRAT      -0.06895    0.01321  -5.219 8.27e-06 ***
## NFATAST     -0.47445    0.07015  -6.763 7.75e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07517 on 35 degrees of freedom
## Multiple R-squared:  0.7207, Adjusted R-squared:  0.6888
## F-statistic: 22.58 on 4 and 35 DF,  p-value: 2.749e-09
```

4 explanatory variables are selected.

IV.6) Lasso Method (+ cross validation)

I perform a cross validation method to select variables (based on the training sample), combined with the lasso method (by fixing alpha=1). As a regularization, Lasso method uses a penalized criterion λ to select the best compromise between model fitting and model complexity. For this task, I arbitrary set a gradient of 50 lambda values that will be generated, from 0 to a strongly penalizing value, and a default number of 10 folds.

```
library(glmnet)
```

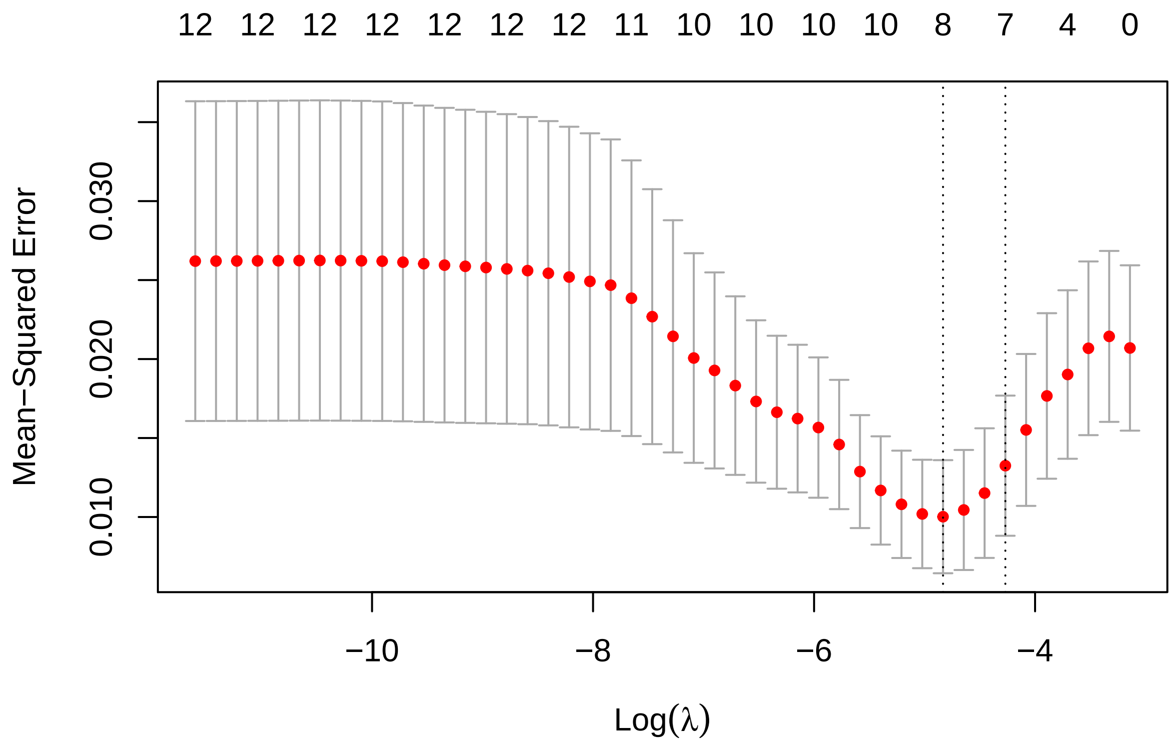
```
## Warning: package 'glmnet' was built under R version 4.0.5
```

```
## Loaded glmnet 4.1-2
```

```
model_cv = cv.glmnet(as.matrix(ukcomp_train[,-1]),
                     ukcomp_train[,1], family="gaussian",
                     nlambda=50, nfolds = 10, alpha=1)
model_cv #shows min and 1SE models
```

```
##
## Call:  cv.glmnet(x = as.matrix(ukcomp_train[, -1]), y = ukcomp_train[,      1], nfolds = 10, family =
##
## Measure: Mean-Squared Error
##
##       Lambda Index Measure       SE Nonzero
## min 0.007963    10 0.01002 0.003579       8
## 1se 0.013994     7 0.01325 0.004437       7
```

```
plot(model_cv)
```



The best model will be the one with the minimum Mean-Squared Error. As I am looking for a compromise between fitting and complexity, I will take the model for which Lambda is equal to the minimum value + 1 times standard error. I then do a new variable selection with Lasso method, with this value of Lambda+1SE, and will keep the selected variables:

```
model_LASSO = glmnet(as.matrix(ukcomp_train[,-1]),
                     as.matrix(ukcomp_train[,1]),family="gaussian",
                     alpha=1,lambda = model_cv$lambda.1se)
model_LASSO$beta   #estimated beta vector
```

```
## 12 x 1 sparse Matrix of class "dgCMatrix"
##                     s0
## GEARRAT  .
## CAPINT   0.018408211
## WCFTDT   .
## LOGSALE  0.038544414
## LOGASST  .
## CURRAT  -0.015600391
## QUIKRAT  .
## NFATAST -0.233644784
## INVTAST  .
## FATTOT  -0.021914001
## PAYOUT  -0.001732096
## WCFTCL   0.218632315
```

7 explanatory variables are selected.

I keep all explanatory variables given with a coefficient (all except those annotated ".") by the previous glmnet() function. I build a final model to obtain unbiased values for coefficients:

```
model_final_LASSO = lm(RETCAP ~ CAPINT + LOGSALE + CURRAT
                        + NFATAST + FATTOT + PAYOUT + WCFTCL, data=ukcomp_train)
summary(model_final_LASSO)
```

```
##
## Call:
## lm(formula = RETCAP ~ CAPINT + LOGSALE + CURRAT + NFATAST + FATTOT +
##     PAYOUT + WCFTCL, data = ukcomp_train)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.134012 -0.030532 -0.000928  0.034740  0.282365
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.06739    0.08510   0.792 0.434226
## CAPINT       0.02544    0.01767   1.440 0.159657
## LOGSALE      0.04261    0.01803   2.363 0.024384 *
## CURRAT      -0.04840    0.01195  -4.049 0.000305 ***
## NFATAST     -0.32985    0.12945  -2.548 0.015836 *
## FATTOT      -0.08916    0.08705  -1.024 0.313434
## PAYOUT      -0.01969    0.01732  -1.137 0.263942
## WCFTCL       0.41277    0.06104   6.762 1.22e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07823 on 32 degrees of freedom
## Multiple R-squared:  0.7235, Adjusted R-squared:  0.663
## F-statistic: 11.96 on 7 and 32 DF,  p-value: 2.198e-07
```

```
model_final_LASSO$coefficients
```

```
## (Intercept)      CAPINT     LOGSALE      CURRAT     NFATAST      FATTOT
##  0.06739121  0.02543606  0.04260988 -0.04839798 -0.32985456 -0.08915667
##      PAYOUT      WCFTCL
## -0.01969132  0.41276807
```

IV.7) Random Forest

```
## Warning: package 'randomForest' was built under R version 4.0.5
```

```
## randomForest 4.6-14
```

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

To build a model from a Random Forest process, I will use default parameters, with 500 trees generated and a number of explanatory variables equal to the square root of the total number of explanatory variables (as suggested for regression models).

```r
p = dim(ukcomp_train[-1])[2] #number of explanatory variables
RF = randomForest(RETCAP~.,data=ukcomp_train, mtry = sqrt(p), ntree = 500, importance = TRUE)
```

```r
RF
```

```
##
## Call:
##  randomForest(formula = RETCAP ~ ., data = ukcomp_train, mtry = sqrt(p),      ntree = 500, importanc
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##           Mean of squared residuals: 0.01477015
##                     % Var explained: 16.58
```

```r
names(RF)
```

```
##  [1] "call"            "type"            "predicted"       "mse"
##  [5] "rsq"             "oob.times"       "importance"      "importanceSD"
##  [9] "localImportance" "proximity"       "ntree"           "mtry"
## [13] "forest"          "coefs"           "y"               "test"
## [17] "inbag"           "terms"
```

```r
RF$importance
```

```
##               %IncMSE IncNodePurity
## GEARRAT -6.611040e-04    0.06106369
## CAPINT   8.662642e-04    0.05683230
## WCFTDT   2.982655e-03    0.10341605
## LOGSALE  5.150147e-04    0.03679733
## LOGASST -2.765134e-05    0.02632158
## CURRAT   2.275546e-04    0.04783633
## QUIKRAT  1.877951e-04    0.02784499
## NFATAST  1.233314e-03    0.06206257
## INVTAST  4.201508e-05    0.02709613
## FATTOT   3.184080e-04    0.05836417
## PAYOUT   5.400781e-04    0.04603080
## WCFTCL   2.684641e-03    0.08501713
```

The % of variable explanation is low. And the % of increase of Mean Squared Error for each variable permutation is also low. The model may likely not show good results on testing error.

V) Computing testing errors and selection of the best model

For each model, I compute:

- predicted values $\hat{Y}$ on testing sample
- residuals, as $\hat{Y} - Y$ vector
- testing error, as the mean of squared residual values

