Abstract

This vignette provides an overview of calculating portfolio returns through time with an emphasis on the math used to develop the `Return.portfolio` function in **PerformanceAnalytics**. We first introduce some basic definitions, then give simple examples of computing portfolio returns in a prices and shares framework as well as a returns and weights framework. We then introduce `Return.portfolio` and demonstrate the function with a few examples.

1 Basic definitions

Suppose we have a portfolio of N assets. The value of asset i, V_i, in the portfolio is defined as

$$V_i = \lambda_i * P_i$$

where:

$$\lambda_i \text{ is the number of shares of asset } i$$
$$P_i \text{ is the price of asset } i$$

The total portfolio value, V_p, is defined as

$$V_P = \sum_{i=1}^{N} V_i$$

The weight of asset i, w_i, in the portfolio is defined as

$$w_i = V_i / V_P$$

where:

$$V_i \text{ is the value of asset } i$$
$$V_P \text{ is the total value of the portfolio}$$

The portfolio return at time t, R_t, is defined as

$$R_t = \frac{V_{p_t} - V_{p_{t-1}}}{V_{p_{t-1}}}$$

$$V_{p_t} \text{ is the portfolio value at time } t$$

2 Simple Example: Prices and Shares Framework

Suppose we have a portfolio of $N = 2$ assets, asset A and asset B. The prices for assets A and B are given as

```
> prices = cbind(c(5, 7, 6, 7),
+                c(10, 11, 12, 8))
> dimnames(prices) = list(paste0("t",0:3), c("A", "B"))
> prices
```

```
   A  B
t0 5 10
t1 7 11
t2 6 12
t3 7  8
```

We wish to form an equal weight portfolio, that is, form a portfolio where

$$w_i = \frac{1}{N} \text{ for } i \in 1, \ldots, N.$$

Let $V_{P0} = 1000$ be the portfolio value at t_0.
Step 1: Compute the number of shares of each asset to purchase.

$$\begin{aligned} w_i &= \frac{V_i}{V_P} \\ &= \frac{\lambda_i * P_i}{V_P} \end{aligned}$$

Solve for λ_i.

$$\lambda_i = \frac{w_i * V_P}{P_i}$$

$$\begin{aligned} \lambda_A &= \frac{w_A * V_P 0}{P_A} = \frac{0.5 * \$1000}{\$5} = 100 \\ \lambda_B &= \frac{w_B * V_P 0}{P_B} = \frac{0.5 * \$1000}{\$10} = 50 \end{aligned}$$

```
> V_P0 = 1000
> N = ncol(prices)
> w = rep(1 / N, N)
> lambda = w * V_P0 / prices["t0",]
> lambda
```

```
  A   B
100  50
```

Step 2: Compute the asset value and portfolio value for $t \in 0, \ldots, 3$.

```
> # Compute the value of the assets
> V_assets <- matrix(0, nrow(prices), ncol(prices), dimnames=dimnames(prices))
> for(i in 1:nrow(prices)){
+   V_assets[i,] = prices[i,] * lambda
+ }
> V_assets

     A   B
t0 500 500
t1 700 550
t2 600 600
t3 700 400

> # Compute the value of the portfolio
> V_P = rowSums(V_assets)
> V_P

  t0   t1   t2   t3
1000 1250 1200 1100
```

Step 3: Compute the portfolio returns for $t \in 1, \ldots, 3$.

```
> # Compute the portfolio returns
> R_t = diff(V_P) / V_P[1:3]
> R_t

         t1          t2          t3
 0.25000000 -0.04000000 -0.08333333
```

Step 4: Compute the weights of each asset in the portfolio for $t \in 0, \ldots, 3$

```
> weights = V_assets / V_P
> weights

           A         B
t0 0.5000000 0.5000000
t1 0.5600000 0.4400000
t2 0.5000000 0.5000000
t3 0.6363636 0.3636364
```

We have shown that calculating portfolio weights, values, and returns is simple in a prices and shares framework. However, calculating these metrics becomes more challenging in a weights and returns framework.

