Visualization

Stefan Glogger

August 2017

Visualization

Functions

Evaluation of Varying Portfolio

We want to visualize the evolvement of a portfolio over each time window.

Be aware of the index shifting: retPlot[j-1, i] take wealth of previous day retOverTime[j-1,] take return of today (j is one step ahead)

Remove numbering of x-axis by *xaxt='n'*.

Generate *retPortSentixVarying*, the returns of portfolios with varying portfolio weights using sentix as third factor with optimal weights. It has the following structure:

time window -> dispersion (sentixGroup) -> return of Portfolio, sharpe ratio

x: portfolio weights R: return of portfolio on each date r: mean return of portfolio over whole time window sd: standard deviation of return of portfolio over whole time window sr: sharpe ratio (weekly) anR: annualized return of portfolio over whole time window anSd: annualized standard deiation anSR: sharpe ratio (annual) fweight: mean of goal function value turnover: turnover of weights (how much of portfolio has to be changed) in percent

Calculation of turnover: NOT USEFUL (not comparable as portfolios evolve differently): we fix portfolio weights in t, hold these weights to t+1 (while portfolio raises to (1+ret)*price_t), and may change weightsin t+1 -> amount changed is (change in weights) * (value of index in t+1) USEFUL: we calculate the percentage points of weights that change in each time step. Divide by 2 as a percentage point is taken from one part of the portfolio and given to another part (so counted twice) -> get amount of portfolio that changes

Use an adoption of *calcTestVar()*

```
calcEvalVarClassic <- function(dat){
    res <- list()
    for(timeWindowName in names(dat)){
        timeWindow <- get(timeWindowName)
        retTimeWindow <- ret[timeWindow,]
        retTimeWindow <- retTimeWindow[-1,]
        colnames(retTimeWindow) <- colnames(ret)

        rf <- mean(retTimeWindow[,"BUND"])

        for(portfolioName in names(dat[[timeWindowName]])){


            R <- rowSums(dat[[timeWindowName]][[portfolioName]]$x * retTimeWindow)

            turnover <- c(0, rowSums(abs(diff(dat[[timeWindowName]][[portfolioName]]$x)))/2) # start of

            r <- mean(R)
            sd <- sd(R)

            anR <- (1+r)^52-1
            anSd <- sqrt((sd^2)*52)


            res[[timeWindowName]][[portfolioName]] <- list(x = dat[[timeWindowName]][[portfolioName]]$x
                                                           R = R, r = r, sd = sd, sr = r/sd,
                                                           anR = anR, anSd = anSd, anSR = anR/anSd,
                                                           turnover = turnover)

        }
    }
    return(res)
}
```

difference in function is that fweight is not there for classic portfolios

```
calcEvalVarSentix <- function(dat){
    res <- calcEvalVarClassic(dat)
    for(timeWindowName in names(dat)){
        for(portfolioName in names(dat[[timeWindowName]])){
            fweight = mean(dat[[timeWindowName]][[portfolioName]]$obj)
            res[[timeWindowName]][[portfolioName]]$fweight <- fweight
        }
    }
    return(res)
}
```

plot performance

We now optimize the plotting for ggplot(). (DOESN'T WORK)

Therefore our dataframe to plot should have the following structure: date: Date value: worth of Portfolio portfolio: Portfolio (SentixGroup)

first in separate list, then in one dataframe

NOTE: returns occur one date later as stated here (in the data)

There has been an issue with *date*. It is getted as character and we need to transform it to integer and then back to date to store date as a numeric value (formatted as a date) and then used as x-axis

```
plotPortfolio <- function(data, timeWindowName){
    datWork <- data[[timeWindowName]]
    timeWindow <- get(timeWindowName)

    colBackground <- colsEvalDates[timeWindowName]

    retPlot <- data.frame(date = as.integer(as.Date(get(timeWindowName))))# date is read as character,
    class(retPlot$date) <- "Date"

    for(s in names(datWork)){
        ret <- cumprod(1+datWork[[s]]$R)
        retPlot[[s]] <- c(100, 100*ret)
    }
    ggplot(retPlot, aes(x=date))+
        geom_line(aes(y=retPlot[,2], color = colnames(retPlot)[2]))


    plotCommand <- paste0(text = "ggplot(retPlot, aes(x=date))+")
    for (i in 2:(ncol(retPlot)-1)){
        plotCommand <- paste0(plotCommand, "geom_line(aes(y=retPlot[,",i,"], color = colnames(retPlot)[
    }
    plotCommand <- paste0(plotCommand, "geom_line(aes(y=retPlot[,",ncol(retPlot),"], color = colnames(r

    eval(parse(text = plotCommand))+
        labs(title = paste("Time:", timeWindowName),
             y = "Value",
             x = "Date") +
        scale_color_discrete(name = "Index")+
        theme(panel.background = element_rect(fill = alpha(colBackground, 0.2)))
}
```

```
plotPortfolioComplete <- function(dat, fileName){
    lateximport <- c(paste0("\\subsection{",fileName,"}"))

    for(d in datesEvalNames){
        plotPortfolio(dat, d)

        title <- paste0(fileName, "-", d, ".pdf")
        pdf(file.path(getwd(), "Plot", title), width = 10, height = 4)
        plot(plotPortfolio(dat, d))
        dev.off()

        lateximport <- c(lateximport, paste0("\\includegraphics[width=\\textwidth]{",title,"}"))
    }

    fileConnection <- file(file.path(getwd(), "Plot", paste0("0",fileName,".txt")))
    writeLines(lateximport, fileConnection)
    close(fileConnection)
}
```

