Analysis

Willie Langenberg

11/23/2021

Task 1

a)

We start by fitting a linear regression model to the data, including all predictors (explanatory variables). From the summary of the model below, we can conclude that every predictor except the intercept are positive and every estimate except one are significant. Almost all p-values are below 0.01 which implies strong evidence against the null hypothesis of no relationship between the individual predictors and the response variable. The SINCH stock have a p-value at 0.56, meaning it exceed any critical limit, thus we would probably want to exclude it from the model. For the other predictors we are better off keeping them in the model than excluding them, so they evidently have an relation to the return of the capital market. However, I am not so sure about the causality between the stocks and the index (OMX). So I am a bit careful about expressing how they influence the OMX. But according to the estimated coeffcients Investor(INVE_B), Atlas Copco (ATCO_A) and SEB(SEB_A) seem to have a somewhat stronger relationship with the response variable. The estimated coefficients is how much the swedish capital market's index is supposed to be in relation to the stocks. So if Atlas Copco have a log return of 1, and all other are zero, we model the log return of OMX to be 0.14 (-intercept).

```
# Fit a linear regression model to all data using all variables.
model <- lm(rOMX ~ ., data = returns)
summary(model)
```

```
##
## Call:
## lm(formula = rOMX ~ ., data = returns)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0124460 -0.0011346  0.0000013  0.0010945  0.0100296
##
## Coefficients:
##               Estimate  Std. Error t value             Pr(>|t|)
## (Intercept) -0.00013658  0.00005483  -2.491              0.01286 *
## rABB         0.06014341  0.00625945   9.608 < 0.0000000000000002 ***
## rNDA_SE      0.10269915  0.00499088  20.577 < 0.0000000000000002 ***
## rTELIA       0.07828631  0.00529661  14.780 < 0.0000000000000002 ***
## rHM_B        0.05993705  0.00303379  19.756 < 0.0000000000000002 ***
## rINVE_B      0.13567375  0.00867218  15.645 < 0.0000000000000002 ***
## rSEB_A       0.08794174  0.00532706  16.508 < 0.0000000000000002 ***
## rSINCH       0.00113976  0.00197162   0.578              0.56331
## rASSA_B      0.08976028  0.00476003  18.857 < 0.0000000000000002 ***
```

```
## rERIC_B      0.05789388  0.00301034  19.232 < 0.0000000000000002 ***
## rAZN         0.02900010  0.00398028   7.286    0.000000000000554 ***
## rSCA_B       0.03496756  0.00410166   8.525 < 0.0000000000000002 ***
## rALFA        0.04167989  0.00432282   9.642 < 0.0000000000000002 ***
## rATCO_A      0.14160054  0.00487397  29.052 < 0.0000000000000002 ***
## rKINV_B      0.00766358  0.00294373   2.603              0.00934 **
## rBOL         0.03054243  0.00321487   9.500 < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.001968 on 1293 degrees of freedom
## Multiple R-squared:  0.9733, Adjusted R-squared:  0.973
## F-statistic:  3146 on 15 and 1293 DF,  p-value: < 0.00000000000000022
```

b)

Obviously the answer is dependent on how accurately we want to mimic the Swedish capital market index. Even though every predictor except SINCH showed a significant influence on the capital market returns, we can not really conclude that they all *have* to be included in the model. Often we want to simplify models if we can, without compromising too much performance. We can not say that the predictors with the highest estimated coefficient need to be included in the model either. We need to do some further analysis. For example we need to check the model for colinearity and other model assumptions. So we can not decide which variables we have to include solely based on the results of *a)*.

c)

We run the forward-stepwise selection method below. We use the BIC criterion to select how many parameters to include.

```
# Finding best amount of variables according to BIC and forward selection.
forwardSelectModels <- regsubsets(rOMX ~., data=returns, method = "forward", nvmax=15)
print(c("Forward Select Models, Lowest BIC:", which.min(summary(forwardSelectModels)$bic)))
```

```
## [1] "Forward Select Models, Lowest BIC:" "13"
```

According to this method we recieved the lowest BIC at 13 variables included. This model excluded SINCH and KINV. So the cost of adding these parameters is greater than their added contribution, according to the BIC criterion.

d)

Now we use the backward-stepwise selection method.

```
# Finding best amount of variables according to BIC and backward selection.
backwardSelectModels <- regsubsets(rOMX ~., data=returns, method = "backward", nvmax=15)
print(c("Backward Select Models, Lowest BIC:", which.min(summary(backwardSelectModels)$bic)))
```

```
## [1] "Backward Select Models, Lowest BIC:" "13"
```

```
bestBackwardModel <- lm(rOMX ~ rABB + rNDA_SE + rTELIA + rHM_B + rINVE_B + rSEB_A + rASSA_B + rERIC_B +
```

The backward-stepwise method resulted in the same amount of variables, and the same model at 13 variables, excluding SINCH and KINV from the model.

e)

Now we fit the ridge regression. The lambda is set using a leave-one-out cross-validation.

