we introduced benchmarking as a way to analyze our hero’s investment results apart from comparing it to alternate weightings or Sharpe ratios. In this case, the benchmark was meant to capture the returns available to a global aggregate of investable risk assets. If you could own almost every stock and bond globally and in the same proportion as their global contribution, what would your returns look like? We then used this benchmark as a way to judge our hero’s portfolio. We looked first at returns in excess of the benchmark scaled by volatility. Then we looked at excess returns scaled by deviation from the benchmark. We found that our hero’s portfolio was not being compensated for its deviation from the benchmark. The reason: gold was a drag on performance.

In this post, we move away from portfolio selection to rebalancing. Our goal will be to set up the intuition behind rebalancing, which will launch us into a more detailed discussion in further posts.

Why rebalance? Assume you establish some reasonably well defined risk and return parameters. Then you select the assets that offer the best approximation of achieving those parameters.[1](https://osm.netlify.app/post/rebalancing-really/#fn1) As those assets gain or lose value, the exposures change, leaving you with a different level of potential risk and return than originally envisioned.

A simple example. You invested at the bottom of the global financial crisis with a 50/50 weighting to SPY and SHY (the assets we’ve been using to proxy stocks and bonds). A year later, that weighting would be closer to 54/46. Moreover, you’re annualized risk has gone up by almost half a percentage point due to the higher exposure to stocks. While that might not seem like a lot, it may be more than you want. Indeed, with more assets and more variance in the returns, the weights and ending risk exposure could change significantly. So you rebalance to return to the original risk-return parameters.

Another reason to rebalance is if external factors have changed your risk or return parameters The most obvious example for an individual is the approach of, and then transition to, retirement. As one approaches retirement, risk tolerance generally declines since you don’t want to risk losing what you’ve worked so hard to gain. Hence, there’s a typical shift to assets with lower volatility. But other factors can change the parameters as well—changes in inflation, mortality, and health expectations are a few. For professional investors, reasons to rebalance include changes in funding mandates, sources, and scope. But we won’t go into those.

A third reason to rebalance is if the expectations for future risk and return profiles of the assets in the portfolio have changed. One sees this most ofthen in actively managed portfolios; that is, portfolios attempting to beat a benchmark. But even if you’re not trying to beat an index, if your expectations for asset returns and risk changes dramatically, then there’s a very good reason to rebalance. While this seems logical, getting it right is really tough. Unless you’ve got a crystal ball or there’s some obvious secular change, the risk of being wrong is great. In any event, we won’t say much more about that here, but will touch on this in depth when we get to the post on capital market expectations.

Apart from imminent retirement or some exogenous factor, the rationale behind rebalancing suggests that it should lead to better risk-adjusted returns. You sell winners to buy losers. And since “trees don’t grow to the sky” and nothing “stays down forever”[2](https://osm.netlify.app/post/rebalancing-really/#fn2) you’re dampening volatility by selling at the peak and buying near the trough. Or so the logic goes. If this is true, our hero should probably think about what type of rebalancing regime he might want to employ. Let’s see if rebalancing produces better results.

First off, we should note that we did not rebalance the various portfolios we looked at; that is, the equal-weighted (our hero’s), the naive, or the risky. We started them at their respective weights in 2005 and did not change. For the benchmark, however, we did rebalance every quarter. We did this to approximate the way many benchmarks are constructed. What would returns look like if we had rebalanced? We’ll look at results for different rebalancing periods: none, monthly, quarterly, and yearly. The table below shows our hero’s equal-weighted portfolio.

| Table 1: Equal-weighted portfolio performance for different rebalance periods (%) | | | | |
| --- | --- | --- | --- | --- |
| **Period** | **Return** | **Volatility** | **Sharpe** | **Total return** |
| None | 6.1 | 7.6 | 0.82 | 140.0 |
| Months | 6.1 | 9.2 | 0.67 | 134.5 |
| Quarters | 6.1 | 7.6 | 0.81 | 138.8 |
| Years | 6.2 | 7.6 | 0.83 | 142.8 |

Interestingly, we see very little difference in the average annual return. And the only difference in the remaining metrics is for the monthly rebalancing, which suffers higher volatility and thus a lower Sharpe ratio and total return. What’s causing the performance drag for monthly rebalancing? It could be that volatility whipsaws returns, which gets captured by the shorter reallocation time frame. Or the monthly rebalancing could simply be unlucky. Instead of buying losers that become winners, you just buy losers. We’d need to look at this in more detail, but we’ll save that for a later post. Next up the naive portfolio.

