Fama-French model

Jesse Keränen

11/23/2021

Prologue

This version is outdated, look ... for better code

This file attempts to form portfolios and factors in the spirit of Eugene Fama and Kenneth French (1993). I did my bachelors thesis about Fama's and French's factor model, but then I worked my data completely in Excel. After that I programmed a java program that forms these file and has also GUI. That time I focused on using things I had learned in programming courses. This time I try to maintain more of a data science state of mind and make the code more efficient. Nevertheless lack of access to suitable databases poses some problems and code needs further testing before further usage. Because I can't access better data I will use excel files from my bachelor thesis as data source.

Loading data

Let's start with loading data.

```
library(readxl)
library(pdfetch)
library(magrittr)
library(data.table)
library(tidyr)
library(zoo)
library(ggplot2)
library(stargazer)

data <- read_excel("extdata/famafrench2.xlsx")
data_dt <- as.data.table(data)


# Tidying data
data_dt <- melt(data_dt,id.vars = "Dates", variable.name =
          "Variable", value.name = "Price")

# Set Price values as numeric
data_dt <- data_dt[, Price := as.numeric(Price)]

data_dt <- data_dt[, Company := as.character(Variable)]
data_dt <- separate(data_dt, col = Company, into = c("Company", "right"),
                    sep = "_")
# Remove unnecessary column
```

```r
data_dt <- data_dt[, Variable := NULL]


data_dt <- data_dt[, Dates := as.Date(Dates)]

# Take a look at this. For some reason Kemira and Pihlajalinna had
# duplicates which needed to be removed.
data_dt <- data_dt[!duplicated(data_dt), ]

data_dt <- dcast(data_dt, Dates + Company~right, value.var = "Price")
colnames(data_dt) <- c("Date", "Company", "ME", "MEBE", "Price")
stargazer(tail(data_dt), type = 'latex', header = F,
          summary = F)
```

Table 1:

	Date	Company	ME	MEBE	Price
1	2020-03-31	VALOE	6.018	-1.459	0.045
2	2020-03-31	VERK	223.535	5.894	5
3	2020-03-31	VIK1V	172.800	0.803	13.137
4	2020-03-31	WRT1V	3,958.630	1.958	6.690
5	2020-03-31	WUF1V	10.858	0.917	1.590
6	2020-03-31	YIT	843.842	0.950	4.042

Calculate break points

Now that we have our data in nice format, we can start calculating breakpoints form our data. We then use these breakpoints in portfolio constructing. Usually in literature stocks are divided in more portfolios, but since our data is rather small we only calculate two breakpoints for both market value and ME/BE value. In the spirit of Fama and French (1993) we should calculate market value breakpoints using information from June and BE/ME breakpoints using values December. Now I nevertheless have so little data so I am going to use data from March for ME/BE and data from end of June for ME.

```r
q <- c(0.33, 0.66)
# Variable is called December, because usually we would use data from
# december
first_december <- data_dt[month(Date) == 3, min(Date)]

decembers <- seq(first_december, max(data_dt$Date), by="years")

first_july <- data_dt[month(Date) == 6, min(Date)]
julys <- seq(first_july, max(data_dt$Date), by = "years")

stargazer(data.table(Decembers = decembers, Julys =julys),
          type = 'latex', header = F, summary = F)
```

```r
# in this data there is one March more than Julys
MEBE_breakpoints = matrix(nrow = length(decembers)-1, ncol = length(q))
ME_breakpoints = matrix(nrow = length(julys), ncol = length(q))
```

Table 2:

	Decembers	Julys
1	2011-03-31	2011-06-30
2	2012-03-31	2012-06-30
3	2013-03-31	2013-06-30
4	2014-03-31	2014-06-30
5	2015-03-31	2015-06-30
6	2016-03-31	2016-06-30
7	2017-03-31	2017-06-30
8	2018-03-31	2018-06-30
9	2019-03-31	2019-06-30
10	2020-03-31	2011-06-30

