Project 1

Filip Axelsson

2021-11-19

```
library(quantmod)
library(glmnet)
library(tidyverse)
library(knitr)
library(leaps)

### specify ticker names and download the data
tickers = c("^OMX", "ABB.ST", "NDA-SE.ST", "SKA-B.ST", "INVE-B.ST",
            "KINV-B.ST", "BOL.ST", "HEXA-B.ST", "SAND.ST", "ALFA.ST", "ATCO-B.ST",
            "TELIA.ST", "SCA-B.ST", "AZN.ST", "SWMA.ST", "GETI-B.ST","ERIC-B.ST","SWED-A.ST","SKF-B.ST"
getSymbols(tickers, src="yahoo", from = '2015-01-01', to = "2021-11-01")
```

```
##  [1] "^OMX"        "ABB.ST"      "NDA-SE.ST"   "SKA-B.ST"    "INVE-B.ST"
##  [6] "KINV-B.ST"   "BOL.ST"      "HEXA-B.ST"   "SAND.ST"     "ALFA.ST"
## [11] "ATCO-B.ST"   "TELIA.ST"    "SCA-B.ST"    "AZN.ST"      "SWMA.ST"
## [16] "GETI-B.ST"   "ERIC-B.ST"   "SWED-A.ST"   "SKF-B.ST"    "ELUX-B.ST"
## [21] "ALIV-SDB.ST"
```

```
###### get adjusted prices and save them in market_data
OMX_ad = OMX$OMX.Adjusted
ABB.ST_ad = ABB.ST$ABB.ST.Adjusted
BOL.ST_ad = BOL.ST$BOL.ST.Adjusted
SAND.ST_ad = SAND.ST$SAND.ST.Adjusted
ALFA.ST_ad = ALFA.ST$ALFA.ST.Adjusted
TELIA.ST_ad = TELIA.ST$TELIA.ST.Adjusted
AZN.ST_ad = AZN.ST$AZN.ST.Adjusted
SWMA.ST_ad = SWMA.ST$SWMA.ST.Adjusted

###### a  modification is needed in the case of Nordea because of '-' in the name
NDA_SE.ST = `NDA-SE.ST`
NDA_SE.ST_ad = NDA_SE.ST$`NDA-SE.ST.Adjusted`

INVE_B.ST = `INVE-B.ST`
INVE_B.ST_ad = INVE_B.ST$`INVE-B.ST.Adjusted`

KINV_B.ST = `KINV-B.ST`
KINV_B.ST_ad = KINV_B.ST$`KINV-B.ST.Adjusted`


HEXA_B.ST = `HEXA-B.ST`
HEXA_B.ST_ad = HEXA_B.ST$`HEXA-B.ST.Adjusted`
```

```
ATCO_B.ST = `ATCO-B.ST`
ATCO_B.ST_ad = ATCO_B.ST$`ATCO-B.ST.Adjusted`

SCA_B.ST = `SCA-B.ST`
SCA_B.ST_ad = SCA_B.ST$`SCA-B.ST.Adjusted`

GETI_B.ST = `GETI-B.ST`
GETI_B.ST_ad = GETI_B.ST$`GETI-B.ST.Adjusted`


SKA_B.ST = `SKA-B.ST`
SKA_B.ST_ad = SKA_B.ST$`SKA-B.ST.Adjusted`


ERIC_B.ST = `ERIC-B.ST`
ERIC_B.ST_ad = ERIC_B.ST$`ERIC-B.ST.Adjusted`

SWED_A.ST = `SWED-A.ST`
SWED_A.ST_ad = SWED_A.ST$`SWED-A.ST.Adjusted`

SKF_B.ST = `SKF-B.ST`
SKF_B.ST_ad = SKF_B.ST$`SKF-B.ST.Adjusted`

ELUX_B.ST = `ELUX-B.ST`
ELUX_B.ST_ad = ELUX_B.ST$`ELUX-B.ST.Adjusted`



market_data=cbind(OMX_ad,ABB.ST_ad,NDA_SE.ST_ad,GETI_B.ST_ad,SKA_B.ST_ad,
                  BOL.ST_ad,SAND.ST_ad,ALFA.ST_ad,TELIA.ST_ad,AZN.ST_ad,
                  SWMA.ST_ad,INVE_B.ST_ad,
                  KINV_B.ST_ad,HEXA_B.ST_ad,ATCO_B.ST_ad,SCA_B.ST_ad)




##### remove NA
market_data=na.omit(market_data)

##### compute log-returns and save them as data frame called "returns",
#### whose colums are renamed, respectively.

k=nrow(market_data)
returns=as.data.frame(
  log(as.matrix(market_data[2:k,]))-log(as.matrix(market_data[1:(k-1),]))
  )
names(returns)=c("rOMX","rABB","rNDA.SE.ST","rGETI_B.ST_ad","rSKA_B.ST_ad",
                 "rBOL.ST_ad","rSAND.ST_ad","rALFA.ST_ad","rTELIA.ST_ad",
                 "rAZN.ST_ad","rSWMA.ST_ad","rINVE_B.ST_ad","rKINV_B.ST_ad",
                 "rHEXA_B.ST_ad","rATCO_B.ST_ad","rSCA_B.ST_ad")
```