3 Example: Weights and Returns Framework

We will use the monthly returns during 1997 of the first 5 assets in the edhec dataset for the following example.

```
> library(PerformanceAnalytics)
> data(edhec)
> R = edhec["1997", 1:5]
> colnames(R) = c("CA", "CTAG", "DS", "EM", "EMN")
> R
```

```
               CA    CTAG      DS      EM    EMN
1997-01-31 0.0119  0.0393  0.0178  0.0791 0.0189
1997-02-28 0.0123  0.0298  0.0122  0.0525 0.0101
1997-03-31 0.0078 -0.0021 -0.0012 -0.0120 0.0016
1997-04-30 0.0086 -0.0170  0.0030  0.0119 0.0119
1997-05-31 0.0156 -0.0015  0.0233  0.0315 0.0189
1997-06-30 0.0212  0.0085  0.0217  0.0581 0.0165
1997-07-31 0.0193  0.0591  0.0234  0.0560 0.0247
1997-08-31 0.0134 -0.0473  0.0147 -0.0066 0.0017
1997-09-30 0.0122  0.0198  0.0350  0.0229 0.0202
1997-10-31 0.0100 -0.0098 -0.0064 -0.0572 0.0095
1997-11-30 0.0000  0.0133  0.0054 -0.0378 0.0041
1997-12-31 0.0068  0.0286  0.0073  0.0160 0.0066
```

Suppose that on 1996-12-31 we wish to form an equal weight portfolio such that the weight for asset i is given as:

$$w_i = \frac{1}{N} \quad \text{for } i \in 1, \ldots, N$$

where N is equal to the number of assets.

```
> N = ncol(R)
> weights = xts(matrix(rep(1 / N, N), 1), as.Date("1996-12-31"))
> colnames(weights) = colnames(R)
> weights
```

```
            CA CTAG  DS  EM EMN
1996-12-31 0.2  0.2 0.2 0.2 0.2
```

There are two cases we need to consider when calculating the beginning of period (bop) value.

Case 1: The beginning of period t is a rebalancing event. For example, the rebalance weights at the end of `1996-12-31` take effect at the beginning of `1997-01-31`. This means that the beginning of `1997-01-31` is considered a rebalance event.

The beginning of period value for asset i at time t is given as

$$V_{bop_{t,i}} = w_i * V_{t-1}$$

where w_i is the weight of asset i and V_{t-1} is the end of period (eop) portfolio value of the prior period.

Case 2: The beginning of period t is not a rebalancing event.

$$V_{bop_{t,i}} = V_{eop_{t-1,i}}$$

where $V_{eop_{t-1,i}}$ is the end of period value for asset i from the prior period.

The end of period value for asset i at time t is given as

$$V_{eop_{t,i}} = (1 + R_{t,i}) * V_{bop_{t,i}}$$

Here we demonstrate this and compute values for the periods 1 and 2. For the first period, $t = 1$, we need an initial value for the portfolio value. Let $V_0 = 1$ denote the initial portfolio value. Note that the initial portfolio value can be any arbitrary number. Here we use $V_0 = 1$ for simplicity.

```
> V_0 = 1
> bop_value = eop_value = matrix(0, 2, ncol(R))
```

Compute the values for $t = 1$.

```
> t = 1
> bop_value[t,] = coredata(weights) * V_0
> eop_value[t,] = coredata(1 + R[t,]) * bop_value[t,]
```

Now compute the values for $t = 2$.

```
> t = 2
> bop_value[t,] = eop_value[t-1,]
> eop_value[t,] = coredata(1 + R[t,]) * bop_value[t,]
```

It is seen that the values for the rest of the time periods can be computed by iterating over $t \in 1, \ldots, T$ where $T = 12$ in this example.

The weight of asset i at time t is calculated as

$$w_{t,i} = \frac{V_{t,i}}{\sum_{i=0}^{N} V_{t,i}}$$

Here we compute both the beginning and end of period weights.

```
> bop_weights = eop_weights = matrix(0, 2, ncol(R))
> for(t in 1:2){
+   bop_weights[t,] = bop_value[t,] / sum(bop_value[t,])
+   eop_weights[t,] = eop_value[t,] / sum(eop_value[t,])
+ }
> bop_weights

         [,1]      [,2]      [,3]      [,4]      [,5]
[1,] 0.200000 0.2000000 0.2000000 0.2000000 0.2000000
[2,] 0.195839 0.2011419 0.1969808 0.2088446 0.1971937

> eop_weights

          [,1]      [,2]      [,3]      [,4]      [,5]
[1,] 0.1958390 0.2011419 0.1969808 0.2088446 0.1971937
[2,] 0.1936464 0.2023282 0.1947562 0.2147071 0.1945622
```

The portfolio returns at time t are calculated as

$$R_{P_t} = \frac{V_t - V_{t-1}}{V_{t-1}}$$

```
> V = c(V_0, rowSums(eop_value))
> R_P = diff(V) / V[1:2]
> R_P

[1] 0.03340000 0.02376201
```

The contribution of asset i at time t is calculated as

$$contribution_{t,i} = \frac{V_{eop_{t,i}} - V_{bop_{t,i}}}{\sum_{i=1}^{N} V_{bop_{t,i}}}$$

```
> contribution = matrix(0, 2, ncol(R))
> for(t in 1:2){
+   contribution[t,] = (eop_value[t,] - bop_value[t,]) / sum(bop_value[t,])
+ }
> contribution

            [,1]        [,2]        [,3]       [,4]        [,5]
[1,] 0.002380000 0.007860000 0.003560000 0.01582000 0.003780000
[2,] 0.002408819 0.005994027 0.002403166 0.01096434 0.001991657
```

