Portfolio Rebalancing Using Python

Posted on December 17, 2018 in [posts](https://evgenypogorelov.com/category/posts.html)

If you are an investor in stocks or bonds you are likely always looking for ways to increase your return or lower your risk, and it turns out there is a way to do both simultaneously called rebalancing. Portfolio rebalancing is something every investor should either do themselves or have done for them. Registered investment advisors like Vanguard and others will provide this service as part of an active management agreement, while robo-advisors such as Betterment will also rebalance automatically. To understand why it's so important, I'll defer to the finance professionals over at [BogleHeads](https://www.bogleheads.org/wiki/Rebalancing) :

Rebalancing is the action of bringing a portfolio that has deviated away from one's target asset allocation back into line. The objective is to maintain a consistent mix of asset classes (most commonly equities vs. fixed income) in order to control risk at the level desired by the investor. This is accomplished by transferring funds from higher-performing classes to lower-performing classes. While potentially counterintuitive, rebalancing ensures that investors "Buy Low" and "Sell High".

Much research has been conducted on the benefits of portoflio rebalancing, here is an excerpt from a fairly recent [Morgan Stanley](https://www.morganstanley.com/articles/rebalancing-effect) research piece:

Rebalancing takes advantage of the long-term effects of mean reversion. By lightening up on stocks after periods of significant outperformance, or topping off positions after periods of underperformance, this discipline helps take advantage of volatility to benefit from these swings. ... A disciplined approach to rebalancing portfolios annually can create additional return and lower volatility versus never rebalancing or rebalancing during different time periods. While investing for the long term requires patience, a disciplined approach to rebalancing can help create value beyond the cyclical trends of the market.

Timely and consistent rebalancing has become a cornerstone of modern portfolio theory. Rebalancing can magnify returns by promoting selling high and buying low, and reduce long-term risk by ensuring the portfolio adheres to its designated risk tolerance. The keys are to rebalance in a timely manner (i.e. annually) and to do it consistently because the benefits of rebalancing compound over time - but rebalancing by hand is a pain, and that can lead to inconsistency. I pursued this project due to the lack of free tools to simply rebalance an existing portfolio. Sure, we can all use a worksheet to do the math each time, but why not automate it to make it more likely we will actually do it? Here, inspired by the work of [kdboller](https://nbviewer.jupyter.org/github/kdboller/pythonsp500/blob/a7066d998ff046c3cc8b26ece3b0efdf00959d57/Investment%20Portfolio%20Python%20Notebook_03_2018_blog%20example.ipynb), I'll use Pandas, the Tiingo API, and some simple math to calculate how to optimally rebalance a portfolio given a target allocation. This is a simple, no-frills portfolio rebalancing exercise which does not factor in important considerations such as tax efficiency, transaction costs, minimum investment amounts, or alternate approaches such as stock-out rebalances, or bond-floor settings. Future versions of this project may contemplate these extra factors.

My portfolio is spread across various accounts and asset classes, so at a minimum this portfolio rebalancer had to consider how to best allocate assets within and across accounts. If new assets were added to the portfolio (based on the target portfolio definition) the rebalancer must also handle distributing those new assets into an existing account with adequate funds for it. A final consideration was basic asset tax location, I wanted the distribution of new assets to prioritize the appropriate accounts, for example, bonds will go in tax-free or tax-deferred acounts first and only to taxable accounts as a last resort. The reason for this tax location is bonds and bond funds are taxed as regular income (a higher rate) versus stocks which are taxed as capital gains (a lower rate) so we would generally prefer the higher taxed asset to be placed in a tax-advantaged account. The account-level distribution logic is constrained by available funds from sales in that same account, this is a simplifying constraint because the code does not currently contemplate IRA or Roth-IRA contribution levels or limits, and as a practical matter manually moving additional funds into a 401k may be impossible. Unfortunately, these additional considerations add to complexity and code length but I felt leaving them out would be too oversimplified and not actually useful for an average portfolio like my own.

The steps and static code are below, but thanks to the amazing folks over at [Binder](https://mybinder.org/) you can run this code interactively right in your browser and play with the rebalancer in real-time and change the inputs to suit your needs. Click the launch Binder link below to launch a new window which will load the code environment (it takes a few minutes).

**Steps:**

1. Set triggers to rebalance (time or threshold or both)
2. Define our current Portfolio (accounttype, accountid, lastrebaldate, ticker, shares, cost basis, assetclass)
3. Define our target allocation (ticker, allocation, assetclass)
4. Factor in any new money being invested
5. Aggregate to security level and calculate rebalance triggers to determine which securities must be rebalanced
6. Calculate initial transactions needed to hit target allocation (1 row per ticker)
7. Disaggregate portfolio back to account level (1 row per ticker per account)
8. Iteratively distribute new securities into accounts, keeping tax location in mind
9. Iteratively distribute across accounts any new securities which do not fit into a single account

**References:**

[pythonsp500 by kdboller](https://nbviewer.jupyter.org/github/kdboller/pythonsp500/blob/a7066d998ff046c3cc8b26ece3b0efdf00959d57/Investment%20Portfolio%20Python%20Notebook_03_2018_blog%20example.ipynb)

[Portfolio Rebalancing by bogleheads wiki](https://www.bogleheads.org/wiki/Rebalancing)

[The Rebalancing Effect by Morgan Stanely](https://www.morganstanley.com/articles/rebalancing-effect)

**Github Repo:** [pogoetic/rebalance](https://github.com/pogoetic/rebalance)

Code

In [2]:

*#Lets import the necessary packages*

**import** **pandas** **as** **pd**

**from** **IPython.display** **import** display

pd**.**set\_option('display.max\_columns', **None**)

**import** **numpy** **as** **np**

**import** **datetime**

**import** **decimal**

**from** **pandas\_datareader** **import** data **as** pdr

**from** **keys** **import** tiingo\_key

*#define todays datetime*

now **=** datetime**.**datetime**.**now()

In [3]:

*#Settings*

new\_money\_in **=** 10000

*#Set our rebalance threshold*

rebal\_threshold **=** **.**05 *#allowable allocation drift*

rebal\_timeframe **=** 180 *#in days*

In [4]:

*#Define target and current allocations*

*#create our target allocation*

columns\_t **=** ['ticker','allocation\_target','assetclass']

positions\_t **=** [['VTSAX',0.5652,'ST'],

['VIGAX',0.0131,'ST'],

['VSMAX',0.0066,'ST'],

['VSEQX',0.0066,'ST'],

['VWIGX',0.0507,'ST'],

['VTRIX',0.0507,'ST'],

['VTIAX',0.1521,'ST'],

['VBTLX',0.035,'BD'],

['VTABX',0.015,'BD'],

['VGSLX',0.05,'RE'],

['VNQI',0.01,'RE'],

['VDE',0.03,'ST'],

['GLD',0.015,'CS']]

*#set our current portfolio*

columns\_c **=** ['accounttype','accountid','lastrebaldate','ticker','assetclass','basisdate','costbasis','shares']

positions\_c **=** [['RIRA','1111','2018-11-16','VBTLX','BD','2018-11-16',1,913.483],

['RIRA','1111','2018-11-16','VTIAX','ST','2018-11-16',1,514.298],

['RIRA','1111','2018-11-16','VTSAX','ST','2018-11-16',10,151.121],

['RIRA','2222','2018-11-16','VBTLX','BD','2018-11-16',1,772.407],

['RIRA','2222','2018-11-16','VTSAX','ST','2018-11-16',20,151.578],

['TAXB','3333','2018-11-16','AAPL','ST','2018-11-16',1,3.14],

['TAXB','3333','2018-11-16','VTSAX','ST','2018-11-16',10,549.871]]

In [5]:

*#lookup table for account type abbreviations*

accounttypes **=** {'TAXB':'Taxable Brokerage', '401K':'401k', 'RIRA':'Roth-IRA', 'TIRA':'Traditional-IRA'}

assetclasses **=** {'ST':'Equity Stocks', 'BD':'Bonds Fixed-Income', 'CS':'Cash and Commodities', 'RE':'Real-Estate', 'ALT':'Alternatives'}

assettypes **=** {'SEC':'Individual Security', 'ETF':'Exchange Traded Fund', 'MF': 'Mutual Fund', 'IF':'Index Fund'}

assetregion **=** {'D':'Domestic','I':'International'}

*#initialize target portfolio*

targetalloc **=** pd**.**DataFrame(columns **=** columns\_t, data **=** positions\_t)

total**=**decimal**.**Decimal(targetalloc**.**allocation\_target**.**sum())

*#check that our target allocation indeed adds to 100%*

**assert** round(total,4) **==** 1,'Target Allocation not 100% : **{}**'**.**format(int(total))

*#initialize current portfolio*

start\_port **=** pd**.**DataFrame(columns **=** columns\_c, data **=** positions\_c)

start\_port**.**lastrebaldate **=** pd**.**to\_datetime(start\_port**.**lastrebaldate)

start\_port**.**basisdate **=** pd**.**to\_datetime(start\_port**.**basisdate)

*#custom apply function*

**def** **f**(x):

d **=** {}

d['lastrebaldate'] **=** x['lastrebaldate']**.**max()

d['assetclass'] **=** x['assetclass']**.**max()

d['basisdate'] **=** x['basisdate']**.**min()

d['costbasis'] **=** (x['costbasis'] **\*** x['shares'])**.**sum()**/**(x['shares']**.**sum() **or** 1) *#weighted avg*

d['shares'] **=** x['shares']**.**sum()

**return** pd**.**Series(d, index**=**['lastrebaldate', 'assetclass', 'basisdate', 'costbasis', 'shares'])

*#aggregate by ticker to account for duplicate securities held in different accounts*

agg\_port **=** start\_port**.**groupby(['ticker'])**.**apply(f)

*#Define list of distinct tickers we care about*

tickers **=** set(targetalloc**.**ticker**.**unique()**.**tolist()**+**start\_port**.**ticker**.**unique()**.**tolist())

In [6]:

*#Lets look at our input portfolios (target and current)*

display(targetalloc)

display(start\_port)

*#And our aggregated portfolio at the ticker level (1 row per ticker)*

display(agg\_port)

|  | | **ticker** | **allocation\_target** | | | **assetclass** | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | | VTSAX | 0.5652 | | | ST | |
| **1** | | VIGAX | 0.0131 | | | ST | |
| **2** | | VSMAX | 0.0066 | | | ST | |
| **3** | | VSEQX | 0.0066 | | | ST | |
| **4** | | VWIGX | 0.0507 | | | ST | |
| **5** | | VTRIX | 0.0507 | | | ST | |
| **6** | | VTIAX | 0.1521 | | | ST | |
| **7** | | VBTLX | 0.0350 | | | BD | |
| **8** | | VTABX | 0.0150 | | | BD | |
| **9** | | VGSLX | 0.0500 | | | RE | |
| **10** | | VNQI | 0.0100 | | | RE | |
| **11** | | VDE | 0.0300 | | | ST | |
| **12** | | GLD | 0.0150 | | | CS | |
|  | **accounttype** | | | **accountid** | **lastrebaldate** | | **ticker** | **assetclass** | **basisdate** | **costbasis** | **shares** |
| **0** | RIRA | | | 1111 | 2018-11-16 | | VBTLX | BD | 2018-11-16 | 1 | 913.483 |
| **1** | RIRA | | | 1111 | 2018-11-16 | | VTIAX | ST | 2018-11-16 | 1 | 514.298 |
| **2** | RIRA | | | 1111 | 2018-11-16 | | VTSAX | ST | 2018-11-16 | 10 | 151.121 |
| **3** | RIRA | | | 2222 | 2018-11-16 | | VBTLX | BD | 2018-11-16 | 1 | 772.407 |
| **4** | RIRA | | | 2222 | 2018-11-16 | | VTSAX | ST | 2018-11-16 | 20 | 151.578 |
| **5** | TAXB | | | 3333 | 2018-11-16 | | AAPL | ST | 2018-11-16 | 1 | 3.140 |
| **6** | TAXB | | | 3333 | 2018-11-16 | | VTSAX | ST | 2018-11-16 | 10 | 549.871 |

|  | **lastrebaldate** | **assetclass** | **basisdate** | **costbasis** | **shares** |
| --- | --- | --- | --- | --- | --- |
| **ticker** |  |  |  |  |  |
| **AAPL** | 2018-11-16 | ST | 2018-11-16 | 1.000000 | 3.140 |
| **VBTLX** | 2018-11-16 | BD | 2018-11-16 | 1.000000 | 1685.890 |
| **VTIAX** | 2018-11-16 | ST | 2018-11-16 | 1.000000 | 514.298 |
| **VTSAX** | 2018-11-16 | ST | 2018-11-16 | 11.777895 | 852.570 |

In [7]:

*#Next we pull the latest prices from Tiingo (YahooFinance is buggy, and IEX does not contain mutual fund data)*

*#Tiingo limits for free API: 500 unique tickers ever, 500 requests/hr, 20,000 requests/day*

*#*[*https://pandas-datareader.readthedocs.io/en/latest/remote\_data.html#tiingo*](https://pandas-datareader.readthedocs.io/en/latest/remote_data.html#tiingo)

*#Tiingo API key required: set 'tiingo\_key' value in python file called 'keys.py' in same directory as this script*

now **=** datetime**.**datetime**.**now()

yesterday **=** now **-** datetime**.**timedelta(3) *#avoids weekends with no data - need better weekend detection*

start **=** datetime**.**datetime(yesterday**.**year, yesterday**.**month, yesterday**.**day)

end **=** datetime**.**datetime(now**.**year, now**.**month, now**.**day)

bad\_tickers **=** []

