Seminar Paper

Jesse Keränen

5/7/2022

Load data and libraries and some constants

```
library(data.table)
library(zoo)
```

```
##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(ggplot2)
library(tidyr)
library(readxl)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v tibble  3.1.7     v dplyr   1.0.9
## v readr   2.1.2     v stringr 1.4.0
## v purrr   0.3.4     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::between()   masks data.table::between()
## x dplyr::filter()    masks stats::filter()
## x dplyr::first()     masks data.table::first()
## x dplyr::lag()       masks stats::lag()
## x dplyr::last()      masks data.table::last()
## x purrr::transpose() masks data.table::transpose()
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:data.table':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year
```

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

```
library(xtable)


load("/Users/jessekeranen/seminar/data/2022_GBR_DS_monthly.RData")
load("/Users/jessekeranen/seminar/data/2022_GBR_DS_static.RData")
load("/Users/jessekeranen/seminar/data/2022_GBR_WS_yearly.RData")

# Load data from Keneth French Website
temp <- tempfile()
base <- "https://mba.tuck.dartmouth.edu/pages/faculty/ken.french/ftp/"
factor <-"F-F_Research_Data_Factors"
format<- "_CSV.zip"
ff_url <- paste(base,factor,format,sep ="")
download.file(ff_url, temp)

FF_3_Factors_dt <- as.data.table(read_csv(unzip(temp),skip = 3) %>%
rename(Date = ...1) %>%
  mutate_at(vars(-Date), as.numeric) %>%
  mutate(Date = ymd(parse_date_time(Date, "%Y%m")))) %>%
  mutate(Date = lubridate::rollback(Date))
```

```
## New names:
## * `` -> `...1`
```

```
## Warning: One or more parsing issues, see `problems()` for details
```

```
## Rows: 1249 Columns: 5
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): ...1
## dbl (4): Mkt-RF, SMB, HML, RF
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
## Warning: 97 failed to parse.
```

```
colnames(FF_3_Factors_dt) <- gsub("-", "_", colnames(FF_3_Factors_dt))

date_for_plot <- as.yearmon("Jul 2015")
```

```
# Copy original data to a variable so that we don't need to change original data
dt <- copy(DS.monthly[, .(Id, ym, Date, RET.USD, MV.USD, RI.USD)])
dt[, month := month(Date)]
dt[, year := year(Date)]
dt[, hcol := ifelse(month >= 7, year-1, year-2)]
```

```
# Lagged market value
dt[, LMV.USD := shift(MV.USD, 1, type = "lag"), by = Id]

# Winsorize function, winsorizes given variables to 1st and 99th percentile
winsorize_border <- 0.01
winsorize <- function(dt, x, percentile){
  dt[, (x) := ifelse(get(x) > quantile(get(x), 1-percentile, na.rm=T),
                     quantile(get(x), 1-percentile, na.rm=T), get(x)), by=Date]
  dt[, (x) := ifelse(get(x) < quantile(get(x), percentile, na.rm=T),
                     quantile(get(x), percentile, na.rm=T), get(x)), by=Date]
}
winsorize(dt, "RET.USD", winsorize_border)


# Merge yearly data to monthly data
dt <- merge(dt, WS.yearly[, .(Id, Year, WC05476, WC05001, WC01001, WC01051,
                              WC01251, WC01101, WC05301, WC02999, WC03480,
                              WC03451, WC03255, WC03251, WC01151, WC03051,
                              WC03063, WC02201, WC02001, WC01551, WC03101,
                              WC03151, WC04551)], by.y = c("Id", "Year"),
            by.x = c("Id", "hcol"), all.x = T)

# For some variables we want to calculate yearly change. With help of this merge we get one year lagged
help2 <- dt[month == 7, .(hcol = hcol, Id = Id, LWC02999 = WC02999,
                          LWC03151 = WC03151, LWC01151 = WC01151)]
help2[, hcol := hcol+1]
dt <- merge(dt, help2, by = c("Id", "hcol"), all.x = T)

# Book to market value
dt[, BM := (WC05476/WC05001)]
# Total book value = book value per share * shares outstanding
dt[, BE := WC05476*WC05301]
# Operating profitability
dt[, OP := WC01551/BE]
# Investment variable
dt[, INV := (WC02999 - LWC02999)/LWC02999]

# Order data.table by Date and decreasing by market value. Makes size allocation easier
setorder(dt, Date, -LMV.USD)

# Momentum variable
dt[, MOM := shift(RI.USD, 2, type = "lag")/shift(RI.USD, 12, type = "lag") - 1,
   by = Id]


size_breaks <- function(dt, quantiles, labels, name) {
  setorder(dt, Date, -LMV.USD)
  dt[, hcjun := ifelse(month >= 7, year, year-1)]

  temp <- dt[month==7 & !is.na(LMV.USD)]
  temp[, agg_weight := cumsum(LMV.USD)/sum(LMV.USD), by = year]
  temp[, tempr := cut(agg_weight, breaks = quantiles, labels = labels), Id]
  temp <- temp[, .(year, Id, tempr, agg_weight)]
```

```r
  setnames(temp, "tempr", name)
  dt <- merge(dt, temp, by.x = c("Id", "hcjun"), by.y = c("Id", "year"), all.x = T)
}
dt <- size_breaks(dt, c(0, 0.9, 1), c("Big", "Small"), "pf.size")
panel_country <- copy(dt)


# Quantiles used in portfolio allocations
quantiles <- c(0, 0.3, 0.7, 1)

# Labels for momentum portfolios
labels <- c("Loser", "Neutral_mom", "Winner")
# We don't use factors function for momentum allocation because it differs a bit from other factors. Mo
panel_country[year > 1990, pf.mom := cut(MOM, breaks = quantile(.SD[pf.size == "Big", MOM], quantiles,
                                                    na.rm = T), labels = labels, na.rm = T), by = Dat


# Function for allocating stocks to portfolios based on given variable
factor <- function(dt, variable, name, labels) {
  hlpvariable2 <- dt[month==7 & year > 1985, .(temp = cut(.SD[, get(variable)], breaks =