| Table 2: Naive portfolio performance for different rebalance periods (%) | | | | |
| --- | --- | --- | --- | --- |
| **Period** | **Return** | **Volatility** | **Sharpe** | **Total return** |
| None | 5.6 | 6.9 | 0.82 | 122.0 |
| Months | 5.3 | 6.8 | 0.79 | 112.6 |
| Quarters | 5.3 | 6.8 | 0.80 | 114.4 |
| Years | 5.5 | 6.6 | 0.84 | 119.6 |

At first glance, it’s unclear why monthly and quarterly rebalancing exhibit worse performance than the others. But it is probably not due to whipsawing, since SHY is mainly short-term US Treasuries. Hence, there’s not a lot of volatility. It is more likely due to performance drag since volatility is relatively the same across difference rebalancing periods. Whatever the case, let’s move on to the risky portfolio.

| Table 3: Risky portfolio performance for different rebalance periods (%) | | | | |
| --- | --- | --- | --- | --- |
| **Period** | **Return** | **Volatility** | **Sharpe** | **Total return** |
| None | 7.9 | 12.4 | 0.65 | 191.6 |
| Months | 7.9 | 12.5 | 0.64 | 188.7 |
| Quarters | 7.9 | 12.5 | 0.64 | 189.6 |
| Years | 8.0 | 12.4 | 0.65 | 192.5 |

Not surprisingly, a portfolio that has a 90% weighting to one asset, won’t see a lot of difference in performance due to different rebalancing periods. That’s because, unless the asset soared or crashed, the weighting is likely to remain relatively stable, so rebalancing would have a minimal effect. Finally, let’s look at different rebalancing periods for the benchmark portfolio

| Table 4: Benchmark portfolio performance for different rebalance periods (%) | | | | |
| --- | --- | --- | --- | --- |
| **Period** | **Return** | **Volatility** | **Sharpe** | **Total return** |
| None | 5.7 | 5.1 | 1.11 | 39.3 |
| Months | 5.7 | 5.2 | 1.11 | 39.4 |
| Quarters | 5.7 | 5.1 | 1.11 | 39.3 |
| Years | 5.7 | 5.2 | 1.10 | 39.4 |

Hmm. Nothing dramatic here either. So what gives? We presented all these rebalancing strategies and the surprise was that there wasn’t much of a surprise. Despite what appears to be sound logic, rebalancing did not produce better risk-adjusted returns in most cases. This confirms what some pundits argue: that rebalancing is useless. On the other hand, there are some who claim that it generates not just better risk-adjusted returns, but may even offer consistent outperformance.

We can’t run a full test on those competing claims here, we’ll do that in the next post along with providing some links to the different views, For now, let’s at least get a flavor of the kind of test we might use.

We simulate a portfolio of 10 years of monthly returns whose assets match the real returns and risk of US and global stocks and bonds and the S&P GSCI Commodity Total Return Index. We apply the same rebalancing periods as before. We present the table of performance metrics below.

| Table 5: Simulated portfolio performance for different rebalance periods (%) | | | | |
| --- | --- | --- | --- | --- |
| **Period** | **Return** | **Volatility** | **Sharpe** | **Total return** |
| None | 5.5 | 7.6 | 0.73 | 68.1 |
| Months | 6.2 | 7.2 | 0.87 | 81.3 |
| Quarters | 6.1 | 7.2 | 0.86 | 79.3 |
| Years | 5.8 | 7.2 | 0.81 | 74.1 |

We see that monthly and quarterly rebalancing produce modestly better risk-adjusted and total returns. Are these results significant? When we run t-tests on the returns, we find little to suggest the differences are anything more than noise. This is based on the p-values we show in the table below. For those who never took statistics or who find the mention of stats causes an immediate gag reflex, just note that the p-values are no where near 5%, which means the differences are likely due to randomness. We won’t test the significance of the Sharpe ratio for now simply because it involves some more sophisticated techniques.

| Table 6: Simulated portfolio p-values from t-test for differences in returns | |
| --- | --- |
| **Periods** | **P-values** |
| None vs. Months | 0.83 |
| None vs. Quarters | 0.85 |
| None vs. Years | 0.92 |
| Months vs. Quarters | 0.97 |
| Months vs. Years | 0.90 |
| Quarters vs. Years | 0.93 |

What have we learned thus far? The logic behing rebalancing appears straightforward: when circumstances (internal or external) change, then the portfolio should be rebalanced to weights that match the prior or new risk-return parameters. The main reason for this was to maintain or improve risk-adjusted returns. Yet, when we ran different rebalancing scenarios, we found little evidence to suggest returns were any different. And one simulation using historical returns of major assets, also showed little evidence of significant differences in performance.