change of weights

```
plotWeightsLines <- function(datName, d, s){
    dat <- datName[[d]][[s]]$x
    dat <- as.data.frame(dat)
    dat$date <- as.Date(rownames(dat))
    plotCommand <- paste0("ggplot(dat, aes(x=date)) +")

    for(i in 1:(ncol(dat)-2)){
        plotCommand <- paste0(plotCommand, "geom_line(aes(y=dat[,",i,"], color = colnames(dat)[", i, "]
    }
    plotCommand <- paste0(plotCommand, "geom_line(aes(y=dat[,",ncol(dat)-1,"], color = colnames(dat)[",

    eval(parse(text = plotCommand))+
        labs(title = paste("Time:", d),
             subtitle = paste("Portfolio:", s),
             y = "Weight",
             x = "Date") +
        scale_color_discrete(name = "Index")
}
```

```
plotWeightsLinesComplete <- function(dat, fileName){
    lateximport <- c(paste0("\\subsection{",fileName,"}"))

    for(d in datesEvalNames){
        lateximport <- c(lateximport, paste0("\\subsubsection{", fileName, " - ", d, "}"))

        for(s in names(dat[[d]])){
            # plotWeightsLines(dat, d, s)

            title <- paste0(fileName, "-", d,"-", s, ".pdf")
            pdf(file.path(getwd(), "Plot", title), width = 10, height = 4)
            plot(plotWeightsLines(dat, d, s))
            dev.off()

            lateximport <- c(lateximport, paste0("\\includegraphics[width=\\textwidth]{",title,"}"))
        }
    }

    fileConnection <- file(file.path(getwd(), "Plot", paste0("0",fileName,".txt")))
    writeLines(lateximport, fileConnection)
    close(fileConnection)
}
```

TODO: change of weights with turnover

TODO: include turnover as bar plot with second y-axis to visualize how much a portfolio has to be changed.

```
plotWeightsLines <- function(datName, d, s){
    dat <- datName[[d]][[s]]$x
    dat <- as.data.frame(dat)
    dat$date <- as.Date(rownames(dat))
    dat$turnover <- datName[[d]][[s]]$turnover

    colBackground <- colsEvalDates[d]

    plotCommand <- paste0("ggplot(dat, aes(x=date)) +")
    for(i in 1:(ncol(dat)-3)){
        plotCommand <- paste0(plotCommand, "geom_line(aes(y=dat[,",i,"], color = colnames(dat)[", i, "]
    }
    plotCommand <- paste0(plotCommand, "geom_line(aes(y=dat[,",ncol(dat)-2,"], color = colnames(dat)[",

    eval(parse(text = plotCommand))+
        ylim(0, 1)+
        geom_bar(aes(y=dat$turnover, colour = "Turnover"), stat = "identity")+
        scale_y_continuous(sec.axis = sec_axis(~., name = "Turnover"))+
        labs(title = paste("Time:", d),
             subtitle = paste("Portfolio:", s),
             y = "Weight ",
             x = "Date") +
        scale_color_discrete(name = "Index") +
        theme(panel.background = element_rect(fill = alpha(colBackground, 0.2)))
}
```

```
plotWeightsLinesComplete <- function(dat, fileName){
    lateximport <- c(paste0("\\subsection{",fileName,"}"))

    for(d in datesEvalNames){
        lateximport <- c(lateximport, paste0("\\subsubsection{", fileName, " - ", d, "}"))

        for(s in names(dat[[d]])){
            # plotWeightsLines(dat, d, s)

            title <- paste0(fileName, "-", d,"-", s, ".pdf")
            pdf(file.path(getwd(), "Plot", title), width = 10, height = 4)
            plot(plotWeightsLines(dat, d, s))
            dev.off()

            lateximport <- c(lateximport, paste0("\\includegraphics[width=\\textwidth]{",title,"}"))
        }
    }

    fileConnection <- file(file.path(getwd(), "Plot", paste0("0",fileName,".txt")))
    writeLines(lateximport, fileConnection)
    close(fileConnection)
}
```

