BlackLittermanAnalysis.R

gordonritter

Tue Oct 30 20:38:21 2018

```
rm(list=ls())
require(rlang) ## for new_environment command
```

```
## Loading required package: rlang
```

```
require(tidyverse)
```

```
## Loading required package: tidyverse
```

```
## -- Attaching packages --------------------------------- tidyverse 1.2.1 --
```

```
## v ggplot2 3.1.0     v purrr   0.2.5
## v tibble  1.4.2     v dplyr   0.7.7
## v tidyr   0.8.2     v stringr 1.3.1
## v readr   1.1.1     v forcats 0.3.0
```

```
## -- Conflicts ------------------------------------ tidyverse_conflicts() --
## x purrr::%@%()         masks rlang::%@%()
## x purrr::%||%()        masks rlang::%||%()
## x purrr::as_function() masks rlang::as_function()
## x dplyr::filter()      masks stats::filter()
## x purrr::flatten()     masks rlang::flatten()
## x purrr::flatten_chr() masks rlang::flatten_chr()
## x purrr::flatten_dbl() masks rlang::flatten_dbl()
## x purrr::flatten_int() masks rlang::flatten_int()
## x purrr::flatten_lgl() masks rlang::flatten_lgl()
## x purrr::invoke()      masks rlang::invoke()
## x dplyr::lag()         masks stats::lag()
## x purrr::list_along()  masks rlang::list_along()
## x purrr::modify()      masks rlang::modify()
## x purrr::rep_along()   masks rlang::rep_along()
## x purrr::splice()      masks rlang::splice()
```

```
## change the following line to whatever your data dir is.
basedir <- '/Users/gordonritter/Dropbox/teaching/examples/black-litterman/'

## gg.plot: convenience wrapper for plotting time series
##
## Assuming df is a data frame, which contains
## a column called "date" in date format, and some data columns 'varnames'
## plot the variables in list 'varnames' as time series, optionally taking cumulative sum
##
gg.plot <- function(df, varnames, use.cumsum=FALSE, Title = NA, legend = FALSE) {
  X <- select(df, c("date", varnames))
  if(class(X$date) != "Date") X$date <- as.Date(X$date)
  if(use.cumsum) {
    for(k in varnames) X[[k]] <- cumsum(X[[k]])
  }
  X <- X %>% gather(key = "variable", value = "value", -date)
  g <- ggplot(X, aes(x = date, y = value, colour = variable)) + theme_minimal() + geom_line()
```

```
  if(!legend) g <- g + theme(legend.position="none")

  if(!is.na(Title)) {
    g <- g + ggtitle(Title) + theme(plot.title = element_text(size=10))
  }
  return(g)
}


all.data <- data.frame(read_csv(paste0(basedir, 'monthlyRets-clean.csv')) %>% rename(date = DATE))
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   DATE = col_date(format = "")
## )
```

```
## See spec(...) for full column specifications.
```

