HW5-Darshan Patel-3:30-5:30PM

Darshan Patel

11/15/2018

Given the uncertainty about the future performance of financial markets, investors typically diversify their porfolios to improve the quality of their returns. In this assignmentm you will be constructing a portfolio composed of two ETFs (exchange traded funds) that tracks US equity and fixed income markets. The equity ETF tracks the widely followed S&P 500 while the fixed income ETF tracks long-term US Treasury bonds. You will use optimizaion techniques to determine what fraction of your money should be allocated to each asset (i.e., portfolio weights).

The Sharpe ratio is a widely used metric to gauge the quality of portfolio returns. It was developed by Willian F. Sharpe (a Nobel Prize winner) and provides one way to construct an optimal portfolio. This computation allows investors to determine expected returns while taking into account a measure of how risky that investment is. Heuristically, it is computed by taking the expected return of an asset/portfolio, subtracting the risk free rate (i.e., how much you would make if you simply kept your money in the bank and let it collect interest) and dividing by the standard deviation of the portfolio returns. The higher the Sharpe ratio value, the better the investment is considered to be. This formula assumes that returns are normally distributed (not always the best assumption to make with financial data).

The data for the two assets were downloaded from Google Finance (ticker symbols: SPY, TLT) and the federal funds interest rate, representing the risk free rate, was taken from the U.S. Federal Reserve Bank website. We will be looking at weekly returns, as opposed to daily returns, as they are less correlated with each other; monthly returns would be even less correlated, however, we would need a much longer time period of data to have enough observations to do a robust analysis. Note that if we used a different time period to compute our optimal rates, or we used daily instead of weekly returns, we may get very different results.

Question 1

Upload the data in "asset_data.txt" into R and call the tibble `data.x`. The columns are defined as follows: `date`, `close.spy` is the closing price for the S&P 500 ETF, `close.tlt` is the closing price for the long term treasury bond ETF and `fed.rate` is the federal funds interest rate in `percent` form. Look at the data type for the column `date`; just like the data types numeric, logical, etc., R has one for date/time data. **Extract only the observations where the federal funds rate is available (so you are left with weekly date); this is the data you will use for the rest of the analysis.** What is the start date and end date of this reduced data set?

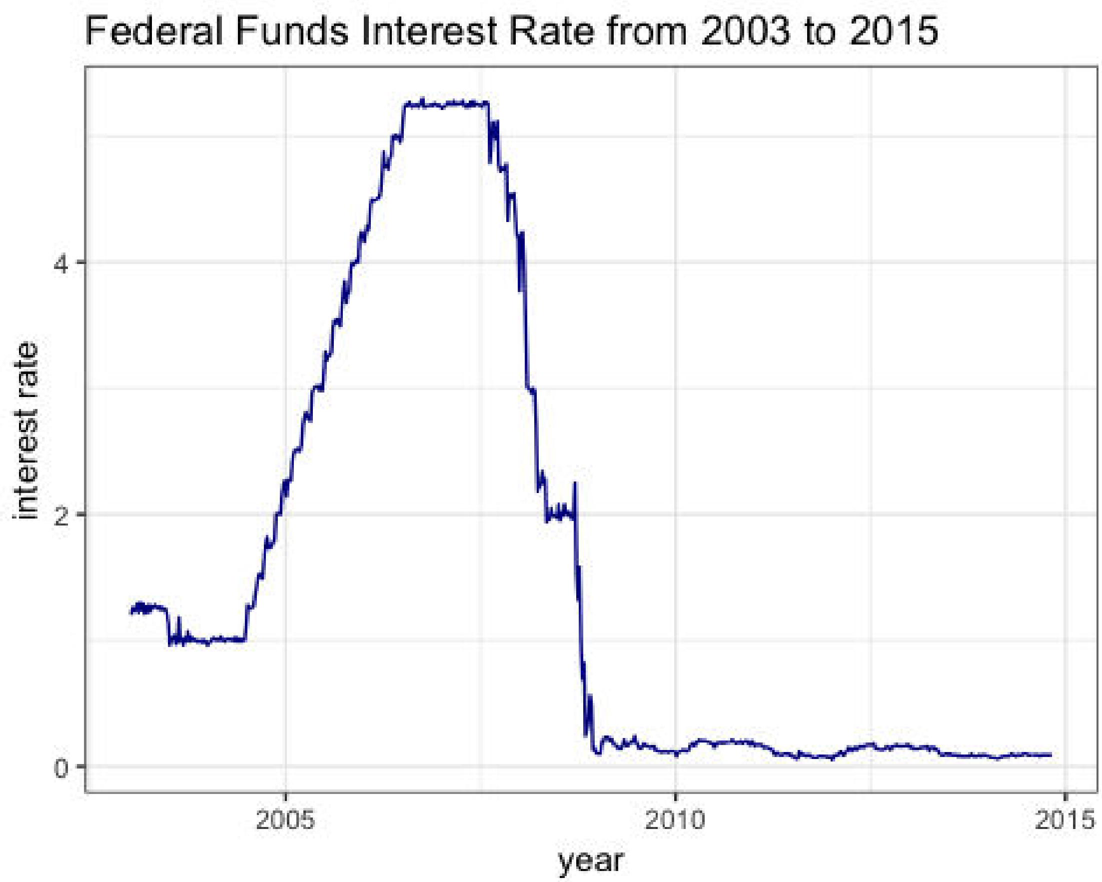
Answer:

```r
# Import tidyverse
library(tidyverse)

## ── Attaching packages ──────────────────────────────────────────────────────────────────────────────────────────────────────────── tidyverse 1.2.1 ──

## ✔ ggplot2 2.2.1     ✔ purrr   0.2.5
## ✔ tibble  1.4.2     ✔ dplyr   0.7.7
## ✔ tidyr   0.8.2     ✔ stringr 1.3.0
## ✔ readr   1.1.1     ✔ forcats 0.3.0

## Warning: package 'tidyr' was built under R version 3.4.4

## Warning: package 'purrr' was built under R version 3.4.4

## Warning: package 'dplyr' was built under R version 3.4.4

## ── Conflicts ──────────────────────────────────────────────────────────────────────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()

# Read in file
data.x <- read_delim("asset_data.txt", delim=",", col_names=TRUE)

## Parsed with column specification:
## cols(
##   date = col_date(format = ""),
##   close.spy = col_double(),
##   close.tlt = col_double(),
##   fed.rate = col_double()
## )

# Retain only the observations where federal funds rate is available
data <- data.x %>% subset(!is.na(fed.rate))

# Print the start and end of this dataset
rbind(head(data, 1), tail(data, 1))

## # A tibble: 2 x 4
##   date       close.spy close.tlt fed.rate
##   <date>         <dbl>     <dbl>    <dbl>
## 1 2003-01-08      91.4      87.0   1.20
## 2 2014-10-29     198.      119.    0.0900
```