summary statistics

print the summary (in matrix to pass it on to LaTeX-Table lateron)

```
summaryClassic <- function(datName, d, roundTo = 2){
    dat <- datName[[d]]

    mat <- matrix(NA, nrow = 3, ncol = length(dat))
    rownames(mat) <- c("Mean Return (an)", "Volatility (an)", "Sharpe Ratio (an)")
    colnames(mat) <- names(dat)

    for(sInd in 1:length(dat)){
        mat[1,sInd] <- round(dat[[sInd]]$anR, roundTo)
        mat[2,sInd] <- round(dat[[sInd]]$anSd, roundTo)
        mat[3,sInd] <- round(dat[[sInd]]$anSR, roundTo)
    }
    return(mat)
}
```

```
library(xtable)
```

```
##
## Attaching package: 'xtable'

## The following object is masked from 'package:timeDate':
```

```
##
##     align
```

```
summaryClassicComplete <- function(dat, fileName, roundTo = 2){
    lateximport <- c(paste0("\\subsection{",fileName,"}"))

    for(d in datesEvalNames){
        lateximport <- c(lateximport, paste0("\\subsubsection{", d, "}"))
        lateximport <- c(lateximport, print(xtable(summaryClassic(dat, d, roundTo))))
        print(summaryClassic(dat, d, roundTo))
    }

    lateximport <- c(lateximport, "\\clearpage")
    fileConnection <- file(file.path(getwd(), "Plot", paste0("0",fileName,".txt")))
    writeLines(lateximport, fileConnection)
    close(fileConnection)
}
```

whole analysis in one command

```
wholeAnalysis <- function(dat, fileName){
    retDat <- calcEvalVarClassic(dat)

    # weights
    plotWeightsLinesComplete(retDat, paste0("Weights-", fileName))

    # performance of portfolio
    plotPortfolioComplete(retDat, paste0("Performance-", fileName))

    # summary statistics
    summaryClassicComplete(retDat, paste0("Summary-", fileName))
}
```