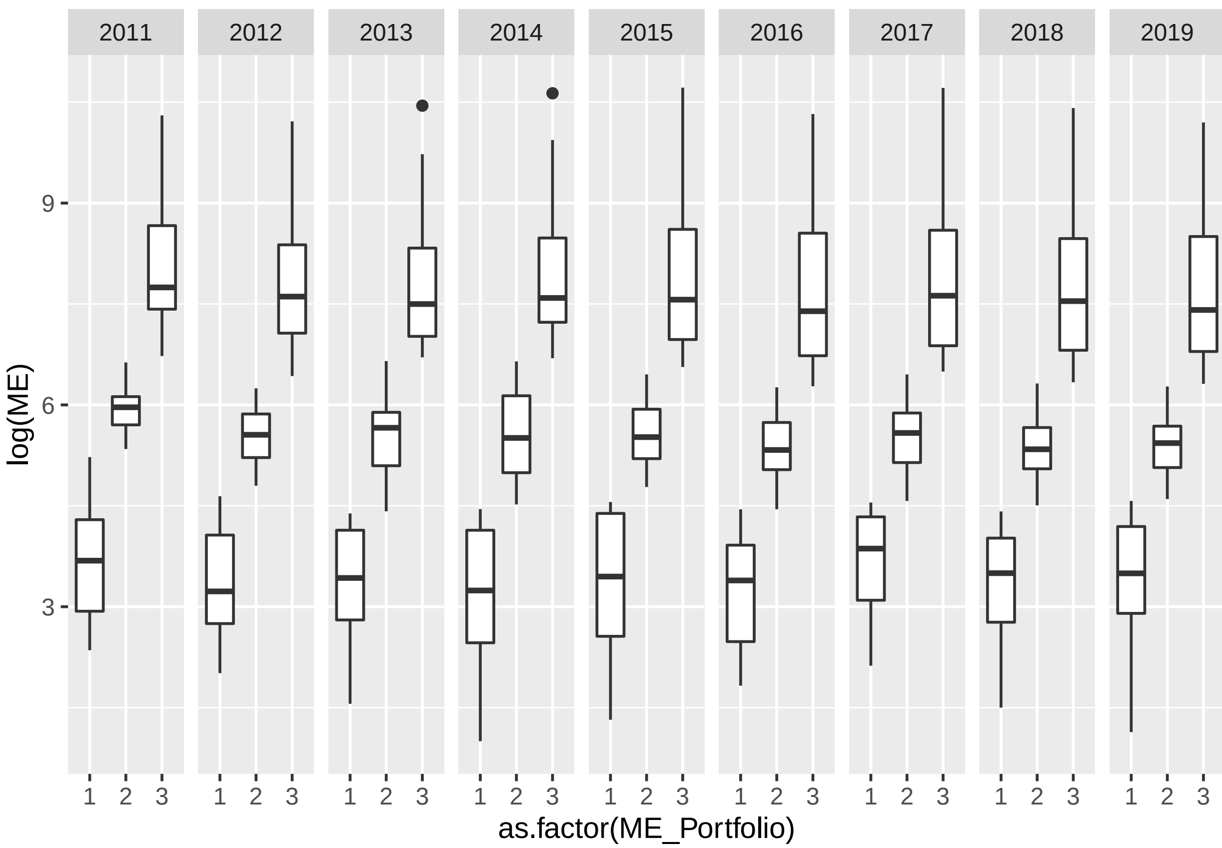
```
for(i in 1:length(julys)){
  MEBE_breakpoints[i,] <- data_dt[Date == decembers[i],
                          quantile(MEBE, q, na.rm = T)]
  ME_breakpoints[i,] <- data_dt[Date == julys[i],
                        quantile(ME, q, na.rm =T)]
}
```

Construct portfolios

We now know the breakpoints so we can start to assign stocks to portfolios.Stock is assigned to a "small" portfolio if it's market value is smaller than first ME break point. If stocks market value is smaller than second break point it is allocated to "medium" portfolio. Otherwise stock goes to "big" portfolio. Each stock is also allocated to a portfolio based on it's ME/BE value. This way we become with 9 size/value portfolios. It's worth mentioning that in my data I have ME/BE values for each stock, whereas in literary BE/ME values are usually used. This shouldn't nevertheless change our results much since these two values are just other way around. Though it changes interpretation of value. Now low ME/BE value indicates value stock and high ME/BE value growth stock.

```
# Order data.table by Company
data_dt <- setorder(data_dt, "Company")

# At the beginning I set every stock to portfolio 3.
data_dt <- data_dt[!is.na(ME) & month(Date) == 7,
          ME_Portfolio := 3]

# I don't like using for loops in R, since there is usually a better way.
# This time I couldn't come up with way doing this without a loop though.
for (j in 1:nrow(ME_breakpoints)){
  for (i in ncol(ME_breakpoints):1){
    data_dt <- data_dt[ME < ME_breakpoints[j, i] &
              month(Date) == 7 & year(Date) == year(first_july)
              -1 + j, ME_Portfolio := i, by=year(Date)]
  }
}

# Now set portfolio value also for other months than July.
# Check that this doesn't mess other variables
```

