FRTB Prototype for IRSWaps

Danny Rasch

2017-03-09

0.1 R Markdown

```
#setwd("c:/FRTB/_Portfolio_01")
#load("myfile.RData")
```

0.2 Including Plots

1 fdsfsd

```
library(tidyquant)
library(ggplot2)
data(FANG)
FANG
```

```
## # A tibble: 4,032 x 8
##    symbol date        open  high   low close    volume adjusted
##    <chr>  <date>     <dbl> <dbl> <dbl> <dbl>     <dbl>    <dbl>
##  1 FB     2013-01-02  27.4  28.2  27.4  28.0  69846400     28.0
##  2 FB     2013-01-03  27.9  28.5  27.6  27.8  63140600     27.8
##  3 FB     2013-01-04  28.0  28.9  27.8  28.8  72715400     28.8
##  4 FB     2013-01-07  28.7  29.8  28.6  29.4  83781800     29.4
##  5 FB     2013-01-08  29.5  29.6  28.9  29.1  45871300     29.1
##  6 FB     2013-01-09  29.7  30.6  29.5  30.6 104787700     30.6
##  7 FB     2013-01-10  30.6  31.5  30.3  31.3  95316400     31.3
##  8 FB     2013-01-11  31.3  32.0  31.1  31.7  89598000     31.7
##  9 FB     2013-01-14  32.1  32.2  30.6  31.0  98892800     31.0
## 10 FB     2013-01-15  30.6  31.7  29.9  30.1 173242600     30.1
## # ... with 4,022 more rows
```

```
## test ############
#### Overwrite FANG
currentPath=getwd()
setwd("C:/OneDrive/PowerBI_test/Portfolio_Slicer/Input")

library(data.table)
Quotes <- fread("C:/OneDrive/PowerBI_test/Portfolio_Slicer/Input/Quotes_detail.csv")
Quotes$date = as.Date(Quotes$date)


#Quotes=as.data.frame(Quotes)
Quotes <- subset(Quotes, symbol == "SIE.DE" | symbol == "CBK.DE" | symbol == "2PP.F"| symbol == "BAYN
```

```
########               does strange things
##Quotes = filter(Quotes, symbol == c("SIE.DE","CBK.DE"))
#t = subset(Quotes, date == "2007-01-02")
#########


## Option 1: Manual ticker
ticker <- c("GDAXI","SSMI","BIO.DE","ZIL2.DE",
    "SIE.DE","IFX.DE","CBK.DE","2PP.F","BAYN.DE","SDF.DE","KBC.DE")
    # Set name for BRK-A to BRK.A
    setSymbolLookup(GDAXI = list(name = "^GDAXI"))




##Option 2:Load ticker from xlsx

#AllocationCurrent = read.csv("AllocationCurrent.csv", sep="")

#(ticker <- as.vector(AllocationCurrent$Symbol))
ticker <- c("SIE.DE", "CBK.DE", "2PP.F")


setwd(currentPath)
```

```
Quotes_annual_returns <- Quotes %>%
    group_by(symbol) %>%
    tq_transmute(select     = adjusted,
                 mutate_fun = periodReturn,
                 period     = "yearly",
                 type       = "arithmetic")
```

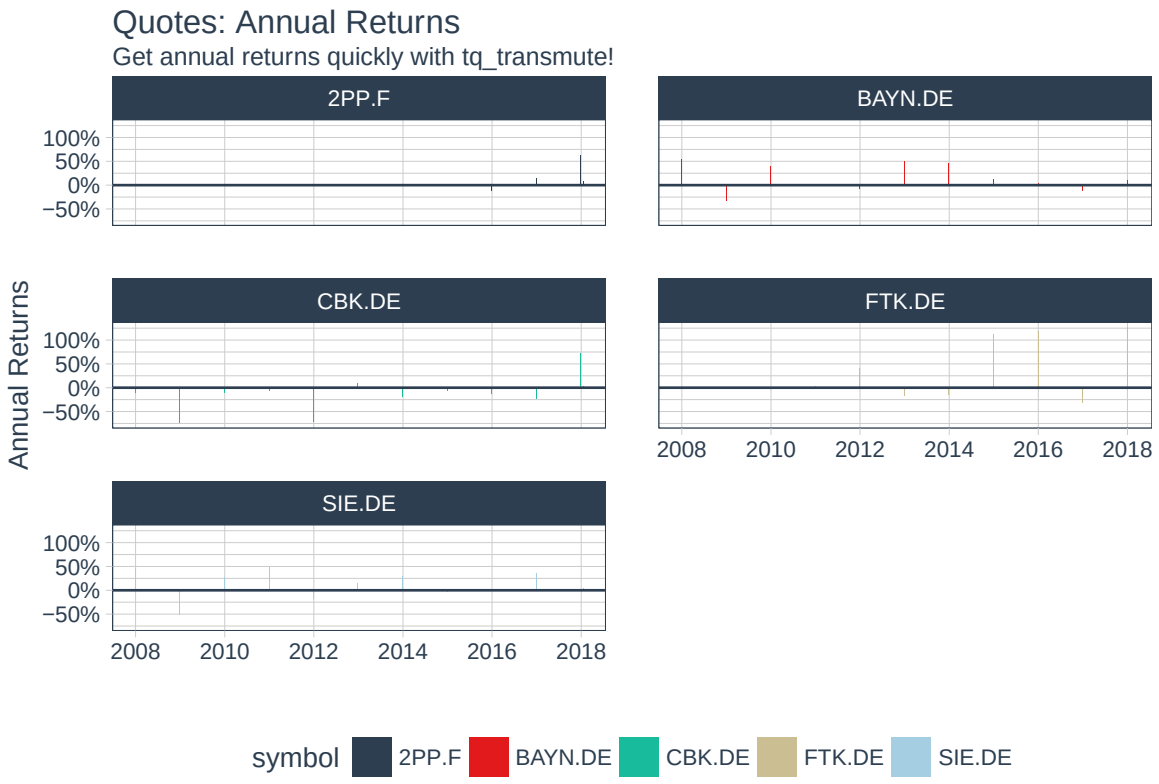
```
## Warning in to_period(xx, period = on.opts[[period]], ...): missing values
## removed from data

## Warning in to_period(xx, period = on.opts[[period]], ...): missing values
## removed from data

## Warning in to_period(xx, period = on.opts[[period]], ...): missing values
## removed from data
```

```
Quotes_annual_returns
```

```
## # A tibble: 49 x 3
## # Groups: symbol [5]
##    symbol date       yearly.returns
##    <chr>  <date>              <dbl>
##  1 SIE.DE 2007-12-28        0.460
##  2 SIE.DE 2008-12-30       -0.506
##  3 SIE.DE 2009-12-30        0.266
##  4 SIE.DE 2010-12-30        0.494
```

```
##  5 SIE.DE 2011-12-30       -0.178
##  6 SIE.DE 2012-12-28        0.158
##  7 SIE.DE 2013-12-30        0.294
##  8 SIE.DE 2014-12-30       -0.0263
##  9 SIE.DE 2015-12-30       -0.00850
## 10 SIE.DE 2016-12-30        0.352
## # ... with 39 more rows
```

```
plot3=Quotes_annual_returns %>%
    ggplot(aes(x = date, y = yearly.returns, fill = symbol)) +
    geom_bar(stat = "identity") +
    geom_hline(yintercept = 0, color = palette_light()[[1]]) +
    scale_y_continuous(labels = scales::percent) +
    labs(title = "Quotes: Annual Returns",
         subtitle = "Get annual returns quickly with tq_transmute!",
         y = "Annual Returns", x = "") +
         facet_wrap(~ symbol, ncol = 2) +
         theme_tq() +
         scale_fill_tq()
plot3
```