                  quantile(.SD[pf.size =="Big", get(variable)], quantiles, na.rm = T), labels =
                  labels), Id), by = year]

  setnames(hlpvariable2, "temp", name)
  # Merge the variable allocation back from July Y to June Y+1
  dt <- merge(dt, hlpvariable2, by.x=c("hcjun", "Id"), by.y=c("year", "Id"), all.x=T)

  return(dt)
}
panel_country <- factor(panel_country, "BM", "pf.bm", c("Low", "Neutral", "High"))
panel_country <- factor(panel_country, "OP", "pf.op", c("Weak", "Neutral_op", "Robust"))
panel_country <- factor(panel_country, "INV", "pf.inv", c("Conservative", "Neutral_inv", "Agressive"))

# Form the 2x3 portfolios combining size with one additional variable
panel_country[, Portfolio := paste0(pf.size,".",pf.bm)]
panel_country[, Portfolio2 := paste0(pf.size,".",pf.op)]
panel_country[, Portfolio3 := paste0(pf.size,".",pf.inv)]
panel_country[, Portfolio4 := paste0(pf.size,".",pf.mom)]


# Function for calculating portfolio returns
portfolioreturns <- function(dt, portfolio){
  portfolio_returns <- dt[, .(RET.USD = weighted.mean(RET.USD, LMV.USD, na.rm = T)),
                          by = c("Date", portfolio)]
  return(dcast(portfolio_returns, paste("... ~ ", portfolio)))
}
portfolio_returns1 <- portfolioreturns(panel_country, "Portfolio")
```

```
## Using 'RET.USD' as value column. Use 'value.var' to override
```

```r
portfolio_returns2 <- portfolioreturns(panel_country, "Portfolio2")
```

```
## Using 'RET.USD' as value column. Use 'value.var' to override
```

```
portfolio_returns3 <- portfolioreturns(panel_country, "Portfolio3")
```

```
## Using 'RET.USD' as value column. Use 'value.var' to override
```

```
# For momentum we use equal weights
portfolio_returns4 <- panel_country[, .(RET.USD = mean(RET.USD, na.rm = T)),
                                    by = c("Date", "Portfolio4")]
portfolio_returns4 <- dcast(portfolio_returns4, ... ~ Portfolio4)
```

```
## Using 'RET.USD' as value column. Use 'value.var' to override
```

```
# Merge all portfolios to one
portfolio_returns <- merge(portfolio_returns1, portfolio_returns2, by = "Date")
portfolio_returns <- merge(portfolio_returns, portfolio_returns3, by = "Date")
portfolio_returns <- merge(portfolio_returns, portfolio_returns4, by = "Date")
```

```
## Warning in merge.data.table(portfolio_returns, portfolio_returns4, by = "Date"):
## column names 'Big.NA.x', 'NA.NA.x', 'Small.NA.x', 'Big.NA.y', 'NA.NA.y',
## 'Small.NA.y' are duplicated in the result
```

```
portfolio_returns[is.na(portfolio_returns)] <- 0

# Market risk factor
market_return_dt <- dt[!is.na(RET.USD) & !is.na(LMV.USD), .(Return_Mkt = weighted.mean(RET.USD, LMV.USD
market_return_dt <- merge(market_return_dt, FF_3_Factors_dt[, .(Date, RF)], by = "Date", all.x = T)
market_return_dt[, RMRF := Return_Mkt - RF]
RMRF_factor <- market_return_dt[, .(Date, RMRF = Return_Mkt-RF)]

# Calculate factors from portfolios
factors <- portfolio_returns[, .(Date,
          SMBbm = (Small.High + Small.Neutral + Small.Low)/3 - (Big.High + Big.Neutral + Big.Low)/3,
          SMBop = (Small.Robust + Small.Neutral_op + Small.Weak)/3 - (Big.Robust + Big.Neutral_op +
                                                                        Big.Weak)/3,
          SMBinv = (Small.Conservative + Small.Neutral_inv + Small.Agressive)/3 - (Big.Conservative +
                                                                    Big.Neutral_inv + Big.Agressive)/3,
          SMBmom = (Small.Winner + Small.Neutral_mom + Small.Loser)/3 - (Big.Winner +
                                                                    Big.Neutral_mom + Big.Loser)/3,
          HML = (Small.High + Big.High)/2 - (Small.Low + Big.Low)/2,
          RMW = (Small.Robust + Big.Robust)/2 - (Small.Weak + Big.Weak)/2,
          CMA = (Small.Conservative + Big.Conservative)/2 - (Small.Agressive + Big.Agressive)/2,
          MOM = (Small.Winner + Big.Winner)/2 - (Small.Loser + Big.Loser)/2)
          ]

factors[, SMB := (SMBbm + SMBop + SMBinv + SMBmom)/4]

# Remove unnecessary columns
factors[, SMBop := NULL]
factors[, SMBbm := NULL]
factors[, SMBinv := NULL]
factors[, SMBmom := NULL]
```

```r
# Merge market factor with other factors
factors_wide <- merge(factors, RMRF_factor, by="Date")

# Melt for plotting
factors <- melt(factors_wide, id.vars = "Date", variable.name = "factor", value.name = "RET.USD")

# Check the factor statistics
factors[factor == "SMB",t.test(RET.USD)]
```

```
##
##  One Sample t-test
##
## data:  RET.USD
## t = 0.42732, df = 491, p-value = 0.6693
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.1822676  0.2835858
## sample estimates:
##  mean of x
## 0.05065911
```

```r
factors[factor == "HML",t.test(RET.USD)]
```

```
##
##  One Sample t-test
##
## data:  RET.USD
## t = 0.91302, df = 491, p-value = 0.3617
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.1265642  0.3462973
## sample estimates:
## mean of x
## 0.1098666
```

```r
factors[factor == "RMW",t.test(RET.USD)]
```

```
##
##  One Sample t-test
##
## data:  RET.USD
## t = 1.8256, df = 491, p-value = 0.06852
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.01227119  0.33412416
## sample estimates:
## mean of x
## 0.1609265
```

```r
factors[factor == "CMA",t.test(RET.USD)]
```

```
##
##  One Sample t-test
##
## data:  RET.USD
## t = 3.3124, df = 491, p-value = 0.000993
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.1124392 0.4403026
## sample estimates:
## mean of x
## 0.2763709
```