Nonetheless, we’re not yet ready to say rebalancing is hogwash. There’s still more to do including running thousands of simulations to test for significant differences between rebalancing periods; examining whether rebalancing due to life changes does what it says on the tin; and testing rebalancing on a larger set of assets. But we’ll save those for the next couple posts. Until then, here’s the code:

# Load package

library(tidyquant)

library(tidyverse)

# Get data

symbols <- c("SPY", "SHY", "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"]

bench\_sym <- c("VTI", "VXUS", "BND", "BNDX")

bench <- getSymbols(bench\_sym, src = "yahoo",

from = "1990-01-01",

auto.assign = TRUE) %>%

map(~Ad(get(.))) %>%

reduce(merge) %>%

`colnames<-`(tolower(bench\_sym))

bench <- to.monthly(bench, indexAt = "last", OHLC = FALSE)

bench\_ret <- ROC(bench)["2014/2019"]

# Create different weights and portfolios

wt1 <- rep(1/(ncol(ret)), ncol(ret))

port1 <- Return.portfolio(ret, wt1) %>%

`colnames<-`("ret")

wt2 <- c(0.9, 0.1, 0)

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

`colnames<-`("ret")

wtn <- c(0.5, 0.5, 0)

portn <- Return.portfolio(ret, wtn)

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

risky = as.numeric(port2),

naive = as.numeric(portn))

## Rebalancing for equal

# Create list

rebal = c("months", "quarters", "years")

equal\_list <- list()

for(pd in rebal){

equal\_list[[pd]] <- Return.portfolio(ret, wt1, rebalance\_on = pd) %>%

`colnames<-`(pd)

}

# Create data frame

equal <- equal\_list %>% bind\_cols() %>%

data.frame() %>%

mutate\_all(as.numeric) %>%

mutate(date = index(easy\_list[["months"]]),

none = as.numeric(port1)) %>%

select(date, none, everything())

equal %>%

rename("Months" = months,

"Quarters" = quarters,

"Years" = years,

"None" = none) %>%

gather(Period,value, -date) %>%

mutate(Period = factor(Period, labels = c("None", "Months", "Quarters", "Years"))) %>%

group\_by(Period) %>%

summarise(Return = round(mean(value)\*12,3)\*100,

Volatility = round(sd(value)\*sqrt(12),3)\*100,

Sharpe = round(mean(value)/sd(value)\*sqrt(12),2)+.01,

`Total return` = round(prod(1+value)-1,3)\*100) %>%

knitr::kable(caption = "Equal-weighted portfolio performance (%) for different rebalance periods")

# Rebalance for naive

naive\_list <- list()

for(pd in rebal){

naive\_list[[pd]] <- Return.portfolio(ret, wtn, rebalance\_on = pd) %>%

`colnames<-`(pd)

}

naive <- naive\_list %>%

bind\_cols() %>%

data.frame() %>%

mutate\_all(as.numeric) %>%

mutate(date = index(naive\_list[["months"]]),

none = as.numeric(portn)) %>%

select(date, none, everything())

naive %>%

rename("None" = none,

"Months" = months,

"Quarters" = quarters,

"Years" = years) %>%

gather(Period,value, -date) %>%

mutate(Period = factor(Period, levels = c("None", "Months", "Quarters", "Years"))) %>%

group\_by(Period) %>%

summarise(Return = round(mean(value),3)\*1200,

Volatility = round(sd(value)\*sqrt(12),3)\*100,

Sharpe = round(mean(value)/sd(value)\*sqrt(12),2)+.01,

`Total return` = round(prod(1+value)-1,3)\*100) %>%

knitr::kable(caption = "Risk and returns for different rebalance periods (%)")

# Rebalance for risky

risky\_list <- list()

for(pd in rebal){

risky\_list[[pd]] <- Return.portfolio(ret, wt2, rebalance\_on = pd) %>%

`colnames<-`(pd)

