Stat222_FinalProject_HenryWong

Henry Wong

May 4, 2014

Part I

Stat222 Final Project: Statistical Analysis on Investment Performance of the S&P 500 Equity-Indexed Annuity (EIA)'s Index Strategies

GitHub's Documentation

---

1 Introduction

I have chosen the topic of "Statistical Analysis on Investment Performance of the S&P 500 Equity-Indexed Annuity (EIA)'s Index Strategies" for my Stat222 Statistics Master's Capstone final project at the University of California, Berkeley. I will use the daily closing prices of S&P 500 index from Yahoo Finance as the main dataset. I will use the main dataset to generate other datasets and some useful statistics to produce plots and figures for supporting the key findings. I have proposed four resarch questions in which their correspoinding key findings will answer the main question of "Whose mind are you going to change about what?".

1.1 Reproducible Research

As recommended by the course instructor Professor Victoria Stodden, we will use GitHub to facilitate reproducible computational results and share our research. This IPython notebook (Stat222_FinalProject_HenryWong.ipynb), the required data file (spx.csv), the PDF created from this notebook (Stat222_FinalProject_HenryWong.pdf), and the associated final presentation PowerPoint (Stat222_FinalPresentation_HenryWong.pptx) can be found here:

`https://github.com/henrykmwong/Stat222FinalProject`

Creating PDF

Please follow the following steps to create PDF from this IPython notebook.

1. Open an UNIX terminal
2. Go to the directory in which this IPython notebook is located

3. Run the following UNIX command

```
ipython nbconvert --to latex
--SphinxTransformer.author='Henry Wong'
--post PDF Stat222_FinalProject_HenryWong.ipynb
```

2 Research Questions

We will focus on the folloiwng four research questions in which their corresponding key findings might change the public mind towards investment on EIA products. We will talk about "Whose mind are we going to change (about what)?" in the next paragraph.

1. How do the distribution, the expected return, and the volatility of the investment performances change by increasing the investment holding period from 1-year to 10-year?
2. Is there any particular day of month has higher or lower expected return and volatility than other days of month?
3. Does the statistically analysis provide a good benchmark for measuring opportunity cost and volatility for investing in the EIA index strategies?
4. Can the chosen index strategies of the S&P 500 EIA outperform the S&P 500 index with 10-year holding period?

3 Whose Mind are We Going to Change (about What)?

Our findings for the four research questions will change the following minds regarding the investment of EIA products.

1. EIA Investors
2. EIA Industry
3. Academia
4. Acturaries

3.1 EIA Investors

Our findings for the research quesitons 1, 2, and 4 might change the mind of the EIA investors. Investors always want to know the risk and return of their investments. They also want to know whether the number of years of holding their investments can have any impact on the relationship between risk and return of their investments. Therefore, potential EIA investors will be interested to our findings for the research question 1. In addition, many EIA products can only be issued on certain days of the month. For instance, some EIA products of American National Insurance Company can only be issued on the 1st, 8th, 15th, and 22nd of the month. Therefore, when there are similar EIA products with different sets of issue days available, potential EIA investors have to decide what EIA products should be purchased and what day of month the products should be issued. Therefore, our findings for the research question 2 will facilitate their decision making. Finally, all the potential EIA investors would be interested in our findings for the research question 4 because they would not consider buying EIA products if they have a better alternative by investing on the index on their own.

3.2 EIA Industry

Our findings for the research quesitons 1 and 4 might change the mind of the EIA industry. The EIA industry will be interested to our findings for research question 1, in particular, whether the number of years of holding EIA investments can have any impact on the relationship between risk and return of the EIA investments. The EIA industry always want their clients to invest on their EIA products as long as possible, therefore the EIA industry would want to know whether increasing investment duration would have positive impact on the risk and return of the EIA investments, in turn, the EIA industry can either drive their sales with marketing on the positive impact or add some incentive features to their existing EIA products in response to the negative impact. The EIA industry is also interested to our findings for the research question 4 with the same reason regarding the sustainability of their business models.

3.3 Academia

Our findings for the research quesiton 1 might change the mind of the academia. Scholars in Economics and Finance are always interested in the relationship between risk and return of investment. Also, scholars in Statistics, in addtional to Economics and Finance, will be interested to the distribution of investment performance, in particular, what distribution the investment performance might follow. With the identified distribution, they can estimate the parameters with methods such as MLE and they can estimate its standard error with bootstrap.

