# HW2-Loran-Ali

2025-03-29

## Introduction

This report analyzes the CAPM (Capital Asset Pricing Model) using data from the Stockholm Stock Exchange. The goal is to assess whether the relationship between return on investment and beta times market return is statistically significant. Beta measures how sensitive an asset is to market movements. A beta below one suggests lower volatility than the market, a beta of one indicates movement in line with the market, and a beta above one implies higher volatility. The analysis uses daily stock prices from 2015 to 2018 and the OMX30 index as a market benchmark. The risk-free rate of return is excluded.

Instructions given in the assignment

```
library("rstudioapi")
```

```
## Warning: package 'rstudioapi' was built under R version 4.4.1
```

```
library("readxl")
library("dplyr")
```

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

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

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

```
library("tidyr")

folder <- dirname(getActiveDocumentContext()$path)
stocks <- read_excel("SSE_Stocks.xlsx")
stocks <- stocks %>% mutate(day = as.Date(day))
stocks <- stocks %>% select(ticker, day, lastad) %>% pivot_wider(names_from = ticker, values_from = last
stocks <- as.data.frame(stocks)
rownames(stocks) <- stocks$day
stocks$day <- NULL
stocks[] <- lapply(stocks, function(col) {
  if (is.factor(col)) col <- as.character(col)
  if (is.character(col)) col[col == "N/A"] <- NA
  as.numeric(col)
})
na_count  <- sapply(stocks, function(x) sum(is.na(x)))
notna_pos <- which(na_count == 0, arr.ind = TRUE)
data <- read_excel("OMX_Stockholm_30.xlsx")
index <- as.data.frame(as.numeric( data$lastad ))
rownames(index) <- data$day
```

```r
colnames(index) <- "OMX_30"
rm(data)
```

```r
set.seed(186) #For reproducibility
stocks_small <- stocks[, notna_pos] #Removes columns with missing observations (NA)

#Produces random sample for analysis
sample_obs <- runif(20, min = 1, max = ncol(stocks_small))
stocks_sample_price <- stocks_small[, sample_obs]
```

Problem 1

```r
stocks_sample <- stocks_sample_price[-nrow(stocks_sample_price),]
#Replaces all the values to percentage changes
for (i in 1:ncol(stocks_sample_price)) {
  stocks_sample[,i] <- 100 * diff(log(stocks_sample_price[,i]))
}
#The same approach for the index.
index_perc <- data.frame(100 * diff(log(index[,1])))
colnames(index_perc) <- colnames(index)
```

Problem 2

```r
Parametars_vector <- as.data.frame(matrix(nrow = 20, ncol = 3))
colnames(Parametars_vector) <- c("alpha", "beta", "variance")
rownames(Parametars_vector) <- colnames(stocks_sample)

for (i in 1:ncol(stocks_sample)) {
  fit <- lm(stocks_sample[,i] ~ index_perc$OMX_30)
  Parametars_vector[i,] <- c(fit$coefficients[1], fit$coefficients[2], var(fit$residuals))
}
```

Problem 3

```r
time_avrage_return <- as.matrix(colMeans(stocks_sample))
second_stage_regression <- lm(time_avrage_return ~ 1 + Parametars_vector$beta)
print(summary(second_stage_regression)$coefficients)
```

```
##                           Estimate Std. Error   t value    Pr(>|t|)
## (Intercept)             0.08985863 0.02836582  3.167849 0.005325642
## Parametars_vector$beta -0.06197463 0.03269291 -1.895660 0.074178169
```