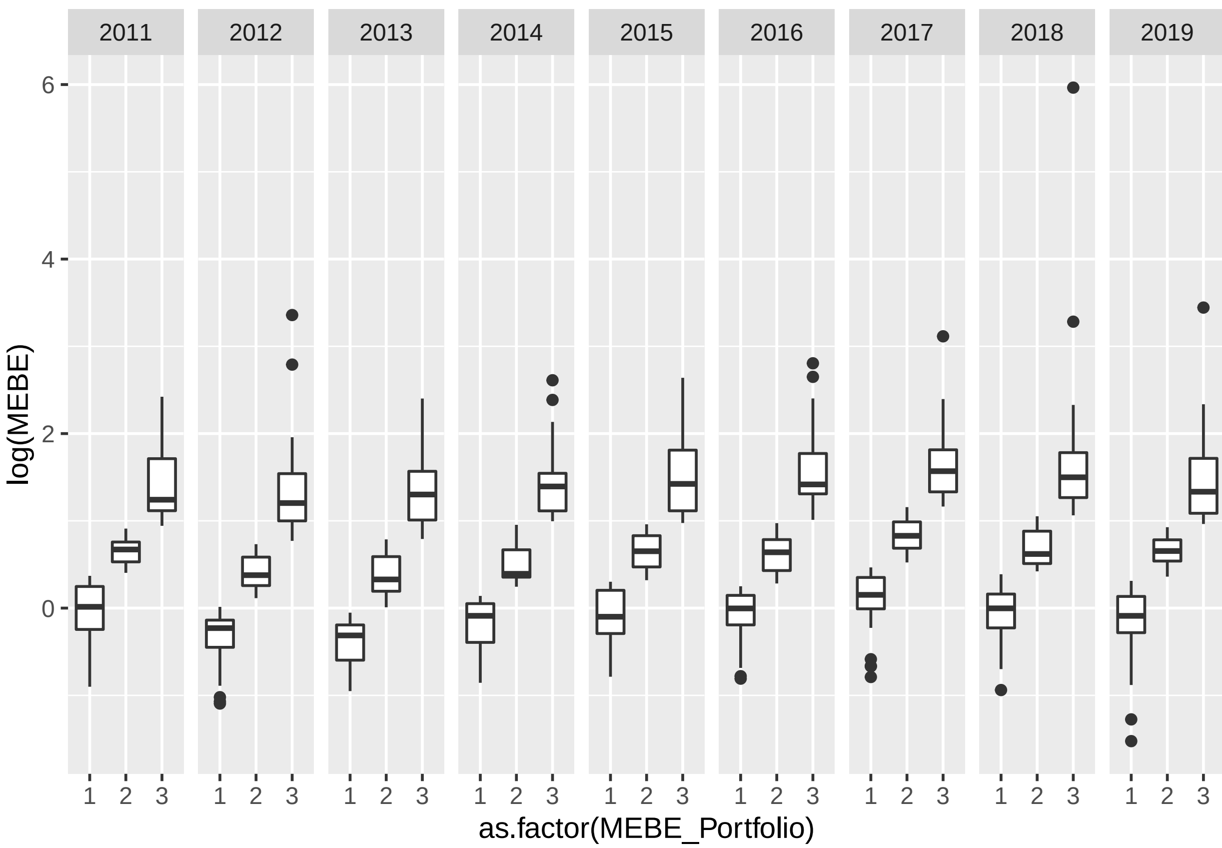
```r
data_dt <- data_dt[, lapply(.SD, function(x) na.locf(x, na.rm=F)), by=Company]

apu <- data_dt[month(Date) == 7]
ggplot(na.omit(apu), aes(as.factor(ME_Portfolio), log(ME))) +
  geom_boxplot() + facet_grid(~year(Date))
```



It seems like our allocation worked, since market values of stocks in different portfolio don't overlap. In nested for loop I set portfolio value for each stock for each year to July. After the loops I copied this value to other months also. This will help us when we start to calculate portfolio returns. Stock was allocated to one portfolio only if it had market value available. Now we do identical thing with the ME/BE values.

```r
data_dt <- data_dt[!is.na(MEBE) & month(Date) == 7, MEBE_Portfolio := 3]

# I don't like using for loops in R, since there is usually a better way.
# This time I couldn't come up with way doing this without a loop though.
for (j in 1:nrow(MEBE_breakpoints)){
  for (i in ncol(MEBE_breakpoints):1){
    data_dt <- data_dt[MEBE < MEBE_breakpoints[j, i] &
    month(Date) == 7 & year(Date) == year(first_july) -1 + j,
    MEBE_Portfolio := i, by=year(Date)]
  }
}

# Now set portfolio value also for other months than July.
data_dt <- data_dt[, lapply(.SD, function(x) na.locf(x, na.rm=F)), by=Company]
```

```r
apu <- data_dt[month(Date) == 7]
ggplot(na.omit(apu), aes(as.factor(MEBE_Portfolio), log(MEBE))) +
  geom_boxplot() + facet_grid(~year(Date))
```