Introduction

In this project we will use real financial data from the Swedish capital market including stock prices and capital market index to preform two differen tasks, one including linear regression models and the other linear classification with servel questions that will be presented futrher into the project. The data is collected from Yahoo Finance, we are also looking in the timeperiod from 2015-01-01 to 2021-11-01. We have picked 15 different stocks and the index OMX30, an overwiev of the stocks is displayed in *Table 1* where also respective stocks GICS sector is listed, that is "Global Industry Classification Standard".

Table 1: Overwiev of each stock's GICS sector

Stock	GICS Sector
OMX	Market index
ABB	Industry
Nordea	Bank
Getinge	(Medical) Technology
Skanska B	Construction
Boliden	Industry
Sandviken	Industry
Alfa Laval	Industry
Telia	Telecommunication
AstraZeneca	Pharmaceuticals
Swedish Match	Tobaco
Invester B	Investment/Holding company
Kinnevik	Investment/Holding company
Hexagon	Technology
Atlas Copco B	Industry
SCA	Forestry

Before we can proceed to complete the tasks, we need to restructure our data by computeing log-returns for each stock as well for the index. The log-returns is defined as follwing:

$$r_t = \ln(P_t) - \ln(P_{t-1})$$

where P_t denote the price/index value on day t and r_t is the log-return on day t, a example of the transformed data is visualized in *Table 2* below.

Table 2: Overwiev of 5 first observations and 5 first stocks/index in the transformed dataset

	rOMX	rABB	rNDA.SE.ST	rGETI_B.ST_ad	rSKA_B.ST_ad
2015-01-05	-0.0061468	-0.0096852	-0.0199562	-0.0057770	0.0023753
2015-01-07	-0.0232753	-0.0327694	-0.0260934	-0.0087285	-0.0047563
2015-01-08	0.0245872	0.0242172	0.0199152	0.0236801	0.0276243
2015-01-09	-0.0084518	-0.0148333	-0.0119016	-0.0011422	-0.0069806
2015-01-12	0.0063352	0.0031083	0.0045507	-0.0342959	0.0150639

Linear regression model

a)

The return of the market index (OMX30) is determined as a linear combination of the returns on the stocks which is included in the index. A economical hypotheses states that there exist only a few stocks on a capital market such that a portfolio constructed by using these stocks will then duplicate the behaviour of the capital market index. One of our aims will be to determine the portfolio based on the stocks included in the OMX30 that can mimic its behavior.
We start out by fitting a linear regression model, which can be written on matrix form as following:

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon},$$

where $\boldsymbol{y}$ corresponds to a vector of response variables, in this case OMX30 is our response vector. The observed values for each stock is given by the design matrix $\boldsymbol{X}$, which is a $n \times (p+1)$ matrix. The errors in the model is given by the vector $\boldsymbol{\varepsilon}$. We also have a paramter vector which correspond to $\boldsymbol{\beta}$ and is considered unknown. However by using the Least Square Estimator, which minimizes RSS (Residual Sum of Squares) we can estimate β. The Least Square Estimator is defined as follwing:

$$\hat{\boldsymbol{\beta}} = argmin_{\beta}||\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}||^2 = argmin_{\beta} \sum_{i=1}^{N}(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij})$$

From this we get the follwing solution:

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

By using the infromation that is written above we get that a linear regression model where all the 15 stocks is used as explantory variables genereates the model which is presented in *Table 3*. We can clearly see that all the estimated parameters are significant, that is we reject that the parameters are non-zero. We also got that the determination of coefficient, $R^2 = 0.9535$ which means that our model explain approximately 95% of the variation. If we consider the amount of parameters in the model we get the adjusted determination of coefficient, which was measured to $R^2_{adj} = 0.9531$. By using this information, mostly the information about that all the parameters is significant we can determine that all of the choosen stocks has an influence on the return of the capital market index, OMX30.

```
# ## multipel linjär regression
full_model <- lm(rOMX ~. , data = returns)


kable(
  summary(full_model)[["coefficients"]],
  caption = "Summary for the linear regression"
)
```