**for** i, t **in** enumerate(tickers):

**try**:

**if** i**==**0:

ohlc **=** pdr**.**get\_data\_tiingo(t, api\_key**=**tiingo\_key)**.**tail(1)**.**close

**else**:

ohlc **=** ohlc**.**append(pdr**.**get\_data\_tiingo(t, api\_key**=**tiingo\_key)**.**tail(1)**.**close)

**except**:

bad\_tickers**.**append(t)

*#print(bad\_tickers)*

ohlc **=** ohlc**.**to\_frame(name**=**'close')

*#drop our date index since its only the latest data*

ohlc2**=**ohlc**.**reset\_index(level**=**1, drop**=True**)

*#Manual fix for known bad\_tickers which Tiingo can't find, adjust to suit your needs*

**if** 'VMFXX' **in** bad\_tickers:

ohlc2**.**loc['VMFXX'] **=** 1.0

In [8]:

*#What did we get?*

display(ohlc2)

|  | **close** |
| --- | --- |
| **symbol** |  |
| **VTABX** | 21.66 |
| **VIGAX** | 68.02 |
| **VSEQX** | 26.77 |
| **VTIAX** | 25.17 |
| **VNQI** | 52.05 |
| **VWIGX** | 24.76 |
| **VTSAX** | 61.08 |
| **VTRIX** | 31.91 |
| **VBTLX** | 10.39 |
| **VDE** | 77.44 |
| **VSMAX** | 62.25 |
| **AAPL** | 156.15 |
| **VGSLX** | 105.06 |
| **GLD** | 120.57 |

In [9]:

*#concatenate target allocation and latest prices with our portfolio*

start\_port\_c **=** pd**.**merge(agg\_port, targetalloc, on **=**'ticker', how **=**'outer')

final\_port **=** pd**.**merge(start\_port\_c, ohlc2, left\_on **=**'ticker', right\_index **=** **True**, how **=** 'left')

*#set target to zero for tickers held but not present in our target allocation, set initial basisdate and costbasis for new securities entering the portfolio*

final\_port**.**fillna(value **=** {'allocation\_target':0.0,'shares':0.0,'basisdate':pd**.**to\_datetime(now**.**strftime("%Y-%m-**%d**")),'costbasis':final\_port**.**close,'assetclass\_x':final\_port**.**assetclass\_y},inplace **=** **True**)

final\_port**.**drop(['assetclass\_y'],axis**=**1,inplace**=True**)

final\_port**.**rename(columns**=**{'assetclass\_x':'assetclass'},inplace**=True**)

*#calculate holding values and current allocation*

final\_port['value'] **=** final\_port**.**close **\*** final\_port**.**shares *#calculate value as price x shares*

final\_port**.**loc[final\_port**.**value**.**isna() **&** final\_port**.**shares**.**isna(),['value']]**=**0.0 *#for securities not currently held but in our target (and close price failed to return), establish zero value*

final\_port['allocation'] **=** final\_port**.**value **/** final\_port**.**value**.**sum()

final\_port['correction'] **=** final\_port**.**allocation\_target **-** final\_port**.**allocation

final\_port['new\_money\_in'] **=** new\_money\_in **\*** final\_port**.**allocation\_target *#Account for new money in*

In [10]:

*#create timedelta int column*

final\_port['timedelta'] **=** (final\_port**.**lastrebaldate **-** pd**.**to\_datetime(now**.**strftime("%Y-%m-**%d**")))**.**dt**.**days

final\_port**.**timedelta**.**fillna(0,inplace**=True**)

*#define rebalance flags to determine if we must rebalance*

final\_port['rebal\_flag\_thresh'] **=** np**.**where((abs(final\_port**.**correction)**<=**rebal\_threshold) **&** (final\_port**.**allocation **>** 0),0,1)

final\_port['rebal\_flag\_time'] **=** np**.**where(final\_port**.**timedelta **>=** rebal\_timeframe,1,0)

final\_port['rebal\_flag\_exit'] **=** np**.**where((final\_port**.**allocation **>** 0) **&** (final\_port**.**allocation\_target**==**0),1,0) *#force rebal securities not present in our target portfolio*

final\_port['rebal\_flag\_newmoney'] **=** np**.**where(final\_port**.**new\_money\_in**>**0,1,0)

final\_port['rebal\_flag'] **=** np**.**where(final\_port**.**rebal\_flag\_thresh **+** final\_port**.**rebal\_flag\_time **+** final\_port**.**rebal\_flag\_exit **+** final\_port**.**rebal\_flag\_newmoney **>=** 1,1,0)

*#Subset of securities we need to rebalance, and those we need to leave alone*

rebal\_port **=** final\_port[final\_port**.**rebal\_flag**==**1]**.**copy()

stable\_port **=** final\_port[final\_port**.**rebal\_flag**==**0]**.**copy()

In [11]:

*#Calculate our current allocation, target, and the change we need to hit target*

total\_val **=** rebal\_port**.**value**.**sum()

rebal\_port['allocation'] **=** rebal\_port**.**value**/**rebal\_port**.**value**.**sum()

rebal\_port['allocation\_target'] **=** rebal\_port**.**allocation\_target**/**rebal\_port**.**allocation\_target**.**sum()

rebal\_port['correction'] **=** rebal\_port**.**allocation\_target **-** rebal\_port**.**allocation

*#Factor in any new money entering the portfolio and determine necessary changes in value and shares*

rebal\_port['value\_chg'] **=** (total\_val **\*** rebal\_port**.**correction) **+** rebal\_port**.**new\_money\_in

rebal\_port['shares\_chg'] **=** rebal\_port**.**value\_chg **/** rebal\_port**.**close

rebal\_port**.**loc[rebal\_port**.**value\_chg**.**isna() **&** rebal\_port**.**shares **>** 0,['shares\_chg']]**=-**rebal\_port**.**shares *#sell all shares of securities not in our target portfolio*

*#Round off shares to whole numbers, except when we are fully exiting a position*

rebal\_port['shares\_chg\_round'] **=** rebal\_port**.**shares\_chg

rebal\_port **=** rebal\_port**.**astype({'shares\_chg\_round': int})

rebal\_port['final\_shares\_chg'] **=** rebal\_port**.**shares\_chg

rebal\_port**.**loc[np**.**round(rebal\_port**.**shares\_chg**+**rebal\_port**.**shares)**!=**0,['final\_shares\_chg']]**=**rebal\_port**.**shares\_chg\_round**\***1.0

rebal\_port**.**drop(['shares\_chg\_round'],axis**=**1,inplace**=True**)

*#Calculate initial new shares and values*

rebal\_port['new\_shares'] **=** np**.**round(rebal\_port**.**shares **+** rebal\_port**.**final\_shares\_chg,3)

rebal\_port['new\_value'] **=** rebal\_port**.**new\_shares **\*** rebal\_port**.**close *#due to share rounding, there will be slight variance vs. portfolio starting value*

rebal\_port['new\_value\_chg'] **=** rebal\_port**.**final\_shares\_chg **\*** rebal\_port**.**close

In [12]:

*#Double check our work so far*

*#net of buying and selling should be zero*

**assert**(np**.**round(rebal\_port**.**value\_chg**.**sum(),3)**-**new\_money\_in**==**0)

*#make sure totals match (with rounding error + new money in) from original portfolio and rebalanced portfolio*

**assert**(np**.**round(rebal\_port**.**new\_value**.**sum() **-** rebal\_port**.**value**.**sum(),3)**==**np**.**round((rebal\_port**.**new\_value**.**sum() **+** stable\_port**.**value**.**sum()) **-** final\_port**.**value**.**sum(),3))

In [13]:

*#Merge our rebalanced portfolio with our stable portfolio for our execution portfolio*

stable\_port['value\_chg'] **=** 0

stable\_port['shares\_chg']**=**0

stable\_port['final\_shares\_chg'] **=** 0

stable\_port['new\_value\_chg'] **=** 0

stable\_port['new\_shares'] **=** stable\_port**.**shares

stable\_port['new\_value'] **=** stable\_port**.**value

exec\_port **=** pd**.**concat([rebal\_port,stable\_port],sort**=False**)

exec\_port**.**drop(columns**=**['timedelta','rebal\_flag\_thresh','rebal\_flag\_time','rebal\_flag\_exit','rebal\_flag\_newmoney','value\_chg','shares\_chg'],inplace**=True**)

*#Reset allocations to be based on all securities*

exec\_port['allocation'] **=** exec\_port**.**value**/**exec\_port**.**value**.**sum()

exec\_port['allocation\_target'] **=** exec\_port**.**allocation\_target**/**exec\_port**.**allocation\_target**.**sum()

exec\_port['correction'] **=** exec\_port**.**allocation\_target **-** exec\_port**.**allocation

exec\_port['final\_allocation'] **=** exec\_port**.**new\_value **/** exec\_port**.**new\_value**.**sum()

In [14]:

*#Lets look at all our work to get to our target portfolio*

exec\_port

Out[14]:

|  | **ticker** | **lastrebaldate** | **assetclass** | **basisdate** | **costbasis** | **shares** | **allocation\_target** | **close** | **value** | **allocation** | **correction** | **new\_money\_in** | **rebal\_flag** | **final\_shares\_chg** | **new\_shares** | **new\_value** | **new\_value\_chg** | **final\_allocation** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | AAPL | 2018-11-16 | ST | 2018-11-16 | 1.000000 | 3.140 | 0.0000 | 156.15 | 490.31100 | 0.005905 | -0.005905 | 0.0 | 1 | -3.14 | 0.000 | 0.00000 | -490.311 | 0.000000 |
| **1** | VBTLX | 2018-11-16 | BD | 2018-11-16 | 1.000000 | 1685.890 | 0.0350 | 10.39 | 17516.39710 | 0.210973 | -0.175973 | 350.0 | 1 | -1372.00 | 313.890 | 3261.31710 | -14255.080 | 0.035196 |
| **2** | VTIAX | 2018-11-16 | ST | 2018-11-16 | 1.000000 | 514.298 | 0.1521 | 25.17 | 12944.88066 | 0.155913 | -0.003813 | 1521.0 | 1 | 47.00 | 561.298 | 14127.87066 | 1182.990 | 0.152468 |
| **3** | VTSAX | 2018-11-16 | ST | 2018-11-16 | 11.777895 | 852.570 | 0.5652 | 61.08 | 52074.97560 | 0.627209 | -0.062009 | 5652.0 | 1 | 8.00 | 860.570 | 52563.61560 | 488.640 | 0.567266 |
| **4** | VIGAX | NaT | ST | 2018-12-27 | 68.020000 | 0.000 | 0.0131 | 68.02 | 0.00000 | 0.000000 | 0.013100 | 131.0 | 1 | 17.00 | 17.000 | 1156.34000 | 1156.340 | 0.012479 |
| **5** | VSMAX | NaT | ST | 2018-12-27 | 62.250000 | 0.000 | 0.0066 | 62.25 | 0.00000 | 0.000000 | 0.006600 | 66.0 | 1 | 9.00 | 9.000 | 560.25000 | 560.250 | 0.006046 |
| **6** | VSEQX | NaT | ST | 2018-12-27 | 26.770000 | 0.000 | 0.0066 | 26.77 | 0.00000 | 0.000000 | 0.006600 | 66.0 | 1 | 22.00 | 22.000 | 588.94000 | 588.940 | 0.006356 |
| **7** | VWIGX | NaT | ST | 2018-12-27 | 24.760000 | 0.000 | 0.0507 | 24.76 | 0.00000 | 0.000000 | 0.050700 | 507.0 | 1 | 190.00 | 190.000 | 4704.40000 | 4704.400 | 0.050770 |
| **8** | VTRIX | NaT | ST | 2018-12-27 | 31.910000 | 0.000 | 0.0507 | 31.91 | 0.00000 | 0.000000 | 0.050700 | 507.0 | 1 | 147.00 | 147.000 | 4690.77000 | 4690.770 | 0.050623 |
| **9** | VTABX | NaT | BD | 2018-12-27 | 21.660000 | 0.000 | 0.0150 | 21.66 | 0.00000 | 0.000000 | 0.015000 | 150.0 | 1 | 64.00 | 64.000 | 1386.24000 | 1386.240 | 0.014960 |
| **10** | VGSLX | NaT | RE | 2018-12-27 | 105.060000 | 0.000 | 0.0500 | 105.06 | 0.00000 | 0.000000 | 0.050000 | 500.0 | 1 | 44.00 | 44.000 | 4622.64000 | 4622.640 | 0.049887 |
| **11** | VNQI | NaT | RE | 2018-12-27 | 52.050000 | 0.000 | 0.0100 | 52.05 | 0.00000 | 0.000000 | 0.010000 | 100.0 | 1 | 17.00 | 17.000 | 884.85000 | 884.850 | 0.009549 |
| **12** | VDE | NaT | ST | 2018-12-27 | 77.440000 | 0.000 | 0.0300 | 77.44 | 0.00000 | 0.000000 | 0.030000 | 300.0 | 1 | 36.00 | 36.000 | 2787.84000 | 2787.840 | 0.030086 |
| **13** | GLD | NaT | CS | 2018-12-27 | 120.570000 | 0.000 | 0.0150 | 120.57 | 0.00000 | 0.000000 | 0.015000 | 150.0 | 1 | 11.00 | 11.000 | 1326.27000 | 1326.270 | 0.014313 |

In [16]:

*#Lets add a bar chart here to show the new allocation vs. the target allocation and vs. the original portfolio*

graph\_port **=** exec\_port[['ticker','allocation','allocation\_target','final\_allocation']]**.**copy()

graph\_port**.**plot**.**barh(x**=**'ticker',figsize**=**(20,10))

Out[16]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x12561aa20>

Shape

Description automatically generated

In [17]:

*#Next we turn our ticker-level strategy into account level actions*

*#Join in our rebalanced portfolio and determine how to split value across accounts for a given ticker*

port **=** pd**.**merge(start\_port[['accounttype','accountid','ticker','shares']],

exec\_port[['ticker','assetclass','close','value','final\_shares\_chg','new\_shares','new\_value','new\_value\_chg','final\_allocation']],

how **=** 'right',

left\_on **=** 'ticker',

right\_on **=** 'ticker')

port['value\_orig'] **=** port**.**close **\*** port**.**shares

*#Calculate the value-weight of each ticker by account*

port['tick\_alloc'] **=** port**.**value\_orig **/** port**.**value *#What pct of each ticker is in a given account?*

port['tick\_alloc']**.**fillna(1.0,inplace**=True**)

*#check our sub-allocations*

**assert**(port**.**groupby('ticker')**.**tick\_alloc**.**sum()**.**sum() **==** len(port**.**groupby('ticker')**.**tick\_alloc**.**sum()))

*#Recalculate the values proportionately*

port['final\_shares\_chg\_n'] **=** port**.**final\_shares\_chg **\*** port**.**tick\_alloc

port['new\_shares\_n'] **=** port**.**new\_shares **\*** port**.**tick\_alloc

port['new\_value\_n'] **=** port**.**new\_value **\*** port**.**tick\_alloc

port['new\_value\_chg\_n'] **=** port**.**new\_value\_chg **\*** port**.**tick\_alloc

port['final\_allocation\_n'] **=** port**.**final\_allocation **\*** port**.**tick\_alloc

*#double check our final\_allocation is 100%*

**assert**(np**.**round(port**.**final\_allocation\_n**.**sum(),4)**==**1.0)

*#Now we must double check to ensure we are not allocating buys to accounts with no sells (we cannot just add funds to a Traditional IRA account, for example)*

*#accounts with single securities in them which also exist in other accounts can cause issues if we don't do this*

acctsdf **=** port**.**groupby(['accountid','accounttype'])**.**new\_value\_chg\_n**.**sum()

acctsdf **=** acctsdf**.**reset\_index()**.**rename(columns**=**{'new\_value\_chg\_n':'new\_value\_chg\_sum'})

errordf **=** acctsdf[acctsdf**.**new\_value\_chg\_sum **>** 0]**.**copy() *#a value >0 at the account-level implies we have allocated buys to an account with insufficient sells*

erroraccts **=** errordf**.**accountid**.**values

**if** len(errordf) **>** 0:

**for** t **in** port[port**.**accountid**.**isin(erroraccts)]**.**ticker**.**unique(): *#Loop by security (not by account)*

print("Correcting distribution for single-security accounts edge case: **{}**"**.**format(t))

index **=** (port**.**accountid**.**isin(erroraccts)) **&** (port**.**ticker **==** t)

display(port[port**.**ticker **==** t])

*#adjust numerator and denominator for proper recalculation of asset distribution across accounts*

port**.**loc[index,'new\_shares\_n'] **=** port**.**new\_shares\_n **-** port**.**final\_shares\_chg\_n

port**.**loc[index,'new\_value\_n'] **=** port**.**new\_value\_n **-** port**.**new\_value\_chg\_n

port**.**loc[index,'final\_shares\_chg\_n'] **=** 0

port**.**loc[index,'new\_value\_chg\_n'] **=** 0

*#remove from denominator*

port**.**loc[port**.**ticker **==** t,'value'] **=** port**.**loc[port**.**ticker **==** t,'value'] **-** port[index]**.**value\_orig**.**sum()

*#recalculate values for this ticker*

port**.**loc[port**.**ticker **==** t,'tick\_alloc'] **=** port[port**.**ticker **==** t]**.**value\_orig **/** port[port**.**ticker **==** t]**.**value

port**.**loc[index,'tick\_alloc'] **=** 0 *#set new money allocation to zero for funds with insufficient assets*

port**.**loc[port**.**ticker **==** t,'final\_shares\_chg\_n'] **=** port**.**final\_shares\_chg **\*** port**.**tick\_alloc

port**.**loc[port**.**ticker **==** t,'new\_shares\_n'] **=** port**.**shares **+** port**.**final\_shares\_chg\_n

port**.**loc[port**.**ticker **==** t,'new\_value\_chg\_n'] **=** port**.**new\_value\_chg **\*** port**.**tick\_alloc

port**.**loc[port**.**ticker **==** t,'new\_value\_n'] **=** port**.**value\_orig **+** port**.**new\_value\_chg\_n

port**.**loc[port**.**ticker **==** t,'final\_allocation\_n'] **=** (port**.**new\_value\_n **/** port**.**new\_value) **\*** port**.**final\_allocation

display(port[port**.**ticker **==** t])

*#Cleanup*

port['value'] **=** port**.**value\_orig

port['final\_shares\_chg'] **=** port**.**final\_shares\_chg\_n

port['new\_shares'] **=** port**.**new\_shares\_n

port['new\_value'] **=** port**.**new\_value\_n

port['new\_value\_chg'] **=** port**.**new\_value\_chg\_n

port['final\_allocation'] **=** port**.**final\_allocation\_n

port**.**drop(['value\_orig','tick\_alloc','final\_shares\_chg\_n','new\_shares\_n','new\_value\_n','new\_value\_chg\_n','final\_allocation\_n'],axis**=**1,inplace**=True**)

port**.**fillna({'value':0.0},inplace**=True**)

*#Check our work*

**assert**(np**.**round(port**.**final\_allocation**.**sum(),4)**==**1.0)

**assert**(np**.**round(np**.**sum((port**.**shares**+**port**.**final\_shares\_chg)**-**port**.**new\_shares))**==**0)

**assert**(np**.**round(np**.**sum(port**.**new\_value**-**(port**.**new\_shares**\***port**.**close)))**==**0)

**assert**(np**.**round(np**.**sum(port**.**new\_value\_chg**-**(port**.**final\_shares\_chg**\***port**.**close)))**==**0)

*#Lets look at our final portfolio at the account level*

display(port)

|  | **accounttype** | **accountid** | **ticker** | **shares** | **assetclass** | **close** | **value** | **final\_shares\_chg** | **new\_shares** | **new\_value** | **new\_value\_chg** | **final\_allocation** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | RIRA | 1111 | VBTLX | 913.483 | BD | 10.39 | 9491.08837 | -743.404775 | 170.078225 | 1767.112759 | -7723.975611 | 0.019071 |
| **1** | RIRA | 2222 | VBTLX | 772.407 | BD | 10.39 | 8025.30873 | -628.595225 | 143.811775 | 1494.204341 | -6531.104389 | 0.016125 |
| **2** | RIRA | 1111 | VTIAX | 514.298 | ST | 25.17 | 12944.88066 | 47.000000 | 561.298000 | 14127.870660 | 1182.990000 | 0.152468 |
| **3** | RIRA | 1111 | VTSAX | 151.121 | ST | 61.08 | 9230.47068 | 1.418028 | 152.539028 | 9317.083821 | 86.613141 | 0.100550 |
| **4** | RIRA | 2222 | VTSAX | 151.578 | ST | 61.08 | 9258.38424 | 1.422316 | 153.000316 | 9345.259305 | 86.875065 | 0.100854 |
| **5** | TAXB | 3333 | VTSAX | 549.871 | ST | 61.08 | 33586.12068 | 5.159656 | 555.030656 | 33901.272475 | 315.151795 | 0.365862 |
| **6** | TAXB | 3333 | AAPL | 3.140 | ST | 156.15 | 490.31100 | -3.140000 | 0.000000 | 0.000000 | -490.311000 | 0.000000 |
| **7** | NaN | NaN | VIGAX | NaN | ST | 68.02 | 0.00000 | 17.000000 | 17.000000 | 1156.340000 | 1156.340000 | 0.012479 |
| **8** | NaN | NaN | VSMAX | NaN | ST | 62.25 | 0.00000 | 9.000000 | 9.000000 | 560.250000 | 560.250000 | 0.006046 |
| **9** | NaN | NaN | VSEQX | NaN | ST | 26.77 | 0.00000 | 22.000000 | 22.000000 | 588.940000 | 588.940000 | 0.006356 |
| **10** | NaN | NaN | VWIGX | NaN | ST | 24.76 | 0.00000 | 190.000000 | 190.000000 | 4704.400000 | 4704.400000 | 0.050770 |
| **11** | NaN | NaN | VTRIX | NaN | ST | 31.91 | 0.00000 | 147.000000 | 147.000000 | 4690.770000 | 4690.770000 | 0.050623 |
| **12** | NaN | NaN | VTABX | NaN | BD | 21.66 | 0.00000 | 64.000000 | 64.000000 | 1386.240000 | 1386.240000 | 0.014960 |
| **13** | NaN | NaN | VGSLX | NaN | RE | 105.06 | 0.00000 | 44.000000 | 44.000000 | 4622.640000 | 4622.640000 | 0.049887 |
| **14** | NaN | NaN | VNQI | NaN | RE | 52.05 | 0.00000 | 17.000000 | 17.000000 | 884.850000 | 884.850000 | 0.009549 |
| **15** | NaN | NaN | VDE | NaN | ST | 77.44 | 0.00000 | 36.000000 | 36.000000 | 2787.840000 | 2787.840000 | 0.030086 |
| **16** | NaN | NaN | GLD | NaN | CS | 120.57 | 0.00000 | 11.000000 | 11.000000 | 1326.270000 | 1326.270000 | 0.014313 |

In [18]:

*#Finally, all new tickers need an account to land in*

dport **=** **None**

acctsdf **=** **None**

**if** len(port[port**.**accounttype**.**isnull()])**>**0: *#if we have none, skip this step*

print('Distributing new securities to existing accounts . . .')

dport **=** port**.**copy()

*#account-level fund surplus or deficit - must match these with our orphaned securities*

acctsdf **=** port**.**groupby(['accountid','accounttype'])**.**new\_value\_chg**.**sum()

acctsdf **=** acctsdf**.**reset\_index()**.**rename(columns**=**{'new\_value\_chg':'new\_value\_chg\_sum'})

*#establish sort order so we can allocate tax-efficient account space first*

actype\_sortorder **=** pd**.**DataFrame(data**=**[['RIRA',1],['TIRA',2],['TAXB',3]],columns**=**['accounttype','order'])

acctsdf **=** pd**.**merge(acctsdf,actype\_sortorder,how**=**'left',left\_on**=**'accounttype',right\_on**=**'accounttype')

*#We make a consequential assumption here that any new\_money\_in will be allocated 100% in one of the Taxable accounts (first in list).*

*#if you have a Roth-IRA which has not met its contribution limits for the year, it may be preferrential to distribute the funds there first.*

*#IF YOU HAVE NO TAXABLE ACCOUNT AND YOU WISH TO REBALANCE WITH new\_money\_in > 0 this will cause errors - so we assert here:*

**assert**(new\_money\_in **==** 0 **or** (len(acctsdf[acctsdf**.**accounttype **==** 'TAXB'])**>**0 **and** new\_money\_in **>** 0))

min\_idx **=** acctsdf[acctsdf**.**accounttype **==** 'TAXB']**.**index**.**min()

acctsdf**.**loc[min\_idx,'new\_value\_chg\_sum'] **=** acctsdf**.**loc[min\_idx,'new\_value\_chg\_sum'] **-** new\_money\_in

*#only return accounts that have space*

acctsdf **=** acctsdf[acctsdf**.**new\_value\_chg\_sum**<**0]**.**copy()

*#establish sort order so we can allocate tax-inefficient assets first*

aclass\_sortorder **=** pd**.**DataFrame(data**=**[['ST',3],['BD',1],['CS',4],['RE',2],['ALT',5]],columns**=**['assetclass','order'])

dport **=** pd**.**merge(dport,aclass\_sortorder,how**=**'left',left\_on**=**'assetclass',right\_on**=**'assetclass')

*#We loop twice, first to fit whole securities in accounts with tax location in mind, then again without tax location for anything leftover*

loop **=** 0

**while** loop **<** 2:

loop**+=**1

*#loop through orphaned tickers and place them in accounts until all assets are allocated or we are forced to split a security across accounts*

*# in the first loop we do not allow tax-inefficient assets to wind up in Taxable accounts, in the second loop we relax this constraint*

**for** index, row **in** dport[dport**.**accounttype**.**isnull()]**.**sort\_values(['order','new\_value\_chg'],ascending**=**[**True**,**False**])**.**iterrows():

*#loop through accounts and place the assets*

**for** i, r **in** acctsdf**.**iterrows():

aid **=** r**.**accountid

atype **=** r**.**accounttype

bal **=** r**.**new\_value\_chg\_sum

*#print('Evaluating {}-{} with {} starting bal'.format(aid,atype,bal))*

**if** loop **==** 0 **and** (row**.**assetclass **in** ('BD','RE') **and** atype **==** 'TAXB'):

**continue** *#skip this case, since we don't want to place Bonds and Real-Estate assets in Taxable accounts*

**elif** loop **==** 0 **and** (row**.**assetclass **not** **in** ('BD','RE') **and** atype **!=** 'TAXB'):

**continue** *#skip this case, since we don't want to place tax-efficient assets into tax sheltered accounts*

**if** row**.**new\_value\_chg **+** bal **<=**0: *#it fits*

bal**+=**row**.**new\_value\_chg

print(' FITS **{}** in **{}**-**{}** with **{}** remaining'**.**format(row**.**ticker,aid,atype,bal))

*#update our portfolio*

dport**.**loc[index,'accountid'] **=** aid

dport**.**loc[index,'accounttype'] **=** atype

*#update account bal for next loop*

acctsdf**.**loc[i,'new\_value\_chg\_sum'] **=** bal

**break**

**else**:

print(' **{}** **{}** does not fit in **{}**-**{}**'**.**format(row**.**ticker,row**.**new\_value\_chg,aid,atype))

print('**\n**Lets see what remains in our accounts after 2 loops . . .')

display(acctsdf)

*#Here we are forced to split a security across multiple accounts because no one account can fit it*

*# in this loop we allow tax-inefficient assets to wind up in Taxable accounts, but only as a last resort*

**if** len(dport[dport**.**accounttype**.**isnull()])**>**0:

print('Splitting remaining securities across accounts . . .')