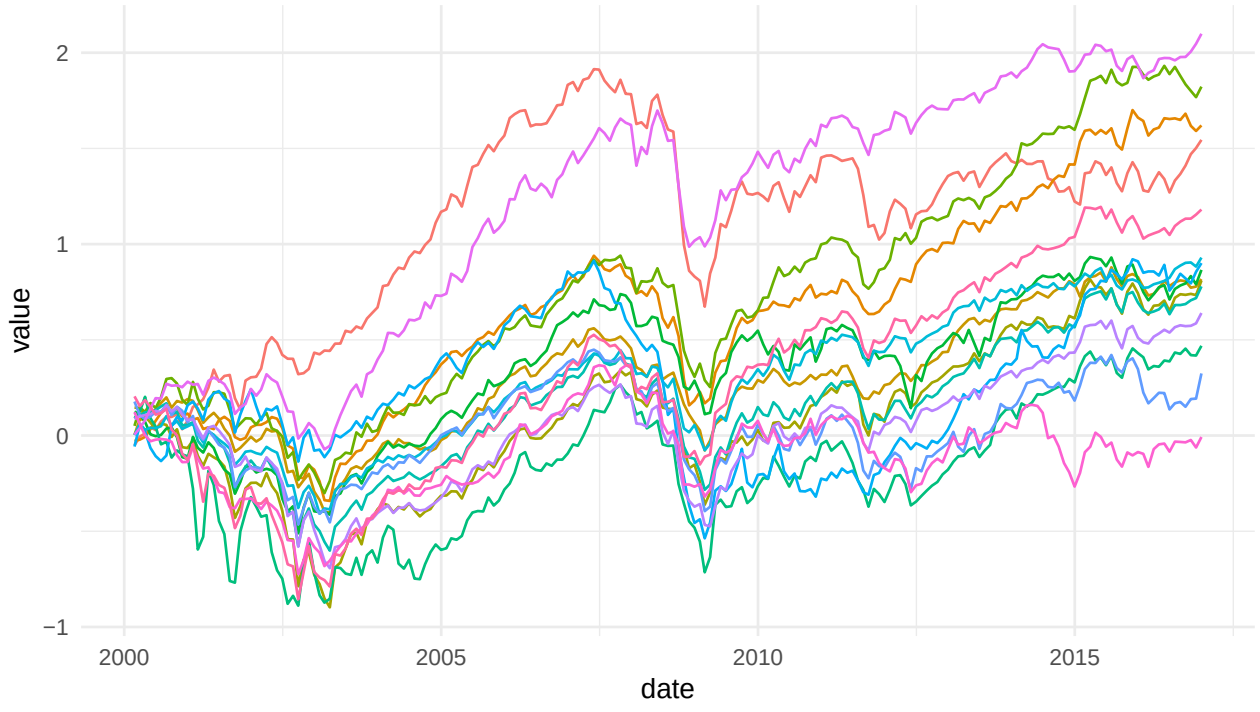
```
all.data$date <- as.Date(all.data$date)

print(paste('first date = ', all.data[1,"date"]))
```

```
## [1] "first date =  2000-02-29"
```

```
print(paste('last date = ', all.data[NROW(all.data),"date"]))
```