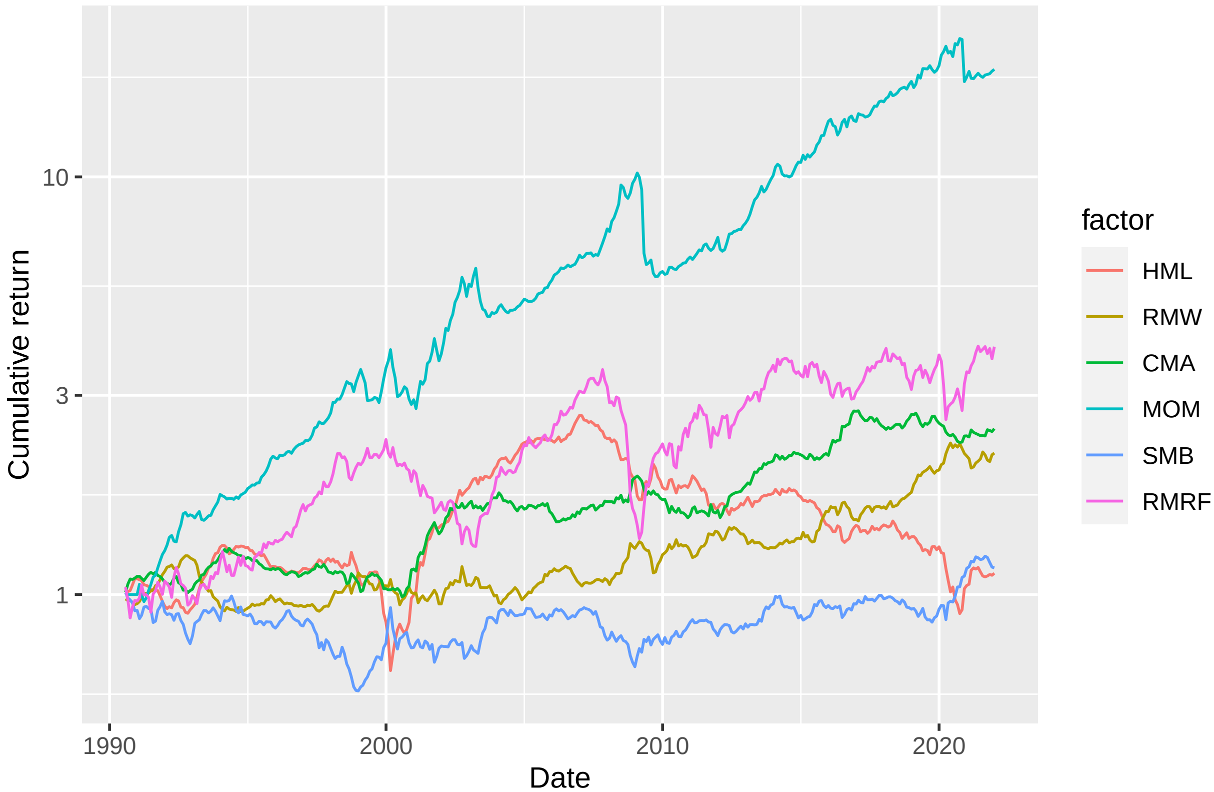
```
factors[factor == "MOM",t.test(RET.USD)]
```

```
##
##  One Sample t-test
##
## data:  RET.USD
## t = 4.5346, df = 491, p-value = 7.259e-06
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.3640777 0.9207990
## sample estimates:
## mean of x
## 0.6424383
```

```
factors[factor == "RMRF",t.test(RET.USD)]
```

```
##
##  One Sample t-test
##
## data:  RET.USD
## t = 2.3669, df = 491, p-value = 0.01832
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.09253377 0.99679816
## sample estimates:
## mean of x
##  0.544666
```

```
# Just for plotting
factors2 <- factors[Date >= "1990-07-31", .(Date, cum_prod = cumprod(RET.USD/100 + 1)), by = factor]
ggplot(factors2, aes(Date, cum_prod)) + geom_line(aes(color = factor)) +
  labs(y = "Cumulative return", title = "") + scale_y_log10()
```