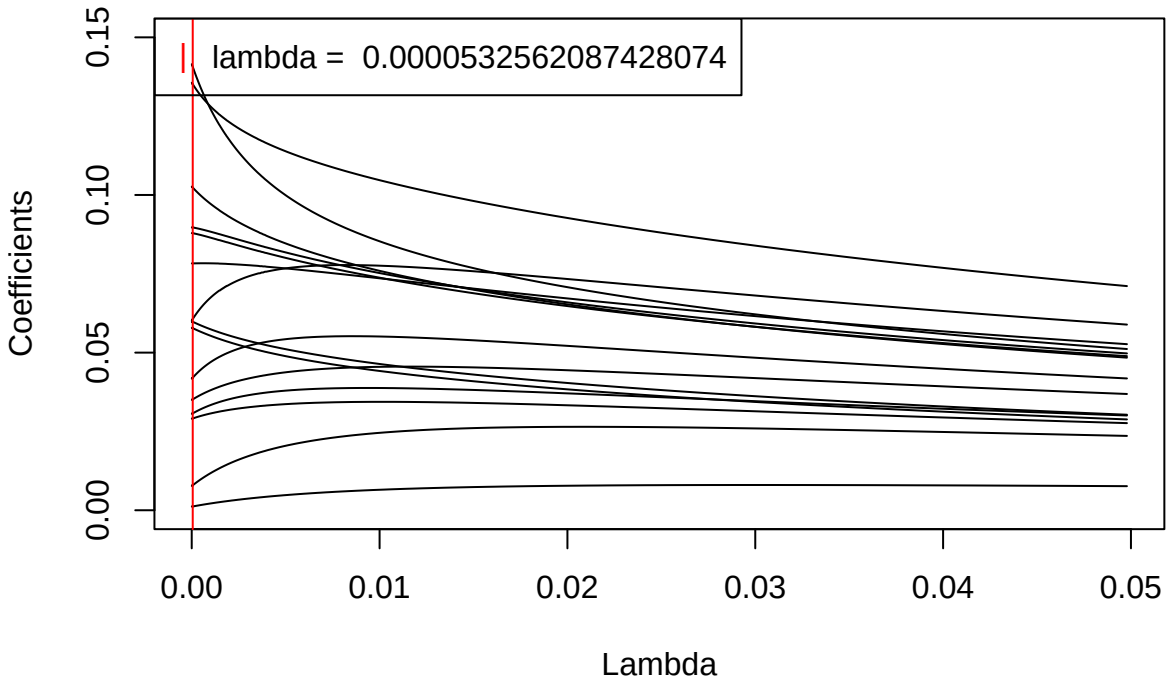
```
# Ridge model
set.seed(990714)
# Cross validation to find best lambda
ridgeModel <- cv.glmnet(y = as.matrix(returns[, 1]), x = as.matrix(returns[, -1]), alpha = 1, nfolds=nr

#Specify grid to plot from
grid_min <- 0
grid_max <- 0.01
grid = exp(seq(-3, -12, length.out=400))

#Train ridge model at each point of grid
ridgeModels <- glmnet(y = as.matrix(returns[, 1]), x = as.matrix(returns[, -1]), alpha = 0, lambda = gr
plot(grid, coef(ridgeModels)[2,], 'l', ylim=range(c(0, 0.15)), xlab= "Lambda", ylab="Coefficients")
apply(coef(ridgeModels)[c(-1,-2),], 1, function(x) lines(x=grid, y=x, 'l'))
```

```
## NULL
```

```
abline(v = ridgeModel$lambda.min, col="red")
legend("topleft", legend = paste("lambda = ", ridgeModel$lambda.min), pch = "|", col = "red")
```



It seems like the best lambda is close to zero, and thus only a minimal amount of shrinkage is performed. It is not optimal to only use ridge regression if we want the structure of the portfolio, since we are always going to get a full model with all predictors. Ridge regression can not select variables, only shrink them.

f)

Now we do the same as before, but with lasso regression.