Classic Optimization

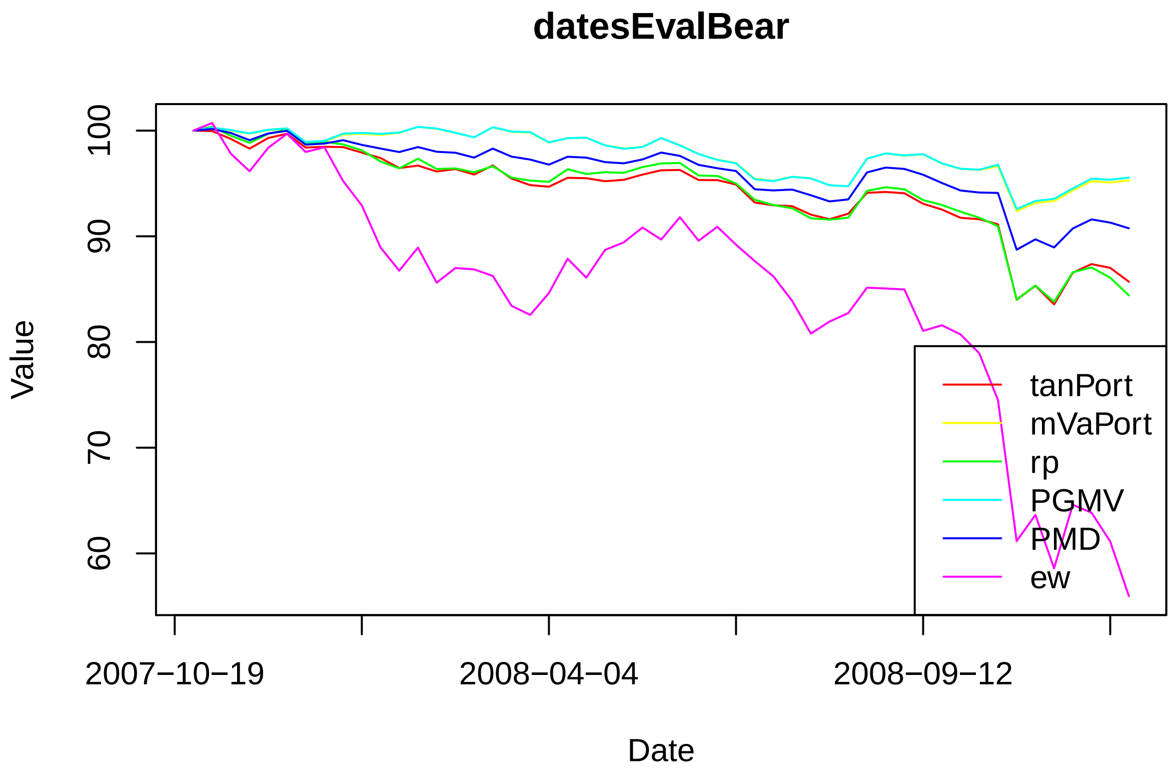
Constant weights over time window

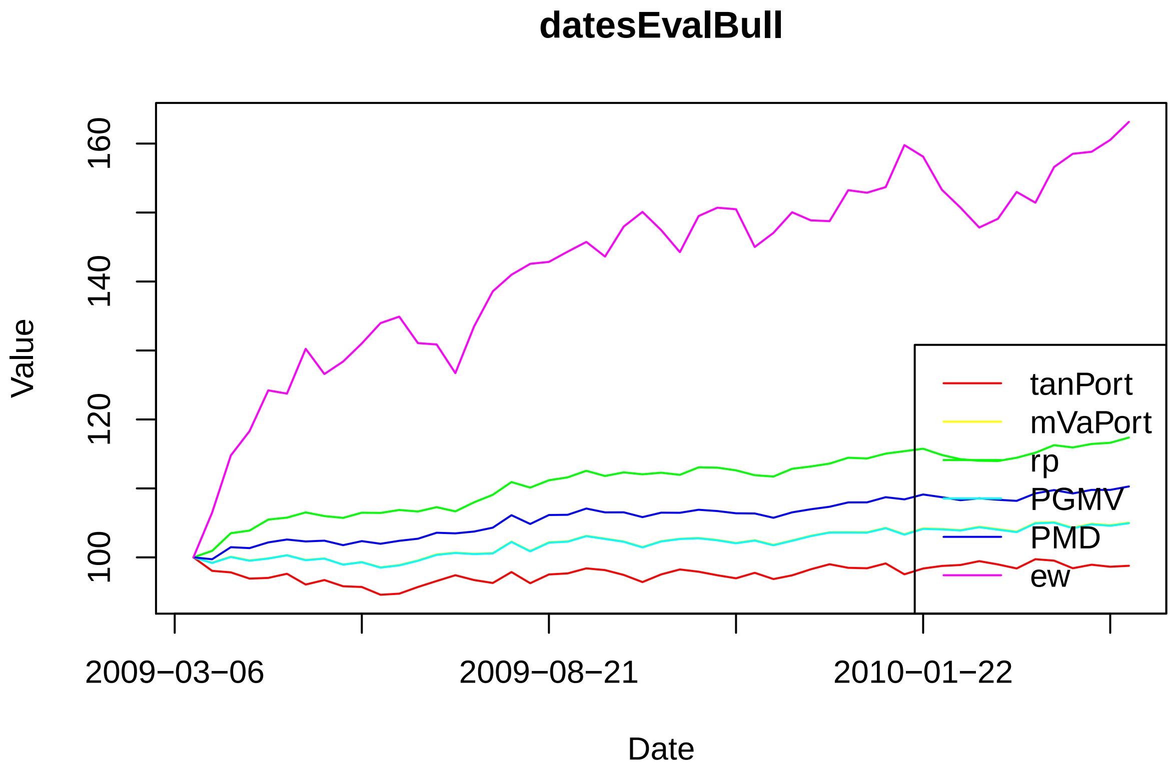
We want to visualize the evolvement of a portfolio over each time window.

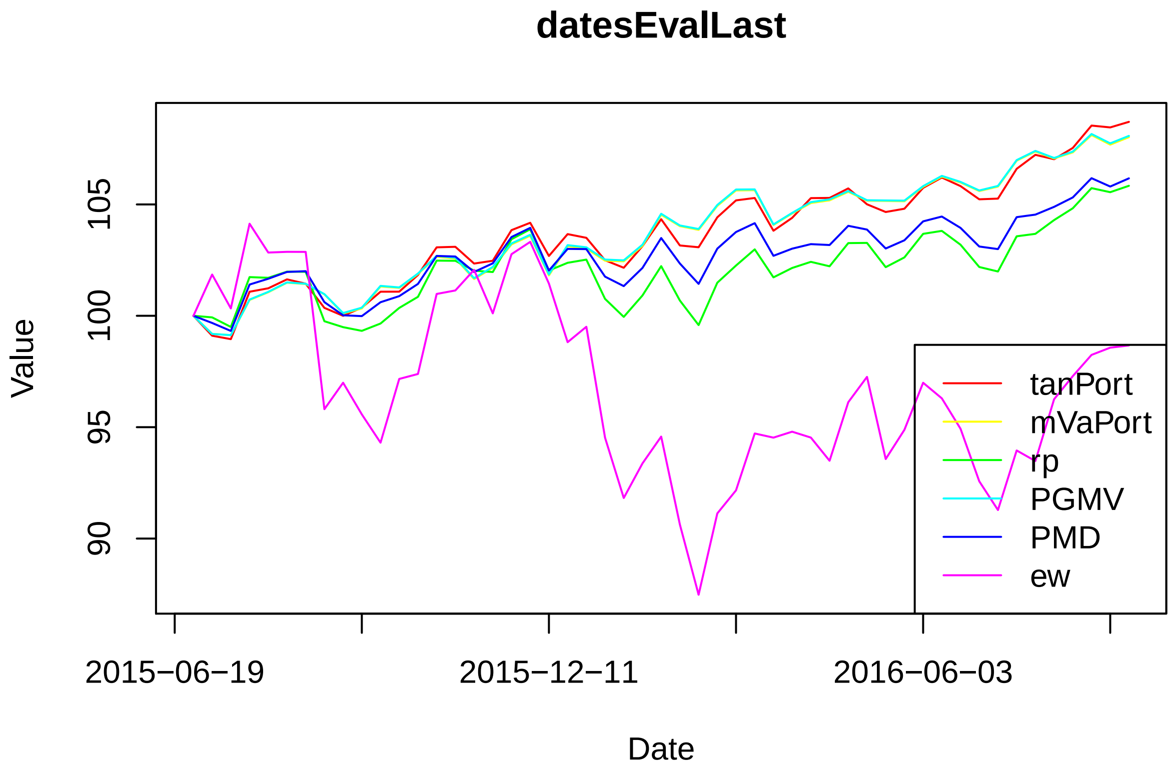
Be aware of the index shifting: retPlot[j-1, i] take wealth of previous day retOverTime[j-1,] take return of today (j is one step ahead)