3.4 Acturaries

In fact, the actuaries who work in the EIA industry would be interested in our findings for all the research quesitons because they are part of the EIA industry and they might be the EIA investors as well. In particular, our findings for the research quesiton 3 might change their mind towards their pricing and hedging work for the EIA products.

4 S&P 500 Equity-Indexed Annuity (EIA) and its Chosen Index Strategies

4.1 S&P 500 EIA

S&P 500 EIA investors invest an amount of money called premium, in return for protection against down markets and the potential for some investment growth. The credited interest is linked to the S&P 500 index and it depends on investment performance of an index strategy that investors choose at the contract effective date. At each contract anniversary, any interest credited to the EIA is automatically locked in, so it becomes part of EIA's new accumulated value. This means that even if the index value declines in later years, the premium and any interest that has been credited to the EIA's contract cannot be lost to future market downturns.

4.2 S&P 500 EIA Index Strategies

We will focus on the following two popular S&P 500 EIA index strategies and compare their statistics with the ones for the S&P 500.

1. S&P 500 1-Year Gain-Trigger with 5% Annual Rate
2. S&P 500 1-Year Point-to-Point Index Strategy with 8% Annual Cap

S&P 500 1-Year Gain-Trigger with 5% Annual Rate

The formula for computing the annualized return is as follows:

$$Annualized\ Return = I(\frac{Annual\ Ending\ Index\ Value}{Annual\ Beginning\ Index\ Value} > 1) \times 5\%$$

where

- $I()$ is an indictor function

In other words, if the ending S&P 500 index value is greater than the beginning S&P 500 index value, 5% interest rate is credited to the account, otherwise, no interest credited.

S&P 500 1-Year Point-to-Point Index Strategy with 8% Annual Cap

The formula for computing the annualized return is as follows:

$$Annualized\ Return = max(min(\frac{Annual\ Ending\ Index\ Value}{Annual\ Beginning\ Index\ Value} - 1,\ 8\%),\ 0)$$

The interest credited to the annuity is equal to the percentage change in the index, subject to an annual cap rate 8%. For instance, if the index value increases by 12% in a year, 8% would be credited, not 12%. Also, if the index value decreases by 10% in a year, the account value would remain the same.

5 Methodologies

We will use our main dataset for the daily closing prices of the S&P 500 index to compute the Expected Return, the Volatility, and the Shape Ratio of the investments in our interest by different investment duration and day of month.

5.1 Expected Return

Expected Return is defined as follows:

$$Expected\ Return = E(R) = \sum_{i=1}^{m} R_i\ p_i$$

where

- m is the number of different R_i
- R_i is the annualized investment return
- p_i is the probability of an occurrence of R_i such that $\sum_{i=1}^{m} p_i = 1$

With our observation dataset from Yahoo Finance, we can use sample mean of return to estimate expected return. The formula for sample mean of return is as follows:

$$Sample\ Mean\ of\ Return = \bar{R} = \frac{1}{n} \sum_{i=1}^{n} R_i$$

where

- n is the sample size

5.2 Volatility

Volatility is defined as follows:

$$Volatility = \sigma_R = \sqrt{\sum_{i=1}^{m} (R_i - E(R))^2\ p_i}$$

With our observation dataset from Yahoo Finance, we can use sample volatility to estimate volatility. The formula for sample volatility is as follows:

$$Sample\ Volatility = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(R_i - \bar{R})^2}$$

5.3 Sharpe Ratio

Sharpe ratio is defined as follows:

$$Sharpe\ Ratio = \frac{E(R_i - R_f)}{\sqrt{Var(R_i - R_f)}}$$

where

- $E(R_i - R_f)$ is the expected value of the excess of investment return over the risk-free rate (e.g. US 10-year Treasury rate).
- $Var(R_i - R_f)$ is the variance of the excess return.

In other words, Sharpe ratio is the expected return on each unit risk (or volatility) taking, therefore, Shape ratio can be interpreted as price of risk. Risk-averse investors always prefer high Sharpe ratio, whereas risk-neutral and risk-loving investors would prefer high expected return.

6 Import the Required Packages

We need to import the following IPython packages for conducting our statistical analysis.