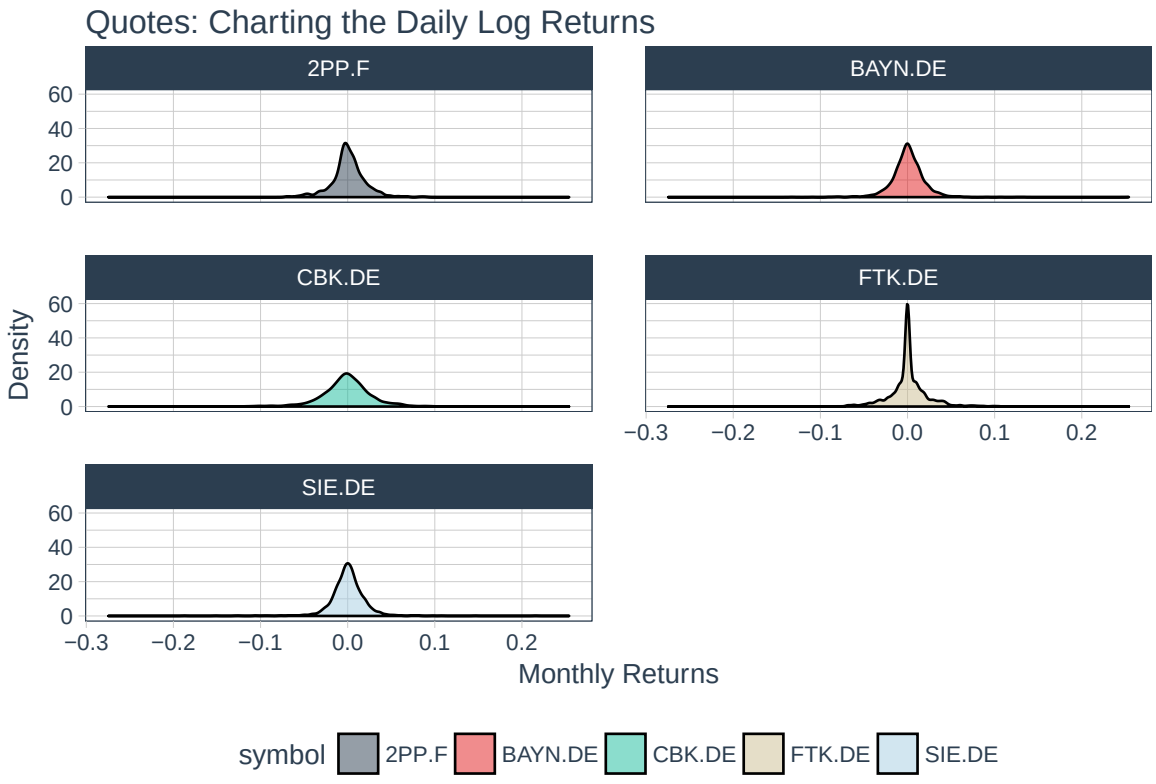



```
Quotes_daily_log_returns <- Quotes %>%
    group_by(symbol) %>%
    tq_transmute(select     = adjusted,
                 mutate_fun = periodReturn,
                 period     = "daily",
```

```
                type       = "log",
                col_rename = "monthly.returns")
```

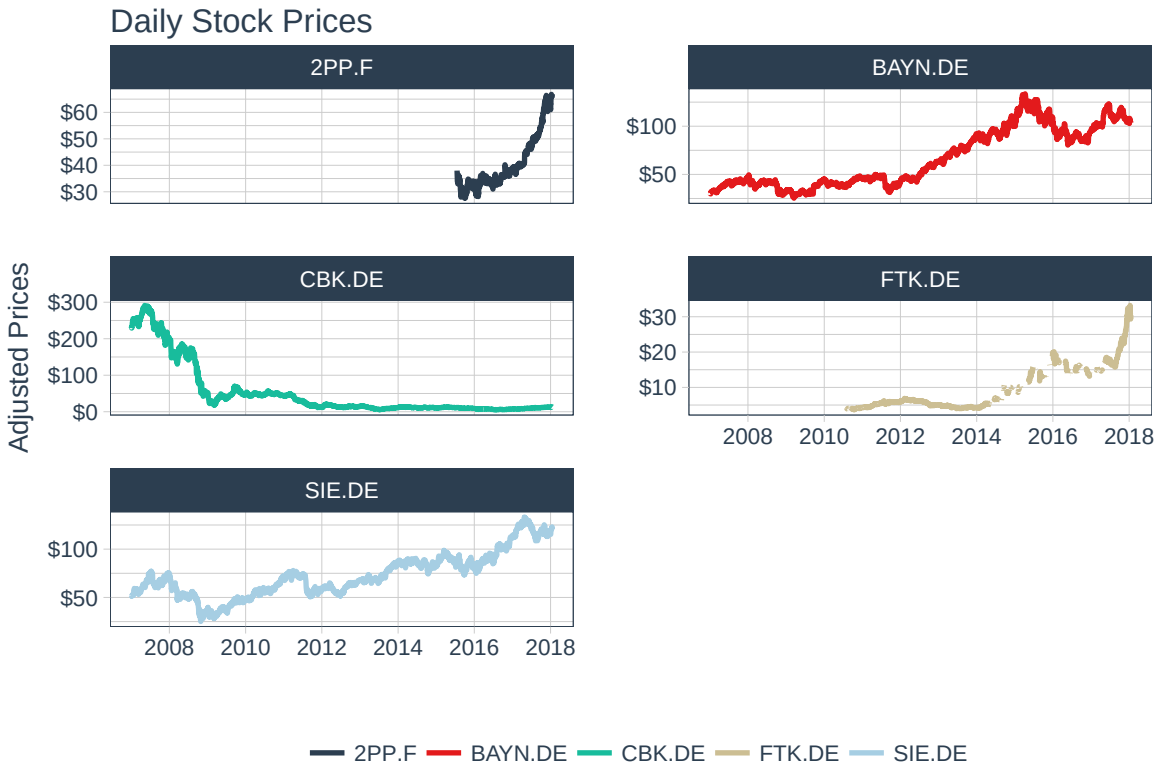
```
## Warning in to_period(xx, period = on.opts[[period]], ...): missing values
## removed from data

## Warning in to_period(xx, period = on.opts[[period]], ...): missing values
## removed from data

## Warning in to_period(xx, period = on.opts[[period]], ...): missing values
## removed from data
```

```
Quotes_daily_log_returns %>%
    ggplot(aes(x = monthly.returns, fill = symbol)) +
    geom_density(alpha = 0.5) +
    labs(title = "Quotes: Charting the Daily Log Returns",
         x = "Monthly Returns", y = "Density") +
         theme_tq() +
         scale_fill_tq() +
         facet_wrap(~ symbol, ncol = 2)
```




```
#Without Periodicity Aggregation

Quotes_daily <- Quotes %>%
    group_by(symbol)
```

```
Quotes_daily %>%
    ggplot(aes(x = date, y = adjusted, color = symbol)) +
    geom_line(size = 1) +
    labs(title = "Daily Stock Prices",
         x = "", y = "Adjusted Prices", color = "") +
         facet_wrap(~ symbol, ncol = 2, scales = "free_y") +
         scale_y_continuous(labels = scales::dollar) +
         theme_tq() +
         scale_color_tq()
```