*#loop through accounts and place portions of asset in each, create a new row in the df for each placement.*

**for** index, row **in** dport[dport**.**accounttype**.**isnull()]**.**sort\_values(['order','new\_value\_chg'],ascending**=**[**True**,**False**])**.**iterrows():

final\_shares\_chg **=** row**.**final\_shares\_chg

asset\_bal **=** row**.**new\_value\_chg

*#if its a tax-inefficent asset, order the accounts by 'order'*

**if** row**.**assetclass **in** ('BD','RE'):

acctsdf **=** acctsdf**.**sort\_values('order',ascending**=True**)

**else**:

acctsdf **=** acctsdf**.**sort\_values('order',ascending**=False**)

**for** i, r **in** acctsdf**.**iterrows():

bal **=** r**.**new\_value\_chg\_sum

**if** asset\_bal**>-**bal:

to\_move **=** **-**bal

pct\_move **=** **-**bal**/**row**.**new\_value\_chg

asset\_bal**+=**bal

**else**:

to\_move **=** asset\_bal

pct\_move **=** asset\_bal**/**row**.**new\_value\_chg

asset\_bal**=**0

print(' **{}** move **{}** or **{}% i**nto account **{}**-**{}**. **{}** bal remaining **{}**'**.**format(row**.**ticker,to\_move,pct\_move,r**.**accountid,r**.**accounttype,row**.**ticker,asset\_bal))

*#update our account to reflect this change*

**if** asset\_bal **>** 0:

acctsdf**.**loc[i,'new\_value\_chg\_sum'] **=** 0.0

**else**:

acctsdf**.**loc[i,'new\_value\_chg\_sum'] **=** to\_move**+**bal

**if** (np**.**floor(pct\_move**\***row**.**new\_shares)**\***row**.**close)**-**row**.**value **>** 0:

*#create new row in our portfolio for this asset in this account*

dport**.**loc[max(dport**.**index)**+**1] **=** [r**.**accounttype,

r**.**accountid,

row**.**ticker,

row**.**shares,

row**.**assetclass,

row**.**close,

row**.**value,

np**.**floor(pct\_move**\***row**.**final\_shares\_chg), *#we round down to get back to whole shares*

np**.**floor(pct\_move**\***row**.**new\_shares),

np**.**floor(pct\_move**\***row**.**new\_shares)**\***row**.**close,

(np**.**floor(pct\_move**\***row**.**new\_shares)**\***row**.**close)**-**row**.**value, *#rounding can cause us to be short of our total allocatable funds*

np**.**floor(pct\_move**\***row**.**new\_value)**/**dport**.**new\_value**.**sum(),

row**.**order]

*#finally delete the original row from the df*

dport**.**drop(dport[dport**.**accounttype**.**isnull()]**.**index,inplace**=True**)

*#double check our work - we just care that distributed funds < total available funds for this ticker*

**assert**(dport[dport**.**ticker**==**row**.**ticker]**.**new\_value\_chg**.**sum() **<** row**.**new\_value\_chg)

Distributing new securities to existing accounts . . .

FITS VTABX in 1111-RIRA with -5068.132470695083 remaining

FITS VGSLX in 1111-RIRA with -445.4924706950824 remaining

VNQI 884.8499999999999 does not fit in 1111-RIRA

FITS VNQI in 2222-RIRA with -5559.379323808596 remaining

VWIGX 4704.400000000001 does not fit in 1111-RIRA

FITS VWIGX in 2222-RIRA with -854.9793238085958 remaining

VTRIX 4690.77 does not fit in 1111-RIRA

VTRIX 4690.77 does not fit in 2222-RIRA

FITS VTRIX in 3333-TAXB with -5484.389205496322 remaining

VDE 2787.84 does not fit in 1111-RIRA

VDE 2787.84 does not fit in 2222-RIRA

FITS VDE in 3333-TAXB with -2696.5492054963215 remaining

VIGAX 1156.34 does not fit in 1111-RIRA

VIGAX 1156.34 does not fit in 2222-RIRA

FITS VIGAX in 3333-TAXB with -1540.2092054963216 remaining

VSEQX 588.9399999999999 does not fit in 1111-RIRA

FITS VSEQX in 2222-RIRA with -266.0393238085959 remaining

VSMAX 560.25 does not fit in 1111-RIRA

VSMAX 560.25 does not fit in 2222-RIRA

FITS VSMAX in 3333-TAXB with -979.9592054963216 remaining

GLD 1326.27 does not fit in 1111-RIRA

GLD 1326.27 does not fit in 2222-RIRA

GLD 1326.27 does not fit in 3333-TAXB

GLD 1326.27 does not fit in 1111-RIRA

GLD 1326.27 does not fit in 2222-RIRA

GLD 1326.27 does not fit in 3333-TAXB

Lets see what remains in our accounts after 2 loops . . .

|  | **accountid** | **accounttype** | **new\_value\_chg\_sum** | **order** |
| --- | --- | --- | --- | --- |
| **0** | 1111 | RIRA | -445.492471 | 1 |
| **1** | 2222 | RIRA | -266.039324 | 1 |
| **2** | 3333 | TAXB | -979.959205 | 3 |

Splitting remaining securities across accounts . . .

GLD move 979.9592054963216 or 0.7388836402062338% into account 3333-TAXB. GLD bal remaining 346.3107945036784

GLD move 346.3107945036784 or 0.26111635979376624% into account 1111-RIRA. GLD bal remaining 0

GLD move 0 or 0.0% into account 2222-RIRA. GLD bal remaining 0

In [19]:

*#Lets see whats left in our accounts, it should be very close to zero*

**if** acctsdf **is** **not** **None**:

display(acctsdf)

|  | **accountid** | **accounttype** | **new\_value\_chg\_sum** | **order** |
| --- | --- | --- | --- | --- |
| **2** | 3333 | TAXB | 0.000000 | 3 |
| **0** | 1111 | RIRA | -99.181676 | 1 |
| **1** | 2222 | RIRA | -266.039324 | 1 |

In [20]:

*#Review our final portfolio with recommended buys/sells in 'final\_shares\_chg' column*

**if** dport **is** **not** **None**:

*#Cleanup*

dport**.**drop(columns**=**['order'],inplace**=True**)

dport **=** dport[['accounttype','accountid','ticker','shares','assetclass','close','value','new\_shares','final\_shares\_chg','new\_value','new\_value\_chg','final\_allocation']]

display(dport)

**else**:

port **=** port[['accounttype','accountid','ticker','shares','assetclass','close','value','new\_shares','final\_shares\_chg','new\_value','new\_value\_chg','final\_allocation']]

display(port)

|  | **accounttype** | **accountid** | **ticker** | **shares** | **assetclass** | **close** | **value** | **new\_shares** | **final\_shares\_chg** | **new\_value** | **new\_value\_chg** | **final\_allocation** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | RIRA | 1111 | VBTLX | 913.483 | BD | 10.39 | 9491.08837 | 170.078225 | -743.404775 | 1767.112759 | -7723.975611 | 0.019071 |
| **1** | RIRA | 2222 | VBTLX | 772.407 | BD | 10.39 | 8025.30873 | 143.811775 | -628.595225 | 1494.204341 | -6531.104389 | 0.016125 |
| **2** | RIRA | 1111 | VTIAX | 514.298 | ST | 25.17 | 12944.88066 | 561.298000 | 47.000000 | 14127.870660 | 1182.990000 | 0.152468 |
| **3** | RIRA | 1111 | VTSAX | 151.121 | ST | 61.08 | 9230.47068 | 152.539028 | 1.418028 | 9317.083821 | 86.613141 | 0.100550 |
| **4** | RIRA | 2222 | VTSAX | 151.578 | ST | 61.08 | 9258.38424 | 153.000316 | 1.422316 | 9345.259305 | 86.875065 | 0.100854 |
| **5** | TAXB | 3333 | VTSAX | 549.871 | ST | 61.08 | 33586.12068 | 555.030656 | 5.159656 | 33901.272475 | 315.151795 | 0.365862 |
| **6** | TAXB | 3333 | AAPL | 3.140 | ST | 156.15 | 490.31100 | 0.000000 | -3.140000 | 0.000000 | -490.311000 | 0.000000 |
| **7** | TAXB | 3333 | VIGAX | NaN | ST | 68.02 | 0.00000 | 17.000000 | 17.000000 | 1156.340000 | 1156.340000 | 0.012479 |
| **8** | TAXB | 3333 | VSMAX | NaN | ST | 62.25 | 0.00000 | 9.000000 | 9.000000 | 560.250000 | 560.250000 | 0.006046 |
| **9** | RIRA | 2222 | VSEQX | NaN | ST | 26.77 | 0.00000 | 22.000000 | 22.000000 | 588.940000 | 588.940000 | 0.006356 |
| **10** | RIRA | 2222 | VWIGX | NaN | ST | 24.76 | 0.00000 | 190.000000 | 190.000000 | 4704.400000 | 4704.400000 | 0.050770 |
| **11** | TAXB | 3333 | VTRIX | NaN | ST | 31.91 | 0.00000 | 147.000000 | 147.000000 | 4690.770000 | 4690.770000 | 0.050623 |
| **12** | RIRA | 1111 | VTABX | NaN | BD | 21.66 | 0.00000 | 64.000000 | 64.000000 | 1386.240000 | 1386.240000 | 0.014960 |
| **13** | RIRA | 1111 | VGSLX | NaN | RE | 105.06 | 0.00000 | 44.000000 | 44.000000 | 4622.640000 | 4622.640000 | 0.049887 |
| **14** | RIRA | 2222 | VNQI | NaN | RE | 52.05 | 0.00000 | 17.000000 | 17.000000 | 884.850000 | 884.850000 | 0.009549 |
| **15** | TAXB | 3333 | VDE | NaN | ST | 77.44 | 0.00000 | 36.000000 | 36.000000 | 2787.840000 | 2787.840000 | 0.030086 |
| **17** | TAXB | 3333 | GLD | NaN | CS | 120.57 | 0.00000 | 8.000000 | 8.000000 | 964.560000 | 964.560000 | 0.010565 |
| **18** | RIRA | 1111 | GLD | NaN | CS | 120.57 | 0.00000 | 2.000000 | 2.000000 | 241.140000 | 241.140000 | 0.003749 |

Conclusion

Through this portfolio rebalancing demonstration using Pandas you can see we have achieved a rebalanced portfolio very closely approximating our desired target allocation. We can see how GLD was added as a new security along with 9 others, and AAPL was removed from the portfolio. The remaining securities were bought or sold as required by our target allocation. We accounted for whole-share rounding because most of our assets in this sample are index funds. The final steps had us consider how to distribute newly acquired securities into existing accounts, this was accomplished through iteratively fitting securities into accounts until they all had a home. This simple rebalancer can be adapted to your needs and I urge you grab the code and see if you can improve upon it. I welcome your thoughts or feedback in the comments.

***No Solicitation or Investment Advice:*** The material contained on this website is for informational purposes only and the author is not soliciting any action based upon such material. The material is not to be construed as an offer or a recommendation to buy or sell a security nor is it to be construed as investment advice. Additionally, the material accessible through this website does not constitute a representation that the investments described herein are suitable or appropriate for any person. This code is provided as-is with no warranties and may contain errors which affect its performance.

The Elusive Rebalancing Bonus – Part 1

August 8, 2020 by [Siamond](https://www.bogleheads.org/blog/author/siamondblog/" \o "View all posts by Siamond)

Passive investors typically rebalance their accounts every now and then to get the various positions back in sync with their target asset allocation. This multi-part article studies various rebalancing methods and analyzes if some of those methods provide(d) a significant rebalancing bonus. Part 1 will focus on basics, describing typical rebalancing methods and illustrating them with some historical trajectories. The next parts of the study will explore rebalancing in more quantitative details.

What is rebalancing?

A passive investor articulating an investment plan will typically come up with a target asset allocation (AA), say something like 50% US equities, 20% international equities and 30% US bonds. Such target AA might stay fixed year over year or possibly follow some sort of glide path (say 1% more bonds every year), but the fact is, every year, there is a target to aim at.

In the mean time, the stock market tends to follow a rather erratic and unpredictable trajectory and if one’s portfolio is perfectly aligned with a given target AA at the beginning of the year, the current positions after a few months or a year will drift away from the intended target.