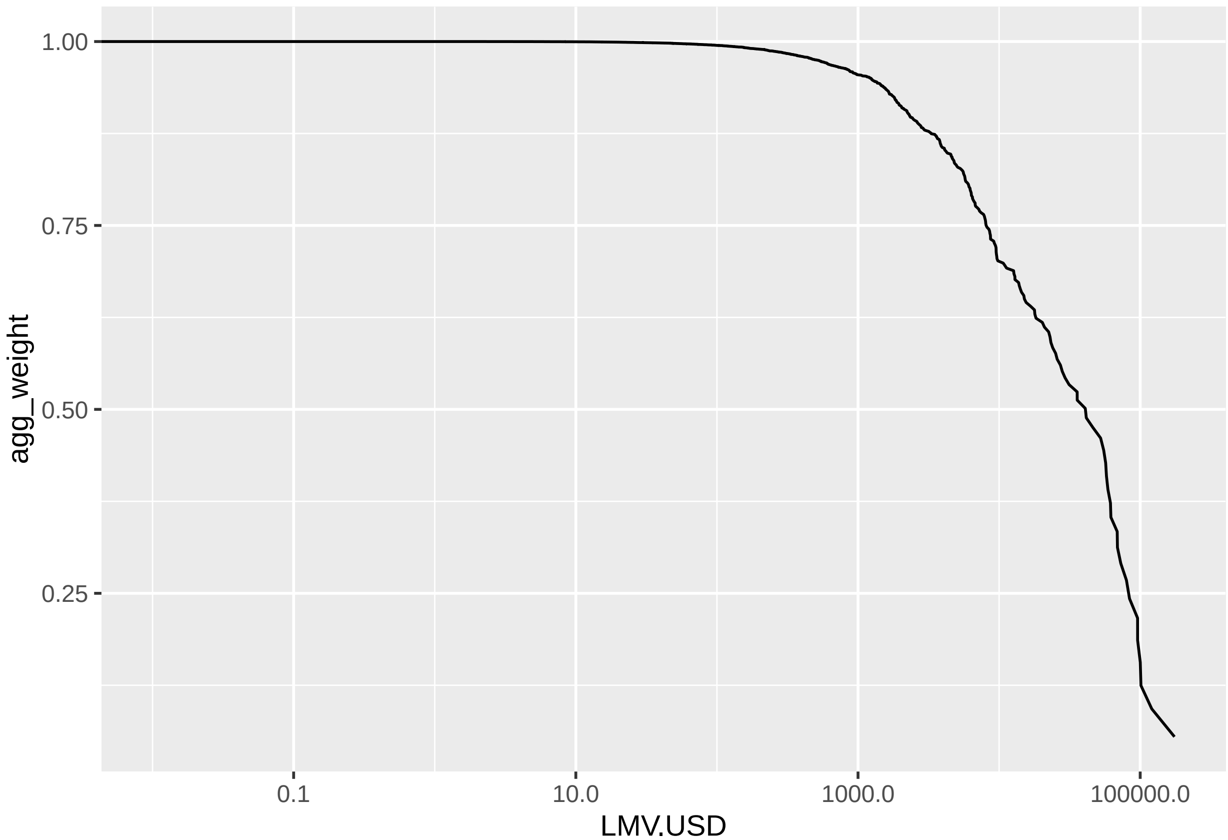


Size allocation:

```
dt2 <- dt[, .(Id, ym, Date, RET.USD, MV.USD, month, year, LMV.USD, BM, OP, INV, MOM, pf.size)]

quantiles <- c(0, 0.75, 0.90, 0.96, 0.99, 1)
dt2 <- size_breaks(dt2, quantiles, c(5, 4, 3, 2, 1), "portf.size")

# Plot to see if our allocation seems reasonable
options(scipen = 999)

# Cumulutave distribution of the market values in 2015 allocation
ggplot(dt2[ym == date_for_plot], aes(LMV.USD, agg_weight)) + geom_line() + scale_x_log10()
```

```
## Warning: Transformation introduced infinite values in continuous x-axis
```

