Using the Bloomebrg and Eikon API
An Application to the Efficient Frontier

Antonio Fasano

April 29, 2021

1 Download Stock Data

Let us make a bit of practice with downloading and storing data. When you store and later read your file, it is suggested that you keep it the default working directory, `mybloomr`, so it moves with BloomR (USB pen) and you don't have to stress with its Windows path.

```r
## Make sure the terminal is on and connect...
con <- blpConnect()

## Last 31 days S&P500
bdh("SPX Index", c("PX_LAST", "VOLUME"), start.date=Sys.Date()-31)

## Set periodicity monthly and go back 5 years
opt <- c("periodicitySelection"="MONTHLY")
start  <- Sys.Date() -365 * 5
bdh("SPX Index", c("PX_LAST", "VOLUME"),
                                    start.date=start, options=opt)

## Set a specific start/end date
opt <- c("periodicitySelection"="MONTHLY")
start  <- as.Date("2019/01/01")
end    <- as.Date("2020/01/31")
bdh("SPX Index", "PX_LAST", start.date=start, end.date=end,
    options=opt)

## Download a portfolio as a list of time series (data frames)
tiks <- c("MSFT", "GE", "GM", "JPM")
start  <- as.Date("2010/01/01")
PL <- bdh(paste(tiks, "US Equity"), "PX_LAST",
          start.date=start, end.date=end, options=opt)

## Merge list into a single data frame by date (GM is shorter)
```

```r
P <- Reduce(function(x,y)
    merge(x,y, "date", suffixes = c("", ncol(x))), PL)
names(P) <- c("date", tiks)

## Save data for later use
saveRDS(P, file="port-bbg.rds")

## Disconnect from Bloomberg if you have finished
blpDisconnect(con)
```

Assembling data from Eikon requires some extra steps. In the future it will be more streamlined.

```r
## Make sure you are logged into Eikon

## Last 31 days S&P500 (dates require also the ISO T00:00:00 time)
ts <- function(d) paste0(d, "T00:00:00")
now <- Sys.Date()
get_timeseries(".SPX", c("TIMESTAMP", "CLOSE"), ts(now-31), ts(now))

## Download all available fields
get_timeseries(".SPX", "*", ts(now-31), ts(now))

## Use monthly periodicity and go back 5 years
start  <- ts(Sys.Date() -365 * 5)
get_timeseries(".SPX", c("TIMESTAMP", "CLOSE"), start, ts(now),
               interval="monthly")

## Set a specific start/end date
start <- "2019-01-01T00:00:00"
end   <- "2020-01-31T00:00:00"
get_timeseries(".SPX", c("TIMESTAMP", "CLOSE"), start, end,
               interval="monthly")

## Download a portfolio as a data frame of stacked time series
tiks <- list("MSFT.O", "GE", "GM", "JPM")# a list not a c() vector!
start <- "2010-01-01T00:00:00"
P.df  <- get_timeseries(tiks, c("TIMESTAMP", "CLOSE"), start, end,
                        interval="monthly")

## Split by ticker-column and remove it
tiks  <- unlist(tiks)
n <- ncol(P.df)
PL <- lapply(split(P.df, P.df[[n]])[tiks], `[`, -n)
```

```r
## Data come as characters
for(i in seq_along(tiks)) {
    PL[[i]]$CLOSE <- as.numeric(PL[[i]]$CLOSE)
    PL[[i]]$TIMESTAMP <- as.Date(PL[[i]]$TIMESTAMP)
}

## Merge list into a single data frame by TIMESTAMP (GM is shorter)
P <- Reduce(function(x,y) merge(x,y, "TIMESTAMP",
                                suffixes = c("", ncol(x))), PL)
names(P) <- c("date", tiks)

## Save data for later use
saveRDS(P, file="port-eikon.rds")
```

The file `port-bbg.rds`, or `port-eikon.rds`, is now in your `mybloomr` folder. If you keep it here, will be easily found by the `load` function later. If you move it to another path, you should specify its path in the `readRDS` function or change the R working directory accordingly. Remember that, if a Windows path is "C:
Users
Antonio", you will write it in R as "C:/Users/Antonio" or "C:

Users

Antonio".

2 Identify a Model

if $\bar{r}$ is the portfolio expected-return vector and w are the related weights, let us assume we want to obtain a target return, μ, with least portfolio risk, measured by its variance. That is:

$$\begin{aligned} \min_{w} \quad & w'\mathbf{S}w \\ \text{sub} \quad & w'\bar{r} = \mu \\ & w'\mathbf{1} = 1 \end{aligned}$$

where $\mathbf{S}$ is the portfolio variance-covariance matrix.