The start date of this reduced data set is January 8th 2003 and the end date is October 29th 2014.

Graph the federal funds interest rate as a time series. Describe what you see in the plot and relate it briefly to the most recent financial crisis.

```
# Time series plot of federal funds interest rate
data %>% ggplot(aes(x = date, y = fed.rate)) + geom_line(color = 'darkblue')
+ ggtitle("Federal Funds Interest Rate from 2003 to 2015 ") + labs(x =
'year', y = 'interest rate') + theme_bw()
```



From 2004 to 2007, the federal funds interest rate is increasing. After 2007, the interest rate started dropping until the end of 2008 and stayed relatively steady up to 2015. This is consistent with the financial crisis of 2007 - 2008. In 2007, house prices in the US started to fall. With the housing market suffering, trusts also began to fall apart. Finally in September 2008, Lehman Brothers, a global bank, collapsed and filed for bankrupty. This can be seen in the above plot where interest rates fell below 1% in the latter half of 2008.

(Source: https://www.economist.com/schools-brief/2013/09/07/crash-course)

Question 2

Now we will split the data into training and test sets. The training data will be used to commpute our portfolio weights and our test set will be used to evaluate our portfolio. Make two separate tibbles: (a) the training set should contain all observations before 2014 and (b) the test set should contain all observations in 2014. (Install and load the R package `lubridate` and use the function `year` to extract the year from the `date` column of your data set `data.x`.) How many observations are in each subset?

Answer:

```r
# Import lubridate
library(lubridate)

## Warning: package 'lubridate' was built under R version 3.4.4

##
## Attaching package: 'lubridate'

## The following object is masked from 'package:base':
##
##     date

# Crate training and testing data based on the year of the date
train <- data %>% subset(year(date) < "2014")
test <- data %>% subset(year(date) == "2014")

# Print the number of observations in the training and testing data
paste("There are",nrow(train),"observations before 2014 and", nrow(test),
"observations in 2014", sep = ' ')

## [1] "There are 570 observations before 2014 and 43 observations in 2014"
```

Question 3

The federal funds interest rate is in percent form so convert it to decimal (i.e., fractional) form. Then, for the S&P 500 and long term treasury bonds ETF assets, compute the returns using the following formula:

$$r_t = \frac{p_t - p_{t-1}}{p_{t-1}}$$

where r_t is the return at time t, p_t is the asset price at time t and p_{t-1} is the asset price at time $t-1$ (i.e., the previous period). Add both sets of returns to your `training set` tibble. These returns are also called total returns.

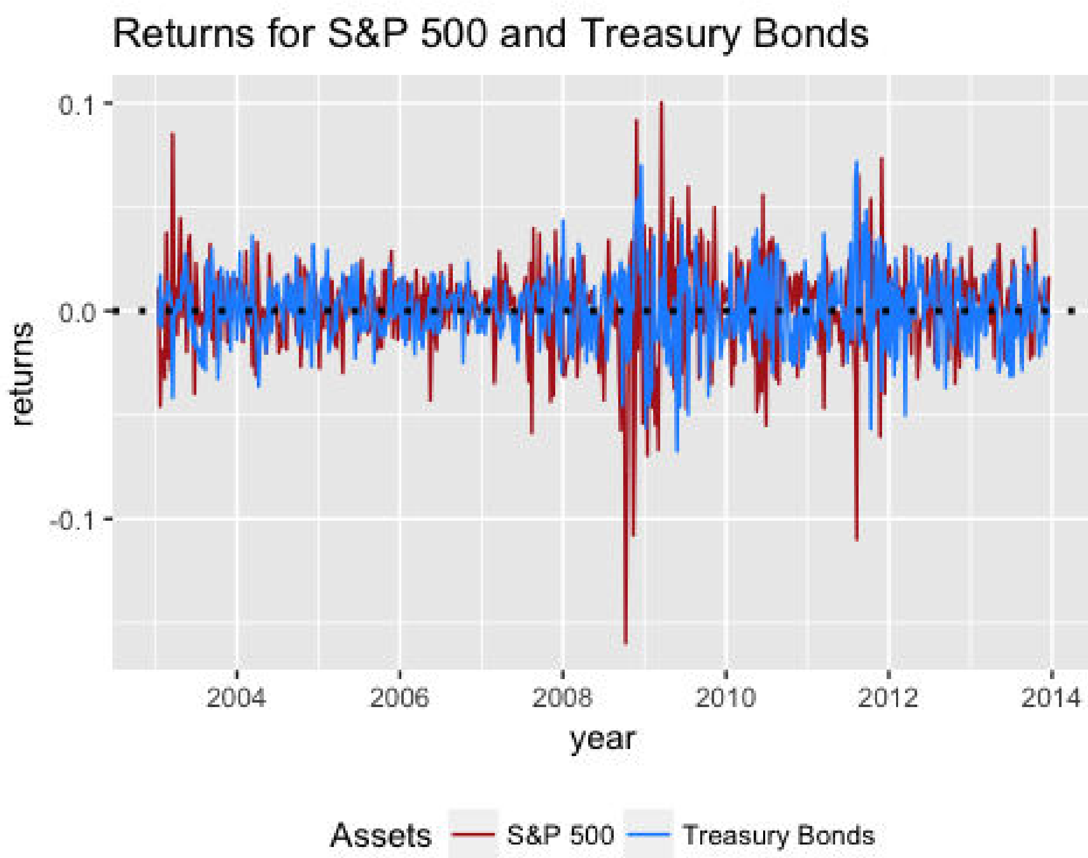
Answer:

```r
# Convert the federal funds interest rate to decimal form
train$fed.rate <- train$fed.rate / 100
```

```r
# Create vectors to store the total returns for both assets
tr_sp_train <- c(NA)
tr_tb_train <- c(NA)

# Iterate over all records to obtain the returns for the S&P 500 assets
for(i in 1:length(train$close.spy)){
  new <- train$close.spy[i]
  prev <-train$close.spy[i-1]
  tr <- (new - prev) / prev
  tr_sp_train <- c(tr_sp_train, tr)
}

# Iterate over all records to obtain the returns for the treasury bonds
assets
for(j in 1:length(train$close.tlt)){
  new <- train$close.tlt[j]
  prev <- train$close.tlt[j-1]
  tr <- (new - prev) / prev
  tr_tb_train <- c(tr_tb_train, tr)
}

# Add the total returns column for each asset to the training dat
train = train %>% mutate("total_returns_sp" = tr_sp_train)
train = train %>% mutate("total_returns_tb" = tr_tb_train)
```