In the illustration below, the target AA is 25% of four assets (say US stocks, Int’l stocks, Bonds, Gold). After some period of time (say a couple of quarters), the current asset allocation starts shifting (say gold suddenly picks up in value while Int’l stocks crater for whatever reason). Rebalancing would then be the fact of exchanging shares (e.g. sell/buy) between the four asset classes to come back to the plan of record (i.e. the target AA). **Rebalancing methods (as discussed below) regulate WHEN to decide to rebalance.**

All-in-one funds (e.g. Target-Date or LifeStrategy kind of funds) automatically perform rebalancing on behalf of their investors, but passive investors choosing to use more than one fund (e.g. a three-funds portfolio, or more) will have to rebalance by themselves.

[Chart, pie chart

Description automatically generated](https://www.bogleheads.org/blog/wp-content/uploads/2020/07/Rebalancing-Basics.jpg)

Note that rebalancing is typically something to perform across one’s entire portfolio, across all investment accounts (taxable and tax-deferred). It is certainly easier to act on tax-deferred accounts where exchanges have no tax impact (nor transaction costs), while more care should be exercised with taxable accounts, to avoid triggering capital gains and other cost issues. This article will not cover tax implications, nor transaction costs.

The baseline method: annual rebalancing

The simplest rebalancing method is to pick a fixed date in the year (early January, some sort of anniversary, whatever) and rebalance one’s portfolio at this date, irrespective of the current state of the portfolio. Many backtesting tools make such implicit assumption, e.g. a rebalancing towards the target AA on January 1st. Here is an illustration of what would have happened to the current AA over a period of 20 years. Click on the image below to see a bigger version.

[A picture containing text, white

Description automatically generated](https://www.bogleheads.org/blog/wp-content/uploads/2020/07/Rebalancing-1995-20-Annual-52w.jpg)

Many graphs in this article will use the same format, so let’s unpack it:

* Such graph shows what happens to a portfolio made of a given initial investment over a time period of 20 years. No addition (contribution) or withdrawal is performed during this period of time, to simplify.
  + Most graphs below will use the same time period (1995-2014) as this is a particularly eventful period.
* The initial (and fixed target) AA is 50% US stocks (blue line), 20% Int’l stocks (green line) and 30% US bonds (purple line).
* The lines wiggle over time as the current AA of the portfolio evolves with the vagaries of the stock market.
* Rebalancing events occur every now and then and are shown with the red triangle mark labeled with the corresponding date. The asset allocation is then reset to its target.
* The underlying data set is a set of weekly returns (total returns, dividends included) for each asset class. The author’s spreadsheet computes on a weekly basis if rebalancing is warranted or not.
  + In this precise case, rebalancing events occur every 52 weeks, to approximate annual rebalancing.
  + Note that 52 weeks is 364 days, not exactly a full year, so there is a little bit of calendar drift over time as the rebalancing labels show.
  + Rebalancing labels show Mondays, but actual rebalancing would obviously happen on the first tradable day of the week.
  + The 15-Dec-08 rebalancing event is the most dramatic event of the graph, due to the financial crisis unfolding in Q4-08.
* The thin orange line near the bottom illustrates the S&P 500 price trajectory. This helps to appreciate when large movements of the stock market occurred (e.g. big bull market run-up; big stock market crises).

To illustrate how things can be different over time, the following chart displays annual rebalancing between 1980 and 1999 (essentially one long bull market with a few hiccups along the way, notably in Oct-87).

[Chart

Description automatically generated](https://www.bogleheads.org/blog/wp-content/uploads/2020/07/Rebalancing-1980-20-Annual-52w.jpg)

Such annual periodic rebalancing method has the great virtue of simplicity and is used by many investors. We will use its properties as the ‘baseline method’ when analyzing more complex methods.

Rebalancing triggers

A common criticism of annual (or periodic) rebalancing is that a fixed rebalancing date is very arbitrary and doesn’t account for those ‘special moments’ in time where markets go haywire, leading to sudden sizable changes in one’s portfolio.

For example, if the stock market drops 30% in just a few weeks (as happened in March 2020), a cold blooded investor might perceive that stock suddenly got much ‘cheaper’ (relatively speaking) and might wish to use its ‘excess’ bonds to quickly buy more stocks while they are ‘cheap’. Fact is rebalancing can be perceived as ‘selling high and buying low’, by virtue of selling shares in positions which seem more highly valued than others (the shares one would buy when rebalancing). Let’s take an example.

Let’s say we start with $1000 with a 60/40 asset allocation, therefore $600 in stocks and $400 in bonds. Let’s say stocks drop by 33% while bonds stay stable. The portfolio is now $402 in stocks and $400 in bonds, essentially a current AA of 50/50. One could be tempted to use the 10% ‘excess’ bonds to restore the 60/40 target allocation, while perceiving that those good stable bonds are used as ‘dry powder’ to buy stocks on a fire sale, hence selling bonds ‘high’ to buy stocks ‘low’. Please note that the author is NOT stating that such perception is correct, just that it speaks strongly to one’s intuition.

Subsequently, one might be tempted to define rebalancing triggers (aka rebalancing bands), i.e. dynamic rules allowing to decide when to rebalance in an event-driven manner instead of a calendar manner. This implies to monitor one’s current AA very frequently (say weekly or monthly), but simple scripts (say with Google Sheet) can easily automate such process. The two main types of triggers are fixed bands and relative bands (which are sometimes combined), a less commonly used type is called adaptive bands. Let’s explore those variants.

Rebalancing with fixed (absolute) bands

Let’s keep exploring the 50/20/30 asset allocation from the previous graph. One could decide to rebalance when one of the asset classes drifts by an absolute 5% or more from their target (i.e. if US stocks go above 55% or below 45%; or if Int’l stock go above 25% or below 15%; of if US bonds go above 35% or below 25%). Assuming a weekly monitoring process of sorts is in place, here is what would have happened.

Chart

Description automatically generated

Note that such ‘fixed band’ approach keeps the virtue of simplicity, but applying the same 5% absolute variation to a 50% (target) position or a 20% (target) position seems rather inconsistent. Still, such approach is certainly a good way to be more reactive to ongoing events, whether they follow quite dramatic changes (e.g. end of 2008) or slower cumulative changes (e.g. big bull market by the end of the 90s or the slowly-unfolding Internet crisis which followed).

Let’s look more carefully at how the Internet and the Financial crises unfolded. The following chart hones on the 2000 to 2011 time period and the scale of the S&P 500 price (thin orange line) is expanded to show the events. The Financial crisis in 08/09 is the most glaring with four rebalancing events in quick succession (the first two just a few weeks apart!), certainly frazzling the nerves of most investors. It then took nearly two years for rebalancing in the reverse direction. Note that rebalancing triggers just perform simple percentage math, which led (in hindsight) to rather poor market timing calls in Q4-08 (while the rebalancing events in 2009 were more fortunate). The Internet crisis unfolded much more slowly, but the same type of observations apply. **Rebalancing is just a discipline, there is NO market timing magic here.**

[Chart

Description automatically generated](https://www.bogleheads.org/blog/wp-content/uploads/2020/08/Growth-Chart-2000-2010-Fixed-5.jpg)

The fixed (absolute) bands approach suffers from clear inconsistencies though. Let’s say that we express the 50% US allocation as two 25% positions of the same asset class (or something very similar, e.g. total-market vs. S&P 500 index). Quite clearly, fixed bands will then not behave the same way and this makes little sense.

A sounder variation of fixed bands consists of assessing a **cumulative drift** from the target AA. Just sum up the positive differences (for each asset class) between the current AA and the target AA, then use such cumulative drift as the trigger to rebalance. This solves the previously described consistency issue (and other similar issues) while keeping the math trivial.

Rebalancing with relative bands

A variation on the previous idea is to use relative bands, say rebalancing when an asset class is more than 20%, relatively speaking, from its target. Using such logic on our 50/30/20 allocation:

* 20% of 50% is 10%, so one would rebalance if US stocks exceeds 60% of the current AA or is below 40%.
* 20% of 20% is 4%, so one would rebalance if Int’l stocks exceeds 24% of the current AA or is below 16%.
* 20% of 30% is 6%, so one would rebalance if US bonds exceeds 36% of the current AA or is below 24%.

Here is what would have happened when starting in 1995, a slightly different pattern than the fixed band approach, but not that dissimilar.

[Chart

Description automatically generated](https://www.bogleheads.org/blog/wp-content/uploads/2020/07/Rebalancing-1995-20-Relative-20.jpg)

This relative band approach is more consistent than the fixed band method, while keeping computations to a very simple formula. Still, for large positions (e.g. 50% of the portfolio), a relative band could end up being uncomfortably large. One could, of course, tune how reactive the algorithm is, by using narrower or wider bands. Here is the same chart using a 15% relative band.

[A picture containing text, white, document

Description automatically generated](https://www.bogleheads.org/blog/wp-content/uploads/2020/07/Rebalancing-1995-20-Relative-15.jpg)

Combining fixed and relative bands

A fairly common practice (popularized by Larry Swedroe) consists of combining fixed (absolute) and relative bands. A ‘5/25’ rebalancing algorithm would use fixed (absolute) 5% bands in combination with 25% relative bands. Fixed bands would be used for any asset class weighing 20% or more in the target AA and relative bands would be used for smaller positions. This is certainly useful when asset allocations use smaller positions (e.g. a 10% tilt towards something like Small-Cap-Value, Gold or REITs). If a position is only 10% of one’s portfolio, waiting for it to go to 15% or 5% (a fixed 5% difference) would take quite a rare chain of events…

As a side note, the 5/25 numbers are often touted as a typical recommendation, but as we could see from the previous sections, a 25% relative band is actually VERY wide (in other words, it could let the current AA drift away a lot from the target AA) and seems a rather odd choice.

With the 50/30/20 allocation we’ve been illustrating, only fixed bands would be activated with such combined algorithm, so there is no point providing a graph of such approach in this specific case. To better observe the dynamics, we can adjust a bit the asset allocation, carving out 10% for Small-Cap-Value while reducing US stocks to 45% and Int’l to 15% (just as an example), as illustrated below.

[Chart

Description automatically generated](https://www.bogleheads.org/blog/wp-content/uploads/2020/07/Rebalancing-1995-20-SW-5-20.jpg)

Looking carefully at the first three rebalancing events, we can see that the ones in Mar-96 and Sep-97 were triggered by the 5% fixed rule, while the next one (in Feb-99) was triggered by the relative rule (with a 20% band) on the smaller SCV position. This being said, if one would disable the relative rule, a rebalancing event would have occurred in Mar-99 based on solely using the fixed rule.

Rebalancing with adaptive bands

As was hinted at, fixed bands as well as relative bands have the virtue of simplicity, but anybody with a bit of a mathematical bent would quickly sense that those are rather inconsistent methods, which are ok for some simple asset allocations, but less so for other cases. As to the combined approach, this seems a rather unsatisfying kludge.

Such considerations led to in-depth discussions on the Bogleheads forum, from which emerged a more mathematically consistent approach dubbed ‘adaptive bands’, which can be implemented with a simple (albeit non intuitive) spreadsheet formula. The point of this blog is not to delve in such detailed discussions, but interested readers can start reading [here](https://www.bogleheads.org/forum/viewtopic.php?t=186203). In essence, a 20% adaptive band would be triggered by a 20% drop (or increase) of a given asset class in absence of changes to the other asset classes. Note that 20% is then an absolute number, which consistently applies irrespective of how large or small the asset class target is in the overall AA. It is also a truly symmetrical band, based on geometric math (while simple relative & fixed bands formulas suffer from asymmetry).

Here is an illustration of 20% adaptive bands with the same parameters as before, followed by the same graph using 30% adaptive bands (hence less frequent rebalancing events).

[A picture containing text, white

Description automatically generated](https://www.bogleheads.org/blog/wp-content/uploads/2020/07/Rebalancing-1995-20-Adaptive-20.jpg)[A picture containing text, map, white

Description automatically generated](https://www.bogleheads.org/blog/wp-content/uploads/2020/07/Rebalancing-1995-20-Adaptive-30.jpg)

In all fairness, although adaptive bands are more consistent than previous schemes and more directly applicable to any type of asset allocation, the primary net effect is probably a form of intellectual satisfaction for math-minded investors. For other investors, just picking fixed and/or relative bands that fit a specific portfolio would probably work better, if only because of the more intuitive nature of corresponding formulas. Such considerations are essentially behavioral, but this is important too. Matching one’s strategy with one’s skills and psychology is the best way to stay the course…

One-way rebalancing

In combination with some of the methods previously described, some people like to add another constraint which can be expressed as “only rebalance out of stocks, never into stocks”. The author assumes that the intent is to mitigate emotions along the lines of “don’t catch a falling knife” when a deep crisis is unfolding in real time and nobody knows (in foresight) the outcome. Avoiding to sell ‘safe’ bonds under such circumstances will probably provide emotional comfort to some.

This is all understandable at the ‘gut feeling’ level, but backtesting shows that really strange situations can occur. In the graph below, the same target AA as before is used, started again in 1995, using fairly narrow 15% relative rebalancing bands, but the time period was extended till mid-2020.

[Chart

Description automatically generated](https://www.bogleheads.org/blog/wp-content/uploads/2020/07/Rebalancing-1995-25-Relative-15-ONEWAY.jpg)

If you’re looking for rebalancing events between 1999 and 2017, well, there were none. Stocks dropped precipitously during the Internet crisis and just never recovered enough before the Financial crisis hit, i.e. not coming back where one could ‘rebalance out of stocks’ (i.e. US stocks exceeding 60%, the target AA). It took until 2017 (nearly 20 years!) for the current AA to catch up with the target AA for US stocks, finally triggering a rebalancing event (then another one, ironically enough right before the Covid-19 crisis started).