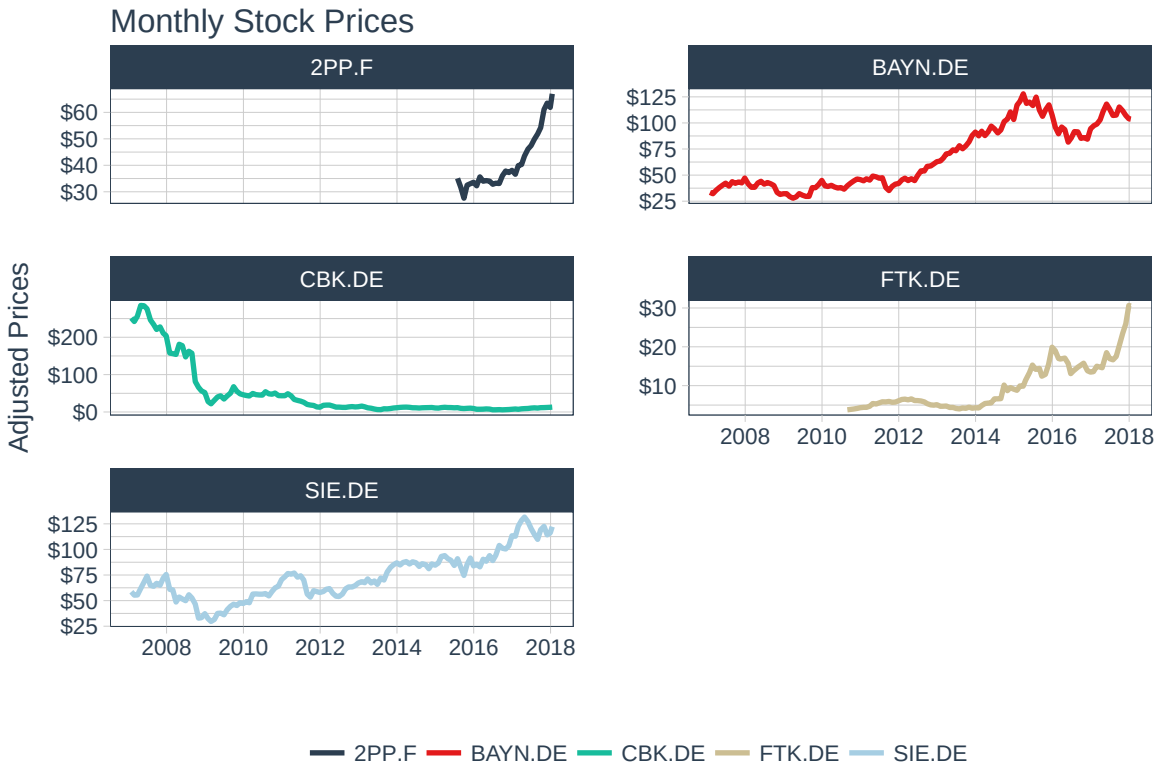

```
#With Periodicity Aggregation

Quotes_monthly <- Quotes %>%
    group_by(symbol) %>%
    tq_transmute(select     = adjusted,
                 mutate_fun = to.period,
                 period     = "months")
```

```
## Warning in fun_transmute(., OHLC = FALSE, ...): missing values removed from
## data
```

```
## Warning in fun_transmute(., OHLC = FALSE, ...): missing values removed from
## data
```

```
## Warning in fun_transmute(., OHLC = FALSE, ...): missing values removed from
## data
```

```
Quotes_monthly %>%
    ggplot(aes(x = date, y = adjusted, color = symbol)) +
    geom_line(size = 1) +
    labs(title = "Monthly Stock Prices",
         x = "", y = "Adjusted Prices", color = "") +
         facet_wrap(~ symbol, ncol = 2, scales = "free_y") +
         scale_y_continuous(labels = scales::dollar) +
         theme_tq() +
         scale_color_tq()
```




```
# Asset Returns
Quotes_returns_monthly <- Quotes %>%
    group_by(symbol) %>%
    tq_transmute(select     = adjusted,
                 mutate_fun = periodReturn,
                 period = "monthly",
                 col_rename = "Ra")
```

```
## Warning in to_period(xx, period = on.opts[[period]], ...): missing values
## removed from data
```

```
## Warning in to_period(xx, period = on.opts[[period]], ...): missing values
## removed from data
```

```
## Warning in to_period(xx, period = on.opts[[period]], ...): missing values
## removed from data
```

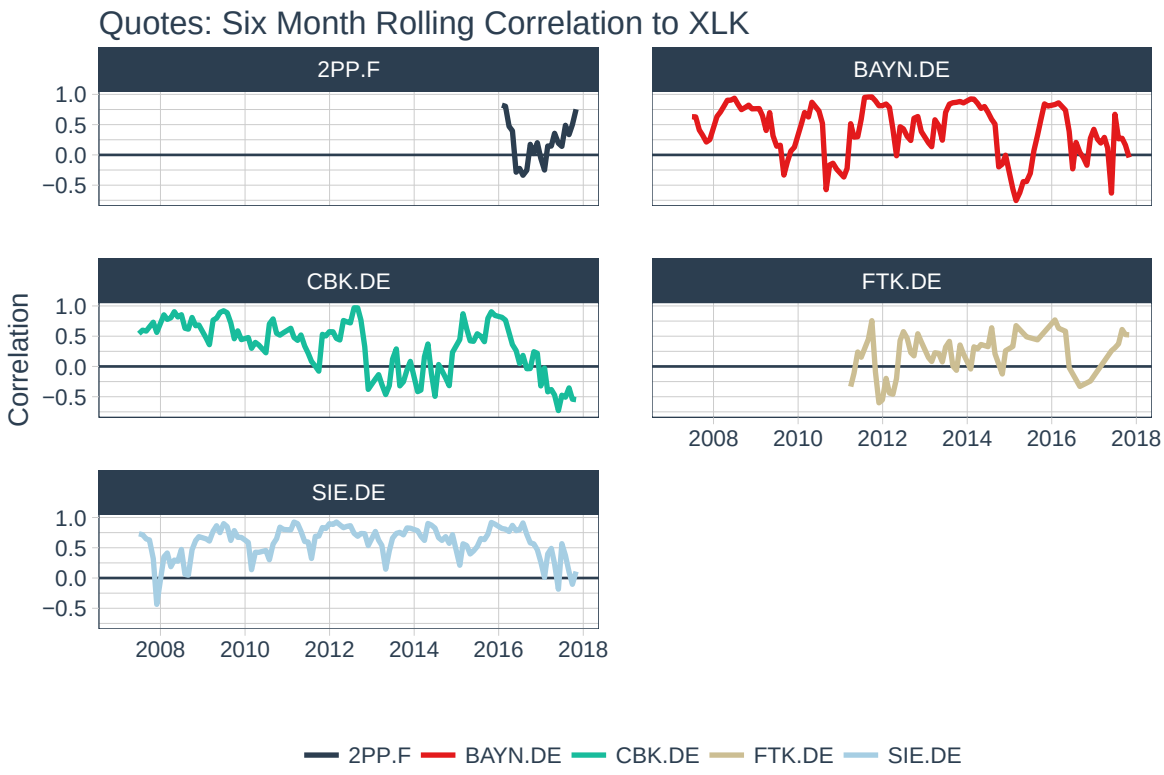
```
# Baseline Returns
baseline_returns_monthly <- "XLK" %>%
    tq_get(get  = "stock.prices",
           from = "2007-01-01",
           to   = "2017-11-01") %>%
           tq_transmute(select     = adjusted,
                 mutate_fun = periodReturn,
                 period = "monthly",
                 col_rename = "Rb")



#Next, join the asset returns with the baseline returns by date.
################################# make date seq from: getsymbols_irregular data.R
####################
#########################


returns_joined <- left_join(Quotes_returns_monthly,
                            baseline_returns_monthly,
                            by = "date")
returns_joined
```

```
## # A tibble: 520 x 4
## # Groups: symbol [?]
##    symbol date             Ra       Rb
##    <chr>  <date>        <dbl>    <dbl>
##  1 SIE.DE 2007-01-31  0.134    0.0171
##  2 SIE.DE 2007-02-28 -0.0563  -0.0270
##  3 SIE.DE 2007-03-30  0.00339  0.00996
##  4 SIE.DE 2007-04-30  0.115    0.0472
##  5 SIE.DE 2007-05-31  0.0980   0.0504
##  6 SIE.DE 2007-06-29  0.0881  -0.00125
##  7 SIE.DE 2007-07-31 -0.122   -0.00195
##  8 SIE.DE 2007-08-31 -0.0147   0.0192
##  9 SIE.DE 2007-09-28  0.0459   0.0378
## 10 SIE.DE 2007-10-31 -0.0280   0.0530
## # ... with 510 more rows
```

```
returns_joined <- na.omit(returns_joined)

Quotes_rolling_corr <- returns_joined %>%
    tq_transmute_xy(x          = Ra,
                    y          = Rb,
                    mutate_fun = runCor,
                    n          = 6,
                    col_rename = "rolling.corr.6")
#And, we can plot the rolling correlations for the Quotes stocks.

Quotes_rolling_corr %>%
    ggplot(aes(x = date, y = rolling.corr.6, color = symbol)) +
```

```
    geom_hline(yintercept = 0, color = palette_light()[[1]]) +
    geom_line(size = 1) +
    labs(title = "Quotes: Six Month Rolling Correlation to XLK",
         x = "", y = "Correlation", color = "") +
         facet_wrap(~ symbol, ncol = 2) +
         theme_tq() +
         scale_color_tq()
```