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.dates as dates
import csv

from pylab import *
from pandas.io.data import DataReader
from datetime import datetime
from __future__ import division

%matplotlib inline
%load_ext ipython_nose
```

7 Data

We will use the daily closing prices of the S&P 500 index from Yahoo Finance as our main dataset. In this section, we will discuss the steps for downloading the main dataset and how we can use the main dataset to generate other datasets and some useful statistics to produce plots and figures for supporting our key findings for the four research questions.

7.1 Download the S&P 500 Data from Yahoo Finance

We use the following IPython codes to download the S&P 500 daily adjusted closing prices from Yahoo Finance. We first need to generate the daily dates for the corresponding S&P 500 daily adjusted closing prices. If the data for the corresponding date is not available, the code will look for the next available data, which is the convention used in the EIA industry. The download will take about 2 hours. After the download, the code will export the download in csv format with file name spx.csv in the same directory where this IPython notebook is located.

```
year = range(1951, 2014)
month = range(1, 13)
day = range(1, 29)

date = []
for i in year:
    for j in month:
        for k in day:
            date.append(datetime(i,j,k))

adj_close = []

for i in year:
    for j in month:
        for k in day:
            temp1 = DataReader("^GSPC",  "yahoo", datetime(i,j,k),
                                       datetime(i+1,j,k))
            temp2 = temp1["Adj Close"][0]
            adj_close.append(temp2)

data = pd.DataFrame({'date': date,
'adj_close': adj_close})
data.to_csv('./spx.csv')
```

Import the Downloaded S&P 500 Data

```
In [2]: spx = pd.read_csv('spx.csv')
```

7.2 Generate Supporting Datasets from the Main Dataset

We will use our main dataset to generate the following supporting datasets.

1. 1-Year Annualized Return of Investment
2. 1-Year Annualized Expected Return and Volatility
3. 10-Year Annualized Return of Investment
4. 10-Year Annualized Expected Return and Volatility
5. 10-year Annualized Expected Return by Day of Month
6. 10-Year Annualized Volatility by Day of Month
7. Sharpe Ratio

1-Year Annualized Return of Investment

With our main dataset for the S&P 500 daily closing prices, we can compute the 1-year annualized return of our chosen EIA's index strategies (Gain-Trigger and Point-to-Point) and the S&P 500 using the aforementioned formulas. The results of our computation will be export to a csv file named t1.csv. The corresponding IPython codes are as follows.

In [3]:
```
date = spx.date[0:(62*12*28)]
day = 62 * 12 * range(1,29)
close0 = spx.adj_close[0:(62*12*28)]
close1 = spx.adj_close[(1*12*28):(63*12*28)]
close1.index = close0.index

n = len(date)

### 1-Year S&P 500
sp1yr = range(0, n)

### 1-Year Gain-Trigger with 5% Annual Rate
gt1yr = range(0, n)

### 1-Year Point-to-Point with 8% Annual Cap
ptp1yr = range(0, n)

for i in range(0, n):
    sp1yr[i] = close1[i] / close0[i] - 1

    if sp1yr[i] > 0:
        gt1yr[i] = 0.05
    else:
        gt1yr[i] = 0

    if sp1yr[i] > 0.08:
        ptp1yr[i] = 0.08
    elif sp1yr[i] > 0:
        ptp1yr[i] = sp1yr[i]
    else:
        ptp1yr[i] = 0

t1data = pd.DataFrame({'date': date,
                       'day': day,
                       'close0': close0,
                       'close1': close1,
                       'gt1yr': gt1yr,
                       'ptp1yr': ptp1yr,
                       'sp1yr': sp1yr
                       })

t1data.to_csv('./t1.csv')
```

1-Year Annualized Expected Return and Volatility

With our supporting dataset for the 1-year annualized return of investment, we can compute the 1-year annualized expected return and volatility of our chosen EIA's index strategies (Gain-Trigger and Point-to-Point) and the S&P 500 using the aforementioned formulas. The results of our computation will be export to a csv file named t2.csv. The corresponding IPython codes are as follows.

In [4]:
```
strategy = range(0,3)
return_mean = range(0,3)
return_sd = range(0,3)

strategy[0:3] = ['gt1yr', 'ptp1yr', 'sp1yr']
return_mean[0:3] = [mean(gt1yr), mean(ptp1yr), mean(sp1yr)]
```

