









|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |























|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |









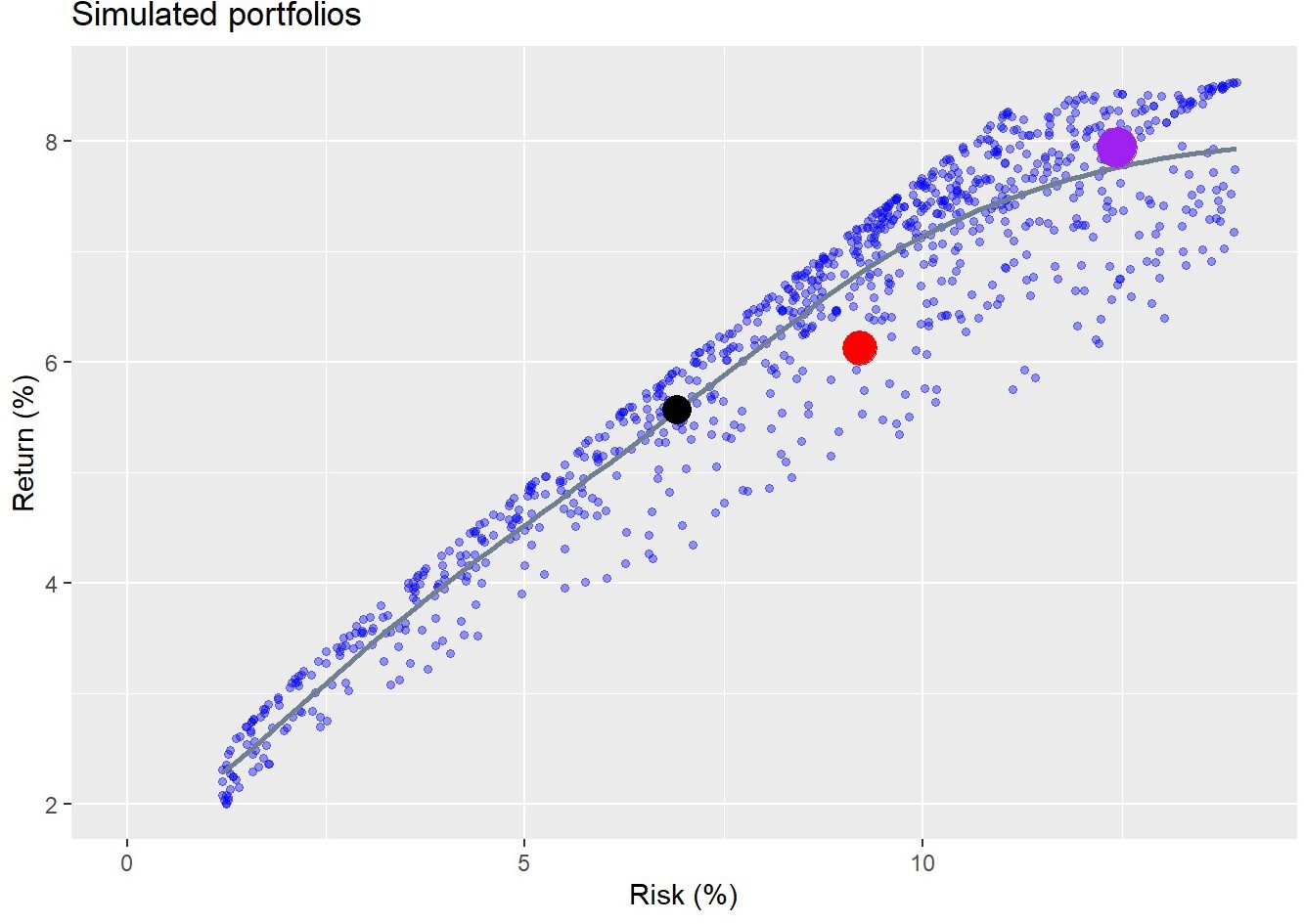








