Problem 4

```r
#By hand
tTesting <- matrix(nrow=2, ncol = 2)
colnames(tTesting) <- c("T-statistic", "P-Value")
rownames(tTesting) <- c("Intercept", "Slope")

degrees_of_freedom <- length(time_avrage_return) - 2

#Intercept, againts 0
tested_value_intercept <- coef(second_stage_regression)[1]
se_tested_value_intercept <- summary(second_stage_regression)$coefficients[1,2]
t_statistic_intercept <-  tested_value_intercept / se_tested_value_intercept
p_value_intercept <- 2 * (1 - pt(abs(t_statistic_intercept), degrees_of_freedom))
tTesting[1,] <- c(t_statistic_intercept, p_value_intercept)
```

```r
#Slope, against mean index
tested_value_slope <- coef(second_stage_regression)[2]
se_tested_value_slope <- summary(second_stage_regression)$coefficients[2,2]
t_statistic_slope <- (tested_value_slope - colMeans(index_perc) ) / se_tested_value_slope
p_value_slope <- 2 * (1 - pt(abs(t_statistic_slope), degrees_of_freedom))
tTesting[2,] <- c(t_statistic_slope, p_value_slope)

print(tTesting)
```

```
##           T-statistic     P-Value
## Intercept    3.167849 0.005325642
## Slope       -1.778895 0.092150198
```

Our data shows that we can reject the null hypothesis that the intercept is zero at the 1% significance level, indicating unexplained returns beyond beta. However, the null hypothesis that the slope equals the average index return holds at the 5% level but is rejected at the 10% level. Rejecting the intercept hypothesis contradicts CAPM's assumption, suggesting a baseline return independent of beta. Meanwhile, the slope aligning with market returns at the 5% level supports CAPM, implying a beta-return trade-off.

Problem 5

```r
plot(time_avrage_return ~ Parametars_vector$beta)
abline(second_stage_regression, col = "red", lwd = 2, )
```

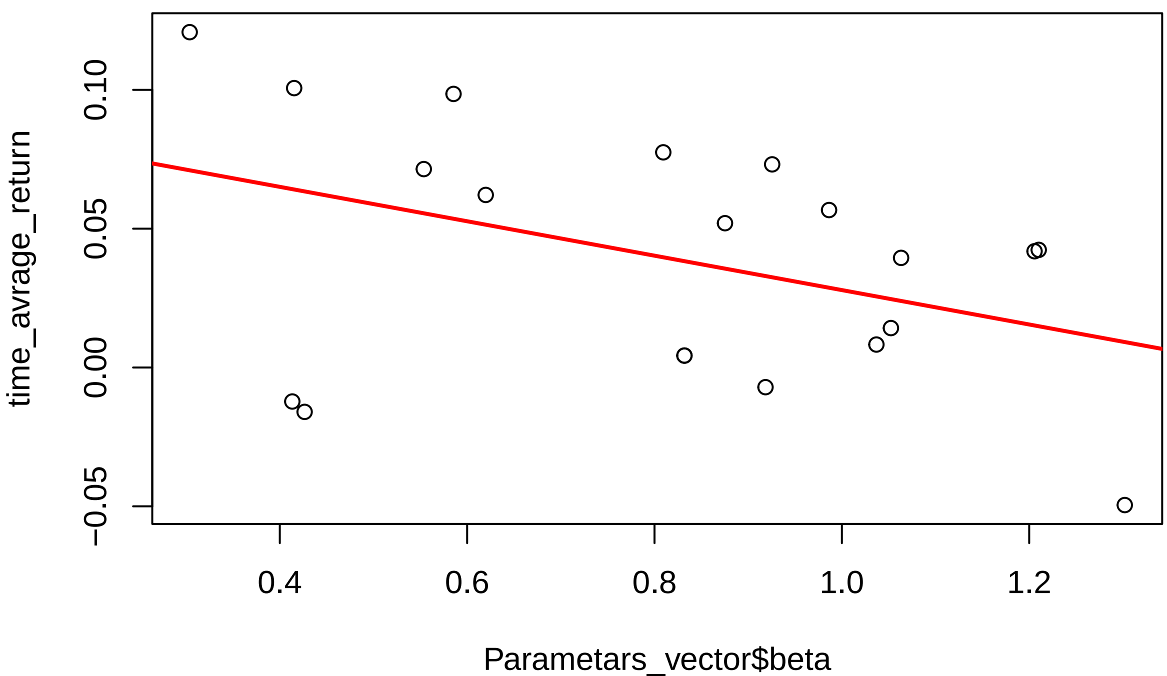


Figure 1 shows the average returns over the period plotted against the estimated beta coefficients. The line represents the estimated relationship from the second regression. The plot illustrates the (lack of) correlation between a stock's beta and its return.