```
return_sd[0:3] = [std(gt1yr), std(ptp1yr), std(sp1yr)]

t2data = pd.DataFrame({'strategy': strategy,
                       'return_mean': return_mean,
                       'return_sd': return_sd
                       })

t2data.to_csv('./t2.csv')
```

10-Year Annualized Return of Investment

With our supporting dataset for the 1-year annualized return of investment, we can compute the 10-year annualized return of our chosen EIA's index strategies (Gain-Trigger and Point-to-Point) and the S&P 500 using the aforementioned formulas. The results of our computation will be export to a csv file named t3.csv. The corresponding IPython codes are as follows.

In [5]:
```
t1 = pd.read_csv('t1.csv')

### Create a function to calculate the annualized 10-year return

def re10yrFun(re1yr, n):
    re1 = re1yr[0:(53*12*28)]
    re2 = re1yr[(1*12*28):(54*12*28)]
    re3 = re1yr[(2*12*28):(55*12*28)]
    re4 = re1yr[(3*12*28):(56*12*28)]
    re5 = re1yr[(4*12*28):(57*12*28)]
    re6 = re1yr[(5*12*28):(58*12*28)]
    re7 = re1yr[(6*12*28):(59*12*28)]
    re8 = re1yr[(7*12*28):(60*12*28)]
    re9 = re1yr[(8*12*28):(61*12*28)]
    re10 = re1yr[(9*12*28):(62*12*28)]

    re2.index = re1.index
    re3.index = re1.index
    re4.index = re1.index
    re5.index = re1.index
    re6.index = re1.index
    re7.index = re1.index
    re8.index = re1.index
    re9.index = re1.index
    re10.index = re1.index

    re10yr = range(0, n)

    for i in range(0, n):
        re10yr[i] = (((1+re1[i])*(1+re2[i])*(1+re3[i])*(1+re4[i])*
                      (1+re5[i])*(1+re6[i])*(1+re7[i])*(1+re8[i])*
                      (1+re9[i])*(1+re10[i]))**(1/10)) - 1

    return re10yr

date = spx.date[0:(53*12*28)]
day = 53 * 12 * range(1,29)
n = len(date)

gt10yr = re10yrFun(t1.gt1yr, n) ### 10-Year Gain-Trigger
ptp10yr = re10yrFun(t1.ptp1yr, n) ### 10-Year Point-to-Point
sp10yr = re10yrFun(t1.sp1yr, n) ### 10-Year S&P 500

t3data = pd.DataFrame({'date': date,
                       'day': day,
                       'gt10yr': gt10yr,
                       'ptp10yr': ptp10yr,
                       'sp10yr': sp10yr
```

```
                             })

t3data.to_csv('./t3.csv')
```

10-Year Annualized Expected Return and Volatility

With our supporting dataset for the 10-year annualized return of investment, we can compute the 10-year annualized expected return and volatility of our chosen EIA's index strategies (Gain-Trigger and Point-to-Point) and the S&P 500 using the aforementioned formulas. The results of our computation will be export to a csv file named t4.csv. The corresponding IPython codes are as follows.

In [6]:
```
strategy = range(0,3)
return_mean = range(0,3)
return_sd = range(0,3)

strategy[0:3] = ['gt10yr', 'ptp10yr', 'sp10yr']
return_mean[0:3] = [mean(gt10yr), mean(ptp10yr), mean(sp10yr)]
return_sd[0:3] = [std(gt10yr), std(ptp10yr), std(sp10yr)]

t4data = pd.DataFrame({'strategy': strategy,
                       'return_mean': return_mean,
                       'return_sd': return_sd
                       })

t4data.to_csv('./t4.csv')
```

10-year Annualized Expected Return by Day of Month

With our supporting dataset for the 10-year annualized return of investment, we can compute the 10-year annualized expected return by day of month of our chosen EIA's index strategies (Gain-Trigger and Point-to-Point) and the S&P 500 using the aforementioned formulas. The results of our computation will be export to a csv file named t5.csv. The corresponding IPython codes are as follows.

In [7]:
```
t3 = pd.read_csv('t3.csv')

### Annualized expected return by day of month
gt10yr_mean = pd.pivot_table(t3, values='gt10yr', rows=['day'],
                             aggfunc='mean')
ptp10yr_mean = pd.pivot_table(t3, values='ptp10yr', rows=['day'],
                              aggfunc='mean')
sp10yr_mean = pd.pivot_table(t3, values='sp10yr', rows=['day'],
                             aggfunc='mean')

t5data = pd.DataFrame({'gt10yr_mean': gt10yr_mean,
                       'ptp10yr_mean': ptp10yr_mean,
                       'sp10yr_mean': sp10yr_mean
                       })

t5data.to_csv('./t5.csv')
```

10-Year Annualized Volatility by Day of Month

With our supporting dataset for the 10-year annualized return of investment, we can compute the 10-year annualized volatility by day of month of our chosen EIA's index strategies (Gain-Trigger and Point-to-Point) and the S&P 500 using the aforementioned formulas. The results of our computation will be export to a csv file named t6.csv. The corresponding IPython codes are as follows.