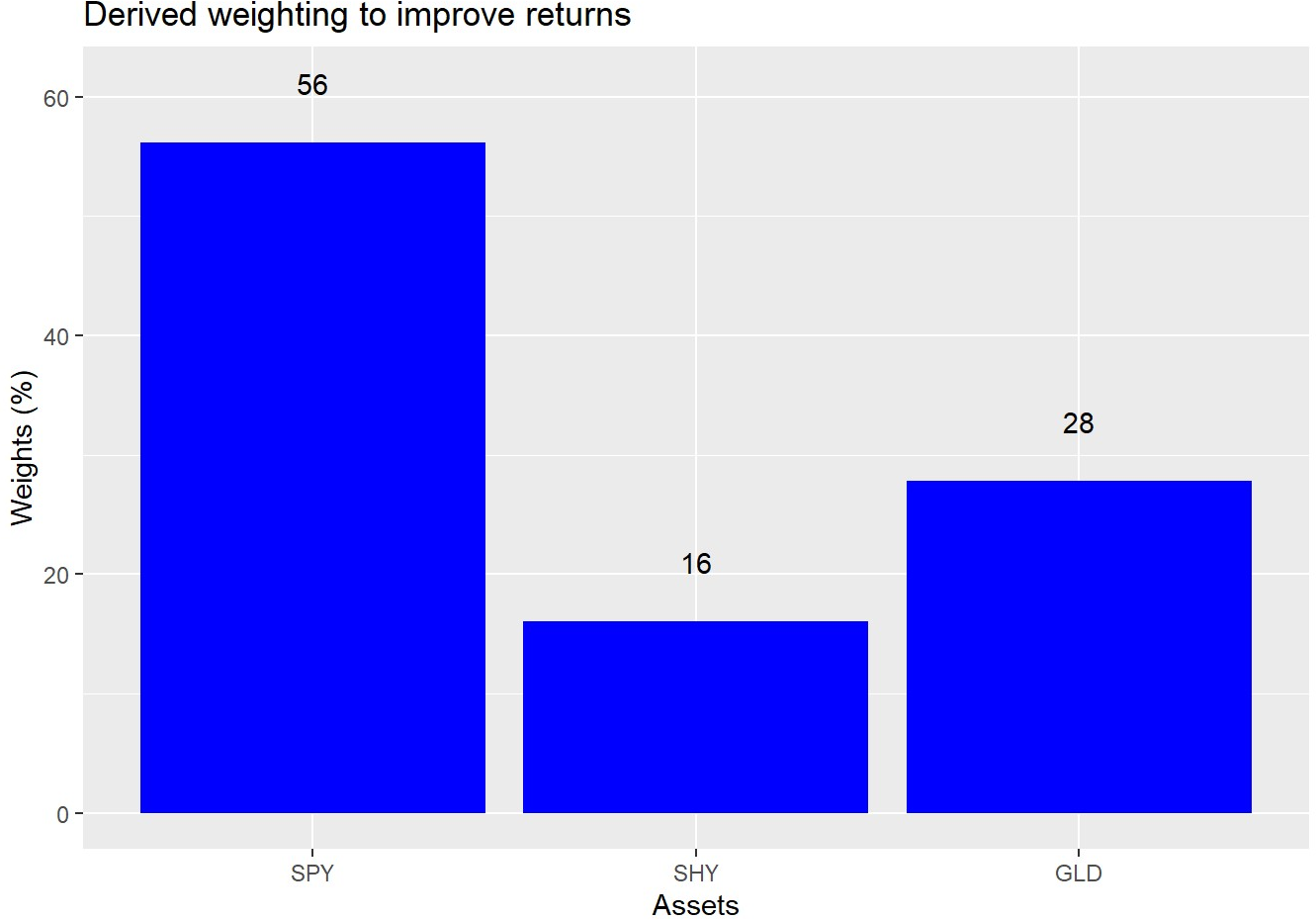






















































# Load package library(tidyquant) library(tidyverse)

# Get data

symbols <- c("SPY", "EEM", "SHY", "IYR", "GLD")