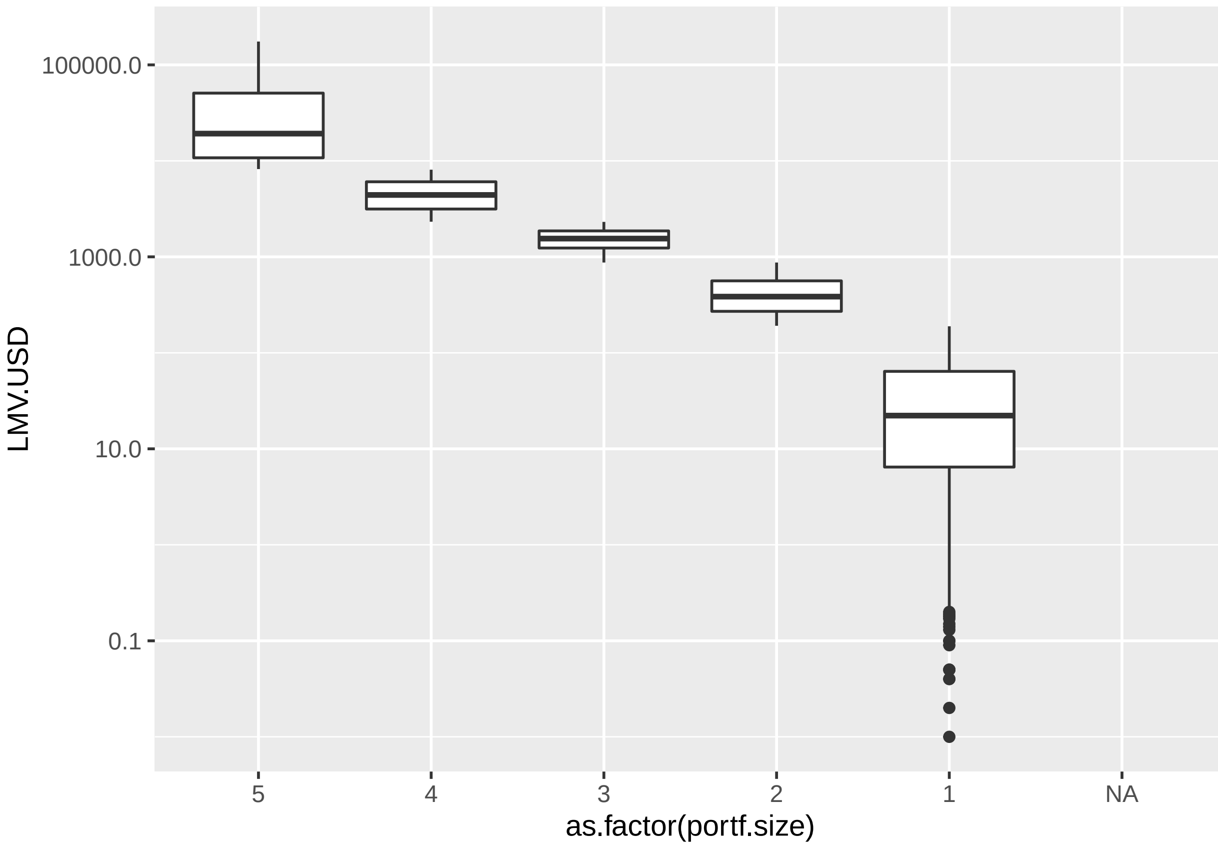
```
## Warning: Removed 4916 row(s) containing missing values (geom_path).
```




```
# Market values of stocks in different portfolios don't seem to overlap
ggplot(dt2[ym == date_for_plot], aes(as.factor(portf.size), LMV.USD)) + geom_boxplot() + scale_y_continu
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
```

```
## Warning: Removed 4919 rows containing non-finite values (stat_boxplot).
```