```
In [8]: gt10yr_sd = pd.pivot_table(t3, values='gt10yr', rows=['day'],
                                   aggfunc='std')
        ptp10yr_sd = pd.pivot_table(t3, values='ptp10yr', rows=['day'],
                                    aggfunc='std')
        sp10yr_sd = pd.pivot_table(t3, values='sp10yr', rows=['day'],
                                   aggfunc='std')

        t6data = pd.DataFrame({'gt10yr_sd': gt10yr_sd,
                               'ptp10yr_sd': ptp10yr_sd,
                               'sp10yr_sd': sp10yr_sd
                               })

        t6data.to_csv('./t6.csv')
```

Sharpe Ratio

With our supporting datasets for the 1-year and 10-year annualized return of investment, we can compute the corresponding Sharpe Ratio of our chosen EIA's index strategies (Gain-Trigger and Point-to-Point) and the S&P 500 using the aforementioned formulas. The results of our computation will be export to a csv file named t7.csv. The corresponding IPython codes are as follows.

```
In [9]: ### Create a list of risk-free rate
        rf = range(0, 31)

        for i in range(0, 31):
            rf[i] = i / 1000

        ### Create a function to the Sharpe Ratio

        def srFun(re, rf, sd):
            n = len(rf)
            sr = range(0, n)

            for i in range(0, n):
                sr[i] = (re - rf[i]) / sd

            return sr

        ### Calculate the Sharpe Ratios for the investment strategies

        gt1yr_sr = srFun(mean(gt1yr), rf, std(gt1yr))
        ptp1yr_sr = srFun(mean(ptp1yr), rf, std(ptp1yr))
        sp1yr_sr = srFun(mean(sp1yr), rf, std(sp1yr))

        gt10yr_sr = srFun(mean(gt10yr), rf, std(gt10yr))
        ptp10yr_sr = srFun(mean(ptp10yr), rf, std(ptp10yr))
        sp10yr_sr = srFun(mean(sp10yr), rf, std(sp10yr))

        t7data = pd.DataFrame({'rf': rf,
                               'gt1yr_sr': gt1yr_sr,
                               'ptp1yr_sr': ptp1yr_sr,
                               'sp1yr_sr': sp1yr_sr,
                               'gt10yr_sr': gt10yr_sr,
                               'ptp10yr_sr': ptp10yr_sr,
                               'sp10yr_sr': sp10yr_sr
                               })

        t7data.to_csv('./t7.csv')
```

7.3 Import the Supporting Datasets

In [10]:
```
t1 = pd.read_csv('t1.csv')
t2 = pd.read_csv('t2.csv')
t3 = pd.read_csv('t3.csv')
t4 = pd.read_csv('t4.csv')
t5 = pd.read_csv('t5.csv')
t6 = pd.read_csv('t6.csv')
t7 = pd.read_csv('t7.csv')
```

8 Nose Test

We want to make sure that we have not made any careless coding mistake for downloading the S&P 500 data from Yahoo Finance and generating data from the downloaded data. Therefore, we define the following 13 nose test functions to examine whether our datasets contain any data point that is not well-defined, for instance, negative S&P 500 index value.

1. Test for non-negative beginning closing prices
2. Test for non-negative ending closing prices
3. Test for non-negative 1-year gain-trigger returns
4. Test for non-negative 1-year point-to-point returns
5. Test for non-negative standard deviation of 1-year returns
6. Test for non-negative annualized 10-year gain-trigger returns
7. Test for non-negative annualized 10-year point-to-point returns
8. Test for non-negative standard deviation of 10-year returns
9. Test for non-negative mean of annualized 10-year gain-trigger returns by day of month
10. Test for non-negative mean of annualized 10-year point-to-point returns by day of month
11. Test for non-negative standard deviation of annualized 10-year gain-trigger returns by day of month
12. Test for non-negative standard deviation of annualized 10-year point-to-point returns by day of month
13. Test for non-negative standard deviation of annualized 10-year S&P 500 returns by day of month

8.1 Define Functions for Nose Tests

In [11]:
```
### Test 1: Test for non-negative beginning closing prices
def test_close0():
    assert min(t1.close0) >= 0