Remove numbering of x-axis by *xaxt='n'*.

```
for(d in datesEvalNames){
    cols <- rainbow(length(xClassicConst[[d]]))
    retOverTime <- 1+ret[get(d),]
    retPlotDates <- get(d)
    retPlotDates <- c(datesAll[which(datesAll==min(retPlotDates))-1], retPlotDates)
    retPlot <- data.frame(Datum = retPlotDates)

    for(i in names(xClassicConst[[d]])){
        retPlot[1,i] <- 100
        for(j in 2:nrow(retPlot)){
            retPlot[j,i] <- retPlot[j-1,i]*crossprod(xClassicConst[[d]][[i]], retOverTime[j-1,])
        }
    }

    ylim = c(min(retPlot[,-1]), max(retPlot[,-1]))
    plot(retPlot[,2], type = "l", ylim = ylim, col = cols[1], main = d, xlab = "Date", ylab = "Value",
    for(i in 3:ncol(retPlot)){
        par(new=T)
        plot(retPlot[,i], type = "l", ylim = ylim, axes = F, xlab = "", ylab = "", col = cols[i-1])
    }
    axis(1, at = c(0, 10, 20, 30, 40, 50), labels = retPlot[c(0, 10, 20, 30, 40, 50)+1,1])
    legend("bottomright", legend = names(xClassicConst[[d]]), col = cols, lty = 1)

    pdf(file.path(getwd(), "Plot", paste0("Performance-ClassicConst-", d, ".pdf")), width = 10, height
    plot(retPlot[,2], type = "l", ylim = ylim, col = cols[1], main = d, xlab = "Date", ylab = "Value",
    for(i in 3:ncol(retPlot)){
        par(new=T)
        plot(retPlot[,i], type = "l", ylim = ylim, axes = F, xlab = "", ylab = "", col = cols[i-1])
    }
    axis(1, at = c(0, 10, 20, 30, 40, 50), labels = retPlot[c(0, 10, 20, 30, 40, 50)+1,1])
    legend("bottomright", legend = names(xClassicConst[[d]]), col = cols, lty = 1)
    dev.off()
}
```




datesEvalBull
Value
160
140
120
100
2009−03−06
2009−08−21
2010−01−22
Date
tanPort
mVaPort
rp
PGMV
PMD
ew