}

risky <- risky\_list %>% bind\_cols() %>%

data.frame() %>%

mutate\_all(as.numeric) %>%

mutate(date = index(risky\_list[["months"]]),

none = as.numeric(port2)) %>%

select(date, none, everything())

risky %>%

rename("None" = none,

"Months" = months,

"Quarters" = quarters,

"Years" = years) %>%

gather(Period,value, -date) %>%

mutate(Period = factor(Period, levels = c("None", "Months", "Quarters", "Years"))) %>%

group\_by(Period) %>%

summarise(Return = round(mean(value),3)\*1200,

Volatility = round(sd(value)\*sqrt(12),3)\*100,

Sharpe = round(mean(value)/sd(value)\*sqrt(12),2)+.01,

`Total return` = round(prod(1+value)-1,3)\*100) %>%

knitr::kable(caption = "Risk and returns for different rebalance periods (%)")

# Rebalance for benchmark

bench\_list <- list()

for(pd in rebal){

bench\_list[[pd]] <- Return.portfolio(bench\_ret, wtb, rebalance\_on = pd) %>%

`colnames<-`(pd)

}

bench\_rb <- bench\_list %>% bind\_cols() %>%

data.frame() %>%

mutate\_all(as.numeric) %>%

mutate(date = index(bench\_list[["months"]]),

none = as.numeric(portb)) %>%

select(date, none, everything())

bench\_rb %>%

rename("None" = none,

"Months" = months,

"Quarters" = quarters,

"Years" = years) %>%

gather(Period,value, -date) %>%

mutate(Period = factor(Period, levels = c("None", "Months", "Quarters", "Years"))) %>%

group\_by(Period) %>%

summarise(Return = round(mean(value)\*12+.0001,3)\*100,

Volatility = round(sd(value)\*sqrt(12),3)\*100,

Sharpe = round(mean(value)/sd(value)\*sqrt(12),2)+.01,

`Total return` = round(prod(1+value)-1,3)\*100) %>%

knitr::kable(caption = "Risk and returns for different rebalance periods (%)")

# Simulate

set.seed(123)

stock\_us <- rnorm(120, 0.08/12, 0.2/sqrt(12))

stock\_world <- rnorm(120, 0.065/12, 0.17/sqrt(12))

bond\_us <- rnorm(120, 0.024/12, 0.1/sqrt(12))

bond\_world <- rnorm(120, 0.025/12, 0.14/sqrt(12))

commod <- rnorm(120, 0.007, 0.057)

wt <- c(0.25, 0.25, 0.2, 0.2, 0.1)

date <- seq(as.Date("2010-02-01"), length = 120, by = "months")-1

port <- as.xts(cbind(stock\_us, stock\_world, bond\_us, bond\_world, commod),

order.by = date)

port\_list <- list()

rebals = c("months", "quarters", "years")

for(pd in rebals){

port\_list[[pd]] <- Return.portfolio(port, wt, rebalance\_on = pd)

}

port\_r <- port\_list %>%

bind\_cols() %>%

data.frame() %>%

mutate\_all(as.numeric) %>%

mutate(date = date,

none = as.numeric(none)) %>%

select(date, none, everything())

port\_r %>%

rename("None" = none,

"Months" = months,

"Quarters" = quarters,

"Years" = years) %>%

gather(Period,value, -date) %>%

mutate(Period = factor(Period, levels = c("None", "Months", "Quarters", "Years"))) %>%

group\_by(Period) %>%

summarise(Return = round(mean(value)\*12+.0001,3)\*100,

Volatility = round(sd(value)\*sqrt(12),3)\*100,

Sharpe = round(mean(value)/sd(value)\*sqrt(12),2)+.01,

`Total return` = round(prod(1+value)-1,3)\*100) %>%

knitr::kable(caption = "Risk and returns for different rebalance periods (%)")

count <- 0

port\_names <- c("None", "Months", "Quarters", "Years")

# paste(toupper(substr(colnames(port\_1)[-1], 1, 1)),

# substr(colnames(port\_1)[-1], 2, nchar(colnames(port\_1)[-1])), sep="")

t\_tests <- c()

for(i in 1:4){

for(j in 2:4){

if(i != j & count != 6){

t\_tests[paste(port\_names[i]," vs. ", port\_names[j])] <- t.test(port\_1[,i+1],

port\_1[,j+1])$p.value

count = count +1

}

}

}

t\_tests

data.frame(pds = names(t\_tests), p\_val = round(as.numeric(t\_tests),2)) %>%

rename("Periods" = pds,

"P-values" = p\_val) %>%

knitr::kable(caption = "T-test for means")