Table 3: Summary for the linear regression

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-0.0002287	0.0000620	-3.690383	0.000231
rABB	0.0568470	0.0070110	8.108287	0.000000
rNDA.SE.ST	0.1476782	0.0049303	29.952959	0.000000
rGETI_B.ST_ad	0.0182775	0.0032680	5.592786	0.000000
rSKA_B.ST_ad	0.0406852	0.0052268	7.783936	0.000000
rBOL.ST_ad	0.0279311	0.0035478	7.872835	0.000000
rSAND.ST_ad	0.0665541	0.0060340	11.029867	0.000000
rALFA.ST_ad	0.0418583	0.0050699	8.256310	0.000000

	Estimate	Std. Error	t value	Pr(>|t|)
rTELIA.ST_ad	0.1027939	0.0061030	16.843079	0.000000
rAZN.ST_ad	0.0362867	0.0046040	7.881635	0.000000
rSWMA.ST_ad	0.0229792	0.0038122	6.027785	0.000000
rINVE_B.ST_ad	0.1923377	0.0095220	20.199372	0.000000
rKINV_B.ST_ad	0.0116183	0.0034883	3.330676	0.000885
rHEXA_B.ST_ad	0.0401135	0.0048292	8.306484	0.000000
rATCO_B.ST_ad	0.1060364	0.0061119	17.349261	0.000000
rSCA_B.ST_ad	0.0411987	0.0045421	9.070470	0.000000

b)

This result can however not be used to anwser the question which of the stocks have to be included in the portfolio to mimic the behavior of the Swedish capital market index OMX30. That is becuase we haven't taken into consider if there is any multicolinearity, that is if there is any realtion between the stocks. If one stock goes up that could correlate that another stock went up. For example, both Kinnevik and Investor is holding companies, that is they invest in other companies. Both these companies have stocks in the other companies that is included in the market index and is used as explantory variables in this model. This information could then be used to make the conclusion that if one of those goes up it's a high possiblity that the holding companies also goes up, therefore we can not say anything about which stocks have to be used to mimic the index by only using the result above. We need to investigate further to be able to determine that, and we also need to investigate if all these 15 variables is necessary to be included in the model, that is we want to see if we can remove some of the variables and still get a good result. Note that becuase all the 15 variables is significant we can use this model to mimic the capital market index however it may not be the best model.

c)

By performing forward-stepwise selection to the regression we can determine if we can eliminate any of the parameters in the model. To be able do this selection we start with the intercept and sequentially add into the model the predictor that most improves the fit. By prefomring this we still get that all 15 parameters should be used in the model, and this leads to the results that all 15 stocks should be included in the portfolio in order to mimic the market capital index. The process is displayed below.

```
# Forward-stepwise selection
 summary(step(full_model, direction = "forward", trace = F))
```

```
##
## Call:
## lm(formula = rOMX ~ rABB + rNDA.SE.ST + rGETI_B.ST_ad + rSKA_B.ST_ad +
##     rBOL.ST_ad + rSAND.ST_ad + rALFA.ST_ad + rTELIA.ST_ad + rAZN.ST_ad +
##     rSWMA.ST_ad + rINVE_B.ST_ad + rKINV_B.ST_ad + rHEXA_B.ST_ad +
##     rATCO_B.ST_ad + rSCA_B.ST_ad, data = returns)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0136946 -0.0013681  0.0000176  0.0014192  0.0113376
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -2.287e-04  6.196e-05  -3.690 0.000231 ***
## rABB           5.685e-02  7.011e-03   8.108 9.74e-16 ***
## rNDA.SE.ST     1.477e-01  4.930e-03  29.953  < 2e-16 ***
```

```
## rGETI_B.ST_ad  1.828e-02  3.268e-03   5.593 2.60e-08 ***
## rSKA_B.ST_ad   4.069e-02  5.227e-03   7.784 1.21e-14 ***
## rBOL.ST_ad     2.793e-02  3.548e-03   7.873 6.14e-15 ***
## rSAND.ST_ad    6.655e-02  6.034e-03  11.030  < 2e-16 ***
## rALFA.ST_ad    4.186e-02  5.070e-03   8.256 2.99e-16 ***
## rTELIA.ST_ad   1.028e-01  6.103e-03  16.843  < 2e-16 ***
## rAZN.ST_ad     3.629e-02  4.604e-03   7.882 5.73e-15 ***
## rSWMA.ST_ad    2.298e-02  3.812e-03   6.028 2.04e-09 ***
## rINVE_B.ST_ad  1.923e-01  9.522e-03  20.199  < 2e-16 ***
## rKINV_B.ST_ad  1.162e-02  3.488e-03   3.331 0.000885 ***
## rHEXA_B.ST_ad  4.011e-02  4.829e-03   8.306  < 2e-16 ***
## rATCO_B.ST_ad  1.060e-01  6.112e-03  17.349  < 2e-16 ***
## rSCA_B.ST_ad   4.120e-02  4.542e-03   9.070  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.00255 on 1695 degrees of freedom
## Multiple R-squared:  0.9535, Adjusted R-squared:  0.9531
## F-statistic:  2317 on 15 and 1695 DF,  p-value: < 2.2e-16
```

```
## Another function that makes the same, just wanted to dubbelcheck ##
# regfit.fwd=regsubsets(rOMX~.,data=returns,nvmax=16, method="forward")
# summary(regfit.fwd)
```

d)