Construct a single time series plot with the returns for both assets plotted. Add a dotted, horizontal line at $y = 0$ to the plot. Compare the two returns series. What do you see?

```r
# Time series plot for total returns of both assets
train %>% select(date, total_returns_sp, total_returns_tb) %>%
  gather(key = Assets, value = total_returns, -date) %>%
  ggplot(aes(x = date, y = total_returns)) + geom_line(aes(color = Assets)) +
  ggtitle("Returns for S&P 500 and Treasury Bonds") +
  labs(x = "year", y = "returns") +
  scale_color_manual(labels = c("S&P 500", "Treasury Bonds"), values =
c("firebrick", "dodgerblue")) +
  geom_hline(yintercept = 0, linetype = "dotted", size = 1) +
  theme(legend.position="bottom")

## Warning: Removed 2 rows containing missing values (geom_path).
```



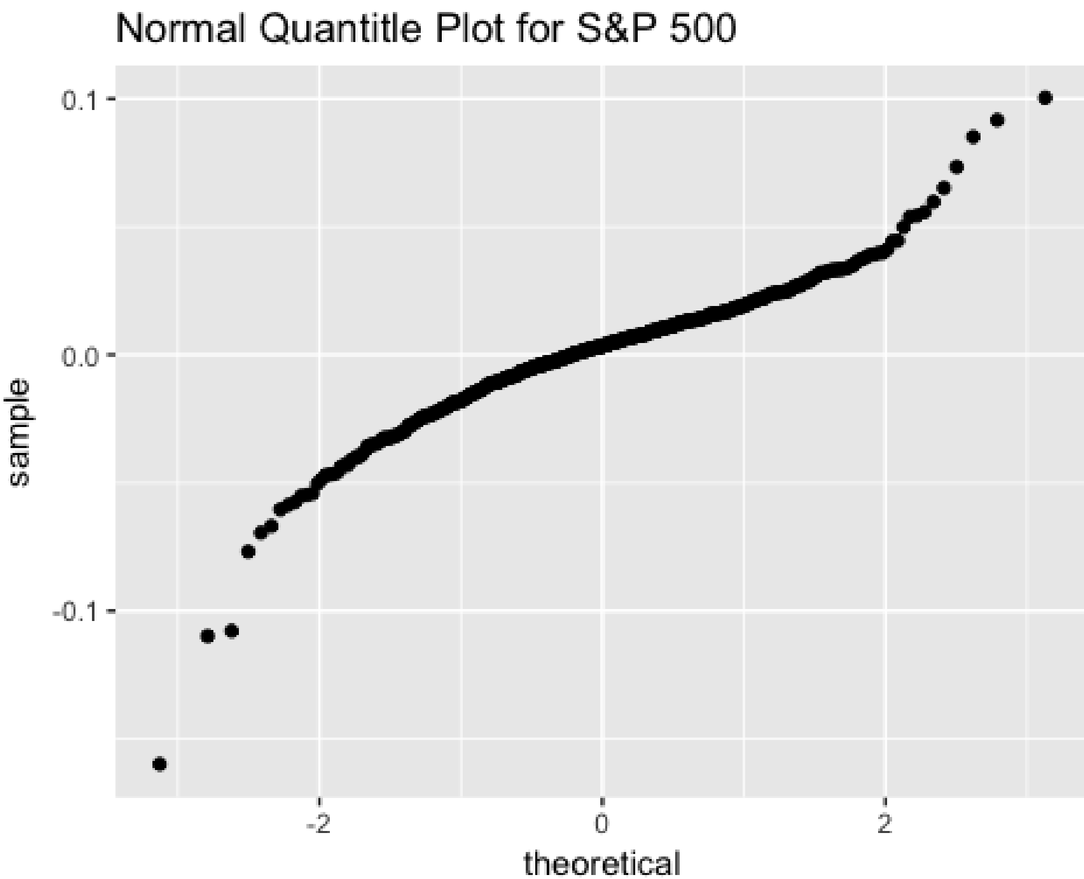
The two return series appear to be near each other for most of the time periods. The S&P 500 asset is more likely to have an extreme return than the treasury bonds asset.

Question 4

The Sharpe ratio calculation assumes that returns are normally distributed. Construct two normal quantile plots, one for `training set` returns of each asset. Is this assumption satisfied? Justify your answer.