Note that contribution can also be calculated as

$$contribution_{t,i} = R_{t,i} * w_{t,i}$$

4 Return.portfolio Examples

```
> args(Return.portfolio)

function (R, weights = NULL, wealth.index = FALSE, contribution = FALSE,
    geometric = TRUE, rebalance_on = c(NA, "years", "quarters",
        "months", "weeks", "days"), value = 1, verbose = FALSE,
    ..., rebal_cost = 0, full_investment = FALSE)
NULL
```

If no `weights` are specified, then an equal weight portfolio is computed. If `rebalance_on=NA` then a buy and hold portfolio is assumed. See `?Return.portfolio` for a detailed explanation of the function and arguments.

```
> # Equally weighted, buy and hold portfolio returns
> Return.portfolio(R)

           portfolio.returns
1997-01-31       0.033400000
1997-02-28       0.023762011
1997-03-31      -0.001413340
1997-04-30       0.003678107
1997-05-31       0.017767315
1997-06-30       0.025914372
1997-07-31       0.036969968
```

```
1997-08-31      -0.005005540
1997-09-30       0.022080944
1997-10-31      -0.012352423
1997-11-30      -0.003843547
1997-12-31       0.012936194

> # Equally weighted, rebalanced quarterly portfolio returns
> Return.portfolio(R, rebalance_on="quarters")

           portfolio.returns
1997-01-31       0.033400000
1997-02-28       0.023762011
1997-03-31      -0.001413340
1997-04-30       0.003680000
1997-05-31       0.017660872
1997-06-30       0.025452430
1997-07-31       0.036500000
1997-08-31      -0.005136602
1997-09-30       0.022049167
1997-10-31      -0.010780000
1997-11-30      -0.002621013
1997-12-31       0.012985944

> # Equally weighted, rebalanced quarterly portfolio returns.
> # Use verbose=TRUE to return additional information
> # including asset values and weights
> Return.portfolio(R, rebalance_on="quarters", verbose=TRUE)

$returns
           portfolio.returns
1997-01-31       0.033400000
1997-02-28       0.023762011
1997-03-31      -0.001413340
1997-04-30       0.003680000
1997-05-31       0.017660872
1997-06-30       0.025452430
1997-07-31       0.036500000
1997-08-31      -0.005136602
1997-09-30       0.022049167
1997-10-31      -0.010780000
1997-11-30      -0.002621013
1997-12-31       0.012985944

$contribution
                    CA          CTAG            DS           EM          EMN
1997-01-31 0.002380000  0.0078600000  0.0035600000  0.015820000 0.0037800000
1997-02-28 0.002408819  0.0059940275  0.0024031662  0.010964341 0.0019916567
1997-03-31 0.001510442 -0.0004248891 -0.0002337074 -0.002576485 0.0003112995
1997-04-30 0.001720000 -0.0034000000  0.0006000000  0.002380000 0.0023800000
1997-05-31 0.003135294 -0.0002938187  0.0046568428  0.006351596 0.0038109577
```

```
1997-06-30 0.004252156  0.0016336242  0.0043610924  0.011874480 0.0033310776
1997-07-31 0.003860000  0.0118200000  0.0046800000  0.011200000 0.0049400000
1997-08-31 0.002635527 -0.0096662672  0.0029028423 -0.001344834 0.0003361293
1997-09-30 0.002444218  0.0038748559  0.0070493383  0.004659301 0.0040214532
1997-10-31 0.002000000 -0.0019600000 -0.0012800000 -0.011440000 0.0019000000
1997-11-30 0.000000000  0.0026626352  0.0010847819 -0.007205240 0.0008368108
1997-12-31 0.001392218  0.0058170647  0.0014782579  0.002942265 0.0013561387

$BOP.Weight
                  CA      CTAG        DS        EM       EMN
1997-01-31 0.2000000 0.2000000 0.2000000 0.2000000 0.2000000
1997-02-28 0.1958390 0.2011419 0.1969808 0.2088446 0.1971937
1997-03-31 0.1936464 0.2023282 0.1947562 0.2147071 0.1945622
1997-04-30 0.2000000 0.2000000 0.2000000 0.2000000 0.2000000
1997-05-31 0.2009804 0.1958792 0.1998645 0.2016380 0.2016380
1997-06-30 0.2005734 0.1921911 0.2009720 0.2043800 0.2018835