Another way to prefomre variable selection is by doing the backward-stepwise selction, which we start with the full model and sequantially delet the predictor that has the least impact on the fit. This process resulted in the same model as before, that is we should use all 15 stocks to build a portfolio that mimics the market capital index. The process is displayed below

```
# # step utför variabel selektion

summary(step(full_model, direction = "backward", trace = F))
```

```
##
## Call:
## lm(formula = rOMX ~ rABB + rNDA.SE.ST + rGETI_B.ST_ad + rSKA_B.ST_ad +
##     rBOL.ST_ad + rSAND.ST_ad + rALFA.ST_ad + rTELIA.ST_ad + rAZN.ST_ad +
##     rSWMA.ST_ad + rINVE_B.ST_ad + rKINV_B.ST_ad + rHEXA_B.ST_ad +
##     rATCO_B.ST_ad + rSCA_B.ST_ad, data = returns)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0136946 -0.0013681  0.0000176  0.0014192  0.0113376
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -2.287e-04  6.196e-05  -3.690 0.000231 ***
## rABB           5.685e-02  7.011e-03   8.108 9.74e-16 ***
## rNDA.SE.ST     1.477e-01  4.930e-03  29.953  < 2e-16 ***
## rGETI_B.ST_ad  1.828e-02  3.268e-03   5.593 2.60e-08 ***
## rSKA_B.ST_ad   4.069e-02  5.227e-03   7.784 1.21e-14 ***
## rBOL.ST_ad     2.793e-02  3.548e-03   7.873 6.14e-15 ***
## rSAND.ST_ad    6.655e-02  6.034e-03  11.030  < 2e-16 ***
```

```
## rALFA.ST_ad    4.186e-02  5.070e-03   8.256 2.99e-16 ***
## rTELIA.ST_ad   1.028e-01  6.103e-03  16.843  < 2e-16 ***
## rAZN.ST_ad     3.629e-02  4.604e-03   7.882 5.73e-15 ***
## rSWMA.ST_ad    2.298e-02  3.812e-03   6.028 2.04e-09 ***
## rINVE_B.ST_ad  1.923e-01  9.522e-03  20.199  < 2e-16 ***
## rKINV_B.ST_ad  1.162e-02  3.488e-03   3.331 0.000885 ***
## rHEXA_B.ST_ad  4.011e-02  4.829e-03   8.306  < 2e-16 ***
## rATCO_B.ST_ad  1.060e-01  6.112e-03  17.349  < 2e-16 ***
## rSCA_B.ST_ad   4.120e-02  4.542e-03   9.070  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.00255 on 1695 degrees of freedom
## Multiple R-squared:  0.9535, Adjusted R-squared:  0.9531
## F-statistic:  2317 on 15 and 1695 DF,  p-value: < 2.2e-16
```

```
## Another function that makes the same, just wanted to dubbelcheck ##
# regfit.bwd=regsubsets(rOMX~.,data=returns,nvmax=16, method="backward")
# summary(regfit.bwd)
```

e)
We would now like to fit the ridge regression to the data, and determine the regularization parameter λ by using leave-one-out cross-validation. Rigde regression allow for some bias while reducing the variance. We define the rigde regression estimator by minimising:

$$||\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}||^2 + \lambda||\boldsymbol{\beta}||^2,$$

where λ is the regularizationparameter and $||\boldsymbol{\beta}||_2^2 = \sum_{i=1}^{p} \beta_i^2$ which is also referd as quadratic or l_2 penalty (norm) The minimiser $\hat{\boldsymbol{\beta}}_r$ exist for any $\lambda > 0$:

$$\hat{\boldsymbol{\beta}}_r = (\boldsymbol{X}^T\boldsymbol{X} + \lambda l_2)^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

In *Figure 1* we can see that with the optimal λ, which was choosed by using leave-one-out cross-validation we obtain that all 15 stocks is not near zero, that is that all 15 stocks should be included in the portfolio to mimic OMX30. Note that ridge regressions is that it enforces the β coefficients to be lower (shrinks them), but it will not set them to zero. That is, the ridge regression will not remove any irrelevant parameters, but instead miminize their impact. This makes it possible to see which parameters has an impact on the index but not deterimine if we should remove any of them from the result. An idea to structure the portfolio is to take the infromation from how big the coeffiecents are relative to each other, and invest from that. With other words that is to have the most of the money in the stock with highest/biggest coefficient and the least money in the stock with the lowest coefficient, since ridge regression penalizes the variables/stocks that have least impact on the index.

```
## Ridge regression

x <- model.matrix(rOMX~., data = returns)[,-1]
y <- as.vector(returns$rOMX)

ridge.mod <- glmnet(x,y,alpha=0)


# Cross validation
cv_ridge <-cv.glmnet(x,y,alpha=0)
best_lambda_r <- cv_ridge[["lambda.min"]]

plot(ridge.mod, xvar = "lambda")
abline(v=log(best_lambda_r), h = 0)
```