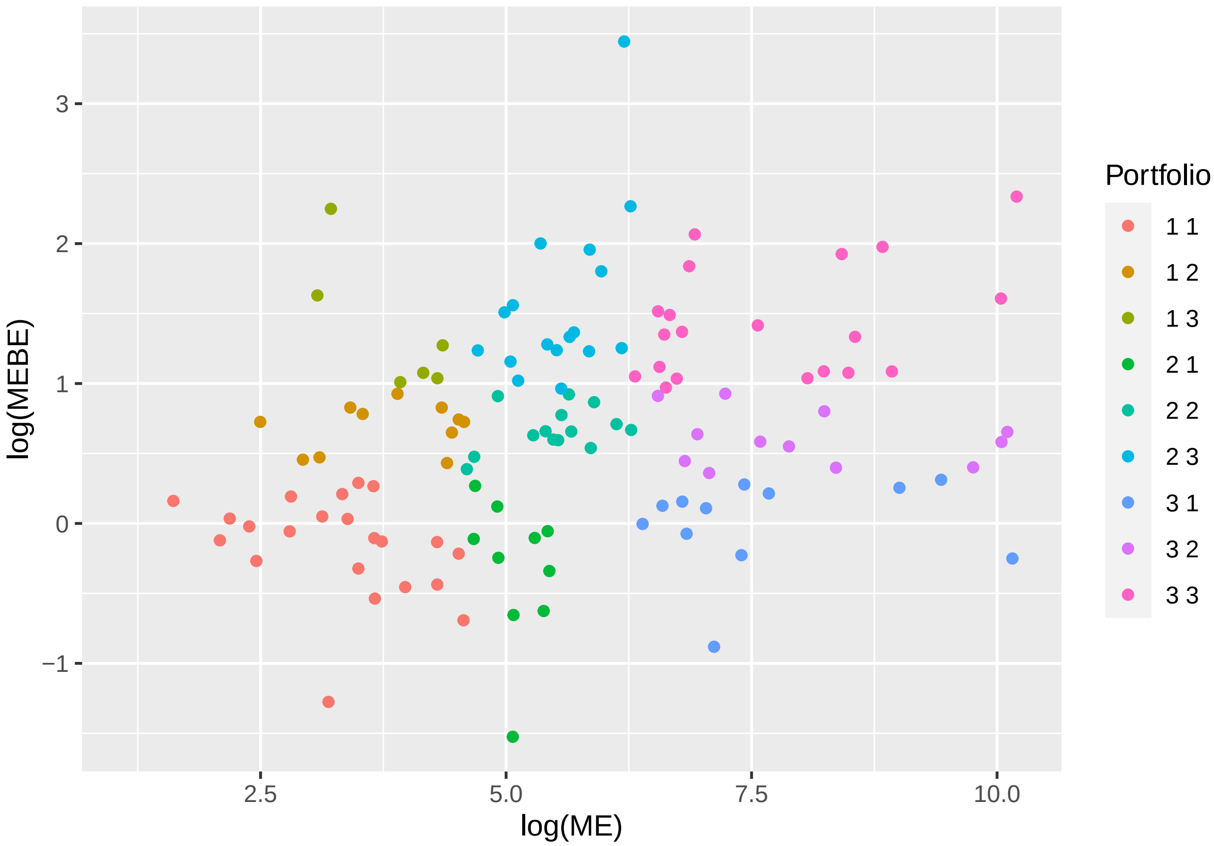


We can see that our data has some big outliers. I think that these are mistakes in data and they need to be removed from data. Negative ME/BE value means that book value of the company is negative, this can of course be possible. This means that company's liabilities exceed its assets. But for example Fama and French (1993) clean all stocks with negative BE/ME value from their data. I only remove stocks with unrealistically low ME/BE value, by setting their portfolio value to NA.

```r
data_dt[MEBE < -2, MEBE_Portfolio := NA]
```

