http://mintegration.eu/

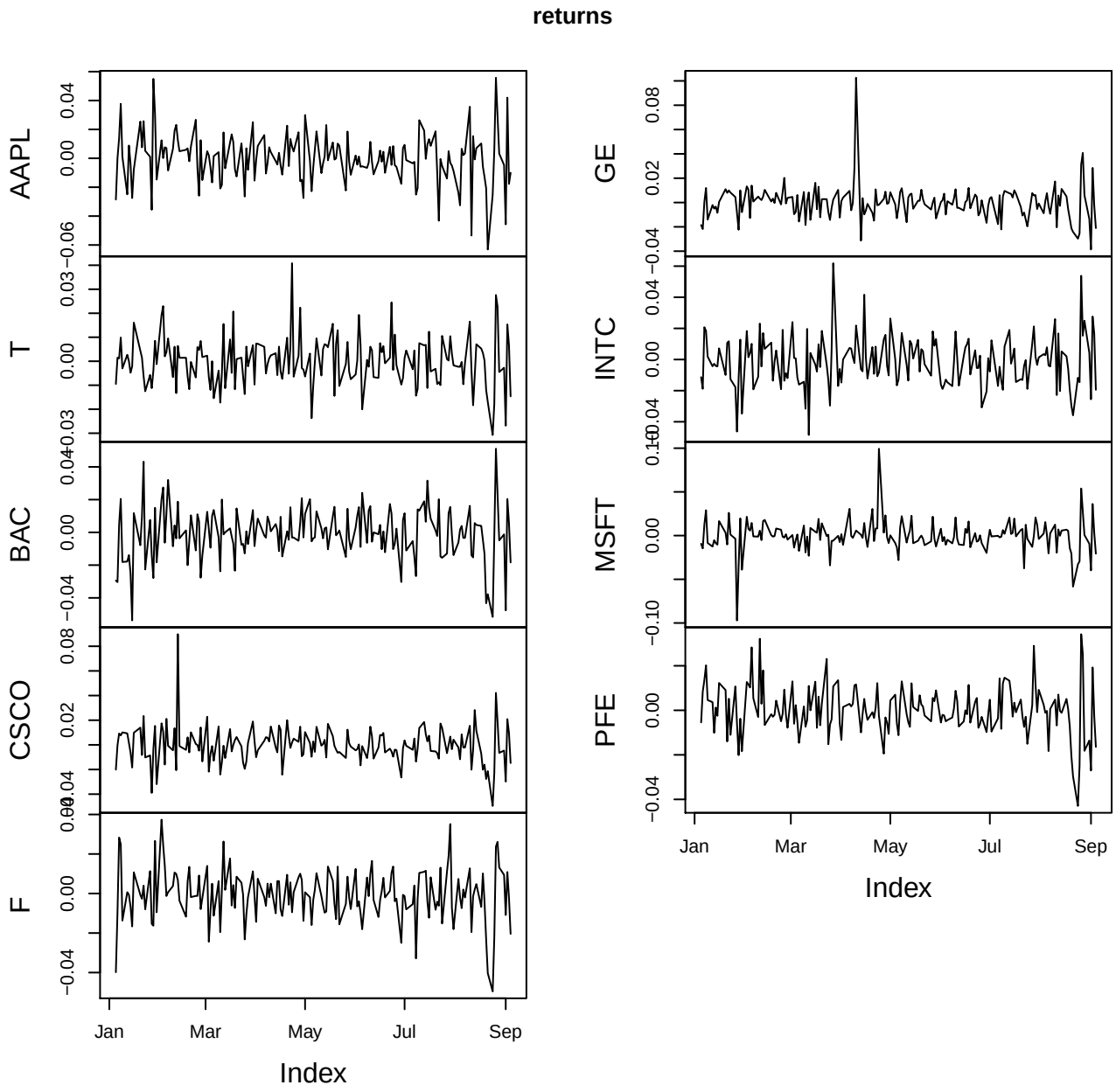
Calculating Risk Parity Weights Using Newton's Method

Author:
Nick Kirk
nk@mintegration.eu

September 7, 2015

```
################################################################################
##                                    _______                                   ##
##                                   /       \                                  ##
##  ____   ____ _____ _____| |_  ___ |   []    )                                 ##
## / ___) / ___)  _  )  _  |  _)/ _ \|   ||   /                                  ##
##  (___ | |   ( (/ (| ( | | |_| |_| |   ||   |                                  ##
## \____)|_|    \____)__||_|\___)___/|___||___|                                  ##
##              2015, Nick Kirk                                                 ##
##              http://mintegration.eu                                          ##
##                                                                              ##
## References:                                                                  ##
## ------------                                                                 ##
## Title:   Risk parity funds suffer a cruel summer                             ##
## Author:  Robin Wigglesworth                                                  ##
## Source:  Finanical Times (FT)                                                ##
##          http://www.ft.com/cms/s/0/                                          ##
##          d210373e-5142-11e5-8642-453585f2cfcd.html?                          ##
##          siteedition=intl#axzz3kbVfJfnX                                      ##
##                                                                              ##
## Title:   Efficient Algorithms for Computing Risk Parity Portfolio Weights    ##
## Authors: Chaves et al                                                        ##
## Source:  Journal of Investing, 2012, 21, no. 3 (fall): 150-163               ##
##          http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2117303          ##
################################################################################