Varying of portfolio weights

```r
wholeAnalysis(xClassicVar, "Classic")
```

```
## % latex table generated in R 3.4.1 by xtable 1.8-2 package
## % Sun Sep 03 17:28:04 2017
## \begin{table}[ht]
## \centering
## \begin{tabular}{rrrrrrr}
##   \hline
##  & tanPort & mVaPort & rp & PGMV & PMD & ew \\
##   \hline
## Mean Return (an) & -0.34 & -0.08 & -0.17 & -0.08 & -0.12 & -0.44 \\
##   Volatility (an) & 0.29 & 0.08 & 0.11 & 0.08 & 0.09 & 0.29 \\
##   Sharpe Ratio (an) & -1.19 & -1.00 & -1.54 & -0.99 & -1.37 & -1.51 \\
##    \hline
## \end{tabular}
## \end{table}
##                    tanPort mVaPort    rp  PGMV   PMD    ew
## Mean Return (an)    -0.34   -0.08 -0.17 -0.08 -0.12 -0.44
## Volatility (an)      0.29    0.08  0.11  0.08  0.09  0.29
## Sharpe Ratio (an)   -1.19   -1.00 -1.54 -0.99 -1.37 -1.51
## % latex table generated in R 3.4.1 by xtable 1.8-2 package
## % Sun Sep 03 17:28:05 2017
## \begin{table}[ht]
## \centering
## \begin{tabular}{rrrrrrr}
##   \hline
##  & tanPort & mVaPort & rp & PGMV & PMD & ew \\
##   \hline
## Mean Return (an) & 0.03 & 0.07 & 0.16 & 0.07 & 0.11 & 0.60 \\
##   Volatility (an) & 0.06 & 0.05 & 0.04 & 0.05 & 0.04 & 0.18 \\
##   Sharpe Ratio (an) & 0.61 & 1.48 & 3.50 & 1.51 & 2.75 & 3.35 \\
##    \hline
## \end{tabular}
## \end{table}
##                   tanPort mVaPort   rp PGMV  PMD   ew
## Mean Return (an)     0.03    0.07 0.16 0.07 0.11 0.60
## Volatility (an)      0.06    0.05 0.04 0.05 0.04 0.18
## Sharpe Ratio (an)    0.61    1.48 3.50 1.51 2.75 3.35
## % latex table generated in R 3.4.1 by xtable 1.8-2 package
## % Sun Sep 03 17:28:05 2017
## \begin{table}[ht]
## \centering
## \begin{tabular}{rrrrrrr}
##   \hline
##  & tanPort & mVaPort & rp & PGMV & PMD & ew \\
##   \hline
## Mean Return (an) & 0.01 & 0.10 & 0.04 & 0.10 & 0.07 & -0.02 \\
##   Volatility (an) & 0.08 & 0.05 & 0.09 & 0.05 & 0.06 & 0.17 \\
##   Sharpe Ratio (an) & 0.15 & 2.10 & 0.42 & 2.07 & 1.04 & -0.11 \\
##    \hline
## \end{tabular}
## \end{table}
##                   tanPort mVaPort   rp PGMV  PMD    ew
## Mean Return (an)     0.01    0.10 0.04 0.10 0.07 -0.02
## Volatility (an)      0.08    0.05 0.09 0.05 0.06  0.17
## Sharpe Ratio (an)    0.15    2.10 0.42 2.07 1.04 -0.11
```

Varying of portfolio weights no risk free asset

```r
wholeAnalysis(xClassicVarNoRf, "Classic-No-Risk-Free")
```

```
## % latex table generated in R 3.4.1 by xtable 1.8-2 package
## % Sun Sep 03 17:28:24 2017
## \begin{table}[ht]
## \centering
## \begin{tabular}{rrrrrrr}
##   \hline
##  & tanPort & mVaPort & rp & PGMV & PMD & ew \\
##   \hline
## Mean Return (an) & -0.61 & -0.52 & -0.50 & -0.52 & -0.52 & -0.50 \\
##   Volatility (an) & 0.41 & 0.34 & 0.34 & 0.34 & 0.35 & 0.34 \\
##   Sharpe Ratio (an) & -1.50 & -1.52 & -1.45 & -1.52 & -1.50 & -1.44 \\
##    \hline
## \end{tabular}
## \end{table}
##                   tanPort mVaPort    rp  PGMV   PMD    ew
## Mean Return (an)    -0.61   -0.52 -0.50 -0.52 -0.52 -0.50
## Volatility (an)      0.41    0.34  0.34  0.34  0.35  0.34
## Sharpe Ratio (an)   -1.50   -1.52 -1.45 -1.52 -1.50 -1.44
## % latex table generated in R 3.4.1 by xtable 1.8-2 package
## % Sun Sep 03 17:28:24 2017
## \begin{table}[ht]
## \centering
## \begin{tabular}{rrrrrrr}
##   \hline
##  & tanPort & mVaPort & rp & PGMV & PMD & ew \\
##   \hline
## Mean Return (an) & 0.84 & 0.80 & 0.72 & 0.80 & 0.75 & 0.73 \\
##   Volatility (an) & 0.23 & 0.20 & 0.21 & 0.20 & 0.21 & 0.21 \\
##   Sharpe Ratio (an) & 3.60 & 3.94 & 3.41 & 3.93 & 3.56 & 3.38 \\
##    \hline
## \end{tabular}
## \end{table}
##                    tanPort mVaPort   rp PGMV  PMD   ew
## Mean Return (an)     0.84    0.80 0.72 0.80 0.75 0.73
## Volatility (an)      0.23    0.20 0.21 0.20 0.21 0.21
## Sharpe Ratio (an)    3.60    3.94 3.41 3.93 3.56 3.38
## % latex table generated in R 3.4.1 by xtable 1.8-2 package
## % Sun Sep 03 17:28:24 2017
## \begin{table}[ht]
## \centering
## \begin{tabular}{rrrrrrr}
##   \hline
##  & tanPort & mVaPort & rp & PGMV & PMD & ew \\
##   \hline
## Mean Return (an) & -0.03 & 0.01 & -0.03 & 0.01 & -0.05 & -0.04 \\
##   Volatility (an) & 0.21 & 0.15 & 0.19 & 0.15 & 0.20 & 0.20 \\
##   Sharpe Ratio (an) & -0.16 & 0.08 & -0.17 & 0.08 & -0.25 & -0.20 \\
##    \hline
## \end{tabular}
## \end{table}
##                    tanPort mVaPort    rp PGMV   PMD    ew
## Mean Return (an)    -0.03    0.01 -0.03 0.01 -0.05 -0.04
## Volatility (an)      0.21    0.15  0.19 0.15  0.20  0.20
## Sharpe Ratio (an)   -0.16    0.08 -0.17 0.08 -0.25 -0.20
```