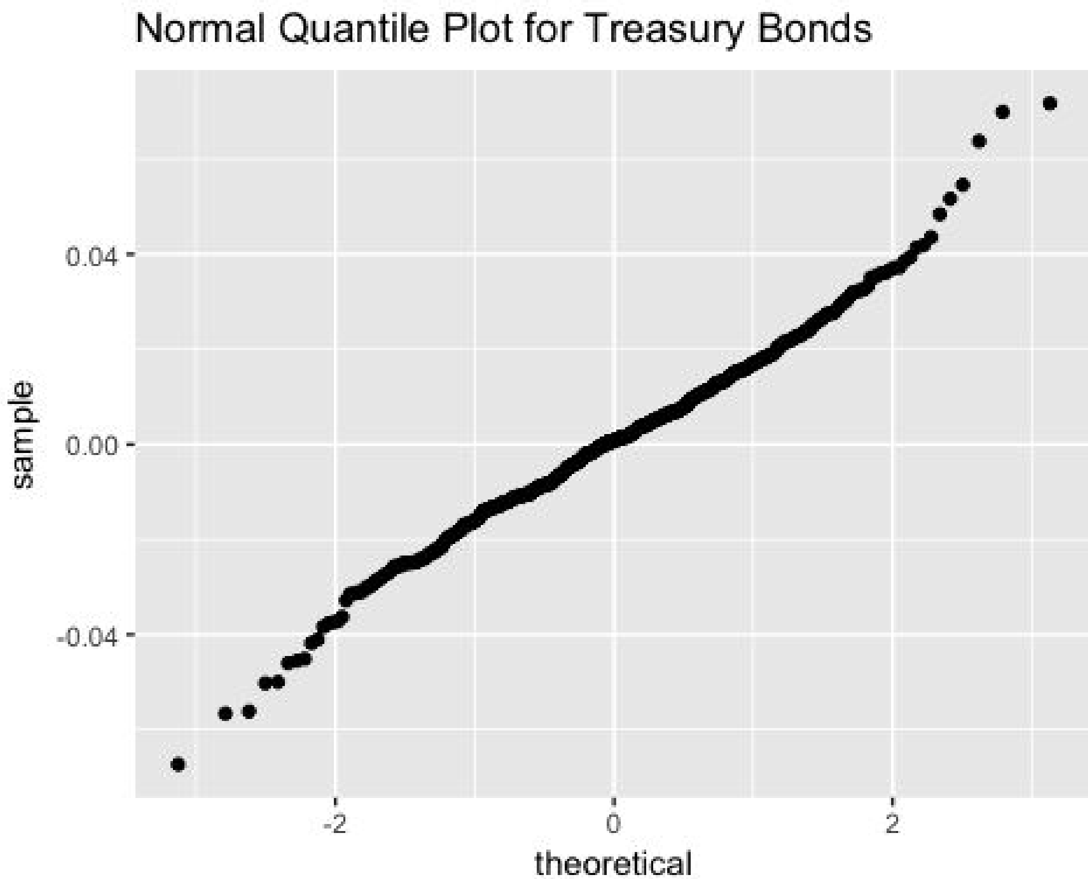
Answer:

```
# Plot quantile plot for the S&P 500 asset
train %>% ggplot(aes(sample = total_returns_sp)) + stat_qq() +
ggtitle("Normal Quantitle Plot for S&P 500")

## Warning: Removed 1 rows containing non-finite values (stat_qq).
```



The distribution for the S&P 500 returns does not appear to look normal since the normal quantile plot does not appear in a straight line.

```
# Plot quantile plot for the treasury bonds asset
train %>% ggplot(aes(sample = total_returns_tb)) + stat_qq() +
ggtitle("Normal Quantile Plot for Treasury Bonds")

## Warning: Removed 1 rows containing non-finite values (stat_qq).
```



The distribution for the Treasury bonds appears to look normally distributed since the normal quantile plot is somewhat straight (compared to the one for the S&P 500 asset). It can be concluded that the assumptions are only half satisfied, for the treasury bonds only but not for S&P 500.

Question 5

Compute the correlation between the S&P 500 and long term treasury bond returns in the `training` set and interpret it.

Answer:

```
# Compute correlation between total returns for both assets
cor(train$total_returns_sp, train$total_returns_tb, method = "pearson", use =
"complete.obs")

## [1] -0.3439013
```