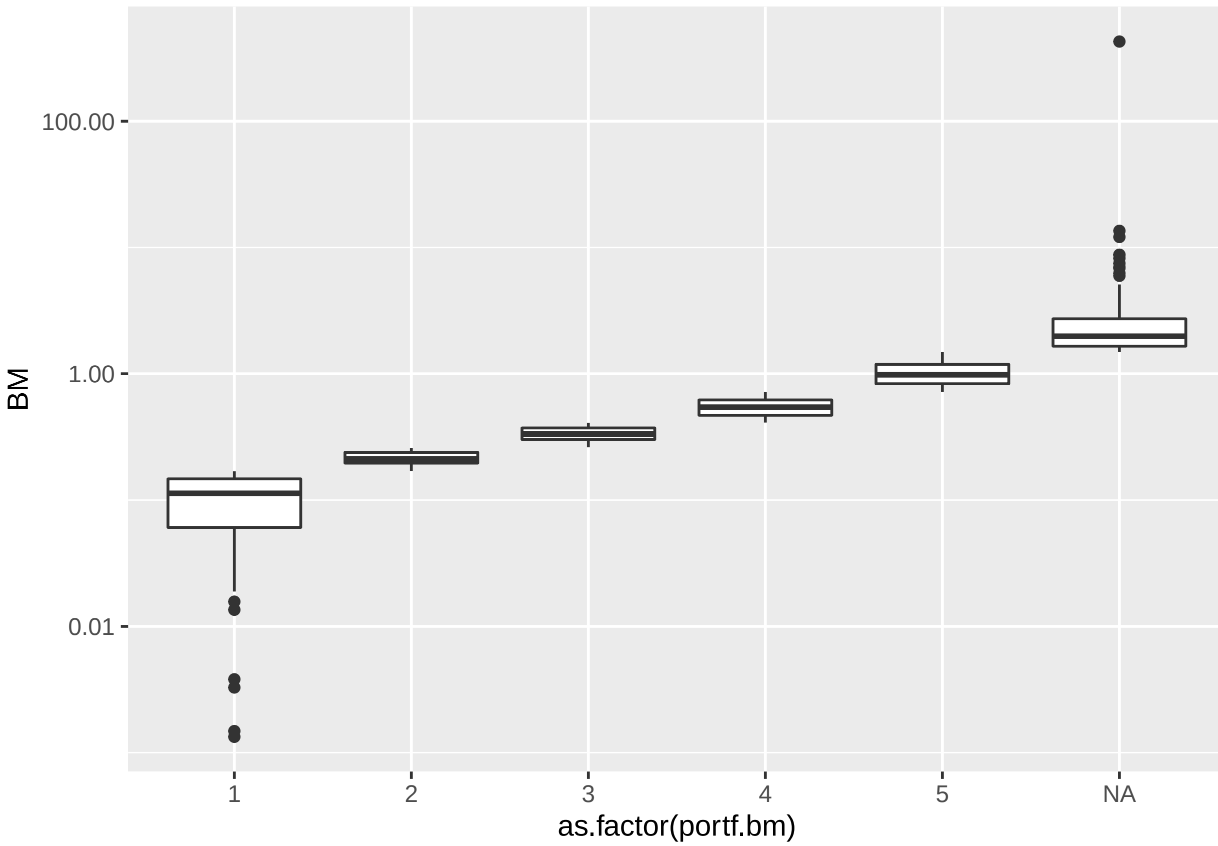
Allocation based no Book to Market-ratio. I tried to look at WC05001 to see if it consistent with stock prices from internet for some stocks. For me it seems like these values are somehow scaled.

```
quantiles <- seq(0, 1, by = 1/5)
dt2 <- factor(dt2, "BM", "portf.bm", c(1, 2, 3, 4, 5))

ggplot(dt2[ym == date_for_plot], aes(as.factor(portf.bm), BM)) + geom_boxplot() + scale_y_log10()
```

```
## Warning in self$trans$transform(x): NaNs produced
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
```

```
## Warning: Removed 5043 rows containing non-finite values (stat_boxplot).
```



By combining Size and Be/Me allocations, each stock will be allocated one of the 25 portfolios.

```r
dt2[, portfolio := paste(as.character(portf.size), as.character(portf.bm), sep = " ")]

# Remove observation that weren't allocted to a portfolio
dt2 <- dt2[!grepl("NA", portfolio)]

# Let's see if there is reasonable amount of stocks in each portfolio each year. Now we can't expect th
tail(table(dt2[year(ym) == 2015, portfolio, by = ym]))
```

```
##           portfolio
## ym          1 1 1 2 1 3 1 4 1 5 2 1 2 2 2 3 2 4 2 5 3 1 3 2 3 3 3 4 3 5 4 1 4 2
##   Jul 2015  90  48  90 141 174  26  24  47  55  43  15  15  30  31  22  18  21
##   Aug 2015  90  48  90 141 174  26  24  47  55  43  15  15  30  31  22  18  21
##   Sep 2015  90  48  90 141 174  26  24  47  55  43  15  15  30  31  22  18  21
##   Oct 2015  90  48  90 141 174  26  24  47  55  43  15  15  30  31  22  18  21
##   Nov 2015  90  48  90 141 174  26  24  47  55  43  15  15  30  31  22  18  21
##   Dec 2015  90  48  90 141 174  26  24  47  55  43  15  15  30  31  22  18  21
##           portfolio
## ym          4 3 4 4 4 5 5 1 5 2 5 3 5 4 5 5
##   Jul 2015  21  18  22  16  13  13  16  12
##   Aug 2015  21  18  22  16  13  13  16  12
##   Sep 2015  21  18  22  16  13  13  16  12
##   Oct 2015  21  18  22  16  13  13  16  12
##   Nov 2015  21  18  22  16  13  13  16  12
##   Dec 2015  21  18  22  16  13  13  16  12
```