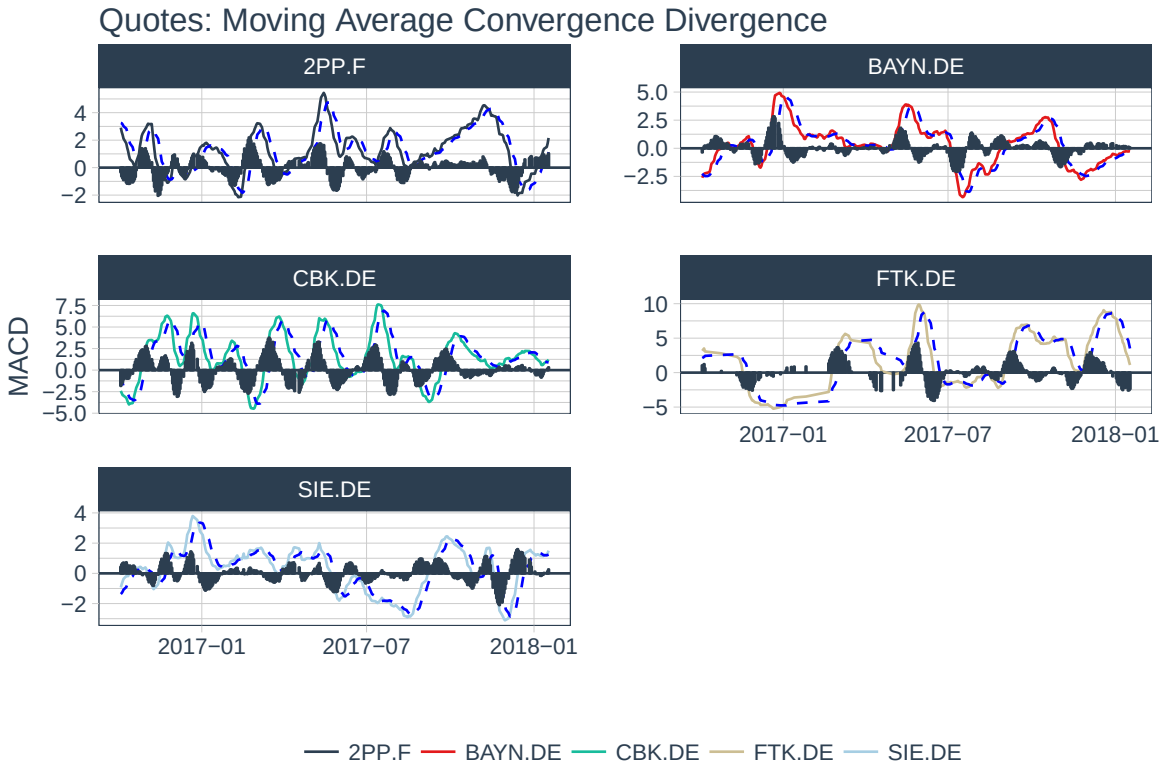
```
## Warning: Removed 25 rows containing missing values (geom_path).
```




```
Quotes <- na.omit(Quotes)

##Example 4: Use TTR MACD to Visualize Moving Average Convergence Divergence

Quotes_macd <- Quotes %>%
    group_by(symbol) %>%
    tq_mutate(select     = close,
              mutate_fun = MACD,
              nFast      = 12,
              nSlow      = 26,
              nSig       = 9,
              maType     = SMA) %>%
              mutate(diff = macd - signal) %>%
              select(-(open:volume))
Quotes_macd
```

```
## # A tibble: 10,493 x 6
## # Groups: symbol [5]
##    symbol date       adjusted  macd signal  diff
##    <chr>  <date>        <dbl> <dbl>  <dbl> <dbl>
##  1 SIE.DE 2007-01-02     51.7    NA     NA    NA
##  2 SIE.DE 2007-01-03     51.5    NA     NA    NA
##  3 SIE.DE 2007-01-04     51.5    NA     NA    NA
##  4 SIE.DE 2007-01-05     51.3    NA     NA    NA
##  5 SIE.DE 2007-01-08     51.4    NA     NA    NA
##  6 SIE.DE 2007-01-09     52.3    NA     NA    NA
##  7 SIE.DE 2007-01-10     51.8    NA     NA    NA
##  8 SIE.DE 2007-01-11     52.3    NA     NA    NA
##  9 SIE.DE 2007-01-12     52.8    NA     NA    NA
## 10 SIE.DE 2007-01-15     52.8    NA     NA    NA
## # ... with 10,483 more rows
```

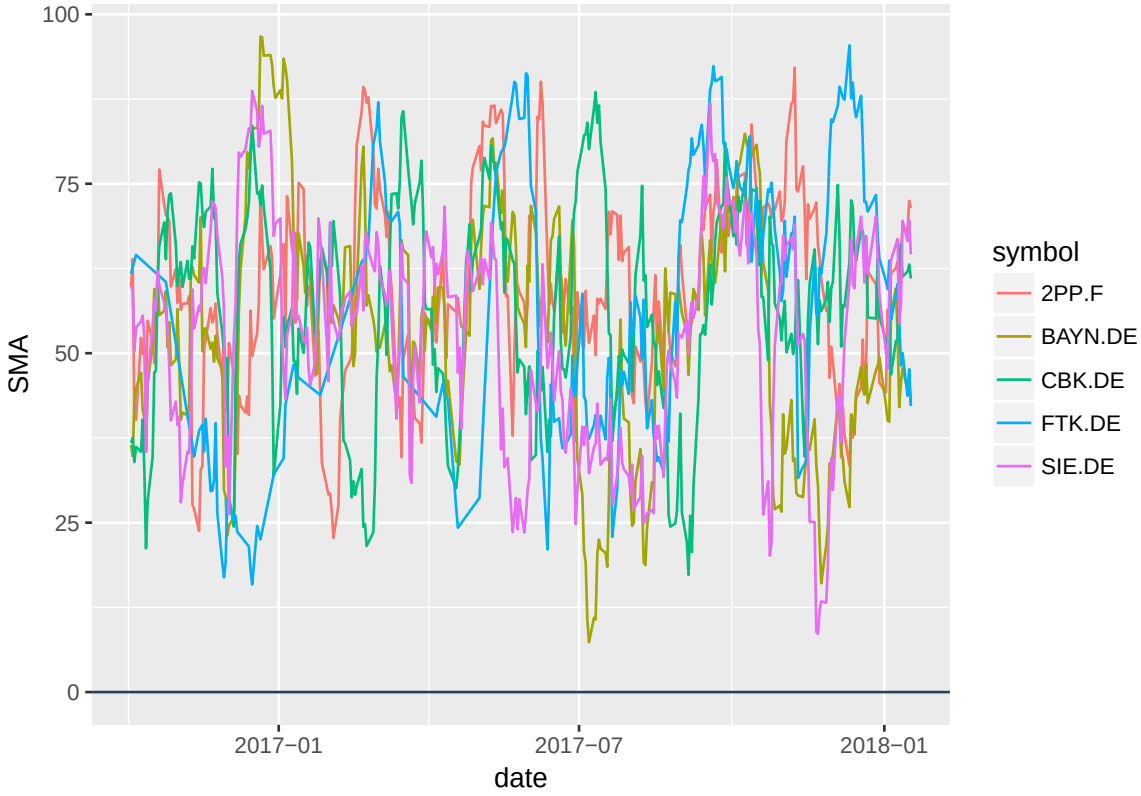
```
#And, we can visualize the data like so.

Quotes_macd %>%
    filter(date >= as_date("2016-10-01")) %>%
    ggplot(aes(x = date)) +
    geom_hline(yintercept = 0, color = palette_light()[[1]]) +
    geom_line(aes(y = macd, col = symbol)) +
    geom_line(aes(y = signal), color = "blue", linetype = 2) +
    geom_bar(aes(y = diff), stat = "identity", color = palette_light()[[1]]) +
    facet_wrap(~ symbol, ncol = 2, scale = "free_y") +
    labs(title = "Quotes: Moving Average Convergence Divergence",
         y = "MACD", x = "", color = "") +
         theme_tq() +
         scale_color_tq()
```




```
Quotes_macd <- Quotes %>%
    group_by(symbol) %>%
    tq_mutate(select     = close,
              mutate_fun = RSI,
              n          = 14,
              maType     = SMA)

Quotes_macd
```

```
## # A tibble: 10,493 x 9
## # Groups: symbol [5]
##    symbol date        open  high   low close  volume adjusted   SMA
##    <chr>  <date>     <dbl> <dbl> <dbl> <dbl>   <int>    <dbl> <dbl>
##  1 SIE.DE 2007-01-02  73.8  73.9  73.2  73.6 3211837     51.7    NA
##  2 SIE.DE 2007-01-03  73.5  73.6  73.0  73.2 3728309     51.5    NA
##  3 SIE.DE 2007-01-04  72.9  73.6  72.7  73.2 4802257     51.5    NA
##  4 SIE.DE 2007-01-05  72.9  74.0  72.9  73.0 5841660     51.3    NA
##  5 SIE.DE 2007-01-08  73.4  73.9  72.6  73.1 4553569     51.4    NA
##  6 SIE.DE 2007-01-09  73.8  75.5  73.2  74.4 8882191     52.3    NA
##  7 SIE.DE 2007-01-10  73.9  74.4  73.4  73.7 5438704     51.8    NA
##  8 SIE.DE 2007-01-11  74.0  74.6  73.0  74.4 5908520     52.3    NA
##  9 SIE.DE 2007-01-12  74.4  75.3  74.2  75.0 5062112     52.8    NA
## 10 SIE.DE 2007-01-15  75.4  75.5  75.1  75.1 2537080     52.8    NA
## # ... with 10,483 more rows
```

```
Quotes_macd %>%
    filter(date >= as_date("2016-10-01")) %>%
    ggplot(aes(x = date)) +
    geom_hline(yintercept = 0, color = palette_light()[[1]]) +
    geom_line(aes(y = SMA, col = symbol))
```




```
#Example 5: Use xts apply.quarterly to Get the Max and Min Price for Each Quarter



Quotes
```

```
##         symbol       date    open    high     low   close  volume adjusted
##      1: SIE.DE 2007-01-02 73.7916 73.8885 73.1714 73.5784 3211837 51.73898
##      2: SIE.DE 2007-01-03 73.5009 73.6366 72.9583 73.1714 3728309 51.45278
##      3: SIE.DE 2007-01-04 72.8710 73.5978 72.6579 73.2102 4802257 51.48006
##      4: SIE.DE 2007-01-05 72.9001 74.0339 72.8710 72.9873 5841660 51.32332
##      5: SIE.DE 2007-01-08 73.4331 73.8885 72.6385 73.1133 4553569 51.41192
##     ---
## 10489: FTK.DE 2018-01-11 33.4500 33.5000 30.9500 31.2500   53871 31.25000
## 10490: FTK.DE 2018-01-12 31.4500 31.5500 27.6000 30.4000  145317 30.40000
## 10491: FTK.DE 2018-01-15 30.3000 30.3500 28.9000 29.3500   57318 29.35000
## 10492: FTK.DE 2018-01-16 29.5500 31.3000 29.4000 30.5500   58608 30.55000
## 10493: FTK.DE 2018-01-17 30.2500 30.6000 29.7000 30.5000   70306 30.50000
```

```
Quotes_max_by_qtr <- Quotes %>%
    group_by(symbol) %>%
    tq_transmute(select = adjusted,
                 mutate_fun = apply.quarterly,
                 FUN = max,
                 col_rename = "max.close") %>%
                 mutate(year.qtr = paste0(year(date), "-Q", quarter(date))) %>%
                 select(-date)
Quotes_max_by_qtr
```