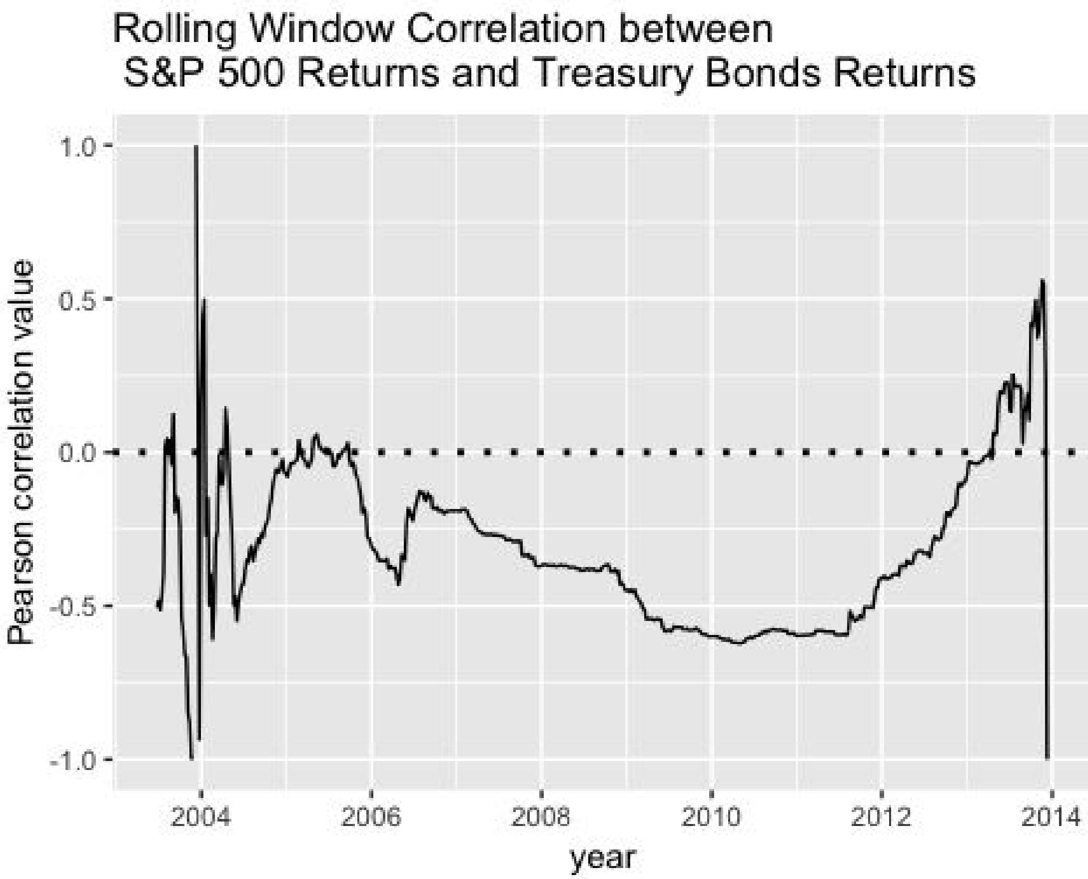
The correlation between the S&P 500 and long term treasury bond returns is -0.3439013. This is a negative correlation meaning when one increases, the other decreases. The absolute value of the correlation is closer to 0 rather than 1 meaning there is high amount of nonlinearality between both returns.

Now, we will compute a rolling-window correlation as follows: compute the correlation between the two asset returns only using the first 24 weeks of data (i.e., weeks 2 to 25), next compute the correlation between the two asset returns for data from week 3 through 26, then week 4 through 27, and so forth.

```
# Create a vector to store rollling window correlation values
rw_corr <- c()

# Note down the number of 24 weeks in the dataset
n_rw <- length(train$date) - 24

# For each size of 24 weeks, compute the rolling window correlation
for(w in 1:n_rw){
  rw_corr <- c(rw_corr, cor(train$total_returns_sp[w:24+w],
train$total_returns_tb[w:24+w],
                           method = "pearson", use = "complete.obs"))
}

# Create a tibble to store the rolling window correlations along with the
last day of the window as its date
rw_dates_corr <- slice(train, 25:n()) %>% select(date) %>% mutate(corr =
rw_corr)

## Warning: package 'bindrcpp' was built under R version 3.4.4
```

Once you compute the rolling-window correlations, make a time series plot of the rolling-window correlation with each point plotted on the *last* day of the window. Add a horizontal, dotted, gray line at 0 to your plot. Is the correlation or rolling-window correlation a better way to describe the relationship between these two assets? Justify your answer.

```
# Plot the time series plot of the rolling window correlation
rw_dates_corr %>% ggplot(aes(x = date, y = corr)) +
  geom_line() +
  ggtitle("Rolling Window Correlation between \n S&P 500 Returns and Treasury
Bonds Returns") +
  labs(x = "year", y = "Pearson correlation value") +
  geom_hline(yintercept = 0, linetype = "dotted", size = 1)

## Warning: Removed 1 rows containing missing values (geom_path).
```



The rolling window correlation does not provide any new information to the correlation calculated above. In the time series plot, it is clear that correlation between the two assets stayed negative through most of the time intervals as well as in the range between -0.5 and 0. Only in a small number of time intervals does the correlation come outside that range.

Question 6

Compute the Sharpe ratios for each asset on the `training set` as follows: Step 0: Let r_t be the return and y_t be the federal funds interest rate for week $t = 1, \dots, T$. Step 1: Compute the excess returns, e_t, for each week in the data set:

$$e_t = r_t - \frac{y_{t-1}}{52}$$

Excess returns are returns that you earn in excess to the risk free rate. Step 2: Convert the excess returns into an excess returns index, g_t:

$$\begin{aligned} g_1 &= 100 \\ g_t &= g_{t-1} \times (1 + e_t) \end{aligned}$$

Step 3: Compute the number of years of data, n , by taking the number of weeks for which you have *returns* (i.e., number of observations in your training set minus 1) and dividing by

52 (since there are 52 weeks in a year); therefore the number of years of data can be a fractional amount. Step 4: Compute the compounded annual growth rate, CAGR:

$$CAGR = \left(\frac{g_T}{g_1}\right)^{\frac{1}{n}} - 1$$

Step 5: Compute the annualized volatility, $\mathscr{v}$:

$$\mathscr{v} = \sqrt{52}SD[e_t]$$

where $SD[e_t]$ is the standard deviation of the excess returns. Step 6: Compute the Sharpe Ratio, SR, which is the ratio of the compounded annual growth rate and the annualized volatility:

$$SR = \frac{CAGR}{\mathscr{v}}$$

Which asset is a better investment? Justify your answer. Answer:

```r
# Create a vector to store the excess returns for both assets
e_t_sp <- c()
e_t_tb <- c()

# Compute the excess returns
for(t in 1:length(train$date)){
  e_t_sp <- c(e_t_sp, train$total_returns_sp[t] - (train$fed.rate[t-1]/52))
  e_t_tb <- c(e_t_tb, train$total_returns_tb[t] - (train$fed.rate[t-1]/52))
}

# Initiate the excess returns index at 100 for both assets
e_r_index_sp <- c(100)
e_r_index_tb <- c(100)

# Compute the excess returns index for each exress return
for(i in 1:length(e_t_sp)){
  e_r_index_sp <- c(e_r_index_sp, tail(e_r_index_sp, 1) * (1 + e_t_sp[i]))
  e_r_index_tb <- c(e_r_index_tb, tail(e_r_index_tb, 1) * (1 + e_t_tb[i]))
}

# Find the number of years of data
n <- (nrow(train) - 1) / 52

# Compute CAGR for both assets
CAGR_sp <- (tail(e_r_index_sp, 1) / head(e_r_index_sp, 1))^(1/n) - 1
CAGR_tb <- (tail(e_r_index_tb, 1) / head(e_r_index_tb, 1))^(1/n) - 1

# Compute the annualized volatility for both assets
v_sp <- sqrt(52) * sd(e_t_sp, na.rm = TRUE)
v_tb <- sqrt(52) * sd(e_t_sp, na.rm = TRUE)
```

```r
# Compute the Sharpe ratio for both assets
sharpe_sp <- CAGR_sp / v_sp
sharpe_tb <- CAGR_tb / v_tb

# Print the Sharpe ratio for both assets
paste("The Sharpe ratio for the S&P 500 asset is", round(sharpe_sp, 2),
      "while the Sharpe ratio for the treasury bonds asset is",
round(sharpe_tb, 2), '.', sep = ' ')

## [1] "The Sharpe ratio for the S&P 500 asset is 0.28 while the Sharpe ratio
for the treasury bonds asset is -0.01 ."
```

The S&P asset is a better investment because it has the higher Sharpe ratio between itself and the treasury bonds. A higher Sharpe ratio indicates a better investment.

Question 7

Write a function which takes the following inputs: (a) a vector of portfolio weights (call this argument x; weights are between 0 and 1), (b) a vector of returns for asset 1, (c) a vector of returns for asset 2 and (d) a vector of the corresponding weekly federal funds interest rates. The function will then do the following: for each weight value in your vector x, you will compute the Sharpe ratio for the corresponding portfolio. To obtain the returns for the portfolio, use the following equation:

$$r_{t,\text{portfolio}} = (x \times r_{t,\text{S\&P 500}}) + ((1 - x) \times r_{t,\text{treasury}})$$

That is, x proportion of the funds will be invested in the S&P 500 ETF and $(1 - x)$ proportion of the funds will be invested into the treasury bond ETF. After you compute the returns for the portfolio, apply the steps in question 6 to get the Sharpe ratio for that portfolio. Your function should output a vector of Sharpe ratios, one for each portfolio weight in x.

Answer:

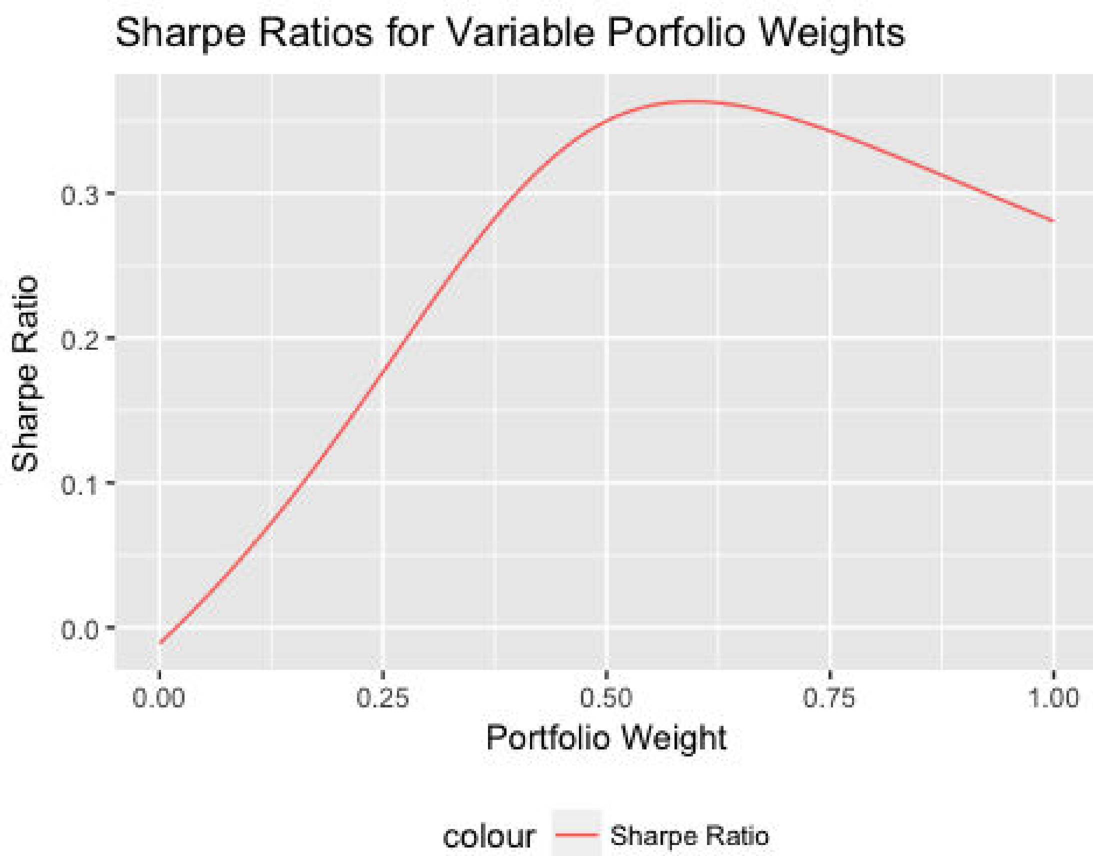
```r
# A function to compute a number of Sharpe ratios for a portfolio given
portfolio weights, returns
# for two assets and the corresponding weekly fed funds interest rates
sharpe_ratio = function(x, returns_asset1, returns_asset2, ff_ir){

  # Store all Sharpe Ratios into a vector
  sr <- c()

  # Iterate over all weights given
  for(i in x){

    # Create a vector to store the returns
    r_port <- c()

    # For each of the federal interest rates, compute the return for the
```

```
portfolio
    for(j in 1:length(ff_ir)){
      r_port <- c(r_port, ((i * returns_asset1[j]) + ((1-i) *
returns_asset2[j])))
    }

    # Create a vector to store excess returns
    e_t <- c()

    # Find the excess returns for each return
    for(t in 1:length(ff_ir)){
      e_t <- c(e_t, r_port[t] - (ff_ir[t-1]/52))
    }

    # Initiate the excess returns index at 100
    e_r_index <- c(100)

    # For all of the returns, find the excess returns index
    for(k in 1:length(e_t)){
      e_r_index <- c(e_r_index, tail(e_r_index, 1) * (1 + e_t[k]))
    }

    # Take note of the number of years of data
    n <- (length(ff_ir) - 1)/52

    # Compute CAGR, annualized volatility and the Sharpe ratio
    CAGR <- (tail(e_r_index, 1) / head(e_r_index, 1))^(1/n) - 1
    v <- sqrt(52) * sd(e_t, na.rm = TRUE)
    sharpe_rat = CAGR / v

    # Add the Sharpe Ratio to the vector that will
    # be returned back to the user
    sr = c(sr, sharpe_rat)
  }

  # Return the vector of Sharpe Ratios
  return(sr)
}
```