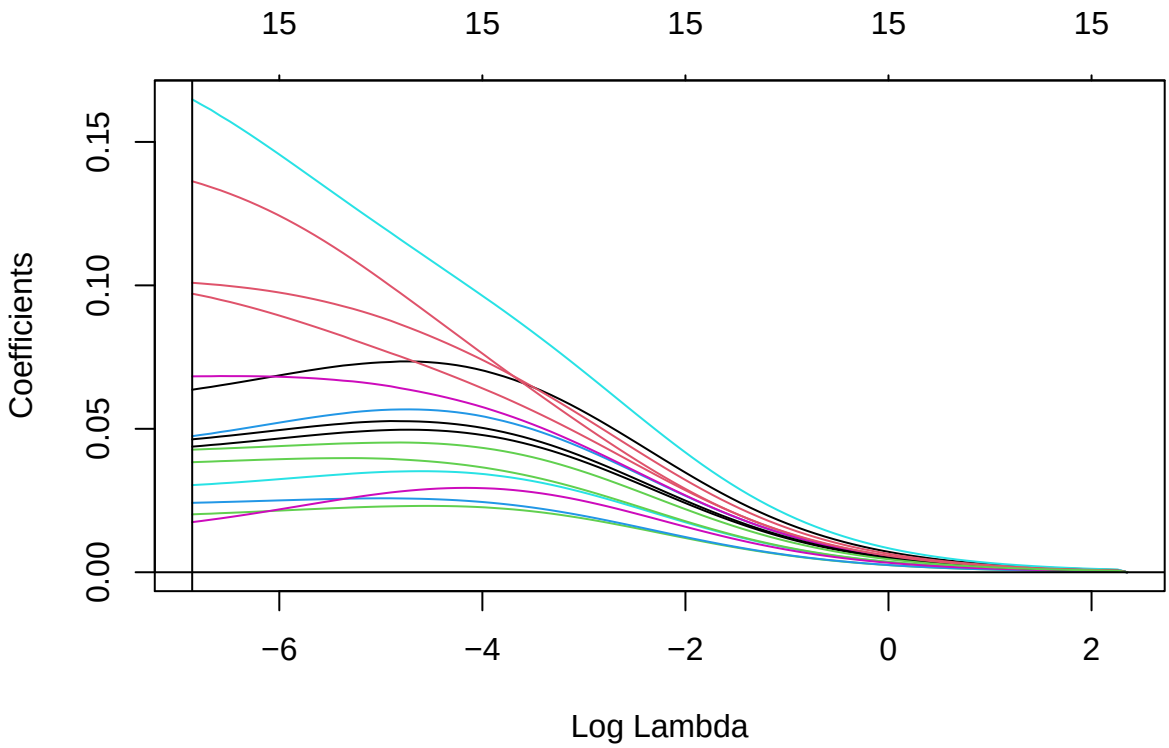


Figure 1: Ridge regression: Estimated coefficients as a function of λ.

```
summary(coef(glmnet(x,y,alpha = 0, lambda = best_lambda_r)))
```

```
## 16 x 1 sparse Matrix of class "dgCMatrix", with 16 entries
##     i j             x
## 1   1 1 -0.0002207504
## 2   2 1  0.0640768057
## 3   3 1  0.1364853593
## 4   4 1  0.0201963864
## 5   5 1  0.0474219754
## 6   6 1  0.0303218771
## 7   7 1  0.0677712765
## 8   8 1  0.0460222793
## 9   9 1  0.1005932619
## 10 10 1  0.0381790965
## 11 11 1  0.0241092492
## 12 12 1  0.1646822303
## 13 13 1  0.0174529994
## 14 14 1  0.0438211959
## 15 15 1  0.0975595640
## 16 16 1  0.0428970299
```

f)
Instead of using ridge regression we now want to use lasso regression, that is "Least Absolute Shrinkage and Selection Operator" regression. It is similiar to ridge regression, but we now replacing the l_2 constraint by an l_1 constraint. The lasso estimator is defined as a minimiser of the following:

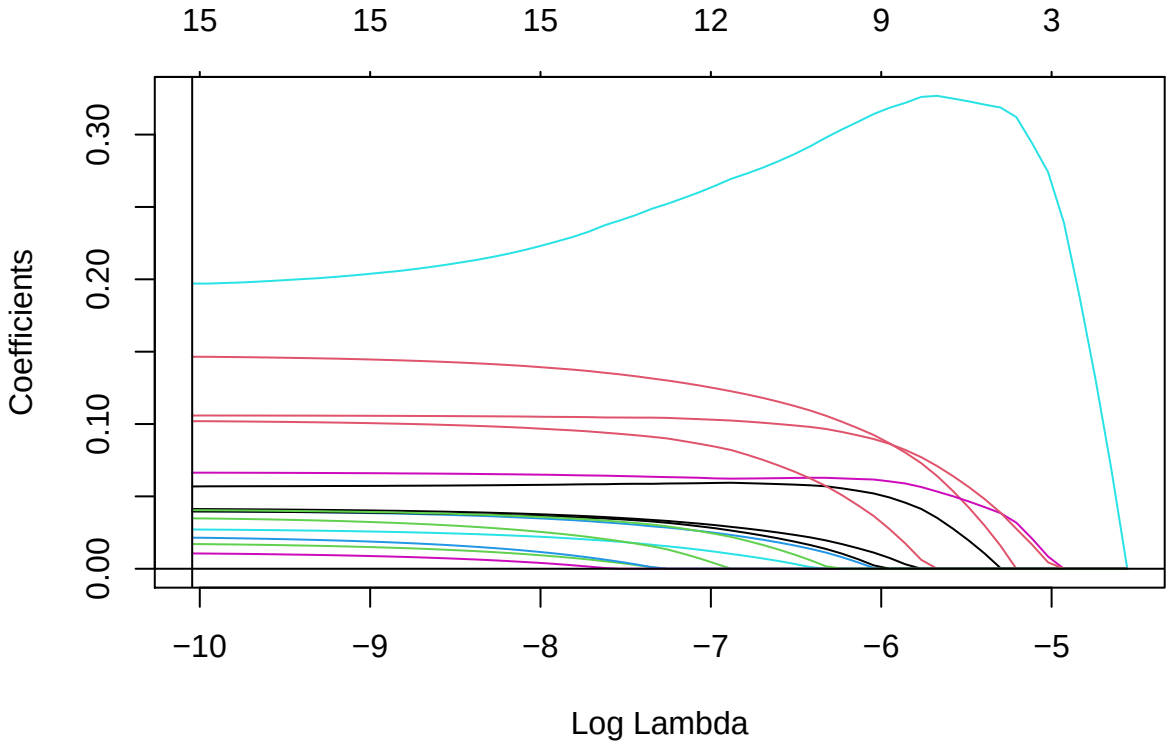
$$||\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}||^2 + \lambda||\boldsymbol{\beta}||_{\mathbf{1}},$$

again we have that λ is the regularization parameter. The parameter $||\boldsymbol{\beta}||_1 = \sum_{i=1}^{p} |\beta_i|$ is l_2 penalty (norm). The minimiser $\hat{\beta}_l$ exists for any $\lambda > 0$ but has a closed form only in some very trivial special cases. Unlike ridge regression, this method overcomes the disadvantage by not only penalize high values of the coefficents β but actually setting them to zero if they aren't relevant. That is it preforme a type of variable selection, which result that fewer variables will be included in the model we started with. In this case we didn't remove any variables when we use the optimal λ which was obtain by leave-one-out cross-validation. The result is therefore the same as the previous regressions, all 15 stocks has an impact on the index. Since all 15 stocks has an impact we could use that as a portfolio to mimic the index, however we can imply the same idea as we did for ridge regression to structure our portfolio. That is to invest the most in the stock with the highest coefficient and the least in the stock with the lowest coefficient since also this lasso regression penalizes the variables with the least impact on the index.

```
## Lasso regression
lasso.mod=glmnet(x,y,alpha=1)

# Cross validation
cv_lasso <- cv.glmnet(x,y,alpha=1)
best_lambda_l <- cv_lasso[["lambda.min"]]

# Plotting lasso regeression
plot(lasso.mod, xvar= "lambda")
abline(v = log(best_lambda_l), h = 0)
```



g)

In *Table 4* we can see a summarize of the estimated coefficient for each type of regression, remember with variable selection linear regression we still obtainded a model with 15 variables. In every type of regression we got the result that all 15 stock should be included in the portfolio, every regression also genterated approximately the same estimated coefficients.

```
kable(
as.matrix(cbind(coef(full_model),coef(glmnet(x,y,alpha = 0, lambda = best_lambda_r)),coef(glmnet(x,y,al
col.names = c("Linear regression", "Ridge regression", "Lasso regression"),
caption = " The result for each regression"
```

```
)
```