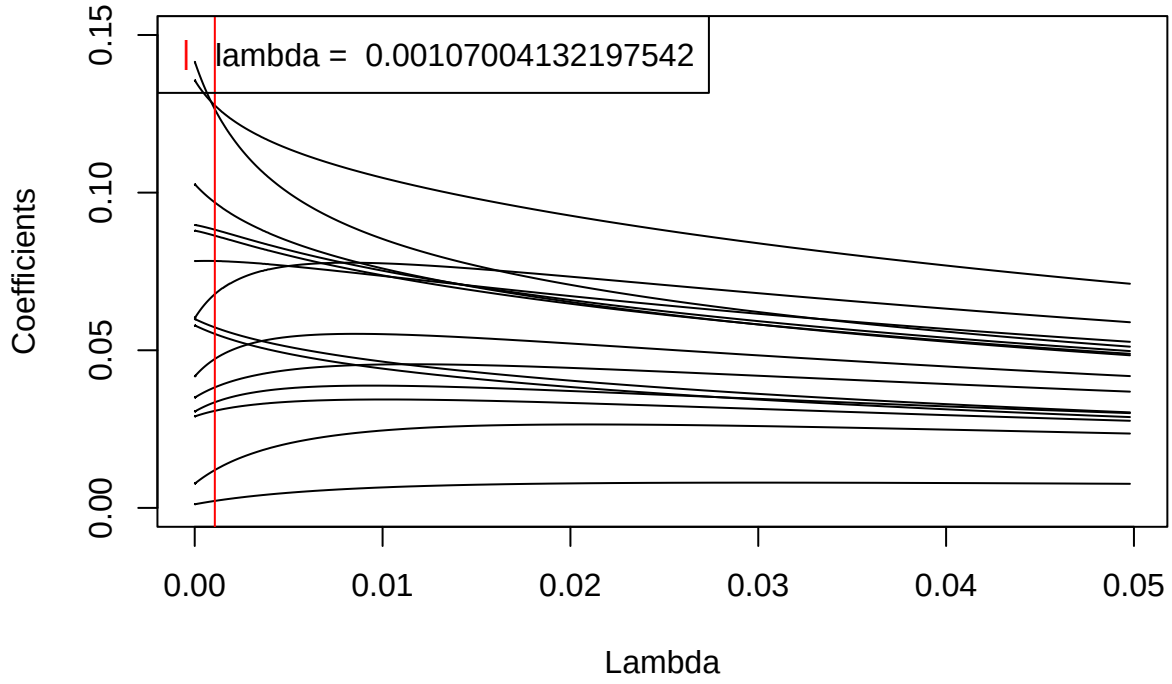
```
# Lasso model
set.seed(990714)
# Cross validation to find best lambda
lassoModel <- cv.glmnet(y = as.matrix(returns[, 1]), x = as.matrix(returns[, -1]), alpha = 0, nfolds=nr

#Specify grid to plot from
grid_min <- 0
grid_max <- 0.01
grid = exp(seq(-3, -12, length.out=400))

#Train lasso model at each point of grid
lassoModels <- glmnet(y = as.matrix(returns[, 1]), x = as.matrix(returns[, -1]), alpha = 0, lambda = gr
plot(grid, coef(lassoModels)[2,], 'l', ylim=range(c(0, 0.15)), xlab= "Lambda", ylab="Coefficients")
apply(coef(lassoModels)[c(-1,-2),], 1, function(x) lines(x=grid, y=x, 'l'))
```

```
## NULL
```

```
abline(v = lassoModel$lambda.min, col="red")
legend("topleft", legend = paste("lambda = ", lassoModel$lambda.min), pch = "|", col = "red")
```



Now we got a slightly bigger lambda. But, still we did not exclude any variable from the full model. However, we might accept that fact because the lasso is at least *able* to select variables.

g)

We dropped no more than two variables after doing each technique. This result could be interpreted as we having predictors that all are explaining the response variable very good. I guess that this result is somewhat expected, since the market index is based on these stocks (and some more). If I were to choose a model, I would use the one in d) where we used the backward-selection, since it chose the least amount of variables (equal with forward-selection). As we stated before, we exclude rSINCH (Sinch AB) and rKINV_B (Kinnevik AB). This model is specified by the coefficients:

```
coef(bestBackwardModel)
```

```
##   (Intercept)          rABB       rNDA_SE        rTELIA         rHM_B
## -0.0001382285  0.0613087767  0.1025359288  0.0783543004  0.0602848868
##       rINVE_B        rSEB_A       rASSA_B       rERIC_B          rAZN
##  0.1429027528  0.0887919769  0.0887411457  0.0578847361  0.0292722025
##        rSCA_B         rALFA       rATCO_A          rBOL
##  0.0357820495  0.0416934061  0.1412650743  0.0308589124
```