```r
# Testing error for selection using Bonferroni correction
pred_test_BONFERRONI <- predict(model_final_BONFERRONI,newdata = ukcomp_test)
residuals_test_BONFERRONI <- pred_test_BONFERRONI - ukcomp_test$RETCAP
test_error_BONFERRONI <- mean(residuals_test_BONFERRONI**2)

# Testing error for selection using Benjamini & Hocheberg correction: same as for Bonferroni

# Testing error for stepwise selection using Fisher criterion
pred_test_STEPWISE_F <- predict(model_final_STEPWISE_F,newdata = ukcomp_test)
residuals_test_STEPWISE_F <- pred_test_STEPWISE_F - ukcomp_test$RETCAP
test_error_STEPWISE_F <- mean(residuals_test_STEPWISE_F**2)

# Testing error for stepwise selection using AIC criterion
pred_test_STEPWISE_AIC <- predict(model_final_STEPWISE_AIC,newdata = ukcomp_test)
residuals_test_STEPWISE_AIC <- pred_test_STEPWISE_AIC - ukcomp_test$RETCAP
test_error_STEPWISE_AIC <- mean(residuals_test_STEPWISE_AIC**2)

# Testing error for stepwise selection using BIC criterion
pred_test_STEPWISE_BIC <- predict(model_final_STEPWISE_BIC,newdata = ukcomp_test)
residuals_test_STEPWISE_BIC <- pred_test_STEPWISE_BIC - ukcomp_test$RETCAP
test_error_STEPWISE_BIC <- mean(residuals_test_STEPWISE_BIC**2)

# Testing error for selection using Lasso method + cross validation
pred_test_LASSO <- predict(model_final_LASSO,newdata = ukcomp_test)
residuals_test_LASSO <- pred_test_LASSO - ukcomp_test$RETCAP
test_error_LASSO <- mean(residuals_test_LASSO**2)

# Testing error for selection using Random Forest
pred_test_RF <- predict(RF,newdata = ukcomp_test)
residuals_test_RF <- pred_test_RF - ukcomp_test$RETCAP
test_error_RF <- mean(residuals_test_RF**2)
```

I then compare all the testing error values:

```r
test_error_BONFERRONI
```

```
## [1] 0.01809331
```

```r
test_error_STEPWISE_F
```

```
## [1] 0.00577021
```

```r
test_error_STEPWISE_AIC
```

```
## [1] 0.00577021
```

```r
test_error_STEPWISE_BIC
```

```
## [1] 0.005759912
```

```
test_error_LASSO
```

```
## [1] 0.006747256
```

```
test_error_RF
```

```
## [1] 0.01137063
```

Minimum test error value is test_error_STEPWISE_BIC. I thus keep this model as the best one.

```
summary(model_final_STEPWISE_BIC)
```

```
##
## Call:
## lm(formula = RETCAP ~ WCFTDT + LOGSALE + CURRAT + NFATAST, data = ukcomp_train)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.130709 -0.034368 -0.005593  0.029094  0.261002
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.02420    0.07971   0.304 0.763187
## WCFTDT       0.61188    0.08257   7.410 1.14e-08 ***
## LOGSALE      0.06096    0.01607   3.794 0.000564 ***
## CURRAT      -0.06895    0.01321  -5.219 8.27e-06 ***
## NFATAST     -0.47445    0.07015  -6.763 7.75e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07517 on 35 degrees of freedom
## Multiple R-squared:  0.7207, Adjusted R-squared:  0.6888
## F-statistic: 22.58 on 4 and 35 DF,  p-value: 2.749e-09
```

```
model_final_STEPWISE_BIC$coefficients
```

```
## (Intercept)      WCFTDT     LOGSALE      CURRAT     NFATAST
##  0.02420409  0.61188470  0.06096181 -0.06894891 -0.47444843
```

In conclusion, I can assume that the main variables explaining the return on capital employed (RETCAP) are :

- WCFTDT : Ratio of working capital flow to total debt
- LOGSALE : log to base 10 of total sales
- CURRAT : current ratio
- NFATAST : Ratio of net fixed assets to total assets

The final linear model obtained to explain RETCAP will thus be:

$$RETCAP = 0.02420409 + 0.6118847.WCFTDT + 0.06096181.LOGSALE - 0.06894891.CURRAT - 0.47444843.NFATAST$$