Table 4: The result for each regression

	Linear regression	Ridge regression	Lasso regression
(Intercept)	-0.0002287	-0.0002208	-0.0002259
rABB	0.0568470	0.0640768	0.0566260
rNDA.SE.ST	0.1476782	0.1364854	0.1463863
rGETI_B.ST_ad	0.0182775	0.0201964	0.0170549
rSKA_B.ST_ad	0.0406852	0.0474220	0.0398072
rBOL.ST_ad	0.0279311	0.0303219	0.0271413
rSAND.ST_ad	0.0665541	0.0677713	0.0666877
rALFA.ST_ad	0.0418583	0.0460223	0.0414921
rTELIA.ST_ad	0.1027939	0.1005933	0.1021927
rAZN.ST_ad	0.0362867	0.0381791	0.0349771
rSWMA.ST_ad	0.0229792	0.0241092	0.0215442
rINVE_B.ST_ad	0.1923377	0.1646822	0.1966666
rKINV_B.ST_ad	0.0116183	0.0174530	0.0106522
rHEXA_B.ST_ad	0.0401135	0.0438212	0.0396939
rATCO_B.ST_ad	0.1060364	0.0975596	0.1056130
rSCA_B.ST_ad	0.0411987	0.0428970	0.0404167

Linear classification

One important task of traders on the capital market is to determine proper times to open position, that is to buy assets and to close postion, that is to sell assets for each stock. This makes it important to open the position when the price of a stock starts to go up and close the position when the price of a stock drops down. Therefore we will use the logistic regression model, where we aim to check wheter it is advantegeous to use a part och stocks determined by the Lassa regression in comparison to all 15 stocks in order to describe the movement of the Swedish captial market index. This is the reason we use 80% of the data to fit two logistic regression models and the other 20% of the data is used for the prediction purposes. THe dependet variable in both regressions is defined by:

$$G_t = \begin{cases} 1, & \text{market return } \leq 0 \text{ on day t, i.e., the market index goes up on day t.} \\ 0, & \text{market return } < 0 \text{ on day t, i.e., the market index goes down on day t.} \end{cases}$$

```
new_returns <- returns %>%
  mutate(G_t = ifelse(rOMX>0,1,0)) # creating the variable G_t




set.seed(123) # Making so the datasplit doesnt change
data <- new_returns[,-1]
index <- sample(1:nrow(data), round(0.80 * nrow(data))) # Splitting the data up to 80/20 train, test
train <- data[index, ]
test <- data[-index, ]
```

a)
We start out with fitting the two logistic regression models to the first 80% of returns data, where the first one uses all 15 stock returns as predictors while the second model only uses those stocks which are determined by the Lasso regression but since we came to the conclusion that all 15 stocks should be included

we therefore only going to use 10 stocks instead. In *Table 5* we can analyze that all stocks is significant execpt Swedish Match ($rSWMA.ST_ad$) and Kinnevik ($rKINV_B.ST_ad$), we also obtainde an $AIC = 490.51$ which is value which makes it possible to compare different models, the lower AIC the better is the model. When preforming logistic regression when only using 10 stocks, we obtain the result which is displayed in *Table 6*, there we can see that all the stocks are significant execpt Swedish Match ($rSWMA.ST_ad$), this model obtained $AIC = 671.81$, which is higher then the model with all 15 stocks. Since AIC is a way to determine which model is better fitted we can now start suspect that the model with all 15 stocks is best fitted for this regression. In both models we see that all the stocks is positive, which suggest that if the stocks goes up, then the index will likely go up. However since we have 2 stocks that have a relative high p-value there is no clear evidence of real association between those stocks and the index.

```
# 15 stocks
glm.fits <- glm(G_t~., data=train,family=binomial)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
kable(
  summary(glm.fits)[["coefficients"]],
  caption = "Logistic regression for all 15 stocks"
)
```