Portfolio returns

```
portfolios <- portfolioreturns(dt2, "portfolio")
```

```
## Using 'RET.USD' as value column. Use 'value.var' to override
```

```
# Because unbalanced portfolio construction there for some months there can be zero stocks in some port
portfolios[is.na(portfolios)] <- 0


#For plotting
portfolios2 <- portfolios[year(Date) > 1990]

portfolios2 <- melt(portfolios2, id.vars = "Date", variable.name = "Portfolio",
     value.name = "Return")
portfolios2[, RET.CUM := cumprod(Return/100 + 1), by = Portfolio]
portfolios2 <- merge(portfolios2, market_return_dt[, .(Date, RF)], by = "Date")
portfolios2[, EXCESS.RET := Return - RF]

ggplot(portfolios2[year(Date) > 1990], aes(Date, RET.CUM, color = Portfolio)) + geom_line() + scale_y_l
```

Data seems weird. For example company id 882323, from 2011-12-31 to 2012-10-31.

```
p <- portfolios2[, mean(Return), by = Portfolio]
separate(p, "Portfolio", into = c("Size", "Value"), sep = " ") %>% spread(Value, V1)
```

Size 1 2 3 4 5 1: 1 -0.1214495 0.2014500 0.4434897 0.5741216 0.8086873 2: 2 0.6739196 0.7801254 0.9246939 0.9605571 1.0322095 3: 3 0.8138651 0.9501446 0.8453216 0.9011807 0.9429079 4: 4 0.7693538 0.9273085 0.9348666 0.8291398 0.8153078 5: 5 0.6183275 0.8844639 0.6456563 0.6595883 0.6603867

```
library(stargazer)
```

```
##
## Please cite as:
```

```
##  Hlavac, Marek (2022). stargazer: Well-Formatted Regression and Summary Statistics Tables.
```

```
##  R package version 5.2.3. https://CRAN.R-project.org/package=stargazer
```

```
portfolio_names <- portfolios2[, unique(Portfolio)]
# One can try different models by choosing different subset of factors here
fac <- as.matrix(factors_wide[year(Date) > 1990, -1])

list <- vector("list", length(portfolio_names))

for (i in 1:length(portfolio_names)){
  list[[i]] <- lm (portfolios2[Portfolio == portfolio_names[i], EXCESS.RET] ~ fac)
}

stargazer(list[1:8], type='latex', header=FALSE, no.space = TRUE,
          column.sep.width = "1pt", font.size = "tiny")
```

Table 1:

	Dependent variable:							
	portfolios2[Portfolio == portfolio_names[i], EXCESS.RET]							
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
facHML	-0.587^{***}	-0.465^{***}	-0.301^{***}	-0.170^{**}	0.195^{***}	-0.423^{***}	-0.132^{***}	0.019
	(0.079)	(0.079)	(0.062)	(0.068)	(0.051)	(0.053)	(0.047)	(0.044)
facRMW	-0.636^{***}	-0.523^{***}	-0.318^{***}	-0.194^{**}	-0.006	-0.316^{***}	-0.088	-0.059
	(0.106)	(0.106)	(0.083)	(0.092)	(0.068)	(0.071)	(0.063)	(0.060)
facCMA	0.004	-0.007	0.122	0.222^{***}	0.171^{***}	-0.184^{***}	0.013	0.061
	(0.094)	(0.095)	(0.074)	(0.082)	(0.061)	(0.063)	(0.057)	(0.053)
facMOM	-0.004	-0.041	-0.017	-0.021	0.026	-0.070^{*}	-0.013	-0.056^{*}
	(0.055)	(0.055)	(0.043)	(0.047)	(0.035)	(0.037)	(0.033)	(0.031)
facSMB	1.351^{***}	1.197^{***}	1.232^{***}	1.182^{***}	1.052^{***}	1.157^{***}	1.001^{***}	1.100^{***}
	(0.065)	(0.065)	(0.051)	(0.056)	(0.042)	(0.043)	(0.039)	(0.036)
facRMRF	1.000^{***}	0.943^{***}	0.951^{***}	0.971^{***}	0.897^{***}	1.025^{***}	0.998^{***}	0.981^{***}
	(0.042)	(0.042)	(0.033)	(0.036)	(0.027)	(0.028)	(0.025)	(0.024)
Constant	-0.755^{***}	-0.388^{**}	-0.265^{*}	-0.197	-0.007	0.062	0.029	0.182^{*}
	(0.188)	(0.189)	(0.149)	(0.164)	(0.122)	(0.127)	(0.113)	(0.106)
Observations	372	372	372	372	372	372	372	372
R^2	0.796	0.764	0.833	0.798	0.854	0.884	0.884	0.901
Adjusted R^2	0.792	0.760	0.830	0.795	0.852	0.882	0.882	0.899
Residual Std. Error (df = 365)	3.397	3.411	2.683	2.948	2.197	2.283	2.037	1.919
F Statistic (df = 6; 365)	236.869^{***}	196.704^{***}	303.087^{***}	240.661^{***}	356.296^{***}	463.192^{***}	463.743^{***}	552.489^{***}
Note:	*p<0.1; **p<0.05; ***p<0.01							

```
stargazer(list[9:16], type='latex', header=FALSE, no.space = TRUE,
          column.sep.width = "1pt", font.size = "tiny")
```

```
stargazer(list[17:25], type='latex', header=FALSE, no.space = TRUE,
          column.sep.width = "1pt", font.size = "tiny")
```