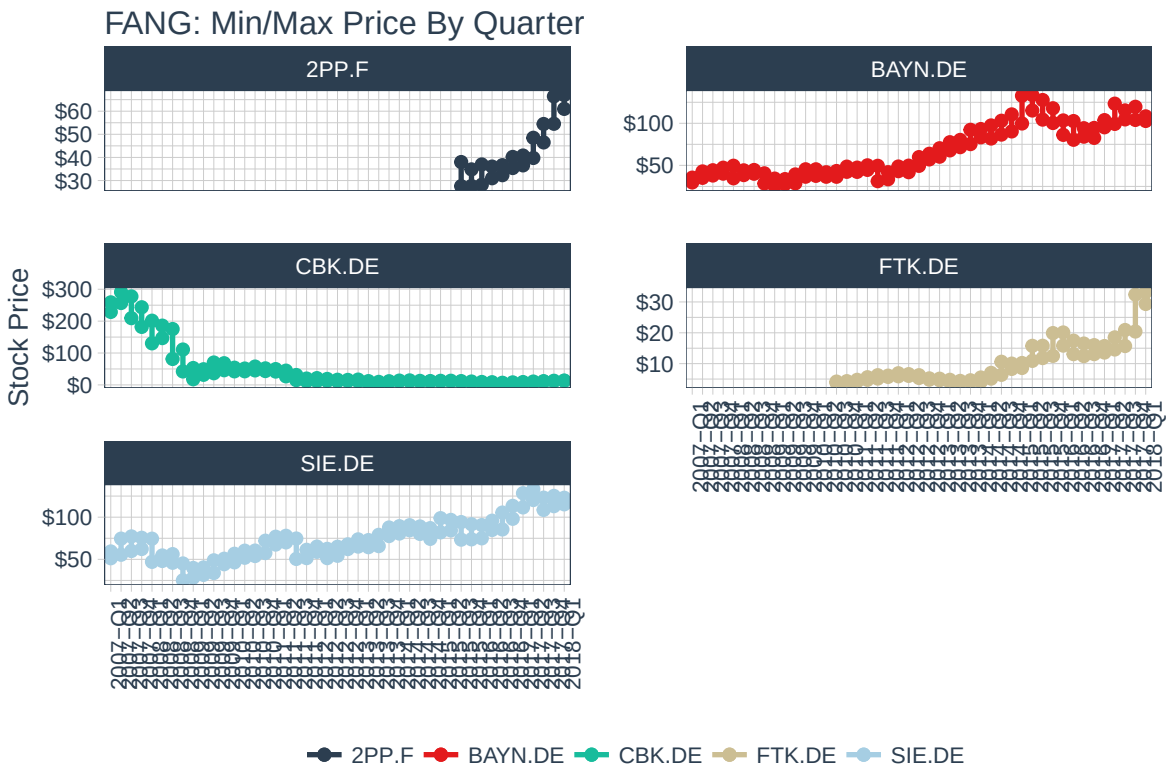
```
## # A tibble: 177 x 3
## # Groups: symbol [5]
##    symbol max.close year.qtr
##    <chr>      <dbl> <chr>
##  1 SIE.DE      59.3 2007-Q1
##  2 SIE.DE      74.5 2007-Q2
##  3 SIE.DE      77.1 2007-Q3
##  4 SIE.DE      75.5 2007-Q4
##  5 SIE.DE      74.5 2008-Q1
##  6 SIE.DE      54.6 2008-Q2
##  7 SIE.DE      56.2 2008-Q3
##  8 SIE.DE      45.1 2008-Q4
##  9 SIE.DE      39.8 2009-Q1
## 10 SIE.DE      40.4 2009-Q2
## # ... with 167 more rows
```

```
#The minimum each quarter can be retrieved in much the same way. The data frames can be joined using

Quotes_min_by_qtr <- Quotes %>%
    group_by(symbol) %>%
    tq_transmute(select = adjusted,
                 mutate_fun = apply.quarterly,
                 FUN = min,
                 col_rename = "min.close") %>%
                 mutate(year.qtr = paste0(year(date), "-Q", quarter(date))) %>%
                 select(-date)

Quotes_by_qtr <- left_join(Quotes_max_by_qtr, Quotes_min_by_qtr,
                           by = c("symbol" = "symbol",
                                  "year.qtr" = "year.qtr"))
Quotes_by_qtr
```

```
## # A tibble: 177 x 4
## # Groups: symbol [?]
##    symbol max.close year.qtr min.close
##    <chr>      <dbl> <chr>        <dbl>
##  1 SIE.DE      59.3 2007-Q1       51.3
##  2 SIE.DE      74.5 2007-Q2       55.5
##  3 SIE.DE      77.1 2007-Q3       59.9
##  4 SIE.DE      75.5 2007-Q4       62.3
##  5 SIE.DE      74.5 2008-Q1       47.0
##  6 SIE.DE      54.6 2008-Q2       48.1
```

```
##  7 SIE.DE      56.2 2008-Q3       45.9
##  8 SIE.DE      45.1 2008-Q4       25.1
##  9 SIE.DE      39.8 2009-Q1       28.2
## 10 SIE.DE      40.4 2009-Q2       31.6
## # ... with 167 more rows
```

```
byQuarterAll <- Quotes_by_qtr %>%
    ggplot(aes(x = year.qtr, color = symbol)) +
    geom_segment(aes(xend = year.qtr, y = min.close, yend = max.close),
                 size = 1) +
                 geom_point(aes(y = max.close), size = 2) +
                 geom_point(aes(y = min.close), size = 2) +
                 facet_wrap(~symbol, ncol = 2, scale = "free_y") +
                 labs(title = "FANG: Min/Max Price By Quarter",
         y = "Stock Price", color = "") +
         theme_tq() +
         scale_color_tq() +
         scale_y_continuous(labels = scales::dollar) +
         theme(axis.text.x = element_text(angle = 90, hjust = 1),
          axis.title.x = element_blank())
byQuarterAll
```