Task 2

We start by splitting the data into an 80/20 training and test dataset. We also create a response variable which is 1 if the market return is positive or equal to zero, and zero else.

```
### Split data into training/test data sets, and label a new variable rOMXClass 1 if rOMX>=0 and 0 else
returns_class = returns
returns_class$rOMXClass = as.integer(returns[,"rOMX"]>=0)
returns_class = subset(returns_class, select = -rOMX)
traintest_ratio = 0.8 * nrow(returns_class)
train_index <- seq_len(traintest_ratio)

train_returns_all <- returns_class[train_index, ]
test_returns_all <- returns_class[-train_index, ]
```

a)

Since the lasso regression led to us including all variables I will instead manually exclude 5 stocks for model_2 (According to the course forum). So I decided to remove all insignificant coefficients (ABB, TELIA, SINCH, SCA and KINV) with p-values greater than 0.05. See the summary of model_1 below. The fitted model is similar to a linear regression model. If our fitted values are above 0.5 they are classified as 1, and 0 if they are less than 0.5. We can not really say that the estimates are a measure how important a predictor is, since we need to check that we have an adequate model first. But we can draw some useful insights from the coefficients. They are all positive, meaning if any of them are increasing, holding the other coefficients fixed, the OMX is also increasing, according to the model. Also the estimates for Investor(INVE_B), Atlas Copco (ATCO_A) and SEB(SEB_A) are seemingly higher than the rest, in this model as well. This is also true for the simper model (see below) with only 10 predictors. It is necessary to check the model further for colinearity, but these three seem to predict the direction of the returns more than the other predictors.

```
## Create a subset with only 10 stocks
train_returns_10 <- subset(train_returns_all, select = -c(rABB, rTELIA, rSINCH, rSCA_B, rKINV_B))
test_returns_10 <- subset(test_returns_all, select = -c(rABB, rTELIA, rSINCH, rSCA_B, rKINV_B) )

## Fit logistic regression with all 15 stocks
print("Model 1")
```

```
## [1] "Model 1"
```

```
model_all <- suppressWarnings(glm(rOMXClass ~ ., data=train_returns_all, family = binomial))
sort(coef(model_all), TRUE)
```

```
##     rINVE_B     rATCO_A      rSEB_A     rNDA_SE     rASSA_B       rHM_B
## 226.7788861 202.3586399 155.4719900 135.3963926  97.2364761  96.0863854
##       rALFA     rERIC_B        rBOL      rTELIA        rAZN        rABB
##  84.0976529  72.1384471  67.7380369  60.8807488  51.6734189  38.7465752
##      rSCA_B     rKINV_B      rSINCH (Intercept)
##  20.4064511   5.8918856   0.3205778  -0.2647944
```

```
print("")
```

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

```
## Fit logistic regression with only 10 stocks
print("Model 2")
```

```
## [1] "Model 2"
```

```
model_10 <- suppressWarnings(glm(rOMXClass ~ ., data=train_returns_10, family = binomial))
sort(coef(model_10), TRUE)
```

```
##     rINVE_B     rATCO_A      rSEB_A     rNDA_SE       rHM_B     rASSA_B
## 248.9005229 198.2186672 161.2167283 119.7154542 100.6740204  93.8471093
##       rALFA     rERIC_B        rBOL        rAZN (Intercept)
##  83.3825115  67.3064041  63.3914669  54.0278110  -0.2429183
```

b)

The error rate for model 1 is 0.07252 and for model 2 0.08015.

```
predicted_all_test = predict(model_all, test_returns_all, type="response")
error_rate_all = sum((round(predicted_all_test)!=test_returns_all$rOMXClass))/nrow(test_returns_all)
print(c("Error rate model_1:", error_rate_all))
```

```
## [1] "Error rate model_1:" "0.0725190839694656"
```

```
predicted_10_test = predict(model_10, test_returns_10, type="response")
error_rate_10 = sum((round(predicted_10_test)!=test_returns_10$rOMXClass))/nrow(test_returns_10)
print(c("Error rate model_2:", error_rate_10))
```

```
## [1] "Error rate model_2:" "0.0801526717557252"
```

c)

The error rate is smaller for the larger model, including all 15 predictors. However, I think it is easy to find a better model, since I just removed the five insignificant coefficients. When trying to only use the returns for Investor(INVE_B), Atlas Copco (ATCO_A) and SEB(SEB_A) I got a similar result, but with slightly higher error rate at 0.10305. So the extra 12 variables did not improve much according to me, and depending on the goal for the model it might be advantageous to consider a smaller model. When trying to include ABB as well in model 2 I got the same error rate as in model 1. My conclusion is that it is not advantageous to use model 1 with all 15 stocks, and since a simpler model is almost always preferable I would prefer model 2 with 11 stocks, when including the ABB stock.