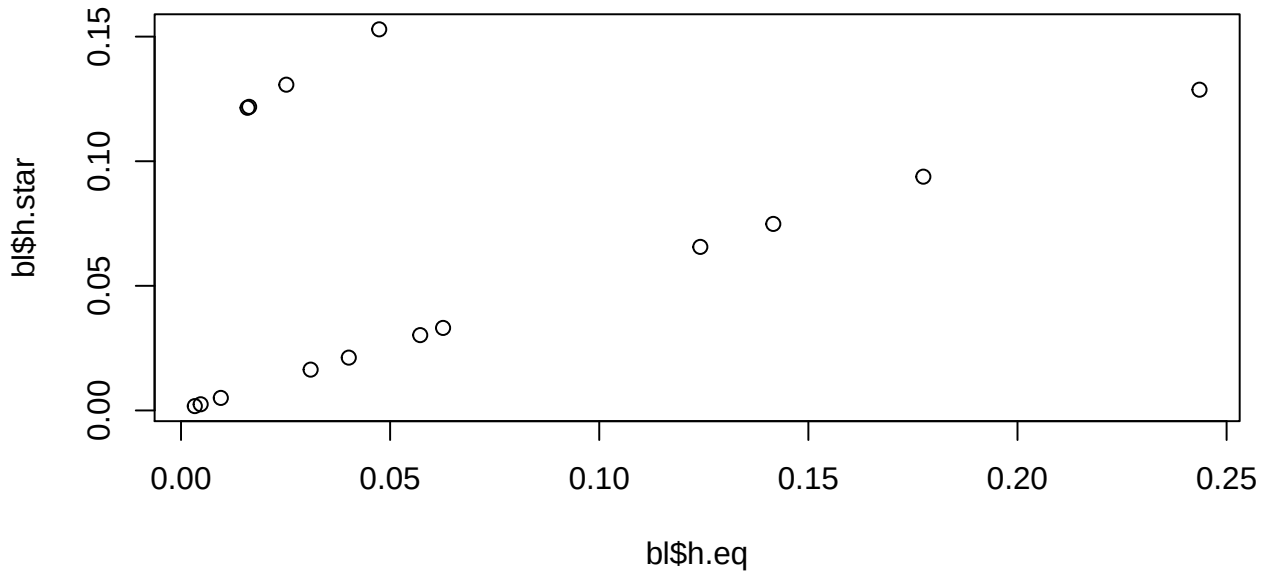
```
## [1] "last date =  2016-12-31"
```

```
totret.cols <- names(all.data)[grep("country.*.totret", names(all.data))]
mcap.cols <- names(all.data)[grep("country.*.previous.usdcap", names(all.data))]

print(gg.plot(all.data, totret.cols, use.cumsum = TRUE))
```



```r
compute.black.litterman <- function(x, tau = 0.01, my.kappa = 1.0, shrinkage = 0.5) {

  n <- NROW(x) # number of dates
  k <- length(totret.cols) # number of countries

  TR <- x[, totret.cols]
  CAP <- x[, mcap.cols]
  names(TR) <- gsub(".totret", "", names(TR))
  names(CAP) <- gsub(".previous.usdcap", "", names(CAP))
  countries <- names(TR)

  vols <- unlist(lapply(countries, function(ctry) sd(TR[1:(n-1), ctry], na.rm=TRUE)))
  S <- matrix(diag(vols, k, k), k, k, dimnames = list(countries, countries))

  Rho <- cor(TR[1:(n-1), ])
  # print(paste0('condition number (pre-shrinkage) = ', kappa(S %*% Rho %*% S)))

  Ident = diag(rep(1, k), k, k); rownames(Ident) <- countries; colnames(Ident) <- countries
  Rho <- shrinkage * Rho + (1 - shrinkage) * Ident

  h.eq <- matrix(data = as.double(CAP[NROW(CAP), ]), nrow = k, ncol = 1, dimnames = list(countries, c('
  h.eq <- h.eq / sum(abs(h.eq))

  Sigma <- S %*% Rho %*% S
  # print(paste0('condition number = ', kappa(Sigma)))

  C <- tau * Sigma
  C.inv <- solve(C)

  Pi <- my.kappa * (1 + tau) * Sigma %*% h.eq

  ## P has one row per view (and only one view for now)
  P <- t(-h.eq)
  fav.list <- paste0('country.', c('nor', 'swe', 'dnk', 'fin'))
  NN <- as.double(length(fav.list))
  for(fav in fav.list) P[1,fav] <- 0
  P <- P / sum(abs(as.double(P)))
  for(fav in fav.list) P[1,fav] <- 1.0 / NN
  tP <- t(P)

  viewname <- paste(fav.list, collapse = '.')
  q <- matrix(0.01, 1, 1, dimnames = list(c(viewname), c(viewname)))
  ## confidence in that view
  uncertainty.in.view <- 0.015
  Omega <- matrix(uncertainty.in.view^2, 1, 1, dimnames = list(c(viewname), c(viewname)))
  Omega.inv <- solve(Omega)

  H <- tP %*% Omega.inv %*% P + C.inv
  H.inv <- solve(H)

  v <- tP %*% (Omega.inv %*% q) + solve(C, Pi)
  h.star <- (1.0 / my.kappa) * solve(H.inv + Sigma, H.inv %*% v)
```

```
  tr.vector <- as.double(TR[NROW(TR),]);

  return(new_environment(data = list(h.eq = h.eq,
              h.star = h.star,
              h.eq.pnl = sum(as.double(h.eq) * tr.vector),
              h.bl.pnl = sum(as.double(h.star) * tr.vector),
              final.date = as.character(x[NROW(x), 'date'])))
}

bl <- compute.black.litterman(all.data);
plot(bl$h.eq, bl$h.star)
```




```
N <- NROW(all.data)
results.list <- list()

for(i in as.integer(N/4):N) {
  data.subset <- all.data[1:i,]
  results.list <- c(results.list, compute.black.litterman(data.subset));
}

get.field <- function(s) { unlist(lapply(results.list, function(X) get(s, envir = X))) }

h.eq.pl <- get.field('h.eq.pnl')
h.bl.pl <- get.field('h.bl.pnl')
actual.dates <- get.field('final.date')

print(gg.plot(data.frame(date = actual.dates, EQ = h.eq.pl, BL = h.bl.pl),
              c('EQ', 'BL'), use.cumsum=TRUE))
```


1.2
0.8
0.4
0.0
value
2005
2010
2015
date