1.1 Portfolio

2 New weights table

```
#tidyverse
library(readxl)
xlsx_example <- read_excel("AllocationCurrent.xlsx")
xlsx_example
```

```
## # A tibble: 12 x 6
##    portfolio stocks Symbol_short Number CurrentPrice Value
##        <dbl> <chr>  <chr>         <dbl>        <dbl> <dbl>
##  1      1.00 CBK.DE CBK.DE         40.0        44.5   1779
##  2      1.00 2PP.F  2PP.F          30.0        36.6   1099
##  3      1.00 SIE.DE SIE.DE         20.0        33.1    661
##  4      1.00 FTK.DE FTK.DE         20.0        25.0    500
##  5      2.00 CBK.DE CBK.DE         40.0        44.5   1779
##  6      2.00 2PP.F  2PP.F          30.0        36.6   1099
##  7      2.00 SIE.DE SIE.DE         20.0        33.1    661
##  8      2.00 MFST   MFST           20.0        40.0    800
##  9      3.00 CBK.DE CBK.DE         40.0        37.5   1500
## 10      3.00 2PP.F  2PP.F          30.0         6.67   200
## 11      3.00 SIE.DE SIE.DE         20.0       120     2400
## 12      3.00 AAPL   AAPL           20.0        40.0    800
```

```
library(openxlsx)
Sys.setenv(R_ZIPCMD = "C:/RBuildTools/3.4/bin/zip")

#wb <- loadWorkbook(file = "OrderBook.xlsx")
#writeData(wb, sheet = 1, x = iris, name = "iris3", startCol = 10)
#Allocation <- read.xlsx("AllocationCurrent.xlsx", namedRegion = "Allocation", detectDates = T)
library(tidyverse)
Allocation <- read.xlsx("AllocationCurrent.xlsx", sheet = 1, startRow = 1, detectDates = T)
Allocation<-Allocation %>%
  group_by(portfolio) %>%
    mutate(weights=Value/sum(Value))


Allocation2 <- read.xlsx("AllocationCurrent.xlsx", sheet = 1, startRow = 1, detectDates = T)


distStocks<-Allocation2 %>%
 distinct(stocks)
distStocks
```

```
##   stocks
## 1 CBK.DE
## 2  2PP.F
## 3 SIE.DE
## 4 FTK.DE
## 5   MFST
```

```
## 6   AAPL
```

```
n = 3
stocks <- distStocks[,1]
weights_table <-  tibble(stocks) %>%
    tq_repeat_df(n = n) %>%
    group_by(portfolio) %>%
    left_join(Allocation) %>%
    replace_na(replace = list(Symbol_short="N/A",weights=0))
```

```
## Joining, by = c("portfolio", "stocks")
```

```
weights_table<-weights_table %>%
  select(portfolio,stocks,weights)

weights_table
```

```
## # A tibble: 18 x 3
## # Groups: portfolio [3]
##    portfolio stocks weights
##        <dbl> <chr>    <dbl>
##  1      1.00 CBK.DE  0.440
##  2      1.00 2PP.F   0.272
##  3      1.00 SIE.DE  0.164
##  4      1.00 FTK.DE  0.124
##  5      1.00 MFST    0
##  6      1.00 AAPL    0
##  7      2.00 CBK.DE  0.410
##  8      2.00 2PP.F   0.253
##  9      2.00 SIE.DE  0.152
## 10      2.00 FTK.DE  0
## 11      2.00 MFST    0.184
## 12      2.00 AAPL    0
## 13      3.00 CBK.DE  0.306
## 14      3.00 2PP.F   0.0408
## 15      3.00 SIE.DE  0.490
## 16      3.00 FTK.DE  0
## 17      3.00 MFST    0
## 18      3.00 AAPL    0.163
```

3 Check missing ticker

```
distStocks<-distStocks%>%
 rename(symbol=stocks)

distStocksAV<-Quotes %>%
 distinct(symbol)
distStocksAV
```

```
##     symbol
## 1:  SIE.DE
```

```
## 2:  CBK.DE
## 3:   2PP.F
## 4: BAYN.DE
## 5:  FTK.DE
```

```
Ticker<-distStocks %>%
  anti_join(distStocksAV)
```

```
## Joining, by = "symbol"
```

3.1 Get intraday data from Alpha Vantage

```
Quotes_complete=Quotes
## Source Telegram API
source("R:/7_DataScience/01_R/12_AlphaVantage/alpha_key.R")
## Get data
for (i in 1:nrow(Ticker)){

S=Ticker[,1][i]
#S="MFST"
QuoteAV<- av_get(symbol = S, av_fun = "TIME_SERIES_DAILY", outputsize = "full")
QuoteAV<- QuoteAV %>%
  mutate(symbol=S)


names(QuoteAV)[1]<-"date"


Quotes_complete<- Quotes_complete %>%
  bind_rows(QuoteAV)
}
```

```
Quotes_returns_monthly_complete <- Quotes_complete %>%
    group_by(symbol) %>%
    tq_transmute(select     = close,
                 mutate_fun = periodReturn,
                 period = "monthly",
                 col_rename = "Ra")
```