setwd("C:/mintegratoR")
source(file = "R/mintegratoR.R")

## [1] "Windows parallel backend 'doParallel' registered \n          - required for 'foreach'"

# ------------------------------------------------------------------------------
# Only call once - Download EOD prices from Quandl
# ------------------------------------------------------------------------------
if (! file.exists("data/creatoR_0002/creatoR_0002_prices.rds")){
    download_EOD(input_file = "data/creatoR_0002/creatoR_0002.csv",
                 output_file = "data/creatoR_0002/creatoR_0002_prices",
                 start_date = "2015-01-01")
}

# Read the saved EOD prices
stocks_xts <- readRDS(file = "data/creatoR_0002/creatoR_0002_prices.rds")

# View the cross-sectional data.
head(stocks_xts)

##            TICKER Adj_Close          Adj_Volume Adj_Open
## 2015-01-02 "AAPL" "107.965380765789" "53204626" "109.999668558504"
## 2015-01-02 "T"    "32.4840117807547" "20557653" "32.3785130711036"
## 2015-01-02 "BAC"  "17.7351850979283" "48915838" "17.8243564196497"
## 2015-01-02 "CSCO" "27.1921584039857" "22926512" "27.4383749777269"
## 2015-01-02 "F"    "14.9141938261211" "24773033" "15.1375183430487"
## 2015-01-02 "GE"   "24.6247099181194" "40919008" "24.8998463417855"
##            Adj_High           Adj_Low
## 2015-01-02 "110.049044475803" "106.010094440752"
## 2015-01-02 "32.5799196986194" "32.3113775285984"
## 2015-01-02 "17.8639881181926" "17.5173098351889"
## 2015-01-02 "27.6944402144179" "26.9656391561437"
## 2015-01-02 "15.195776912682"  "14.7423310457029"
## 2015-01-02 "25.0177619519282" "24.4380102020602"

tail(stocks_xts)

##            TICKER Adj_Close Adj_Volume Adj_Open Adj_High Adj_Low
```

```