It would might be good idea to clean data even more, but since I have only limited amount of data I don't want to be too strict. Now we can combine values from both portfolio allocations to one column. We need to remember first number tells about size and second about value.

```r
data_dt <- unite(data_dt, col = Portfolio, ME_Portfolio,
          MEBE_Portfolio, sep = " ")
# Remove portfolios with insufficient information
data_dt[grep("NA", Portfolio), Portfolio := NA]

# Plot how stocks and portfolios are distributed along the size-value
# horizon.
p <- data_dt[Date == "2019-07-31"]
ggplot(na.omit(p), aes(log(ME), log(MEBE), color=Portfolio)) +
  geom_point()
```



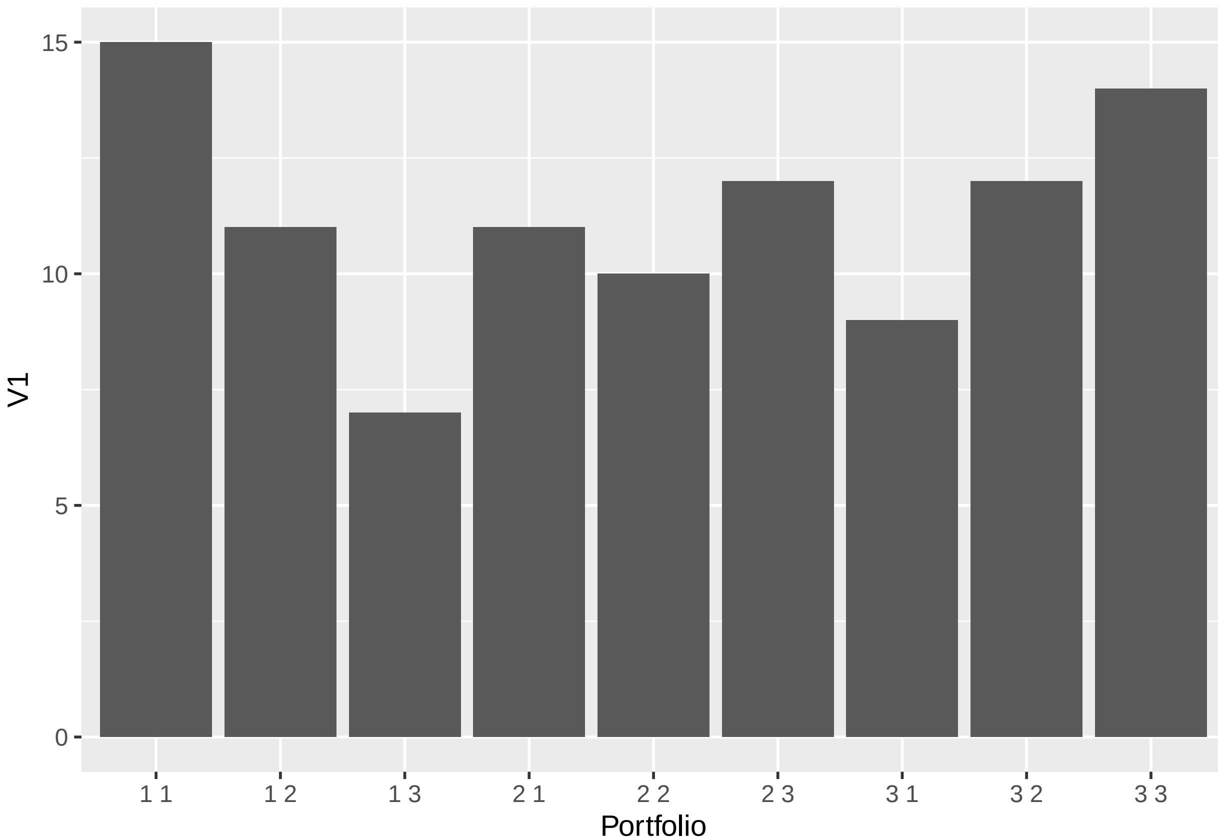
We can see from above graph that problem with little set of data is that number of stock allocated to each portfolio is rather small. We can see that for example in 2019 portfolio “1 3” had only 6 stock. Less than 10 stocks might not be enough to diversify unsystematic risk. Insufficient diversification can affect our results. This is nevertheless just one year. Let’s next take a look how many stocks are allocated to each portfolio on average.

```
# It shouldn't affect greatly on number of stocks what day we use. Only
# way number of stock within portfolio can change between allocations is
# when some stock leaves the stock exchange
ffp <- data_dt[month(Date) == 7]
stock_num <- as.data.table(table(ffp$Portfolio, ffp$Date))
colnames(stock_num) <- c("Portfolio", "Date", "Number")
stargazer(head(stock_num), type = 'latex', header = F,
          summary = F)
```