One might argue that you would only use one-way rebalancing in the depth of a crisis while not being shy to rebalance ‘when things are getting better’, but this is the kind of logic that is hard to put in a formula and relies more on intuition than anything else. And intuition might not be one’s best friend in troubled financial times.

More on data sources

This analysis is based on straightforward data sources, to avoid conclusions to be affected by other types of ‘noise’ (e.g. a fund or an index changing strategy mid-way, expense ratios varying greatly over time, etc).

In addition, to test rebalancing triggers, one needs data sources with extensive AND frequent/reliable history (e.g. weekly data, dividends included). As an acceptable approximation, monthly data is acceptable for some of the older years, notably for slow-changing bonds. To keep the weekly rhythm consistent across series (notably for trigger-based rebalancing methods), the author interpolated monthly returns to equivalent weekly returns (which should work reasonably well for bonds).

The author honed on the following indices as data sources. Note that most tests only used Total-US-Market (TSM), International Stocks and US Aggregate Bonds. With such data set, the author was able to study ~40 years of history (from 1980 to mid-2020). This is a marked change from known rebalancing studies which are often limited to time periods of 20 years or less (again, compare the first two charts to see how things can differ between two time periods). It is also a marked change from studies solely focusing on two asset classes (e.g. US stocks vs. bonds) where older data is available, but suffers from significant issues with granularity (e.g. due to history only recording quarterly dividends) and a certain lack of realism (most passive investors use 3 funds/asset-classes or more).

|  |  |
| --- | --- |
| Wilshire 5000 Total Market TR USD | TSM |
| Russell 2000 TR USD | SCB |
| Russell 2000 Value TR USD | SCV |
| MSCI EAFE GR USD | Int’l |
| FTSE Nareit All Equity REITs TR USD | REITs |
| LBMA Gold Price AM USD | Gold |
| Bloomberg Barclays US Aggregate Bond TR USD | Bonds |
| Bloomberg Barclays US Treasury Long TR USD | LTTs |

Using such indices has the great virtue of simplicity and consistency, and remains fairly realistic as modern passive index funds have quite a remarkable track record of very closely following their index benchmark. The absence of an expense ratio (ER) is not an issue since modern funds have a very low ER, plus most of the computations are made of relative comparisons between numbers (therefore a roughly identical -small- ER adjustment to all data series would not change the outcomes).

Still, to check that the tests weren’t somehow skewed by such index-centric approach, the author also used another data set made of real-life funds (see below). Some of the choices may seem odd, but it is quite difficult to find funds with enough history back in the 80s. Even with such long-lived funds, this led the author to have to approximate some of the missing early years (e.g. use small-caps in lieu of small-caps value) while accepting some historical disruptive changes (e.g. NAESX used to be an active fund before it became a passive index fund in 1990, etc). For gold, there was just no good solution the author could find besides the price index, but few tests used this series anyway.

|  |  |  |
| --- | --- | --- |
| Vanguard 500 Index Fund Investor Shares | TSM | VFINX |
| Vanguard Small-Cap Index Fund Investor Shares | SCB | NAESX |
| DFA US Small Cap Value I | SCV | DFSVX |
| Vanguard International Growth Fund Investor Shares | Int’l | VWIGX |
| Vanguard Intermediate-Term Tax-Exempt Fund Investor Shares | Bonds | VWITX |
| Fidelity Real Estate Investment Portfolio | REITs | FRESX |
| LBMA Gold Price AM USD | Gold | LBMA index |

Long story short, the top level findings were very similar to the findings determined with the index-centric data set. The latter being more consistent, more extensive and closer to modern passive investing, the author decided to use the index-centric data set for the graphs and numbers being published in this entire article.

For general context, here are two growth charts tracking the investment of $1 on 31-Dec-79, until 30-Jun-2020, for the various asset classes of interest represented by their index. Note that the vertical scale is logarithmic.

[Chart, line chart

Description automatically generated](https://www.bogleheads.org/blog/wp-content/uploads/2020/07/Growth-Chart1-1980-2020.jpg)[Chart, line chart

Description automatically generated](https://www.bogleheads.org/blog/wp-content/uploads/2020/07/Growth-Chart2-1980-2020.jpg)

The Elusive Rebalancing Bonus – Part 2

August 8, 2020 by [Siamond](https://www.bogleheads.org/blog/author/siamondblog/" \o "View all posts by Siamond)

Passive investors typically rebalance their accounts every now and then to get the various positions back in sync with their target asset allocation. This multi-part article studies various rebalancing methods and analyzes if some of those methods provided a significant rebalancing bonus. [Part 1](https://www.bogleheads.org/blog/2020/08/08/the-elusive-rebalancing-bonus---part-1) focused on describing the most typical rebalancing methods and illustrating them. In Part 2, we will perform a more systematic quantitative analysis, using the simple annual rebalancing method as a baseline. [Part 3](https://www.bogleheads.org/blog/2020/08/08/the-elusive-rebalancing-bonus-part-3/) will explore a few miscellaneous topics (e.g. frequent rebalancing, less conventional asset allocations).

What should we compare to?

Some rebalancing literature tends to compare a rebalancing approach with ‘unbalancing’ (letting one’s asset allocation drift without ever rebalancing). This doesn’t make much sense, as an ever drifting asset allocation (AA) has little to do with passive investors staying the course. Unless it is truly deliberate to let one’s AA drift to no end, e.g. to simplify one’s life when discipline doesn’t matter any more, the great majority of passive investors have a target AA and one has to rebalance at some point to stick to the plan.

Here is an example of an initial AA (50% US stocks, 30% US Bonds, 20% International) which was never rebalanced for a period of 20 years, starting in 1980. After a few years of relative stability, the current AA started to drift, with the allocation shifting towards significantly more (US) stocks and less bonds (while International varied a good deal).

[Chart

Description automatically generated](https://www.bogleheads.org/blog/wp-content/uploads/2020/07/Rebalancing-1980-20-NEVER.jpg)

A more sensible approach probably consists in comparing the outcome of a given rebalancing method with a realistic baseline, i.e. the very straightforward annual rebalancing method. Here is what would have happened in the same time period when rebalancing every 52 weeks (to approximate annual rebalancing with our weekly returns data set), simply enforcing good solid discipline, with all its behavioral benefits.

[Chart

Description automatically generated](https://www.bogleheads.org/blog/wp-content/uploads/2020/07/Rebalancing-1980-20-Annual-52w.jpg)

Many people perceive that more sophisticated methods (notably when using rebalancing triggers) would allow to extract a rebalancing bonus of sorts (i.e. better returns) by virtue of ‘selling high and buying low’. At the intuitive level, this certainly has appeal, but does (or did) this really hold true? Let’s run the numbers and investigate.

An intuitive definition of a rebalancing bonus

So far, we focused on specific time periods (e.g. 20 years periods, starting in 1980 or in 1995) for illustrative purposes. A more systematic analysis is in order to fully analyze the intrinsic properties of rebalancing methods.

To do so, the author designed a spreadsheet allowing to run all cycles of 20 years contained in the 1980-2020 time interval, starting cycles on January 1st of each year. Details about data sources were provided in Part 1 of the article. Aggregate statistics are then compiled across all those cycles. Click on the image below for a larger display. Excel buffs will recognize a data table (although it was actually implemented as a macro, to speed things up).

[Table

Description automatically generated](https://www.bogleheads.org/blog/wp-content/uploads/2020/08/Full-Stats-20y-R-BAND-20.jpg)

You can see the outcomes of every cycle, per starting year, illustrated by various metrics. In this example, relative bands (20%) were analyzed for our usual 50/20/30 portfolio. As you can see by the final portfolio values (‘Portfolio Last Yr’ column) and by the annualized returns (‘CAGR’ column), the starting date made a large difference for investors (e.g. due to the ‘roaring 90s’ bull market, followed by two decades of much rougher waters). The number of rebalancing events was pretty much the same across cycles, roughly once every other year on average.

Now what about a rebalancing bonus of sorts? The rightmost column provides the annualized returns (CAGR) for a periodic (every 52 weeks, quasi-annual) rebalancing method. The column to its left computes the difference between annualized returns for relative bands and such annual periodic rebalancing. The average difference was 0.07%, which isn’t entirely negligible, but not that significant either (an order of magnitude similar to modern passive funds’ expense ratios).

**Let’s define the ‘rebalancing bonus’ as such average difference across all investment cycles, compared to the baseline method, i.e. annual periodic rebalancing.** This seems a solid intuitive definition, the rebalancing bonus computing the improvement on returns compared to the baseline method. Note that rebalancing bonuses varied a good deal depending on the cycle (i.e. the starting date) and sometimes got negative.

Starting dates, bonus variability and significance

Backtesting results tend to be very sensitive to start and end dates. By cherry-picking a start/end date, one can prove more or less everything you want. The process described so far (and the data table above) tried to minimize such effect by aggregating the results of all periods of 20 years contained in the 1980-2020 time interval. But we always started 20 years cycles at the beginning of a given year.

The author performed a sensitivity test, starting 20 years cycles every 3 months instead of every year. The results were only very mildly impacted on average (e.g. by 0.01% or 0.02%), but individual cycles did display some non-negligible variation between quarters of a given year. All tests performed in the rest of the article used this refined approach of multiplying the number of starting points, on a quarterly basis.

Using the same AA and the same 20% relative bands as before, here are the corresponding aggregate statistics for such extended collection of cycles.

[](https://www.bogleheads.org/blog/wp-content/uploads/2020/08/Quarterly-Start-Aggr-Metrics-1980-2020-50-20-30.jpg)

As you can see, the average bonus (across all cycles) is a tad misleading as there is a ~0.1% standard deviation around such average and squarely negative bonuses are definitely part of the list of possible outcomes. Similar results were obtained when varying the AA or the rebalancing method. This essentially means that any average bonus result shown in this article which is lower than 0.1% is just not that significant.

Comparing multiple scenarios

The spreadsheet used by the author allows to compare sets of scenarios, each scenario being defined by specific parameter settings. For each scenario, the full collection of 20 years cycles was computed and then summarized through a few metrics. Using our 50/20/30 target AA, here are the results. Click on the image below for a larger display.

[Graphical user interface, application, table, Excel

Description automatically generated](https://www.bogleheads.org/blog/wp-content/uploads/2020/08/Metrics-Scenarios-AA-50-20-30.jpg)

At the first glance, fairly typical bands extracted a rather puny rebalancing bonus (on average). For example, SCN1b (20% relative bands) ended up with an average bonus of 0.07%. It took pretty wide bands (hence very few rebalancing events) to start getting in the range of rebalancing bonuses higher than 0.1%. For example, SCN3b (10% fixed bands) ended up with a rebalancing bonus of 0.17%, while only rebalancing 4 times. As to one-way rebalancing on annual basis, it lagged the baseline method by a rather significant 0.21%.

Let’s try with a balanced target AA, half stocks, half bonds, stocks being split 35% US and 15% Int’l. As you can see below, the results turn out fairly similar to the previous test.

[Graphical user interface, application, table, Excel

Description automatically generated](https://www.bogleheads.org/blog/wp-content/uploads/2020/08/Metrics-Scenarios-AA-35-15-50.jpg)

Finally, let’s try with an AA holding a much larger position in bonds (70%), plus 20% US stocks and 10% Int’l stocks. The results turn out again quite similar to the previous test.

[Graphical user interface, application, table, Excel

Description automatically generated](https://www.bogleheads.org/blog/wp-content/uploads/2020/08/Metrics-Scenarios-AA-20-10-70.jpg)

In all cases, only the 10% fixed rebalancing band exhibited a significant rebalancing bonus. Such band is pretty wide though and one would argue this is allowing too much of a drift compared to the target AA. Here is an example of a trajectory starting in 1995 with our usual 50/20/30 target AA. One can hardly call such approach ‘staying the course’.

[Chart

Description automatically generated](https://www.bogleheads.org/blog/wp-content/uploads/2020/07/Rebalancing-1995-20-Fixed-10.jpg)

The author performed more tests with simple asset allocations (e.g. 3 asset classes) and results with a meaningful rebalancing bonus were only obtained with few rebalancing events, hence allowing for a similar excessive drift compared to target. While tests with a more reasonable amount of drift only displayed very small rebalancing bonuses, barely significant.

Is this even a fair comparison?

There is clearly a lesson here from both types of results (better bonus with large bands; worst results with one-way rebalancing). In both cases, the algorithm lets the current AA drift far away from the target AA (see Part 1 for illustrations of such drift). In the first case, given that stocks tend to grow faster on average than bonds, the AA tends to drift towards more stocks than targeted. In the second case, by the very nature of one-way rebalancing, the AA tends to drift towards more bonds than targeted.

Let’s explore a bit further this hypothesis. If you look again at the numbers provided in the previous section, you will notice a “AA-TSM Delta” metric. For each cycle (starting year + 20 years investment period), the current AA for every week is assessed and the US stocks part of it recorded. Such information is then averaged across the investment period and the delta to the target AA (for US stocks) is computed. Those average deltas per cycle are themselves averaged out for the full set of cycles. This essentially represents the average departure from the target AA (for US stocks) for a given scenario. A similar metric (“AA-TIM Delta”) is computed for Int’l stocks and another one (“AA-TBM Delta”) for US bonds.

Looking at tables above, we see that even the simple periodic rebalancing (52 weeks, i.e nearly annual) displays a small average departure of 0.2% (for bonds) from the 50/30/20 target AA. Which makes sense as stocks tend to grow faster than bonds and the current AA is allowed to drift during the year, while being rebalanced to target once a year.