Table 5: Logistic regression for all 15 stocks

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-0.2224473	0.1223908	-1.8175168	0.0691380
rABB	33.6369789	15.6414709	2.1504997	0.0315157
rNDA.SE.ST	133.9021976	14.2948669	9.3671525	0.0000000
rGETI_B.ST_ad	16.6688215	7.2948188	2.2850220	0.0223115
rSKA_B.ST_ad	31.8189283	11.3826435	2.7953900	0.0051837
rBOL.ST_ad	35.7411999	8.5705400	4.1702390	0.0000304
rSAND.ST_ad	42.6415856	13.0600055	3.2650511	0.0010944
rALFA.ST_ad	48.0609529	11.6801118	4.1147682	0.0000388
rTELIA.ST_ad	52.2265449	14.5571405	3.5876926	0.0003336
rAZN.ST_ad	43.8884054	9.9241278	4.4223942	0.0000098
rSWMA.ST_ad	2.8577954	4.6610501	0.6131227	0.5397952
rINVE_B.ST_ad	183.7947020	25.4015177	7.2355795	0.0000000
rKINV_B.ST_ad	11.8370295	10.7776477	1.0982943	0.2720760
rHEXA_B.ST_ad	26.2038865	10.2561068	2.5549545	0.0106202
rATCO_B.ST_ad	113.0343043	16.3204824	6.9259168	0.0000000
rSCA_B.ST_ad	25.2176785	9.6316480	2.6182101	0.0088392

```
# 10 stocks, rABB,rNDA.SE.ST,rGETI_B.ST_ad, rSKA_B.ST_ad, rBOL.ST_ad,
# rSAND.ST_ad, rALFA.ST_ad, rTELIA.ST_ad, rAZN.ST_ad, rSWMA.ST_ad
glm.fits_2=glm(G_t~.-rINVE_B.ST_ad-rKINV_B.ST_ad-rHEXA_B.ST_ad-rATCO_B.ST_ad-rSCA_B.ST_ad, data=train,fa
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
kable(
  summary(glm.fits_2)[["coefficients"]],
  caption = "Logistic regression for 10 stocks"
)
```

Table 6: Logistic regression for 10 stocks

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-0.072698	0.1005375	-0.7230934	0.4696225
rABB	63.222058	13.8125298	4.5771527	0.0000047
rNDA.SE.ST	110.624520	11.1404557	9.9299815	0.0000000
rGETI_B.ST_ad	37.205602	6.6054925	5.6325251	0.0000000
rSKA_B.ST_ad	42.100114	9.3441246	4.5055172	0.0000066
rBOL.ST_ad	35.397577	7.1771631	4.9319733	0.0000008
rSAND.ST_ad	86.770413	10.4842278	8.2762807	0.0000000
rALFA.ST_ad	60.024292	9.7461053	6.1587978	0.0000000
rTELIA.ST_ad	72.349180	12.6543979	5.7173151	0.0000000
rAZN.ST_ad	40.504044	8.6331218	4.6917031	0.0000027
rSWMA.ST_ad	8.896310	6.7805186	1.3120398	0.1895067

b)
We would now use the remaning 20% of the the return data, to determine the error rate by using the two logistic regressions above. That is we want to compute the proportion of cases with incorrect classifications to the total number of considered cases for each model. Note that 1 means the index goes up, and 0 means the index goes down. To be albe to make a prediction as to wheter the index goes up or down depending on the stocks movement, we most convert these into two class labels 0 (down) or 1 (up). We then create a vector of class predictors based on whether the predicted probability of the index increases is greater than or less than 0.5. Given this predictors we can then display the table below, which can be used to determine how many observations were correctly or incorrectly classified. The table gives us that 315 of 342 were correctly classified when using all 15 stocks, which corresponds to 92% correclty predicted movement of the index given the log-return of the stocks. The test error rate is therefore approximately 7.9%. The regression predicted 17 of 193 observations for the index goes up wrong, that is a error rate of 8.8%. The regression also classifed 10 of 149 "down" wrong, which corresponds to 6.7%.

```
# For all 15 stocks
glm.probs <- predict(glm.fits, test, type = "response")


glm.pred <- rep("0",342)
glm.pred[glm.probs>.5]="1"
Direction_test <- test$G_t
table(glm.pred,Direction_test)
```

```
##         Direction_test
## glm.pred   0   1
##        0 152  14
##        1   6 170
```

```
mean(glm.pred==Direction_test)
```

```
## [1] 0.9415205
```

```
mean(glm.pred!=Direction_test) # Test error rate
```

```
## [1] 0.05847953
```

We would now want to do the same thing but for the model with only 10 stocks. We see in the table below that 308 of 342 observations was correctly classified, that is a 90% correclty predicted movement of the index given the log-return of the stocks. The test error rate for this regression resulted in therefore approximately 9.9%. This regression classified 23 of 198 "up" wrong, which is 11.6% and 11 of 144 "down" wrong which is 7.6%.

```r
# For 10 stocks
glm.probs_2 <- predict(glm.fits_2, test, type = "response")

glm.pred_2 <- rep("0",342)
glm.pred_2[glm.probs_2>.5]="1"
Direction_test_2 <- test$G_t
table(glm.pred_2,Direction_test_2)
```

```
##           Direction_test_2
## glm.pred_2   0   1
##          0 152  23
##          1   6 161
```

```r
mean(glm.pred_2==Direction_test_2)
```

```
## [1] 0.9152047
```

```r
mean(glm.pred_2!=Direction_test_2) # Test error rate
```

```
## [1] 0.08479532
```

e)

The conclusion that can be made by using the result above is that the logistic model with all 15 stocks genereated the lowest error rate, both overall, and for each group ("up" or "down"), therefore that model should be used to determine proper times to open positions and to close postions.