Use `stat.function()` to plot the function you just wrote. Weights between 0 and 1 should be on the x-axis and the Sharpe ratio should be on the y-axis. The `training set` data should be used as the input (b), (c) and (d) above.

```
# Import tidyverse
library(tidyverse)

# Plot the Sharpe Ratio function with weights from 0 and 1 and
# using te assets and interest rates from the training set
ggplot(data.frame(x = c(0,1)), aes(x)) +
```

```
  stat_function(aes(color = "Sharpe Ratio"),
                fun = sharpe_ratio,
                args=list(returns_asset1 = train$total_returns_sp,
                          returns_asset2 = train$total_returns_tb,
                          ff_ir = train$fed.rate)) +
  ggtitle("Sharpe Ratios for Variable Porfolio Weights") +
  labs(x = "Portfolio Weight", y = "Sharpe Ratio") +
  theme(legend.position="bottom")
```



Do you see a portfolio weight that produces the maximum Sharpe ratio?

Yes. There seems to be a weight that produces the maximum Sharpe ratio.

Question 8

Using the `training set`, use `optimize()` to determine the optimum weight for each asset using the function above; how much of the funds should be allocated to each asset? What is the Sharpe ratio of the overall portfolio? According to your analysis, is it better to invest in S&P 500 only, long term treasury bonds only, or your combined portfolio? Justify your answer.

Answer:

```r
# Define the function Sharpe ratio so that it only depends on a variable x,
the weight
f <- function(x){return(sharpe_ratio(x, train$total_returns_sp,
train$total_returns_tb, train$fed.rate))}

# Optimize the function to find the optimal portfolio weight
optimize(f, lower = 0, upper = 1, maximum = TRUE)

## $maximum
## [1] 0.5958502
##
## $objective
## [1] 0.3634139
```

According to the optimization, 59.58% should be alloted to S&P 500 and the remaining should go to the treasury bonds. The Sharpe ratio of the overall portfolio is 0.3634, which is definitely positive and high. Thus the portfolio is a good investment. According to the analysis, this optimized allocation produces a higher Sharpe ratio than just investing in one over the other. The Sharpe ratio for investing in just S&P 500 or treasury bonds is only 0.28 and -0.01 respectively. The combined portfolio performed better. Therefore it is better to invest in the combined portfolio.

Question 9

For the remainder of this assignment, we will be evaluating our portfolio using the `test set` data. We will be comparing three strategies: investing only in the S&P 500, investing only in long term treasury bonds and investing in the combined portfolio (computed above). In your `test set` convert the federal funds interest rate from percent to decimal form and compute the returns series for each of the three assets. Next, compute the excess returns index for each asset in the test set.