Table 2:

	Dependent variable:							
	portfolios2[Portfolio == portfolio_names[i], EXCESS.RET]							
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
facHML	0.150***	0.596***	−0.664***	−0.449***	0.263***	0.442***	0.605***	−0.460***
	(0.043)	(0.040)	(0.057)	(0.051)	(0.053)	(0.054)	(0.053)	(0.060)
facRMW	−0.019	0.133**	−0.216***	−0.246***	0.325***	0.124*	0.193***	−0.309***
	(0.058)	(0.054)	(0.076)	(0.069)	(0.071)	(0.072)	(0.072)	(0.080)
facCMA	0.062	0.105**	−0.183***	−0.093	−0.090	−0.050	−0.059	−0.163**
	(0.051)	(0.048)	(0.068)	(0.061)	(0.063)	(0.064)	(0.064)	(0.071)
facMOM	−0.037	−0.088***	−0.145***	−0.005	−0.109***	−0.008	−0.025	0.099**
	(0.030)	(0.028)	(0.039)	(0.035)	(0.037)	(0.037)	(0.037)	(0.041)
facSMB	1.147***	1.008***	0.886***	0.885***	0.750***	0.880***	0.843***	0.518***
	(0.035)	(0.033)	(0.046)	(0.042)	(0.043)	(0.044)	(0.044)	(0.049)
facRMRF	1.004***	0.975***	1.089***	1.026***	1.058***	1.038***	1.064***	1.027***
	(0.023)	(0.022)	(0.030)	(0.027)	(0.028)	(0.029)	(0.028)	(0.032)
Constant	0.168	0.239**	0.251*	0.273**	0.074	0.071	0.095	0.074
	(0.103)	(0.097)	(0.135)	(0.122)	(0.127)	(0.129)	(0.128)	(0.143)
Observations	372	372	372	372	372	372	372	372
R^2	0.912	0.921	0.865	0.870	0.856	0.862	0.872	0.805
Adjusted R^2	0.911	0.920	0.863	0.868	0.854	0.860	0.870	0.802
Residual Std. Error (df = 365)	1.853	1.745	2.442	2.204	2.283	2.323	2.303	2.576
F Statistic (df = 6; 365)	630.114***	712.672***	391.201***	407.413***	361.575***	380.708***	415.399***	250.702***
Note:	*p<0.1; **p<0.05; ***p<0.01							

Table 3:

	Dependent variable:								
	portfolios2[Portfolio == portfolio_names[i], EXCESS.RET]								
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
facHML	−0.158***	0.111*	0.440***	0.579***	−0.503***	−0.167***	0.063	0.280***	0.590***
	(0.056)	(0.059)	(0.058)	(0.059)	(0.052)	(0.051)	(0.050)	(0.056)	(0.065)
facRMW	0.011	0.110	0.196**	0.137*	0.316***	0.181***	0.046	−0.237***	−0.449***
	(0.075)	(0.079)	(0.077)	(0.079)	(0.069)	(0.069)	(0.067)	(0.076)	(0.087)
facCMA	0.096	0.097	0.148**	−0.045	0.003	0.139**	−0.126**	−0.090	0.007
	(0.067)	(0.071)	(0.069)	(0.070)	(0.062)	(0.062)	(0.060)	(0.068)	(0.078)
facMOM	−0.052	−0.091**	−0.153***	−0.024	−0.040	0.006	0.031	−0.0004	−0.069
	(0.039)	(0.041)	(0.040)	(0.041)	(0.036)	(0.036)	(0.035)	(0.039)	(0.045)
facSMB	0.328***	0.243***	0.337***	0.323***	−0.265***	−0.195***	−0.120***	−0.237***	−0.328***
	(0.046)	(0.049)	(0.047)	(0.048)	(0.042)	(0.042)	(0.041)	(0.046)	(0.053)
facRMRF	1.011***	0.997***	1.073***	1.017***	0.881***	0.986***	1.089***	0.961***	0.959***
	(0.030)	(0.031)	(0.031)	(0.031)	(0.027)	(0.027)	(0.027)	(0.030)	(0.035)
Constant	0.231*	0.248*	0.097	0.056	0.003	0.154	−0.080	0.085	0.167
	(0.133)	(0.142)	(0.138)	(0.140)	(0.123)	(0.123)	(0.120)	(0.135)	(0.155)
Observations	372	372	372	372	372	372	372	372	372
R^2	0.801	0.782	0.835	0.821	0.775	0.799	0.850	0.813	0.818
Adjusted R^2	0.797	0.779	0.832	0.818	0.771	0.796	0.848	0.810	0.815
Residual Std. Error (df = 365)	2.403	2.555	2.490	2.529	2.224	2.222	2.164	2.434	2.799
F Statistic (df = 6; 365)	244.477***	218.561***	307.407***	279.428***	209.185***	242.104***	345.758***	264.576***	272.593***
Note:	*p<0.1; **p<0.05; ***p<0.01								