symbols\_low <- tolower(symbols)

prices <- getSymbols(symbols, src = "yahoo",

from = "1990-01-01",

auto.assign = TRUE) %>% map(~Ad(get(.))) %>%

reduce(merge) %>%

`colnames<-`(symbols\_low)

prices\_monthly <- to.monthly(prices, indexAt = "last", OHLC = FALSE) ret <- ROC(prices\_monthly)["2005/2019"]

naive <- ret[,c("spy", "shy")]

basic <- ret[,c("spy", "shy", "gld")]

# Create different weights and portflios wt1 <- rep(1/(ncol(basic)), ncol(basic)) port1 <- Return.portfolio(basic, wt1) %>%

`colnames<-`("ret")

wt2 <- c(0.9, 0.10, 0)

port2 <- Return.portfolio(basic, weights = wt2) %>%

`colnames<-`("ret")

wtn <- c(0.5, 0.5)

portn <- Return.portfolio(naive, wtn)

port\_comp <- data.frame(date = index(port1), equal = as.numeric(port1), wtd = as.numeric(port2),

naive = as.numeric(portn))

port\_comp %>% gather(key,value, -date) %>% group\_by(key) %>%

mutate(value = cumprod(value+1)) %>% ggplot(aes(date, value\*100, color = key)) + geom\_line() +

scale\_color\_manual("", labels = c("Equal", "Naive", "Risky"),

values = c("blue", "black", "red")) +

labs(x = "",

y = "Index",

title = "Three portfolios, which is best?", caption = "Source: Yahoo, OSM estimates") +

theme(legend.position = "top",

plot.caption = element\_text(hjust = 0))

# Portfolio summary table port\_comp %>%

rename("Equal" = equal,

"Naive" = naive, "Risky" = wtd) %>%

gather(Asset, value, -date) %>% group\_by(Asset) %>%

summarise(`Mean (%)` = round(mean(value, na.rm = TRUE),3)\*1200,

`Volatility (%)` = round(sd(value, na.rm = TRUE)\*sqrt(12),3)\*100,

`Risk-adjusted (%)` = round(mean(value, na.rm = TRUE)/sd(value, na.rm=TRUE)\*sqrt

`Cumulative (%)` = round(prod(1+value, na.rm = TRUE),3)\*100) %>% knitr::kable(caption = "Annualized performance metrics")

# Semi-deviation functions down\_dev <- function(vec){

mean\_vec <- mean(vec, na.rm = TRUE) down\_vec <- vec[vec < mean\_vec]

dev <- sqrt(mean((down\_vec - mean\_vec)^2)) dev

}

up\_dev <- function(vec){

mean\_vec <- mean(vec, na.rm = TRUE) up\_vec <- vec[vec > mean\_vec]

dev <- sqrt(mean((up\_vec - mean\_vec)^2)) dev

}

# Semi-deviation table port\_comp %>%

rename("Equal" = equal,

"Naive" = naive, "Risky" = wtd) %>%

gather(Asset, value, -date) %>% group\_by(Asset) %>%

summarise(`Down volatility (%)` = round(down\_dev(value)\*sqrt(12),3)\*100,

`Up volatility(%)` = round(up\_dev(value)\*sqrt(12),3)\*100) %>% knitr::kable(caption = "Annualized performance metrics")

## Portfolio simulations

# Portfolio

mean\_ret <- apply(ret[,c("spy", "shy", "gld")],2,mean)

cov\_port <- cov(ret[,c("spy", "shy", "gld")])

port\_exam <- data.frame(ports = colnames(port\_comp)[-1],

ret = as.numeric(apply(port\_comp[,-1],2, mean)), vol = as.numeric(apply(port\_comp[,-1], 2, sd)))