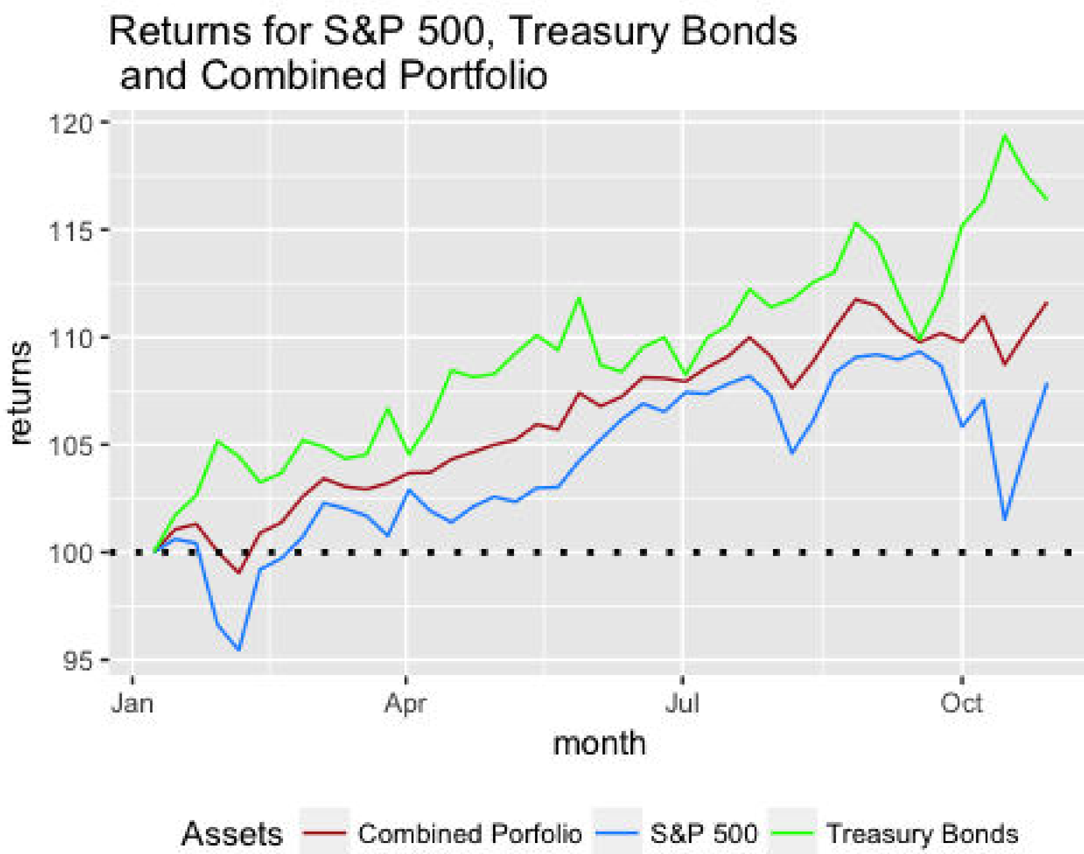
Answer:

```r
# Convert the federal rates into decimal form
test$fed.rate <- test$fed.rate / 100

# Create vectors to store total returns for both assets
tr_sp_test <- c(NA)
tr_tb_test <- c(NA)

# Find the total returns for both assets
for(i in 1:length(test$date)){
  new <- test$close.spy[i]
  prev <- test$close.spy[i-1]
  tr <- (new - prev) / prev
  tr_sp_test <- c(tr_sp_test, tr)

  new <- test$close.tlt[i]
  prev <- test$close.tlt[i-1]
```

```
  tr <- (new - prev) / prev
  tr_tb_test <- c(tr_tb_test, tr)
}

# Create the optimal portfolio returns
optimal_port_tr <- (0.5958 * tr_sp_test) + ((1 - 0.5958) * tr_tb_test)

# Create vectors to store excess returns
e_r_sp_test <- c()
e_r_tb_test <- c()
e_r_combined_test <- c()

# Compute the excess returns for all assets and portfolio
for(t in 1:length(test$date)){
  e_r_sp_test <- c(e_r_sp_test, tr_sp_test[t] - (test$fed.rate[t-1]/52))
  e_r_tb_test <- c(e_r_tb_test, tr_tb_test[t] - (test$fed.rate[t-1]/52))
  e_r_combined_test <- c(e_r_combined_test, optimal_port_tr[t] -
(test$fed.rate[t-1]/52))
}

# Instantiate the exess returns index at 100 for all 3 assets and portfolio
e_r_index_sp <- c(100)
e_r_index_tb <- c(100)
e_r_index_combined <- c(100)

# Compute the excess return index for all 3 assets and portfolio
for(i in 1:length(test$date)){
  e_r_index_sp <- c(e_r_index_sp, tail(e_r_index_sp, 1) * (1 +
e_r_sp_test[i]))
  e_r_index_tb <- c(e_r_index_tb, tail(e_r_index_tb, 1) * (1 +
e_r_tb_test[i]))
  e_r_index_combined <- c(e_r_index_combined, tail(e_r_index_combined, 1) *
(1 + e_r_combined_test[i]))
}

# Remove the NA value at the beginning of the index vectors
e_r_index_sp <- head(e_r_index_sp, -1)
e_r_index_tb <- head(e_r_index_tb, -1)
e_r_index_combined <- head(e_r_index_combined, -1)

# Create a tibble of the excess retturns index for all of the assets and
portfolio
test_e_r_index <- as.tibble(data.frame("date" = test$date, "total_returns_sp"
= e_r_index_sp,
                            "total_returns_tb" = e_r_index_tb,
"total_returns_combined" = e_r_index_combined))
```

Plot the excess returns index for each asset on the same time series plot. Add a dotted, horizontal line at $y = 100$. Describe what you see.

```
# Plot the excess returns index for all assets and portfolio
test_e_r_index %>% select(date, total_returns_sp, total_returns_tb,
total_returns_combined) %>%
  gather(key = Assets, value = total_returns, -date) %>%
  ggplot(aes(x = date, y = total_returns)) + geom_line(aes(color = Assets)) +
  ggtitle("Returns for S&P 500, Treasury Bonds \n and Combined Portfolio") +
  labs(x = "month", y = "returns") +
  scale_color_manual(labels = c("Combined Porfolio", "S&P 500", "Treasury
Bonds"),
                     values = c("firebrick", "dodgerblue", "green")) +
  geom_hline(yintercept = 100, linetype = "dotted", size = 1) +
  theme(legend.position="bottom")
```



Question 10

The excess returns index can be interpreted as follows: if you invested in $100 in at time $t = 1$, the index value at time T represents how much you have earned in addition to (i.e., in excess of) the risk free interest rate. If you invested $100 in each asset (portfolio, all in long term treasury bonds, or all in S&P 500) in the first week of January 2014, how much would you have at the end of the `test set` period for each asset in addition to risk-free interest rate? Did your portfolio perform well in the test set? Justify your answer.

Answer:

```r
# Print the last excess returns index
tail(test_e_r_index, 1)
```

```
## # A tibble: 1 x 4
##   date       total_returns_sp total_returns_tb total_returns_combined
##   <date>                <dbl>            <dbl>                  <dbl>
## 1 2014-10-29             108.             116.                   112.
```

At the end of the test set time period, there would be $100.813 in S&P 500 asset if S&P 500 was invested alone, $108.7683 in treasury bonds asset if treasury bonds was invested alone and $104.3339 if we invested in an optimal porfolio of both assets. Looking at this, the portfolio did not do so well as compared to the treasury bonds asset returns. The treasury bonds asset returns is about $4 more than the returns from the portfolio. Compared to the original investment of $100, the treasury bonds investment grew by 8.7% while the portfolio only grew by 4.3%. Therefore the portfolio did not do well in the test set.