## 2015-09-04 "CSCO" "25.52"   "26904250" "25.42"  "25.65"  "25.35"
## 2015-09-04 "F"    "13.56"   "31136736" "13.68"  "13.7"   "13.4"
## 2015-09-04 "GE"   "24"      "35625029" "24.18"  "24.18"  "23.85"
## 2015-09-04 "INTC" "28.52"   "29117537" "28.57"  "28.9"   "28.41"
## 2015-09-04 "MSFT" "42.61"   "37138775" "42.81"  "43.04"  "42.195"
## 2015-09-04 "PFE"  "31.37"   "22561147" "31.45"  "31.72"  "31.13"

# Get the list of assets
(asset_list <- unique(stocks_xts[, "TICKER"]))

## [1] "AAPL" "T"    "BAC"  "CSCO" "F"    "GE"   "INTC" "MSFT" "PFE"

# Calculate the log returns
calc_log_ret(asset_list)
print(stocks_xts[10:18, c("TICKER", "Log_Ret")])

##            TICKER Log_Ret
## 2015-01-05 "AAPL" "-0.028576023641862"
## 2015-01-05 "T"    "-0.00949280341198788"
## 2015-01-05 "BAC"  "-0.0294805930094673"
## 2015-01-05 "CSCO" "-0.0201214031994215"
## 2015-01-05 "F"    "-0.0398459085472025"
## 2015-01-05 "GE"   "-0.0185265065296059"
## 2015-01-05 "INTC" "-0.0113401851422239"
## 2015-01-05 "MSFT" "-0.00934636443244496"
## 2015-01-05 "PFE"  "-0.00544088396155384"

# Save each asset to the global env
save_assets(asset_list)
ls()

##  [1] "AAPL"              "asset_list"        "BAC"
##  [4] "calc_log_ret"      "compute_plot_MCTR" "compute_plot_PCTR"
##  [7] "CSCO"              "display_weights"   "download_EOD"
## [10] "F"                 "fy"                "GE"
## [13] "INTC"              "Jinv"              "MSFT"
## [16] "newton_Jinv"       "PFE"               "plotG2"
## [19] "save_assets"       "stocks_xts"        "T"

# Form a timeseries of only the returns
returns <- NULL
for (i in 1:length(asset_list)) {
    returns <- cbind(returns, get(asset_list[i])[-1, "Log_Ret"])
}
colnames(returns) <- asset_list
head(returns)

##                  AAPL         T       BAC       CSCO          F         GE
## 2015-01-05 -2.858e-02 -0.009493 -0.029481 -0.0201214 -0.0398459 -0.0185265
## 2015-01-06  9.411e-05  0.001489 -0.030376 -0.0003696 -0.0095304 -0.0217802
## 2015-01-07  1.392e-02  0.001190  0.004734  0.0091997  0.0283229  0.0004154
## 2015-01-08  3.770e-02  0.009900  0.020451  0.0076629  0.0249520  0.0119712
## 2015-01-09  1.072e-03 -0.002990 -0.018092  