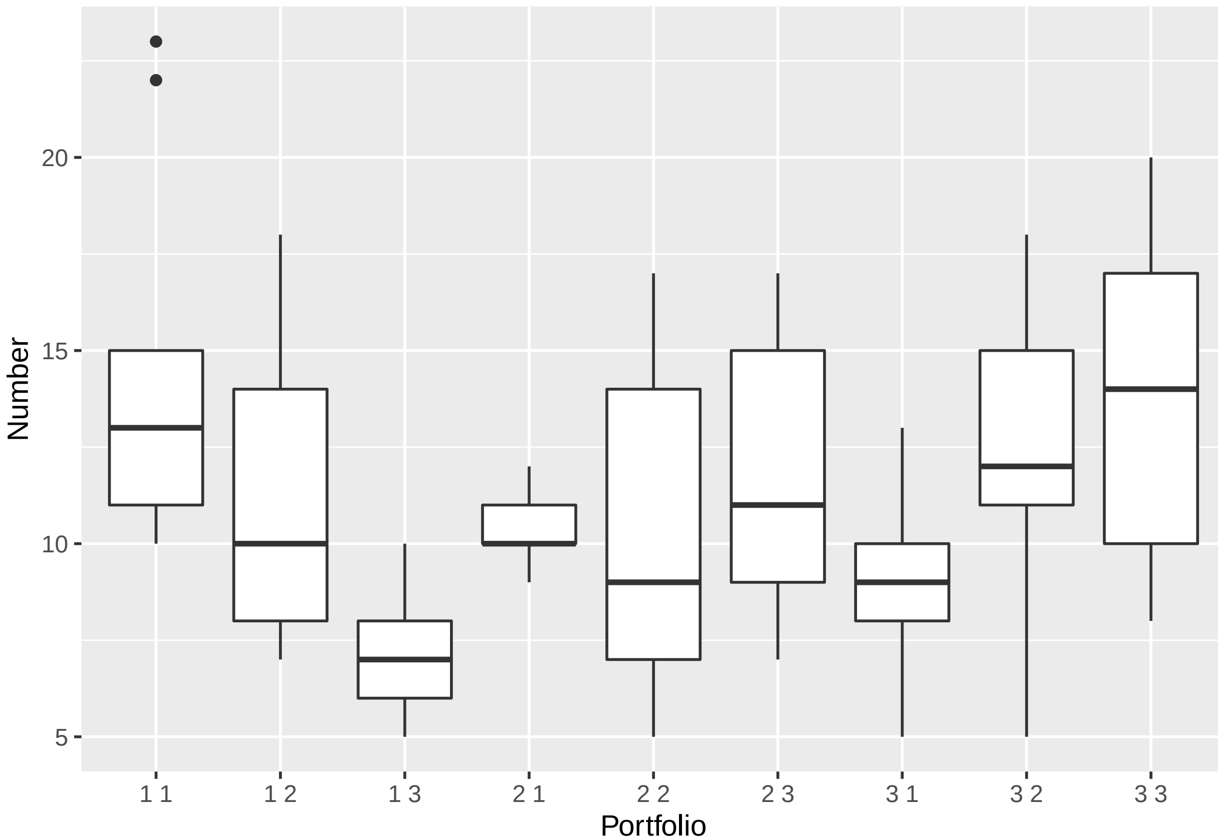
Table 3:

	Portfolio	Date	Number
1	1 1	2011-07-29	10
2	1 2	2011-07-29	7
3	1 3	2011-07-29	7
4	2 1	2011-07-29	12
5	2 2	2011-07-29	6
6	2 3	2011-07-29	7

```
ggplot(stock_num[, round(mean(Number), 0), by = Portfolio],
       aes(Portfolio, V1)) + geom_bar(stat="identity")
```




```
ggplot(stock_num, aes(Portfolio, Number)) + geom_boxplot()
```



We have our stock allocated to nine portfolio for each year. Soon we can start calculating actual portfolio returns. We are going to calculate monthly weighted returns. Let's calculate value of each portfolio each month so we can easier calculate weighted returns. After we calculate weight of each asset in portfolio.

```
for (i in 1:length(unique(data_dt$Date))){
  data_dt[Date == Date[i], Portfolio_ME := sum(ME),
          by = Portfolio]
}

data_dt[, Weight := ME/Portfolio_ME]
```

Calculate portfolio returns

We have done already many things, but to be able to calculate portfolios returns we need to calculate also returns of individual stocks. Next we calculate simply monthly returns for each stock weight them by weight of the stock.

```
data_dt[, Return := Price/shift(Price) -1, by = Company]
data_dt[!is.na(Weight), Weighted_Return := as.numeric(Return)*
          as.numeric(Weight)]
```