Now let’s check what happened for the rebalancing method exhibiting the large bonus (10% fixed bands). Using the 50/20/30 target AA, the average bonds delta is 1.8%, significantly higher than the average delta for annual rebalancing. It is hardly surprising that this method displayed a non-negligible rebalancing bonus on average, as it somewhat amounts to using an AA with more stocks. In a similar fashion, one-way rebalancing displayed a significant underperformance largely because it amounted to a more bonds-heavy AA. But then, if one truly wants to have more or less stock (or bonds) exposure, why bother with large rebalancing bands, better pick a target AA matching one’s true goals, while using a simple rebalancing process. This would also be more deterministic as rebalancing bonuses are quite variable and highly dependent on the exact investment period.

The drift towards more stocks (or more bonds) that specific rebalancing methods display doesn’t fully explain rebalancing bonuses. One can run a test tweaking a target AA by 1.5% to 2% more stocks than the regular 50/20/30 AA, while using annual rebalancing. And then compare to rebalancing with 10% fixed bands for the regular AA. Part of the rebalancing bonus disappears, but not all of it. Still, the current AA drifting between rebalancing events has clearly a significant bearing on the results.

Perceptions and bottomline

The conclusion of this analysis is that **the exact form of rebalancing may not matter much as long it is minimally sensible**. It appears that there is not a lot of significant rebalancing bonus to be gained by using more sophisticated methods (e.g. rebalancing triggers/bands), plus part of the bonus is a side-effect of the current asset allocation drifting (typically towards more stocks), in addition of being fairly random.

In other words, people using a very simple annual rebalancing method should be commended for keeping things simple and have little reason to change what they are doing. People who are more intellectually satisfied with rebalancing triggers (or find behavioral benefits in doing so – see below) are also fine, but they need to be aware that they won’t magically get higher returns by doing so.

This result might be perceived as counter-intuitive though. Rebalancing discussions are full of ‘keep some dry powder’ and ‘sell high and buy low’ perspectives. The trouble with such perspectives is that they are driven by intuition and emotions, hence highly subject to various [behavioral biases](https://www.bogleheads.org/wiki/Behavioral_pitfalls). Recency bias and anchoring are probably two of the main culprits here.

Say the stock market suddenly drops 30% (as in March 2020). Investors keeping their cool would be highly tempted to use cash or bonds to buy in cheapER stocks and rebalancing seems like a very disciplined way of doing so. There is certainly nothing wrong about rebalancing at such time if your preferred rebalancing method tells you to proceed (the author did!), but is that truly a sure winner move? Probably not, otherwise it would indicate a form of market timing that works and there is just no such thing (in Part 1, we’ve seen how rebalancing triggers can lead to poor market timing calls). Maybe in this precise case, we got lucky and it was indeed a good move, but then our quantitative analysis showed that such luck is a fickle thing…

**The bottomline is that rebalancing is a very useful discipline to stay the course, to stick to one’s investment plan. But any perception of ‘selling high, buying low’ is mostly one’s intuition (behavioral biases included) playing games with one’s brain, while there is just little concrete reality to a ‘rebalancing bonus’ of sort, besides side-effects of the AA drifting away from its target.**

Some real life considerations

In this entire study, we made the simplifying assumption of looking at the trajectory of an initial amount invested at T0, staying invested for 20 years, with a portfolio being fully rebalanced every now and then. Real life doesn’t work like that, due to cash flows. Accumulators will (or should) add to their retirement kitty on a fairly regular basis. Retirees will withdraw from their savings on a fairly regular schedule.

When adding some new savings to one’s portfolio, it would be logical to use cash to buy more shares from the asset class which is ‘under water’, checking the current AA compared to target. It is a partial rebalancing event of sorts.

When withdrawing some savings from one’s portfolio, it would be logical to use cash to sell shares from the asset class which is ‘over water’, when checking the current AA compared to target. It is also a partial rebalancing event of sorts.

Such partial rebalance transaction driven by cash flows will typical occur once a month or once a quarter. Consequently, the portfolio will be rebalanced (at least partly) much more often than the previous simulations suggested. One could call such process ‘soft rebalancing’, which typically only acts on one asset class (or maybe two) at a time, exchanging shares for cash or cash for shares (as opposed to a ‘hard rebalancing’ event where the entire target AA is restored, selling shares to buy shares).

Such consideration leads some passive investors to a slightly different approach than discussed before. Soft-rebalance during the year based on cash flows. Once a year, re-assess your current AA vs. its target. Only hard-rebalance if some trigger (e.g. fixed bands, relative bands, etc) is exceeded during this annual check. In practice, such approach leads to very few proactive rebalancing operations over the years.

Due to its typical frequency, soft rebalancing with cash flows will probably slim down any potential ‘rebalancing bonus’ one might gain, while helping to more tightly stay the course. Such effect has not been quantified in this article, which solely focuses on hard rebalancing events.

What about behavioral considerations?

In the author’s experience (during accumulation time as well as retirement time), there are strong behavioral reasons for which choosing a rebalancing method which fits well one’s skills and psychology is important. The incentive is not to chase an elusive rebalancing ‘bonus’ of sorts, but the following considerations became increasingly important to the author, notably in a time of crisis:

* A rebalancing trigger feels like an opportunity, a positive sentiment.
* It gives something to do, an action to take (it is easy to say ‘stay the course’, but most of us itch to **do something**), and… better do something which is NOT damaging.
* It is a mechanical process, which should come with no hesitation, no regret, no hidden ‘market timing’ attempt.

Some may be more inclined to use an annual rebalancing process for its simplicity and find comfort in NOT acting during the year, this is perfectly understandable. Others will find more comfort in the more reactive methods based on simple forms of rebalancing triggers (e.g. fixed/relative bands). Others (like the author) will be more inclined to trust a more consistent method like adaptive bands or cumulative drift. As long as this all helps to stay the course and stay close to your target AA, all those choices are perfectly valid if they suit your personality. **In other words, just… pick whatever rebalancing method you’re comfortable with, stick with it and things will work out.**

The Elusive Rebalancing Bonus – Part 3

August 8, 2020 by [Siamond](https://www.bogleheads.org/blog/author/siamondblog/" \o "View all posts by Siamond)

Passive investors typically rebalance their accounts every now and then to get the various positions back in sync with their target asset allocation. In [Part 1](https://www.bogleheads.org/blog/2020/08/08/the-elusive-rebalancing-bonus---part-1), we explained and illustrated various rebalancing methods, then documented the data sources used in this study. In [Part 2](https://www.bogleheads.org/blog/2020/08/08/the-elusive-rebalancing-bonus---part-2), we pondered if some of those methods provided a significant rebalancing bonus compared to an annual rebalancing model. In Part 3, we will explore miscellaneous topics (e.g. frequent rebalancing, less conventional asset allocations).

Frequent rebalancing

We discussed annual periodic rebalancing a good deal. We also observed that in real life, one would typically partly rebalance quite often by virtue of regular contributions (accumulators) or regular withdrawals (retirees). This begs the question about the possible underperformance of frequent rebalancing. Let’s run a test with rebalancing periods of 1 week, 2 weeks, 4 weeks (~ 1 month), up to 208 weeks (~ 4 years). The bonus is relative to the performance of 52 weeks (~ 1 year) rebalancing. The asset allocation (AA) is 50% US stocks, 20% International stocks, 30% US bonds.

[Graphical user interface, application, table, Excel

Description automatically generated](https://www.bogleheads.org/blog/wp-content/uploads/2020/08/Rebalancing-1980-20-Varying-Periodicity.jpg)

Consistently with our ‘AA drift’ observation (see Part 2), unfrequent rebalancing displayed a small rebalancing bonus, while more frequent rebalancing showed a very slight underperformance. It is a little puzzling to see that rebalancing every single week didn’t meaningfully harm the investor though. This being said, those average bonus numbers were very small and undoubtedly sensitive to input data (e.g. data set, time interval being studied). The author varied the AA in multiple ways and results stayed quite consistent, although in some cases tiny bonuses became negative and the reverse way around.

We should also remember that some of our data sources lack in weekly granularity in the early years, possibly skewing a tad the weekly rebalancing findings. We do have full granularity all the way at the monthly level though, so let’s double-click a bit and explore each cycle of 20 years being tested, when rebalancing every 4 weeks. See Part 1 for detailed explanations about such table. For simplicity, this graph only shows cycles starting at the beginning of a calendar year.

[Table

Description automatically generated](https://www.bogleheads.org/blog/wp-content/uploads/2020/08/Rebalancing-1980-20-Periodicity-4w-details.jpg)

Although the rebalancing bonus stayed quite consistently negative when comparing 4 weeks periodicity with annual periodicity, there are clearly TWO distinct patterns here. Rebalancing cycles (20 years long) starting in the 80s didn’t make much of a difference, while rebalancing cycles starting in 1989 and later showed a more significant difference, with a strong discontinuity between the 1988 cycle and the 1989 cycle.

What happened in Jan 1st, 1989 plus 20 years? This was the end of 2008 with the financial crisis hitting hard and fast. This shows that even with 40 years of solid weekly/monthly data to analyze, a specific event can impact the results quite significantly. We are playing with tiny numbers here (the bonuses -or lack thereof- being observed) and one might very well reach misguided conclusions by assuming such tiny numbers convey a lot of significance.

Stepping back from the details, it appears that frequent rebalancing isn’t much of a problem, and accumulators/retirees shouldn’t be shy of doing the intuitive thing for their cash flows (buy ‘low’ asset classes; sell ‘high’ asset classes) – if only because one would typically feel good about it… As explained in Part 2, behavioral considerations about rebalancing are as important as numerical consequences.

Gold – and one-way rebalancing

Let’s run a test with a more extreme AA using 25% US stocks, 25% Int’l stocks, 25% bonds and 25% gold and our usual ’20 years cycles’. Remember that gold is a very peculiar asset class, with a mind of its own!

[A screenshot of a computer

Description automatically generated with medium confidence](https://www.bogleheads.org/blog/wp-content/uploads/2020/08/Metrics-Scenarios-25-25-25-25-Gold.jpg)

Regular rebalancing methods came up with similar results as before (a tad better, actually). What is striking though is the hefty rebalancing bonus of the fixed 10% bands and the eye-popping bonus coming with one-way rebalancing.

One-way rebalancing delivering a positive bonus is an interesting side-effect, although probably not intended by proponents of such method. Gold is very volatile, but tends to not return much above inflation on average; one-way rebalancing implies rare rebalancing events, which let the AA drift a good deal towards the more productive asset classes during most cycles, while aggressively drifting towards gold during those deep crises where investors sought gold as a refuge. One can hardly call that ‘stay the course’, as illustrated by the cycle starting in January 1995.

[Chart

Description automatically generated](https://www.bogleheads.org/blog/wp-content/uploads/2020/08/Rebalancing-1995-20-One-way-Gold.jpg)

The author would strongly suggest to forget about one-way rebalancing, the side-effects are just plain weird and at odds with staying-the-course principles. Arguing that one would only use such one-way approach during a deep crisis isn’t convincing as there is just no good way (without hindsight) to factually assess when a deep crisis is over for good, leading the investor to make emotion-driven calls in a time of duress – NOT a recipe for success.

As the [Bogleheads wiki page](https://www.bogleheads.org/wiki/Rebalancing" \l "Modified_Approaches) suggests, an alternate method to deal with ‘catch a falling knife’ fears is the following:

*Keep a floor level of bonds, i.e. once the total amount of bonds has dropped to a fixed amount, stop rebalancing into stocks.*

This seems a much sounder way to proceed to deal with behavioral challenges (the March 2020 deep dive definitely reinforced this perception, leading the author to slightly tune his IPS in this respect, with a floor expressed as a fixed amount of dollars – as opposed to a percentage).

A US allocation with strong tilts

So far, we mostly focused on ‘three funds’ asset allocations mixing US stocks, International stocks and US bonds. A subset of passive investors take a different tack though, some solely focusing on US assets, others making a significant bet towards narrow segments of the market which displayed a significant premium in the past.

To stay fairly consistent with our previous tests, let’s look at a US-only asset allocation with 70% stocks and 30% bonds. The first test uses a simple total-market allocation for US stocks. As you can see below, the rebalancing bonuses are slightly higher (albeit small for the most part) than for previous scenarios involving international stocks, which makes sense as the US outperformed International in the past 3 decades (while being strongly outperformed in the 80s), hence more drift away from bonds on average.

[A screenshot of a computer

Description automatically generated with medium confidence](https://www.bogleheads.org/blog/wp-content/uploads/2020/08/Metrics-Scenarios-US-70-30.jpg)

The second test includes a significant real estate (REITs) position (10%). It also introduces a Small-Caps and a Small-Caps-Value tilt (25% of the US stocks allocation, hence 15% of the total allocation for each, leaving 30% for total-market stocks). US bonds stay unchanged at 30%.

[A screenshot of a computer

Description automatically generated with medium confidence](https://www.bogleheads.org/blog/wp-content/uploads/2020/08/Metrics-Scenarios-US-70-30-TILTED.jpg)

With such a diverse asset allocation, both relative and fixed bands start to display somewhat odd results. As discussed in Part 1, such simple methods are good for very simple (e.g. 2 or 3 funds) asset allocations, but suffer from consistency issues for more complex AAs (fixed bands being the most troublesome). Adaptive bands use a more mathematically consistent formula and are better suited for such a tilted portfolio. A more detailed analysis would be required to explain this point in details, but this would exceed the scope of this article (check [this thread](https://www.bogleheads.org/forum/viewtopic.php?t=186203) on the Bogleheads forum). Another (simpler) way to provide more consistency with complex AAs is to use the cumulative drift method (sum up absolute positive drifts from target AA for each asset class, compare to threshold).

