R Notebook

Now we want to port our work to a Shiny application so that an end user is able to:

1) choose tickers and portfolio weights
2) choose a start date
3) choose a rebalancing frequency
4) chart the growth of a dollar in the portfolio since the chosen start date

The input sidebar is identical to that of our app on returns distribtution We let the use choose 5 ticker symbols, 5 weights, a start date and rebalance period. The user then clicks 'submit' to fire up the reactives.

```
portfolio_growth_xts <- eventReactive(input$go, {

  symbols <- c(input$stock1, input$stock2, input$stock3, input$stock4, input$stock5)

  prices <- getSymbols(symbols, src = 'yahoo', from = input$date,
             auto.assign = TRUE, warnings = FALSE) %>%
  map(~Ad(get(.))) %>%
  reduce(merge) %>%
  `colnames<-`(symbols)

  w <- c(input$w1/100, input$w2/100, input$w3/100, input$w4/100, input$w5/100)

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

  asset_returns_xts <- na.omit(Return.calculate(prices_monthly, method = "log"))

  portfolio_growth_xts <-
    Return.portfolio(asset_returns_xts,
                     wealth.index = 1,
                     weights = w,
                     rebalance_on = input$rebalance) %>%
    `colnames<-`("growth")

})
```

Our substantive work has been completed and we now want to display the chart of the portfolio growth over time. Outside of Shiny, this would be a simple passing of the xts object to `highcharter`.

As with `ggplot`, Shiny uses a custom function for building reactive highcharter charts called `renderHighchart()`. Once we invoke that `renderHighchart()`, our code looks very similar to our previous visualization work as we use `hc_add_series(portfolio_growth_xts(), name = "Dollar Growth", color = "cornflowerblue")` to add our portfolio growth xts object to a chart.

```
renderHighchart({

  highchart(type = "stock") %>%
  hc_title(text = "Growth of a Dollar") %>%
  hc_add_series(portfolio_growth_xts(), name = "Dollar Growth", color = "cornflowerblue") %>%
  hc_navigator(enabled = FALSE) %>%
  hc_scrollbar(enabled = FALSE)

})
```

Next, we use `ggplot()` to create the same visual as above but in the tidy world. The code flow is quite similar to how we would normally create a line chart, except we first need to convert our `xts` object to a tibble with `tk_tbl(preserve_index = TRUE, rename_index = "date")`. Then we make our usual call to `ggplot(aes(x = date))` and add a geom with `geom_line(aes(y = growth), color = "cornflowerblue")`.

Note that the `renderPlot()` function is playing the same role as `renderHighchart()` above - it is alerting the Shiny app that a reactive plot is forthcoming after user inputs, instead of a static plot that is unchanging.

```
renderPlot({
portfolio_growth_xts() %>%
    tk_tbl(preserve_index = TRUE, rename_index = "date") %>%
    ggplot(aes(x = date) +
    geom_line(aes(y = growth), color = "cornflowerblue") +
    ylab("dollars") +
    ggtitle("Growth of Dollar over time")
})
```