This solves to the optimal weight vector:

$$w^* = \mathbf{S}^{-1}\frac{(\mu c - b)\bar{r} + (a - \mu b)\mathbf{1}}{ac - b^2}$$

where:

$$\begin{aligned} a &= \bar{r}'\mathbf{S}^{-1}\bar{r} \\ b &= \bar{r}'\mathbf{S}^{-1}\mathbf{1} \\ c &= \mathbf{1}'\mathbf{S}^{-1}\mathbf{1} \end{aligned}$$

3 Optimise your Portfolio Weights

We will now apply the formulas from the previous section to optimise our the downloaded portfolio, using the average of the expected stock returns as a target.

```
## Load your data file
P <- readRDS("port-bbg.rds")

## Get a matrix without dates
P.m <- as.matrix(P[names(P) != "date"])

## For simplicity, we don't use arithmetic returns
R <- diff(log(P.m))

## Expected values
r <- matrix(colMeans(R))*12

## Covariance Matrix
S <- cov(R)*12
S1 <- solve(S)

## Vector of ones
one <- matrix(rep(1, length(r)))

## We use the average historical return as target return
u <- mean(r)

## Optimal weights
A <- t(r) %*% S1 %*% r
B <- t(r) %*% S1 %*% one
C <- t(one) %*% S1 %*% one
num <- r %*% (u*C-B) + one %*% (A-u*B)
den <- A*C - B^2
w <- S1 %*% num %*% (1/den)

## Portfolio return for u
u <- t(r) %*% w

## Portfolio variance for u
v <- t(w) %*% S %*% w
```

We have obtained the optimal weights w for our portfolio, and they are:

$$0.4107, 0.3376, 0.2308, 0.0209$$

The related mean and variance are:

$$0.0804, 0.0306$$

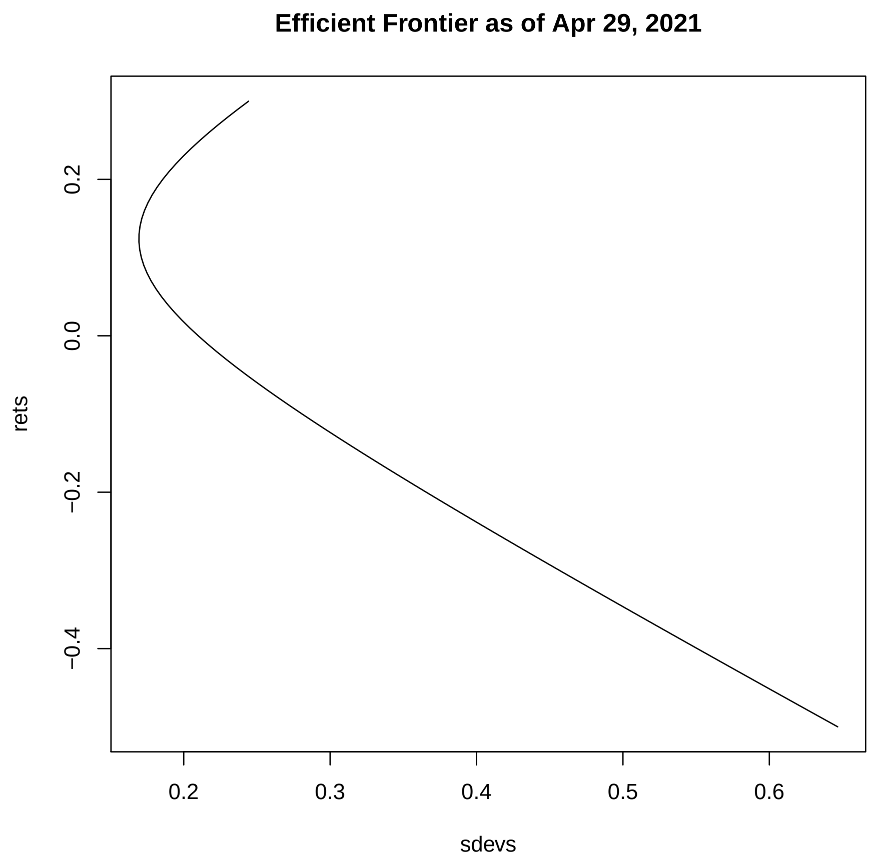
We can now think of a function, $w()$, generating for each target the optimal weights and from these compute the minimum level of portfolio risk, $sd()$.

```r
## Make the weight function
w <- function(u) {
    A <- t(r) %*% S1 %*% r
    B <- t(r) %*% S1 %*% one
    C <- t(one) %*% S1 %*% one
    num <- r %*% (u*C-B) + one %*% (A-u*B)
    den <- A*C - B^2
    S1 %*% num %*% (1/den)
}

## St. dev. function
sd <- function(u) {
    (t(w(u)) %*% S %*% w(u))^.5
}
```

The function $sd()$, repeatedly applied to target returns, gives us the efficient frontier, which is the upper part of the plotted envelope.

```r
## Make the plot
rets <- seq(-0.5, 0.3, by=0.01)
sdevs <- sapply(rets, sd)
plot(sdevs, rets, type="l", main=paste("Efficient Frontier as of",
                                 format(Sys.Date(), "%b %d, %Y")))
```


Efficient Frontier as of Apr 29, 2021
rets
0.2
0.0
−0.2
−0.4
0.2
0.3
0.4
0.5
0.6
sdevs