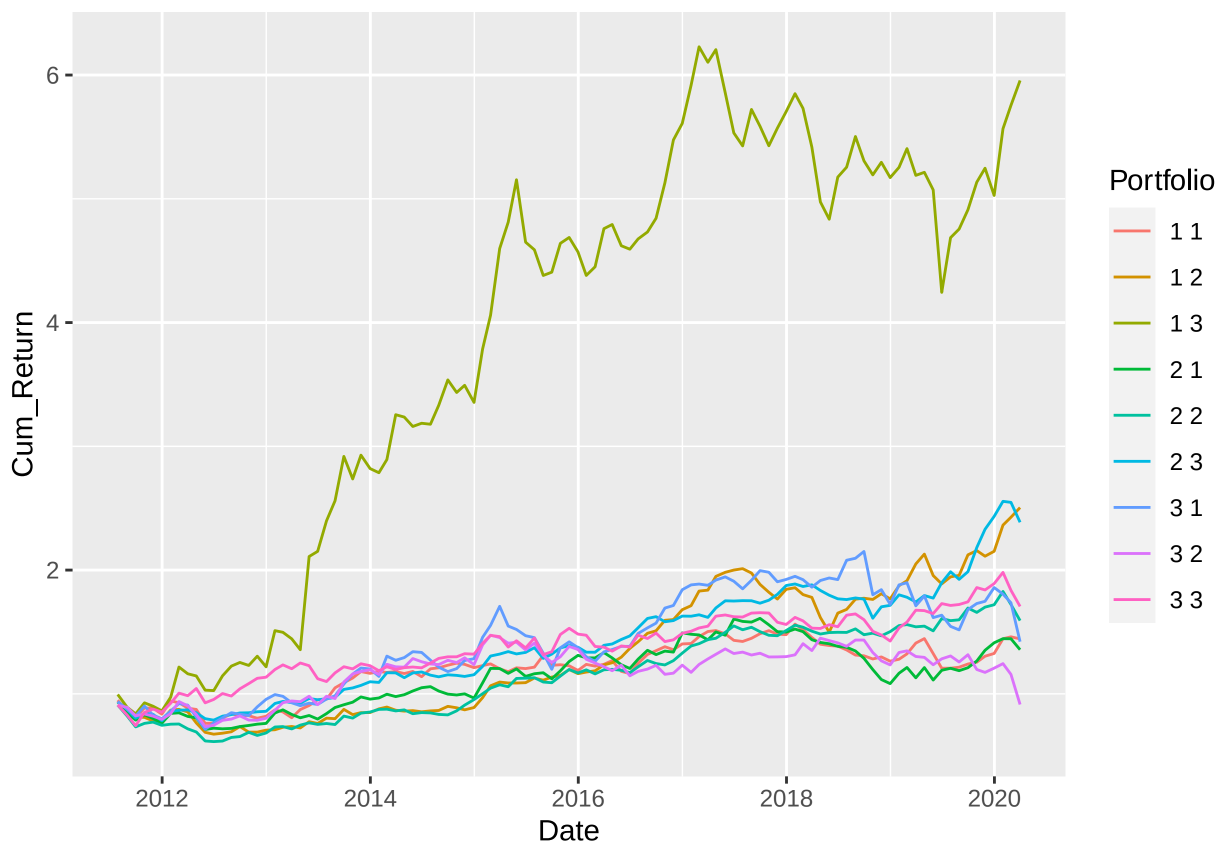
Now we finally have all the information we need to be able to calculate portfolio returns.

```
portfolio_returns <- data.table(Portfolio = character(), V1 =
                    numeric(), Date=as.Date(character()))
```

```
for (i in 1:length(unique(data_dt$Date))){
  apu <- data_dt[Date == Date[i] & !is.na(Portfolio),
         sum(Weighted_Return), by = Portfolio]
  apu <- apu[, Date := data_dt$Date[i]]
  portfolio_returns <- rbind(portfolio_returns, apu)
}
colnames(portfolio_returns) <- c("Portfolio", "Return", "Date")
```

After calculating the portfolio returns we can start the interesting part. Let's first look for a patterns in portfolio returns.

```
portfolio_returns[, Cum_Return := cumprod(1+Return), by = Portfolio]

ggplot(portfolio_returns, aes(Date, Cum_Return, color=Portfolio)) +
  geom_line()
```



Most notably we see that portfolio "1 3" has performed best. As we previously saw this portfolio had small number of stocks in it. So high returns on one stock can drive performance of whole portfolio.