4 Last day of month

```
library(lubridate)

data=Quotes_returns_monthly_complete
data$date2 <- ymd(data$date)
data$DATE_Month_End <- paste(format(data$date, format="%y-%m"),"-", days_in_month(data$date), sep="")
#OR
last_day <- function(date) {
  ceiling_date(date, "month") - days(1)
```

```
}


data2<-data %>%
  mutate(test=last_day(date))%>%
  select(symbol,test,Ra)

test2=spread(data2, symbol, Ra)
```

```
# Single Portfolio depending on n
#https://cran.r-project.org/web/packages/tidyquant/vignettes/TQ05-performance-analysis-with-tidyquant

ticker=distStocks

#stock_returns_monthly <- ticker %>%
#    tq_get(get  = "stock.prices",
#           from = "2010-01-01",
#           to   = "2015-12-31") %>%
#           group_by(symbol) %>%
#           tq_transmute(select     = adjusted,
#                 mutate_fun = periodReturn,
#                 period     = "monthly",
#                 col_rename = "Ra")

#replace
stock_returns_monthly=Quotes_returns_monthly_complete



#Second, get baseline asset returns, which is the exact same as Steps 1B and 2B from the Single Portf
#baseline_returns_monthly <- "XLK" %>%
#baseline_returns_monthly <- "GDAXI" %>%
#    tq_get(get  = "stock.prices",
#           from = "2010-01-01",
 #          to   = "2015-12-31") %>%
  #         tq_transmute(select     = adjusted,
   #              mutate_fun = periodReturn,
    #             period     = "monthly",
      #           col_rename = "Rb")

baseline_returns_monthly
```

```
## # A tibble: 130 x 2
##    date             Rb
##    <date>        <dbl>
##  1 2007-01-31  0.0171
##  2 2007-02-28 -0.0270
##  3 2007-03-30  0.00996
##  4 2007-04-30  0.0472
##  5 2007-05-31  0.0504
##  6 2007-06-29 -0.00125
##  7 2007-07-31 -0.00195
```

```
##  8 2007-08-31  0.0192
##  9 2007-09-28  0.0378
## 10 2007-10-31  0.0530
## # ... with 120 more rows
```

```
stock_returns_monthly_multi <- stock_returns_monthly %>%
    tq_repeat_df(n = n)
```

```
## Ungrouping data frame groups: symbol
```

```
stock_returns_monthly_multi
```

```
## # A tibble: 2,406 x 4
## # Groups: portfolio [3]
##    portfolio symbol date             Ra
##        <int> <chr>  <date>        <dbl>
##  1         1 SIE.DE 2007-01-31  0.113
##  2         1 SIE.DE 2007-02-28 -0.0563
##  3         1 SIE.DE 2007-03-30  0.00339
##  4         1 SIE.DE 2007-04-30  0.115
##  5         1 SIE.DE 2007-05-31  0.0980
##  6         1 SIE.DE 2007-06-29  0.0881
##  7         1 SIE.DE 2007-07-31 -0.122
##  8         1 SIE.DE 2007-08-31 -0.0147
##  9         1 SIE.DE 2007-09-28  0.0459
## 10         1 SIE.DE 2007-10-31 -0.0280
## # ... with 2,396 more rows
```

```
#Erros massage when not all assets available
portfolio_returns_monthly_multi <- stock_returns_monthly_multi %>%
    tq_portfolio(assets_col  = symbol,
                 returns_col = Ra,
                 weights     = weights_table,
                 col_rename  = "Ra")
```

```
## Warning in PerformanceAnalytics::Return.portfolio(., weights = weights, :
## NA's detected: filling NA's with zeros

## Warning in PerformanceAnalytics::Return.portfolio(., weights = weights, :
## NA's detected: filling NA's with zeros

## Warning in PerformanceAnalytics::Return.portfolio(., weights = weights, :
## NA's detected: filling NA's with zeros
```

```
portfolio_returns_monthly_multi
```

```
## # A tibble: 747 x 3
## # Groups: portfolio [3]
##    portfolio date          Ra
##        <dbl> <date>     <dbl>
##  1      1.00 2000-01-31     0
##  2      1.00 2000-02-29     0
##  3      1.00 2000-03-31     0
##  4      1.00 2000-04-28     0
##  5      1.00 2000-05-31     0
```

```
##  6      1.00 2000-06-30     0
##  7      1.00 2000-07-31     0
##  8      1.00 2000-08-31     0
##  9      1.00 2000-09-29     0
## 10      1.00 2000-10-31     0
## # ... with 737 more rows
```

```
RaRb_multiple_portfolio <- left_join(portfolio_returns_monthly_multi,
                                     baseline_returns_monthly,
                                     by = "date")
RaRb_multiple_portfolio
```

```
## # A tibble: 747 x 4
## # Groups: portfolio [?]
##    portfolio date          Ra    Rb
##        <dbl> <date>     <dbl> <dbl>
##  1      1.00 2000-01-31     0    NA
##  2      1.00 2000-02-29     0    NA
##  3      1.00 2000-03-31     0    NA
##  4      1.00 2000-04-28     0    NA
##  5      1.00 2000-05-31     0    NA
##  6      1.00 2000-06-30     0    NA
##  7      1.00 2000-07-31     0    NA
##  8      1.00 2000-08-31     0    NA
##  9      1.00 2000-09-29     0    NA
## 10      1.00 2000-10-31     0    NA
## # ... with 737 more rows
```

```
RaRb_multiple_portfolio <- na.omit(RaRb_multiple_portfolio)
```

```
library(tidyquant)
P_CAPM=RaRb_multiple_portfolio %>%
    tq_performance(Ra = Ra, Rb = Rb, performance_fun = table.CAPM)
P_CAPM
```

```
## # A tibble: 3 x 13
## # Groups: portfolio [3]
##   portfolio ActivePremium    Alpha AnnualizedAlpha  Beta `Beta-` `Beta+`
##       <dbl>         <dbl>    <dbl>           <dbl> <dbl>   <dbl>   <dbl>
## 1      1.00        -0.116 -0.00500         -0.0589 0.564   0.695   0.456
## 2      2.00        -0.146  0.00310          0.0383 0.699   0.685  -0.258
## 3      3.00        -0.102 -0.00630         -0.0727 1.09    1.14    1.30
## # ... with 6 more variables: Correlation <dbl>, `Correlationp-value`
## #   <dbl>, InformationRatio <dbl>, `R-squared` <dbl>, TrackingError <dbl>,
## #   TreynorRatio <dbl>
```

```
P_SR=RaRb_multiple_portfolio %>%
    tq_performance(Ra = Ra, Rb = NULL, performance_fun = SharpeRatio)
```

```
## VaR calculation produces unreliable result (inverse risk) for column: 1 : -0.867981002338723
```

```
## ES calculation produces unreliable result (risk over 100%) for column: 1 : 1.96110630304345
```

```
P_VaR <- RaRb_multiple_portfolio %>%
    tq_performance(Ra = Ra, Rb = NULL, performance_fun = VaR, portfolio_method="component")
```

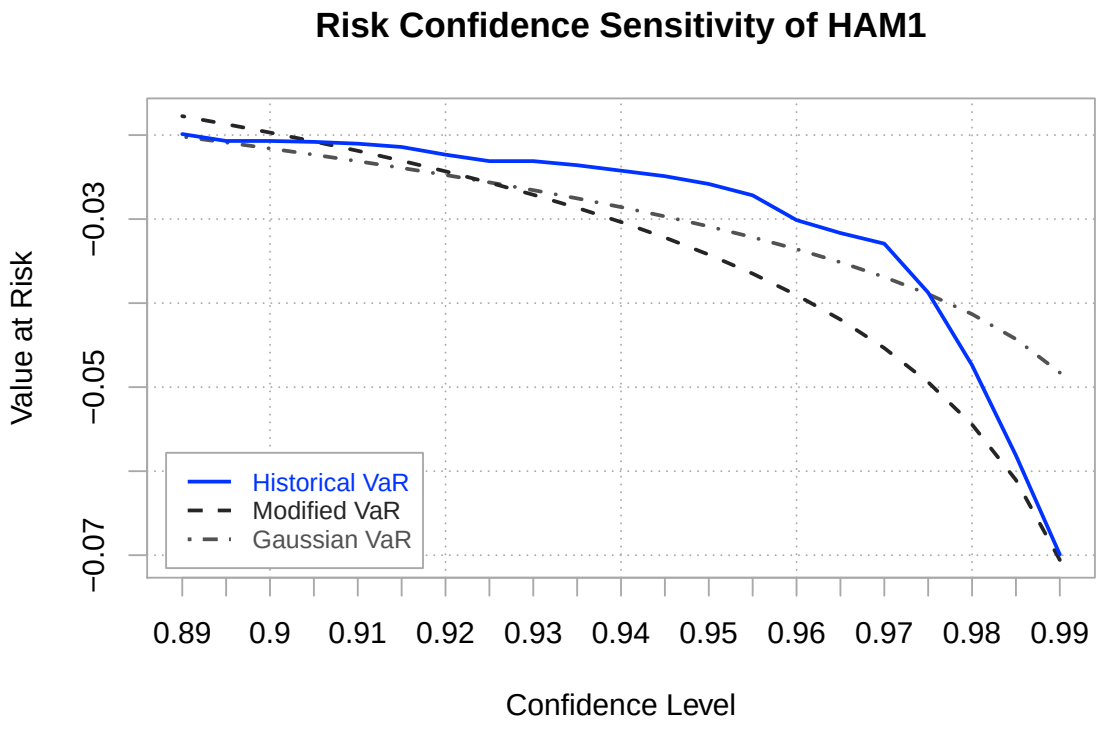
```
## no weights passed in, assuming equal weighted portfolio
```

```
## no weights passed in, assuming equal weighted portfolio
## no weights passed in, assuming equal weighted portfolio
```

```
P_VaR
```

```
## # A tibble: 3 x 4
## # Groups: portfolio [3]
##   portfolio contribution MVaR          pct_contrib_MVaR
##       <dbl> <list>       <list>        <list>
## 1      1.00 <dbl [1]>    <dbl [1 x 1]> <dbl [1 x 1]>
## 2      2.00 <dbl [1]>    <dbl [1 x 1]> <dbl [1 x 1]>
## 3      3.00 <dbl [1]>    <dbl [1 x 1]> <dbl [1 x 1]>
```

```
#tq_performance_fun_options()

P_Stats<-RaRb_multiple_portfolio %>%
    tq_performance(Ra = Ra, Rb = NULL, performance_fun = table.Stats)

data(managers)
chart.VaRSensitivity(managers[,1,drop=FALSE],
        methods=c("HistoricalVaR", "ModifiedVaR", "GaussianVaR"),
        colorset=bluefocus, lwd=2)
```



VaR(R, p = 0.95, method = c("modified", "gaussian","historical", "kernel"), clean = c("none", "boudt", "geltner"), portfolio_method = c("single", "component","marginal"), weights = NULL, mu = NULL, sigma = NULL, m3 = NULL, m4 = NULL, invert = TRUE, ...)