# Weighting that ensures more variation and random weighthing to stocks set.seed(123)

wts <- matrix(nrow = 1000, ncol = 3)

for(i in 1:1000){

a <- runif(1,0,1) b <- c()

for(j in 1:2){

b[j] <- runif(1,0,1-sum(a,b))

}

if(sum(a,b) < 1){

inc <- (1-sum(a,b))/3 vec <- c(a+inc, b+inc)

}else{

vec <- c(a,b)

}

wts[i,] <- sample(vec,replace = FALSE)

}

# Calculate random portfolios

port <- matrix(nrow = 1000, ncol = 2) for(i in 1:1000){

port[i,1] <- as.numeric(sum(wts[i,] \* mean\_ret))

port[i,2] <- as.numeric(sqrt(t(wts[i,] %\*% cov\_port %\*% wts[i,])))

}

colnames(port) <- c("returns", "risk") port <- as.data.frame(port)

# Graph with points port %>%

ggplot(aes(risk\*sqrt(12)\*100, returns\*1200)) + geom\_point(color = "blue", size = 1.2, alpha = 0.4) +

geom\_smooth(method = "loess", formula = y ~ log(x), se = FALSE, color = "slategrey") + geom\_point(data = port\_exam, aes(port\_exam[1,3]\*sqrt(12)\*100,

port\_exam[1,2]\*1200), color = "red", size = 6) +

geom\_point(data = port\_exam, aes(port\_exam[2,3]\*sqrt(12)\*100,

port\_exam[2,2]\*1200), color = "purple", size = 7) +

geom\_point(data = port\_exam, aes(port\_exam[3,3]\*sqrt(12)\*100,

port\_exam[3,2]\*1200), color = "black", size = 5) +

scale\_x\_continuous(limits = c(0,14)) + labs(x = "Risk (%)",

y = "Return (%)",

title = "Simulated portfolios")

# Dominated portfolios naive\_dom <- port %>%

filter(risk < port\_exam[3,3]+0.0005, risk > port\_exam[3,3]-0.0005) %>%

summarise(round(max(returns) - port\_exam[3,2],4)\*1200+.02) %>% as.numeric()

equal\_dom <- port %>%

filter(risk < port\_exam[1,3]+0.0005, risk > port\_exam[1,3]-0.0005) %>%

summarise(round(max(returns) - port\_exam[1,2],3)\*1200) %>% as.numeric()

risky\_dom <- port %>%

filter(risk < port\_exam[2,3]+0.0005, risk > port\_exam[2,3]-0.0005) %>%

summarise(round(max(returns) - port\_exam[2,2],4)\*1200+.02) %>% as.numeric()

# Finad max and equivalent risk for Equal risk slice equal\_max <- port %>%

filter(risk < port\_exam[1,3]+0.0005, risk > port\_exam[1,3]-0.0005) %>%

mutate(returns = returns\*1200,

risk = risk \* sqrt(12)\*100) %>% arrange(desc(returns)) %>%

slice(1)

# Find wieghts for dominant portfolio eq\_wt <- port %>%

mutate(spy\_wt = wts[,1], shy\_wt = wts[,2], gld\_wt = wts[,3],

returns = returns \* 1200,

risk = risk \* sqrt(12) \*100) %>% filter(returns == equal\_max$returns,

risk == equal\_max$risk) %>% select(spy\_wt, shy\_wt, gld\_wt)

# Graph weights eq\_wt %>%

rename("SPY" = spy\_wt,

"SHY" = shy\_wt, "GLD" = gld\_wt) %>%

gather(key,value) %>%

ggplot(aes(factor(key, level = c("SPY", "SHY", "GLD")), value\*100)) + geom\_bar(stat = 'identity', fill = "blue") +

geom\_text(aes(label = round(value,2)\*100), nudge\_y = 5) + labs(x = "Assets",

y = "Weights (%)",

title = "Derived weighting to improve returns")