0.0101267 -0.0137123 -0.0140498
## 2015-01-12 -2.495e-02  0.002691 -0.017826  0.0093124  0.0006572 -0.0020829
##                 INTC      MSFT       PFE
## 2015-01-05 -0.011340 -0.009346 -0.005441
## 2015-01-06 -0.018813 -0.014678  0.008309
## 2015-01-07  0.020758  0.012625  0.013593
## 2015-01-08  0.018430  0.028994  0.020203
## 2015-01-09  0.001906 -0.008441  0.004605
## 2015-01-12 -0.004362 -0.012581  0.003669

# Plot the returns
plot.zoo(returns)
```



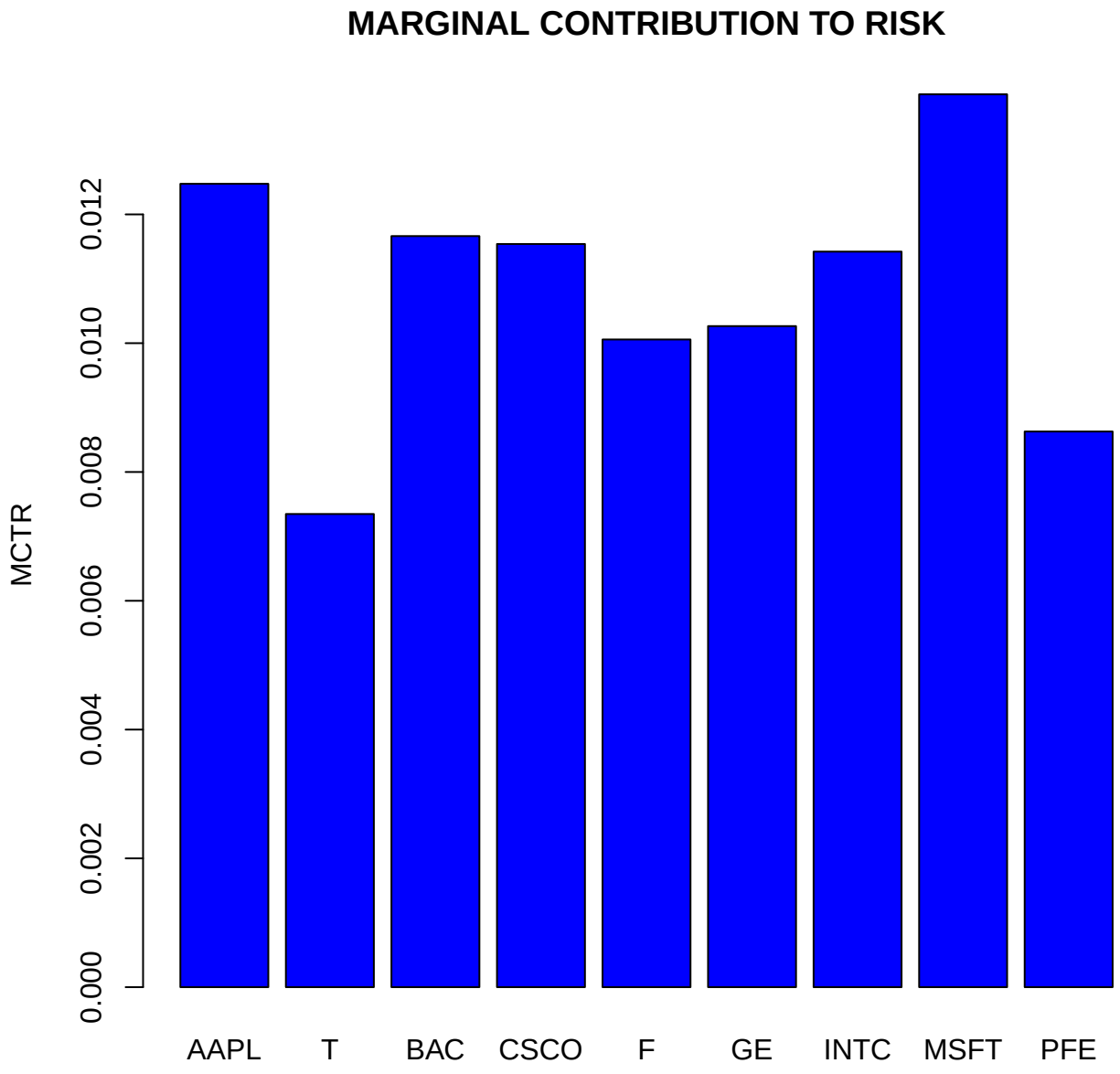

```
# ##############################################################################
# Method for calculating risk parity weights using Newton's Method.
# ##############################################################################

# N assets
n <- ncol(returns)

# Sigma (i.e. Covariance matrix)
Sigma <- var(returns)

#
# Initial values
#

# 1/N (i.e. an equally-weighted portfolio)
weights_init <- matrix(rep(1/n, n))
variance_portfolio <- t(weights_init) %*% Sigma %*% weights_init
lambda_init <- as.numeric(variance_portfolio / n)
y_init <- rbind(weights_init, lambda_init)
```

```r
# Newton iteration based on use of the inverse Jacobian function
y <- newton_Jinv(y_init, Sigma)

# Risk parity weights
weights <- display_weights()
```

```
##    AAPL     T BAC  CSCO     F    GE  INTC  MSFT   PFE
## 1 0.093 0.158 0.1 0.101 0.115 0.113 0.102 0.084 0.135
## [1] "Sum is equal to: 1"
```

```r
# Variance of the risk parity portfolio
variance_portfolio = t(weights) %*% Sigma %*% weights

# Compute and plot each asset's MCTR
mctr <- compute_plot_MCTR()
```

```
## [1] "Marginal contributions to risk (MCTR): "
##  AAPL     T   BAC  CSCO     F    GE  INTC  MSFT   PFE
## 0.012 0.007 0.012 0.012 0.010 0.010 0.011 0.014 0.009
```



```
# Compute and plot each asset's PCTR
compute_plot_PCTR(mctr)

## [1] "Percentage contributions to risk (PCTR): "
##  AAPL     T   BAC  CSCO     F    GE  INTC  MSFT   PFE
## 11.11 11.11 11.11 11.11 11.11 11.11 11.11 11.11 11.11
## [1] "Sum of weights (percentage) equals: 100"
```