4.1 Customizing tq_performance

```
library(knitr)
```

```
## Warning: package 'knitr' was built under R version 3.4.3
```

```
library(kableExtra)
```

```
## Warning: package 'kableExtra' was built under R version 3.4.3
```

```
args(SharpeRatio)
```

```
## function (R, Rf = 0, p = 0.95, FUN = c("StdDev", "VaR", "ES"),
##     weights = NULL, annualize = FALSE, ...)
## NULL
```

```
RaRb_multiple_portfolio %>%
    tq_performance(Ra              = Ra,
                   performance_fun = SharpeRatio)
```

```
## VaR calculation produces unreliable result (inverse risk) for column: 1 : -0.867981002338723
```

```
## ES calculation produces unreliable result (risk over 100%) for column: 1 : 1.96110630304345
```

```
## # A tibble: 3 x 4
## # Groups: portfolio [3]
##   portfolio `ESSharpe(Rf=0%,p=95%)` `StdDevSharpe(Rf=0~ `VaRSharpe(Rf=0%,~
##       <dbl>                   <dbl>                <dbl>              <dbl>
## 1      1.00                 0.00531               0.0155            0.00902
## 2      2.00                 0.0102                0.0467           NA
## 3      3.00                 0.0133                0.0589            0.0313
```

```
tbl=RaRb_multiple_portfolio %>%
    tq_performance(Ra              = Ra,
                   performance_fun = SharpeRatio,
                   Rf              = 0.03 / 12)
```

```
## VaR calculation produces unreliable result (inverse risk) for column: 1 : -0.867981002338723
## ES calculation produces unreliable result (risk over 100%) for column: 1 : 1.96110630304345
```

```
kable(tbl)
```

portfolio	ESSharpe(Rf=0.2%,p=95%)	StdDevSharpe(Rf=0.2%,p=95%)	VaRSharpe(Rf=0.2%,p=95%)
1	-0.0143625	-0.0419905	-0.0243762
2	0.0077305	0.0352679	NA
3	0.0063251	0.0280297	0.0149266

```
kable(tbl, format = "latex", booktabs = T)
```

portfolio	ESSharpe(Rf=0.2%,p=95%)	StdDevSharpe(Rf=0.2%,p=95%)	VaRSharpe(Rf=0.2%,p=95%)
1	-0.0143625	-0.0419905	-0.0243762
2	0.0077305	0.0352679	NA
3	0.0063251	0.0280297	0.0149266

```r
dt <- mtcars[1:5, 1:6]

dt %>%
  kable("html") %>%
  kable_styling()
```

mpg

cyl

disp

hp

drat

wt

Mazda RX4

21.0

6

160

110

3.90

2.620

Mazda RX4 Wag

21.0

6

160

110

3.90

2.875

Datsun 710

22.8

4

108

93

3.85

2.320

Hornet 4 Drive

21.4

6

258

110

3.08

3.215

Hornet Sportabout

18.7

8

360

175

3.15

3.440

```
kable(dt, format = "latex", booktabs = T)
```

	mpg	cyl	disp	hp	drat	wt
Mazda RX4	21.0	6	160	110	3.90	2.620
Mazda RX4 Wag	21.0	6	160	110	3.90	2.875
Datsun 710	22.8	4	108	93	3.85	2.320
Hornet 4 Drive	21.4	6	258	110	3.08	3.215
Hornet Sportabout	18.7	8	360	175	3.15	3.440

5 CHARTS

6 CHARTS

7 Multiple Portfolios

```
portfolio_growth_monthly_multi <- stock_returns_monthly_multi %>%
    tq_portfolio(assets_col = symbol,
                 returns_col = Ra,
                 weights = weights_table,
                 col_rename = "investment.growth",
                 wealth.index = TRUE) %>%
                 mutate(investment.growth = investment.growth * 10000)
```

```
## Warning in PerformanceAnalytics::Return.portfolio(., weights = weights, :
## NA's detected: filling NA's with zeros

## Warning in PerformanceAnalytics::Return.portfolio(., weights = weights, :
## NA's detected: filling NA's with zeros

## Warning in PerformanceAnalytics::Return.portfolio(., weights = weights, :
## NA's detected: filling NA's with zeros
```

```
portfolio_growth_monthly_multi %>%
    ggplot(aes(x = date, y = investment.growth, color = factor(portfolio))) +
    geom_line(size = 2) +
    labs(title = "Portfolio Growth",
         subtitle = "Comparing Multiple Portfolios",
         caption = "Portfolio 3 is a Standout!",
         x = "", y = "Portfolio Value",
         color = "Portfolio") +
         geom_smooth(method = "loess") +
         theme_tq() +
         scale_color_tq() +
         scale_y_continuous(labels = scales::dollar)
```


```
tq_performance_fun_options()
```

```
## $table.funs
##  [1] "table.AnnualizedReturns" "table.Arbitrary"
##  [3] "table.Autocorrelation"   "table.CAPM"
##  [5] "table.CaptureRatios"     "table.Correlation"
##  [7] "table.Distributions"     "table.DownsideRisk"
##  [9] "table.DownsideRiskRatio" "table.DrawdownsRatio"
## [11] "table.HigherMoments"     "table.InformationRatio"
## [13] "table.SFM"               "table.SpecificRisk"
## [15] "table.Stats"             "table.TrailingPeriods"
## [17] "table.UpDownRatios"      "table.Variability"
```

```
##
## $CAPM.funs
##  [1] "CAPM.alpha"       "CAPM.beta"        "CAPM.beta.bear"
##  [4] "CAPM.beta.bull"   "CAPM.CML"         "CAPM.CML.slope"
##  [7] "CAPM.dynamic"     "CAPM.epsilon"     "CAPM.jensenAlpha"
## [10] "CAPM.RiskPremium" "CAPM.SML.slope"   "TimingRatio"
## [13] "MarketTiming"
##
## $SFM.funs
## [1] "SFM.alpha"       "SFM.beta"        "SFM.CML"         "SFM.CML.slope"
## [5] "SFM.dynamic"     "SFM.epsilon"     "SFM.jensenAlpha"
##
## $descriptive.funs
## [1] "mean"           "sd"             "min"            "max"
## [5] "cor"            "mean.geometric" "mean.stderr"    "mean.LCL"
## [9] "mean.UCL"
##
## $annualized.funs
## [1] "Return.annualized"        "Return.annualized.excess"
## [3] "sd.annualized"            "SharpeRatio.annualized"
##
## $VaR.funs
## [1] "VaR"  "ES"   "ETL"  "CDD"  "CVaR"
##
## $moment.funs
##  [1] "var"              "cov"              "skewness"
##  [4] "kurtosis"         "CoVariance"       "CoSkewness"
##  [7] "CoSkewnessMatrix" "CoKurtosis"       "CoKurtosisMatrix"
## [10] "M3.MM"            "M4.MM"            "BetaCoVariance"
## [13] "BetaCoSkewness"   "BetaCoKurtosis"
##
## $drawdown.funs
## [1] "AverageDrawdown"   "AverageLength"     "AverageRecovery"
## [4] "DrawdownDeviation" "DrawdownPeak"      "maxDrawdown"
##
## $Bacon.risk.funs
## [1] "MeanAbsoluteDeviation" "Frequency"             "SharpeRatio"
## [4] "MSquared"              "MSquaredExcess"        "HurstIndex"
##
## $Bacon.regression.funs
##  [1] "CAPM.alpha"       "CAPM.beta"        "CAPM.epsilon"
##  [4] "CAPM.jensenAlpha" "SystematicRisk"   "SpecificRisk"
##  [7] "TotalRisk"        "TreynorRatio"     "AppraisalRatio"
## [10] "FamaBeta"         "Selectivity"      "NetSelectivity"
##
## $Bacon.relative.risk.funs
## [1] "ActivePremium"    "ActiveReturn"     "TrackingError"
## [4] "InformationRatio"
##
## $Bacon.drawdown.funs
## [1] "PainIndex"     "PainRatio"     "CalmarRatio"   "SterlingRatio"
```

```
## [5] "BurkeRatio"    "MartinRatio"   "UlcerIndex"
##
## $Bacon.downside.risk.funs
##  [1] "DownsideDeviation"     "DownsidePotential"
##  [3] "DownsideFrequency"     "SemiDeviation"
##  [5] "SemiVariance"          "UpsideRisk"
##  [7] "UpsidePotentialRatio"  "UpsideFrequency"
##  [9] "BernardoLedoitRatio"   "DRatio"
## [11] "Omega"                 "OmegaSharpeRatio"
## [13] "OmegaExcessReturn"     "SortinoRatio"
## [15] "M2Sortino"             "Kappa"
## [17] "VolatilitySkewness"    "AdjustedSharpeRatio"
## [19] "SkewnessKurtosisRatio" "ProspectRatio"
##
## $misc.funs
## [1] "KellyRatio"   "Modigliani"   "UpDownRatios"
```