Sentix Optimization

```
wholeAnalysis(xDispVarEval, "Sentix")
```

```
## % latex table generated in R 3.4.1 by xtable 1.8-2 package
## % Sun Sep 03 17:28:46 2017
## \begin{table}[ht]
## \centering
## \begin{tabular}{rrrrrrr}
##   \hline
##  & P1 & P6 & I1 & I6 & G1 & G6 \\
##   \hline
## Mean Return (an) & 0.05 & 0.07 & 0.04 & 0.06 & 0.05 & 0.07 \\
##   Volatility (an) & 0.07 & 0.07 & 0.06 & 0.07 & 0.07 & 0.07 \\
##   Sharpe Ratio (an) & 0.79 & 0.89 & 0.59 & 0.84 & 0.78 & 0.90 \\
##    \hline
## \end{tabular}
## \end{table}
##                     P1   P6   I1   I6   G1   G6
## Mean Return (an)  0.05 0.07 0.04 0.06 0.05 0.07
## Volatility (an)   0.07 0.07 0.06 0.07 0.07 0.07
## Sharpe Ratio (an) 0.79 0.89 0.59 0.84 0.78 0.90
## % latex table generated in R 3.4.1 by xtable 1.8-2 package
## % Sun Sep 03 17:28:46 2017
## \begin{table}[ht]
## \centering
## \begin{tabular}{rrrrrrr}
##   \hline
##  & P1 & P6 & I1 & I6 & G1 & G6 \\
##   \hline
## Mean Return (an) & 0.11 & 0.16 & 0.11 & 0.09 & 0.11 & 0.16 \\
##   Volatility (an) & 0.06 & 0.10 & 0.05 & 0.05 & 0.06 & 0.10 \\
##   Sharpe Ratio (an) & 1.89 & 1.59 & 2.01 & 1.71 & 1.85 & 1.59 \\
##    \hline
## \end{tabular}
## \end{table}
##                     P1   P6   I1   I6   G1   G6
## Mean Return (an)  0.11 0.16 0.11 0.09 0.11 0.16
## Volatility (an)   0.06 0.10 0.05 0.05 0.06 0.10
## Sharpe Ratio (an) 1.89 1.59 2.01 1.71 1.85 1.59
## % latex table generated in R 3.4.1 by xtable 1.8-2 package
## % Sun Sep 03 17:28:46 2017
## \begin{table}[ht]
## \centering
## \begin{tabular}{rrrrrrr}
##   \hline
##  & P1 & P6 & I1 & I6 & G1 & G6 \\
##   \hline
## Mean Return (an) & 0.12 & 0.13 & 0.12 & 0.12 & 0.12 & 0.13 \\
##   Volatility (an) & 0.06 & 0.11 & 0.06 & 0.06 & 0.06 & 0.11 \\
##   Sharpe Ratio (an) & 1.84 & 1.25 & 1.81 & 2.06 & 1.83 & 1.25 \\
##    \hline
## \end{tabular}
## \end{table}
##                     P1   P6   I1   I6   G1   G6
## Mean Return (an)  0.12 0.13 0.12 0.12 0.12 0.13
## Volatility (an)   0.06 0.11 0.06 0.06 0.06 0.11
## Sharpe Ratio (an) 1.84 1.25 1.81 2.06 1.83 1.25
```

All together

sentix with classic portfolio with varying weights

Performance

```
retPortClassicVarying <- calcEvalVarClassic(xClassicVar)
retPortSentixVarying <- calcEvalVarClassic(xDispVarEval)

retAllVarying <- retPortClassicVarying
for(timeWindowName in names(retAllVarying)){
    retAllVarying[[timeWindowName]] <- append(retAllVarying[[timeWindowName]], retPortSentixVarying[[ti
}
```

```
plotPortfolioComplete(retAllVarying, "Performance-All")
```