1997-07-31 0.2000000 0.2000000 0.2000000 0.2000000 0.2000000
1997-08-31 0.1966811 0.2043608 0.1974723 0.2037627 0.1977231
1997-09-30 0.2003458 0.1956998 0.2014097 0.2034629 0.1990818
1997-10-31 0.2000000 0.2000000 0.2000000 0.2000000 0.2000000
1997-11-30 0.2042013 0.2001981 0.2008855 0.1906148 0.2041002
1997-12-31 0.2047379 0.2033939 0.2025011 0.1838916 0.2054756

$EOP.Weight
                  CA      CTAG        DS        EM       EMN
1997-01-31 0.1958390 0.2011419 0.1969808 0.2088446 0.1971937
1997-02-28 0.1936464 0.2023282 0.1947562 0.2147071 0.1945622
1997-03-31 0.1954330 0.2021890 0.1947978 0.2124308 0.1951493
1997-04-30 0.2009804 0.1958792 0.1998645 0.2016380 0.2016380
1997-05-31 0.2005734 0.1921911 0.2009720 0.2043800 0.2018835
1997-06-30 0.1997416 0.1890138 0.2002366 0.2108869 0.2001210
1997-07-31 0.1966811 0.2043608 0.1974723 0.2037627 0.1977231
1997-08-31 0.2003458 0.1956998 0.2014097 0.2034629 0.1990818
1997-09-30 0.1984151 0.1952691 0.2039618 0.2036323 0.1987216
1997-10-31 0.2042013 0.2001981 0.2008855 0.1906148 0.2041002
1997-11-30 0.2047379 0.2033939 0.2025011 0.1838916 0.2054756
1997-12-31 0.2034876 0.2065290 0.2013644 0.1844387 0.2041802

$BOP.Value
                  CA      CTAG        DS        EM       EMN
1997-01-31 0.2000000 0.2000000 0.2000000 0.2000000 0.2000000
1997-02-28 0.2023800 0.2078600 0.2035600 0.2158200 0.2037800
1997-03-31 0.2048693 0.2140542 0.2060434 0.2271506 0.2058382
1997-04-30 0.2112921 0.2112921 0.2112921 0.2112921 0.2112921
1997-05-31 0.2131092 0.2077001 0.2119260 0.2138065 0.2138065
1997-06-30 0.2164337 0.2073886 0.2168638 0.2205414 0.2178474
1997-07-31 0.2213080 0.2213080 0.2213080 0.2213080 0.2213080
1997-08-31 0.2255792 0.2343873 0.2264866 0.2337012 0.2267743
```

```
1997-09-30 0.2286020 0.2233008 0.2298159 0.2321588 0.2271598
1997-10-31 0.2332393 0.2332393 0.2332393 0.2332393 0.2332393
1997-11-30 0.2355716 0.2309535 0.2317465 0.2198980 0.2354550
1997-12-31 0.2355716 0.2340252 0.2329980 0.2115858 0.2364204

$EOP.Value
                  CA      CTAG        DS        EM       EMN
1997-01-31 0.2023800 0.2078600 0.2035600 0.2158200 0.2037800
1997-02-28 0.2048693 0.2140542 0.2060434 0.2271506 0.2058382
1997-03-31 0.2064673 0.2136047 0.2057962 0.2244247 0.2061675
1997-04-30 0.2131092 0.2077001 0.2119260 0.2138065 0.2138065
1997-05-31 0.2164337 0.2073886 0.2168638 0.2205414 0.2178474
1997-06-30 0.2210221 0.2091514 0.2215698 0.2333548 0.2214419
1997-07-31 0.2255792 0.2343873 0.2264866 0.2337012 0.2267743
1997-08-31 0.2286020 0.2233008 0.2298159 0.2321588 0.2271598
1997-09-30 0.2313909 0.2277221 0.2378595 0.2374752 0.2317484
1997-10-31 0.2355716 0.2309535 0.2317465 0.2198980 0.2354550
1997-11-30 0.2355716 0.2340252 0.2329980 0.2115858 0.2364204
1997-12-31 0.2371735 0.2407183 0.2346988 0.2149712 0.2379808
```

5 Transaction Costs

The `Return.portfolio` function supports the application of transaction costs during rebalancing events. This is specified via the `rebal_cost` parameter. It allows the user to define a proportional transaction cost applied to the fractional change in weights.

When `rebal_cost` is non-zero, the absolute change in weights (the turnover) is calculated for each asset at the rebalance date. The total turnover is multiplied by the `rebal_cost` to calculate the transaction cost drag, which is then subtracted from the total portfolio value at the beginning of the new period. This effectively reduces the portfolio return by the amount of the transaction costs incurred during rebalancing. The cost is also proportionally allocated to the individual asset contributions, so that the sum of the contributions still correctly equals the portfolio return.