Load factors

Only looking for patterns in returns, doesn't tell us too much. We might want to take a deeper look whether our factors affect returns of our portfolio. Until now we have constructed portfolios whose returns we want to explain. Next we need factors that might explain these returns. Fama and French (1993) use difference of returns of the large size portfolio and small size portfolios, as well as difference between high value portfolios and low value portfolios as these factors. Of course they also use excess market returns, which is familiar for us from capital asset model. We could quite easily calculate these differences, but since Finnish capital markets are quite integrated to European capital market it is interesting to regress our returns with European factors. These we can conveniently download from Kenneth French's data library.

```
factors <- read_excel("extdata/factors.xlsx")
factors_dt <- as.data.table(factors)
```

Regression

Now it's time for regressions. We run simply linear OLS regressions for each portfolio. Next I calculate excess portfolio returns by subtracting the risk free rate from returns and after that I start regressing.

```
portfolio_returns <- portfolio_returns[, RF :=
                    Return-factors_dt[, RF], by = Portfolio]

portfolios <- portfolio_returns[, unique(Portfolio)]
portfolios
```

[1] "1 2" "2 2" "2 1" "2 3" "1 1" "3 2" "3 3" "3 1" "1 3"

```
# This makes our summary table little bit nicer looking.
f <- as.matrix(factors_dt[,2:4])

list <- vector("list", length(portfolios))
for (i in 1:length(portfolios)){
  list[[i]] <- lm (portfolio_returns[Portfolio == portfolios[i],
             RF] ~ f)
}
stargazer (list, type='latex', header=FALSE, no.space = TRUE,
           column.sep.width = "1pt", font.size = "tiny")
```

Table 4:

	Dependent variable:								
	portfolio_returns[Portfolio == portfolios[i], RF]								
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
fMkt-RF	0.278***	0.423***	0.612***	0.538***	0.289***	0.741***	0.801***	0.888***	0.396*
	(0.103)	(0.076)	(0.092)	(0.066)	(0.096)	(0.100)	(0.079)	(0.108)	(0.201)
fSMB	0.095	0.106	0.420*	0.371**	0.163	0.143	−0.270	−0.199	0.160
	(0.251)	(0.185)	(0.225)	(0.161)	(0.234)	(0.244)	(0.192)	(0.263)	(0.490)
fHML	−0.016	0.075	−0.043	−0.185	0.224	−0.018	−0.618***	0.185	0.258
	(0.196)	(0.144)	(0.175)	(0.125)	(0.182)	(0.190)	(0.150)	(0.205)	(0.382)
Constant	0.008**	0.004	0.002	0.006**	0.004	−0.002	0.002	0.004	0.020**
	(0.004)	(0.003)	(0.004)	(0.003)	(0.004)	(0.004)	(0.003)	(0.004)	(0.008)
Observations	105	105	105	105	105	105	105	105	105
R^2	0.086	0.309	0.370	0.440	0.161	0.416	0.517	0.499	0.070
Adjusted R^2	0.058	0.289	0.352	0.423	0.136	0.399	0.502	0.485	0.042
Residual Std. Error (df = 101)	0.042	0.031	0.038	0.027	0.039	0.041	0.032	0.044	0.082
F Statistic (df = 3; 101)	3.148**	15.063***	19.806***	26.455***	6.475***	23.971***	36.013***	33.588***	2.525*

Note: *p<0.1; **p<0.05; ***p<0.01

Running these regressions show us some interesting results. First of all our model explain better returns of portfolios constructed from stock with large market value. This was to be expected. In many studies stock with as small market value as stock within our smallest portfolios are removed from data set. One can argue that assumptions of efficient markets don't concern too small stocks.Secondly, surprisingly market factor is most statistically significant, whereas out other factors are rarely significant.

Epilogue

It's is interesting how even if FF3 is quite straight forward extension of CAP-model, first bases heavily on empirical findings where as latter is more theoretic based. As one could guess in many papers factor model has turned out to explain especially portfolio returns significantly better than its ancestor. This isn't though case in our study. We shouldn't draw too much conclusions from results of this test. It would be interesting to test different combinations of factors or run necessity test for factors. Maybe the most biggest thing affecting reliability of our results is our data. First of all it too small and consists quite much NAs. NAs doesn't seem to be ranbom. They seem to more general within small stocks and stocks that have left the stock exchange. One solution could be looking data from various sources and maybe expanding data to concern all stocks from Nordic countries. One thing affecting our results is that our factors are in U.S. dollars where as our other data is in euros.

References:

https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.139.5892