Summary Statistics

```
summaryClassicComplete(retAllVarying, "SummaryAll")
```

```
## % latex table generated in R 3.4.1 by xtable 1.8-2 package
## % Sun Sep 03 17:28:50 2017
## \begin{table}[ht]
## \centering
## \begin{tabular}{rrrrrrrrrrrrr}
##   \hline
##  & tanPort & mVaPort & rp & PGMV & PMD & ew & P1 & P6 & I1 & I6 & G1 & G6 \\
##   \hline
## Mean Return (an) & -0.34 & -0.08 & -0.17 & -0.08 & -0.12 & -0.44 & 0.05 & 0.07 & 0.04 & 0.06 & 0.05 &
##   Volatility (an) & 0.29 & 0.08 & 0.11 & 0.08 & 0.09 & 0.29 & 0.07 & 0.07 & 0.06 & 0.07 & 0.07 & 0.0
##   Sharpe Ratio (an) & -1.19 & -1.00 & -1.54 & -0.99 & -1.37 & -1.51 & 0.79 & 0.89 & 0.59 & 0.84 & 0.
##    \hline
## \end{tabular}
## \end{table}
##                    tanPort mVaPort    rp  PGMV   PMD    ew   P1   P6   I1
## Mean Return (an)     -0.34   -0.08 -0.17 -0.08 -0.12 -0.44 0.05 0.07 0.04
## Volatility (an)       0.29    0.08  0.11  0.08  0.09  0.29 0.07 0.07 0.06
## Sharpe Ratio (an)    -1.19   -1.00 -1.54 -0.99 -1.37 -1.51 0.79 0.89 0.59
##                      I6   G1   G6
## Mean Return (an)   0.06 0.05 0.07
## Volatility (an)    0.07 0.07 0.07
## Sharpe Ratio (an) 0.84 0.78 0.90
## % latex table generated in R 3.4.1 by xtable 1.8-2 package
## % Sun Sep 03 17:28:50 2017
## \begin{table}[ht]
## \centering
## \begin{tabular}{rrrrrrrrrrrrr}
##   \hline
##  & tanPort & mVaPort & rp & PGMV & PMD & ew & P1 & P6 & I1 & I6 & G1 & G6 \\
##   \hline
## Mean Return (an) & 0.03 & 0.07 & 0.16 & 0.07 & 0.11 & 0.60 & 0.11 & 0.16 & 0.11 & 0.09 & 0.11 & 0.16
##   Volatility (an) & 0.06 & 0.05 & 0.04 & 0.05 & 0.04 & 0.18 & 0.06 & 0.10 & 0.05 & 0.05 & 0.06 & 0.1
##   Sharpe Ratio (an) & 0.61 & 1.48 & 3.50 & 1.51 & 2.75 & 3.35 & 1.89 & 1.59 & 2.01 & 1.71 & 1.85 & 1
##    \hline
## \end{tabular}
## \end{table}
##                    tanPort mVaPort   rp PGMV  PMD   ew   P1   P6   I1   I6
## Mean Return (an)     0.03    0.07 0.16 0.07 0.11 0.60 0.11 0.16 0.11 0.09
## Volatility (an)      0.06    0.05 0.04 0.05 0.04 0.18 0.06 0.10 0.05 0.05
## Sharpe Ratio (an)    0.61    1.48 3.50 1.51 2.75 3.35 1.89 1.59 2.01 1.71
##                       G1   G6
## Mean Return (an)  0.11 0.16
## Volatility (an)   0.06 0.10
## Sharpe Ratio (an) 1.85 1.59
## % latex table generated in R 3.4.1 by xtable 1.8-2 package
## % Sun Sep 03 17:28:50 2017
## \begin{table}[ht]
## \centering
## \begin{tabular}{rrrrrrrrrrrrr}
##   \hline
##  & tanPort & mVaPort & rp & PGMV & PMD & ew & P1 & P6 & I1 & I6 & G1 & G6 \\
##   \hline
## Mean Return (an) & 0.01 & 0.10 & 0.04 & 0.10 & 0.07 & -0.02 & 0.12 & 0.13 & 0.12 & 0.12 & 0.12 & 0.1
##   Volatility (an) & 0.08 & 0.05 & 0.09 & 0.05 & 0.06 & 0.17 & 0.06 & 0.11 & 0.06 & 0.06 & 0.06 & 0.1
##   Sharpe Ratio (an) & 0.15 & 2.10 & 0.42 & 2.07 & 1.04 & -0.11 & 1.84 & 1.25 & 1.81 & 2.06 & 1.83 &
##    \hline
## \end{tabular}
## \end{table}
##                     tanPort mVaPort   rp PGMV  PMD    ew   P1   P6   I1   I6
## Mean Return (an)       0.01    0.10 0.04 0.10 0.07 -0.02 0.12 0.13 0.12 0.12
## Volatility (an)        0.08    0.05 0.09 0.05 0.06  0.17 0.06 0.11 0.06 0.06
## Sharpe Ratio (an)      0.15    2.10 0.42 2.07 1.04 -0.11 1.84 1.25 1.81 2.06
##                       G1   G6
## Mean Return (an)  0.12 0.13
## Volatility (an)   0.06 0.11
## Sharpe Ratio (an) 1.83 1.25
```

```
rm(retPortClassicVarying, retPortSentixVarying, retAllVarying)
```

cleanup

```
rm(calcEvalVarClassic, calcEvalVarSentix, plotPortfolio, plotPortfolioComplete, plotWeightsLines, plotW

detach("package:xtable", unload = T)
```