The higher-level outcome of such ’tilted’ scenario is that rebalancing bonuses were higher than in most previous scenarios we’ve looked at, which isn’t surprising as we’re using higher-performance asset classes (that is, for the 1980-2020 time interval), hence generating more drift towards such winners in-between rebalancing events. The drifting AA effect isn’t a full explanation of the bonuses being observed though, and maybe the algorithm did make a couple of good market timing calls (whether it is luck or skill remains to be determined!).

Summary

The primary findings of Part 3 can be summarized as:

* Frequent rebalancing (by design or as a side effect of regular contributions or withdrawals) does not impact rebalancing in a significantly negative manner.
* One-way rebalancing is just a bad idea. If one really wants some security against doomsday ‘falling knife’ scenarios, a *floor level of bonds* expressed in absolute dollars (or local currency) seem a much better approach.
* US-only scenarios work in a fairly similar manner as more diversified asset allocations, displaying slightly higher bonuses primarily due to AA drift between rebalancing events.
* If one starts to add more complexity to the AA (e.g. using several tilts), the consistency of adaptive bands and cumulative drift methods start becoming more significant. Although the simplicity of annual rebalancing is hard to argue against and works fine with any type of AA.

**Portfolio Rebalancing: The Whys And The Hows**

Apr. 08, 2014 7:00 AM ET[26 Comments](https://seekingalpha.com/article/2130243-portfolio-rebalancing-the-whys-and-the-hows#comments)1 Like

**Summary**

* Ideally, to eliminate style drift investors should rebalance daily. However, because the real world involves costs, investors should reduce, not eliminate, style drift to an acceptable level.
* Investors should rebalance wherever there is sufficient cash to make the effort worthwhile, thus eliminating any tax issues and either eliminating or minimizing trading costs.
* An investor’s investment policy statement (IPS) should include targeted asset allocations as well as minimums and maximums and a rebalancing table.

Rebalancing is the process by which a portfolio's "style drift" caused by market movements is eliminated or minimized. Style drift causes the risk and expected return of the portfolio to change. Thus, if you want to eliminate style drift, you should rebalance daily. However, in the real world there are often costs (trading expenses and in taxable accounts taxes) including the time and effort to rebalance. Thus, you want to reduce (not eliminate) style drift to an acceptable level. That raises the question of how to determine when you should rebalance.

**First**, you should rebalance whenever you have sufficient cash to make the effort worthwhile. That eliminates any tax issues and it either eliminates or minimizes trading costs (depending on the investment vehicle and the custodian). This is the way many endowments (such as the [Yale Endowment](http://www.myplaniq.com/LTISystem/jsp/portfolio/ViewPortfolio.action?ID=407)) rebalance, using the cash flows they donors as well as the interest and dividends provided by their portfolios.

**Second**, you can choose not to automatically reinvest dividends, interest, and distributions. Instead, take them in cash and rebalance using the proceeds. In that way you limit transactions to only purchases, avoiding sales and the tax consequences that can accompany them in taxable accounts. This is generally a good choice, though it may not be appropriate for small accounts where the custodian is charging a minimum fixed transaction fee. If the fee is large relative to the size of the distributions, you would be better served reinvesting the proceeds as there's no transaction fee.

**Third**, before rebalancing that requires the realization of taxable gains, you should determine if in the near future you will have new investable funds (e.g., bonus, maturing bond, the sale of security) that will become available. If this is so, you should consider delaying rebalancing, as long as the delay is a relatively short one. The exception to the guidance on minimizing the realization of gains when performing rebalancing is when there are long-term gains that can be realized from the sale of taxable fixed income investments. Here the "conventional wisdom" gets spun on its head. If there is a long-term gain that could be realized (whether for rebalancing purposes or not) investors in all but the lowest bracket should take it. The reason is that doing so converts future interest income that would have been taxed at the ordinary income tax rate (e.g., 35 percent) into a long-term capital gain that is taxed at a rate of only 20 percent. Of course, the loss of the present value of having to pay taxes early, as well as the costs of the transaction should be considered.

**Fourth,** I would generally recommend not rebalancing if it meant having to realize significant short-term capital gains (unless offsetting losses were available). Instead, you should consider waiting until the gains become long term. If taxes are an issue you might also consider rebalancing to minimum and maximum levels, instead of to the targeted level.

**Fifth**, rebalancing should be considered in tax-deferred accounts first, if doing so avoids realizing capital gains. However, if capital losses are available (and that should be determined at individual lot levels), then the rebalancing should be done in the taxable accounts. It is important to note that the belief "if I have all asset classes represented in tax-deferred accounts, that will allow me to tax-efficiently rebalance in the future," isn't correct. The reason is that the tax inefficiencies deriving from the wrong location are greater than any benefit when rebalancing is needed.

**Sixth**, since rebalancing is about managing style drift, it shouldn't be based on *time*. Instead, it should be based on how much style drift you are willing to accept as the price for keeping costs (especially taxes) low.

**The Rebalancing Table**

Included in your investment policy statement (NYSEARCA:[IPS](https://seekingalpha.com/symbol/IPS?source=content_type%3Areact%7Csection%3Amain_content%7Cbutton%3Abody_link)) should be an asset allocation and rebalancing table. The table should include not only the target levels for each asset class, but also the minimum and maximum levels to which the allocations will be allowed to drift. Some drift should be allowed to occur because rebalancing generally involves costs, including transactions fees and taxes in taxable accounts.

I suggest you use a 5/25 percent rule in an asset class's allocation before rebalancing is considered. That is, rebalancing should be considered if the change in an asset class's allocation is greater than either an absolute 5 percent or 25 percent of the original percentage allocation. Note that the actual percentages used are not as important as having a specific plan and having the discipline to adhere to the plan. In other words, a 4/20 rule might be as appropriate as a 5/25 rule.

**Application of 5/25 Rule:** Assume an asset class was given an allocation of *10* percent. Applying the 5 percent rule, one would not rebalance unless that asset class's allocation had either risen to 15 percent (*10* percent + 5 percent) or fallen to 5 percent (*10* percent - 5 percent). Using the 25 percent rule one would, however, reallocate if it had risen or fallen by just 2.5 percent (*10* percent x 25 percent) to either 12.5 percent (*10* percent +2.5 percent) or 7.5 percent (*10* percent - 2.5 percent). In this case the 25 percent figure was the governing factor. If one had a 50 percent asset class allocation, the 5/25 percent rule would cause the 5 percent figure to be the governing factor since 5 percent is less than 25 percent of 50 percent, which is 12.5 percent. In other words, one rebalances if either the 5 percent or the 25 percent test indicates the need to do so.

The need for rebalancing should be checked at three levels.

* At the broad level of equities and fixed income.
* At the level of domestic and international asset classes.
* At the more narrowly defined individual asset class level (such as emerging markets, real estate, small-cap, value and so on).

Applying this rule you can produce a rebalancing table that looks like the one below:

**Sample Rebalancing Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Asset Class** | **Minimum Allocation (%)** | **Target Allocation (%)** | **Maximum Allocation (%)** |
| U.S. large | 7.5 | 10 | 12.5 |
| U.S. large value | 7.5 | 10 | 12.5 |
| U.S. small | 7.5 | 10 | 12.5 |
| U.S. small value | 7.5 | 10 | 12.5 |
| Real Estate | 7.5 | 10 | 12.5 |
| **Total U.S.** | **45** | **50** | **55** |
| International large value | 3.75 | 5 | 6.25 |
| International small | 3.75 | 5 | 6.25 |
| International small value | 3.75 | 5 | 6.25 |
| Emerging markets | 3.75 | 5 | 6.25 |
| **Total International** | **15** | **20** | **25** |
| **Total Equity** | **65** | **70** | **75** |
| Nominal Bonds | 7.5 | 10 | 12.5 |
| TIPS | 15 | 20 | 25 |
| **Total Fixed Income** | **25** | **30** | **35** |

It's important to understand that rebalancing is more important when asset classes that are out of balance have low correlation (the strength of the linear relationships between asset classes). For example, the annual correlation between the S&P 500 and Treasury bonds has been about 0 (there's no correlation). That makes rebalancing when these two asset classes are out of balance very important. However, the annual correlation between U.S. small value stocks and U.S. small stocks has been 0.97. And the annual correlation between U.S. small value stocks and U.S. large value stocks is about 0.9. In either of these cases, if these asset classes exceed their tolerance ranges at the same time, and your over the max in one and under in another, and there are significant costs that would be incurred, you can decide to wait, accepting the style drift because the correlations are so high. On the other hand, the correlation between U.S. stocks and emerging markets stocks is much lower (about 0.6), making it more important to rebalance. In other words, rebalancing is as much an art, balancing the benefits of controlling risk with the costs of doing so. Rebalancing in the most cost and tax efficient manner is one of the ways a good financial advisor can add value.

**Myths about Rebalancing**

Unfortunately, what passes for conventional investment wisdom often isn't true. A good example is that there are two myths about the rebalancing. These myths need to be exposed.

**Reversion to the Mean**

The first myth is that rebalancing is a "reversion to the mean" strategy. An example will demonstrate that this is false. Consider a portfolio with an asset allocation of 50 percent stocks/50 percent bonds. Assume that stocks have returned 10 percent and are expected to return 10 percent while bonds have returned 6 percent and are expected to return 6 percent. The first year stocks return 9 percent and bonds return 7 percent. A strategy that is based on reversion to the mean of returns would sell bonds (since they produced above average returns) to buy stocks (since they produced below average returns). However, since the portfolio now would have an asset allocation of greater than 50 percent for stocks, rebalancing would require that stocks be sold to buy more bonds - or buy sufficient bonds to increase the bond allocation to 50 percent.

**A Way to Increase Returns**

The second myth about rebalancing is that it increases returns. An example will demonstrate why that will not be the case most of the time. The majority of the time rebalancing will require investors to sell some of the *higher* expected returning asset class to purchase more of the *lower* expected returning asset class. For example, most of the time we would expect to have to sell stocks to buy fixed income assets. Similarly, we should expect that the majority of the time we will have to sell value stocks to buy growth stocks, small stocks to buy large stocks and/or emerging market stocks to buy developed market stocks. In each case, the majority of the time we will be selling the higher expected returning asset class to buy the lower expected returning asset class. While achieving the objective of restoring the portfolio's risk profile, rebalancing, in each of these cases, lowers the expected return of the portfolio. (This is why many recommend infrequent rebalancing - because they are focusing on maximization of returns, not controlling risk.) Of course, this won't always be true. When bonds outperform stocks, rebalancing will increase the expected return of the portfolio since you are reducing the allocation of the lower expected returning asset class in order to increase the allocation of the higher expected returning asset class.

**Conclusion**

The bottom line is that an important part of the winning investment strategy is the establishment of an IPS that includes your targeted asset allocations as well as minimums and maximums. The only way to adhere to that IPS is to rebalance. Be sure to rebalance whenever you have available cash. Otherwise, check for the need to rebalance monthly, or at least quarterly. If you don't, you could see the portfolio drift to an unacceptable level of risk.

### [Rebalancing: adaptive bands (a new approach)](https://www.bogleheads.org/forum/viewtopic.php?p=2825293#p2825293)

* [**Quote**](https://www.bogleheads.org/forum/posting.php?mode=quote&f=10&p=2825293)

[Post](https://www.bogleheads.org/forum/viewtopic.php?p=2825293#p2825293) by **[siamond](https://www.bogleheads.org/forum/memberlist.php?mode=viewprofile&u=37504)** » Sat Mar 05, 2016 11:34 pm

Let’s say that we have one asset category (or asset class) targeted at X% of the portfolio in a given Asset Allocation. Let’s say that after a significant price variation, the asset category of interest represents Y% of the portfolio. What could be a good criterion to decide to rebalance the portfolio?  
  
The [rebalancing wiki page](https://www.bogleheads.org/wiki/Rebalancing) describes deviations from the target by a certain **absolute percentage**:  
=> ABS(Y-X) > 5% would be a typical formula to implement this, with a 5% threshold  
  
The wiki page also describes deviations from the target by a certain **relative percentage**:  
=> ABS(Y/X-1) > 15% would be a typical formula to implement this, with a 15% threshold  
  
Let’s take a scenario where the asset category of interest varies by 25% (upwards or downwards). To keep the discussion symmetric, we need to remember that a 25% rise is the same variation as a 20% drop. (1/(1+25%)-1 = -20%). In other words, after a 25% rise, a 20% drop would bring the price to the same point, and conversely. Ok, so say we have a 25% price variation on our asset category (SCV, REIT, bonds, all equities, whatever), seems like time to rebalance!  
  
On [another thread](https://www.bogleheads.org/forum/viewtopic.php?f=10&t=185596), Kevin nicely illustrated the fact that both the absolute percentage approach and the relative percentage approach display idiosyncrasies, and that the outcome is quite dependent on the exact Asset Allocation (X%). Let me show the same effect while using slightly different graphs.  
  
Let’s assume that the asset category varies by 25%, while the rest of the portfolio does nothing (not fully realistic, but this helps the thinking). See the first graph below, if the target allocation is less than 30% of the portfolio, the relative band threshold is reached. Otherwise, it is NOT. That doesn’t seem terribly satisfying; the price of the asset category tanked or increased by a good deal (25%), whether you have a small or a large investment in this category, it seems like time to rebalance. Even if we were to assume that the rest of the portfolio actually went up, the curve would shift a bit, but this remains unsatisfying, notably with high allocations.  
  