### Test 2: Test for non-negative ending closing prices
def test_close1():
    assert min(t1.close1) >= 0

### Test 3: Test for non-negative 1-year gain-trigger returns
def test_gt1yr():
    assert min(t1.gt1yr) >= 0

### Test 4: Test for non-negative 1-year point-to-point returns
def test_ptp1yr():
    assert min(t1.ptp1yr) >= 0

### Test 5: Test for non-negative
###         standard deviation of 1-year returns
def test_return_sd_1yr():
    assert min(t2.return_sd) >= 0

### Test 6: Test for non-negative
###         annualized 10-year gain-trigger returns
def test_gt10yr():
    assert min(t3.gt10yr) >= 0
```

```
### Test 7: Test for non-negative
###         annualized 10-year point-to-point returns
def test_ptp10yr():
    assert min(t3.ptp10yr) >= 0

### Test 8: Test for non-negative
###         standard deviation of 10-year returns
def test_return_sd_10yr():
    assert min(t4.return_sd) >= 0

### Test 9: Test for non-negative mean of
###         annualized 10-year gain-trigger returns by day of month
def test_gt10yr_mean():
    assert min(t5.gt10yr_mean) >= 0

### Test 10: Test for non-negative mean of
###          annualized 10-year point-to-point returns by day of month
def test_ptp10yr_mean():
    assert min(t5.ptp10yr_mean) >= 0

### Test 11: Test for non-negative standard deviation of
###          annualized 10-year gain-trigger returns by day of month
def test_gt10yr_sd():
    assert min(t6.gt10yr_sd) >= 0

### Test 12: Test for non-negative standard deviation of
###          annualized 10-year point-to-point returns by day of month
def test_ptp10yr_sd():
    assert min(t6.ptp10yr_sd) >= 0

### Test 13: Test for non-negative standard deviation of
###          annualized 10-year S&P 500 returns by day of month
def test_sp10yr_sd():
    assert min(t6.sp10yr_sd) >= 0
```

8.2 Run the Nose Tests

We use the following IPython code to execute the 13 nose test functions we have defined. As shown in the output, all the 13 tests passed. Therefore, we can use the supporting datasets we have generated to produce the figures that help answering the four research questions we have proposed.

In [12]:
```
%nose
```

Out [12]:

```
13/13 tests passed
```

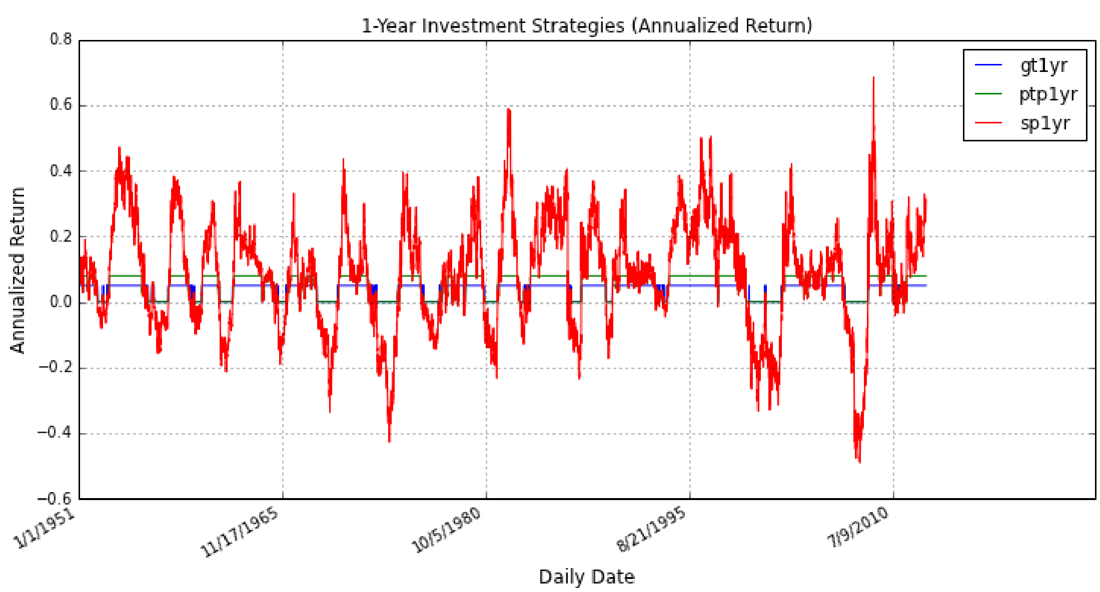
9 Analysis on Risk and Return of Investment by Duration

9.1 1-Year Annualized Return of Investment

With the supporting dataset (t1.csv) for the 1-year annualized return of investment, we can produce a plot of the 1-year annualized return of our chosen EIA's index strategies (Gain-Trigger and Point-to-Point) and the S&P 500 versus the beginning daily trading day of 1-year holding period with the following IPython codes.

In [13]:

```
f1 = t1[['gt1yr', 'ptp1yr', 'sp1yr', 'date']].set_index('date')
f1.plot(title = '1-Year Investment Strategies (Annualized Return)',
        figsize=(12,6))
plt.gcf().autofmt_xdate()
plt.xlabel('Daily Date', fontsize=12)
plt.ylabel('Annualized Return', fontsize=12)
plt.savefig('f1_HenryWong.png')
```



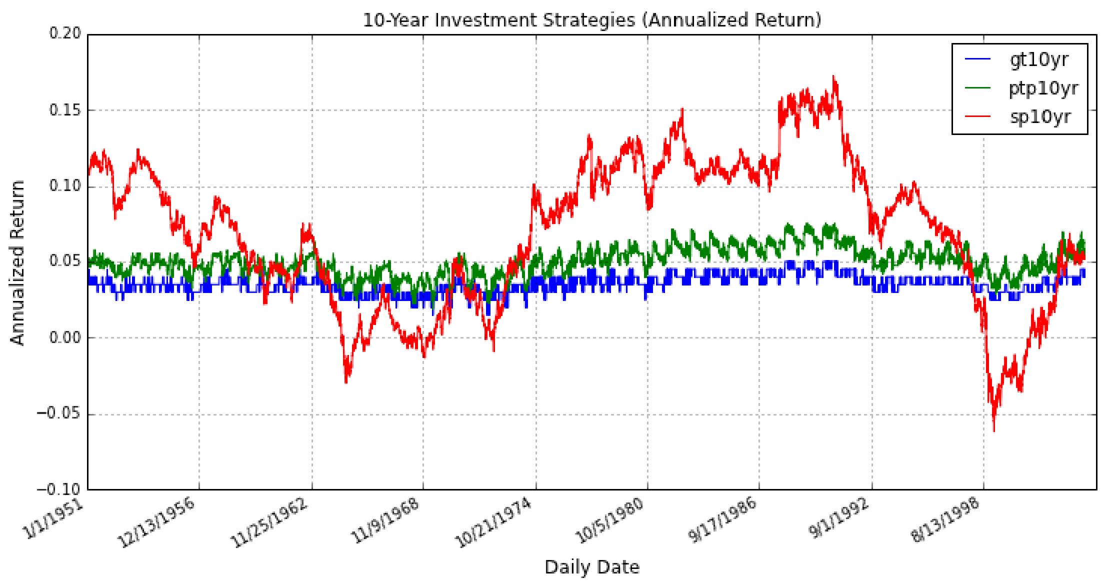
Interpretation and Key Finding

As shown in the plot, there are many ups and downs in the time series for the S&P 500 1-year annualized return, whereas the times series for the EIA index strategies are stable over time.

9.2 10-Year Annualized Return of Investment

With the supporting dataset (t3.csv) for the 10-year annualized return of investment, we can produce a plot of the 10-year annualized return of our chosen EIA's index strategies (Gain-Trigger and Point-to-Point) and the S&P 500 versus the beginning daily trading day 10-year holding period with the following IPython codes.

In [14]:
```
f2 = t3[['gt10yr', 'ptp10yr', 'sp10yr', 'date']].set_index('date')
f2.plot(title = '10-Year Investment Strategies (Annualized Return)',
        figsize=(12,6))
plt.gcf().autofmt_xdate()
plt.xlabel('Daily Date', fontsize=12)
plt.ylabel('Annualized Return', fontsize=12)
plt.savefig('f2_HenryWong.png')
```



Interpretation and Key Finding

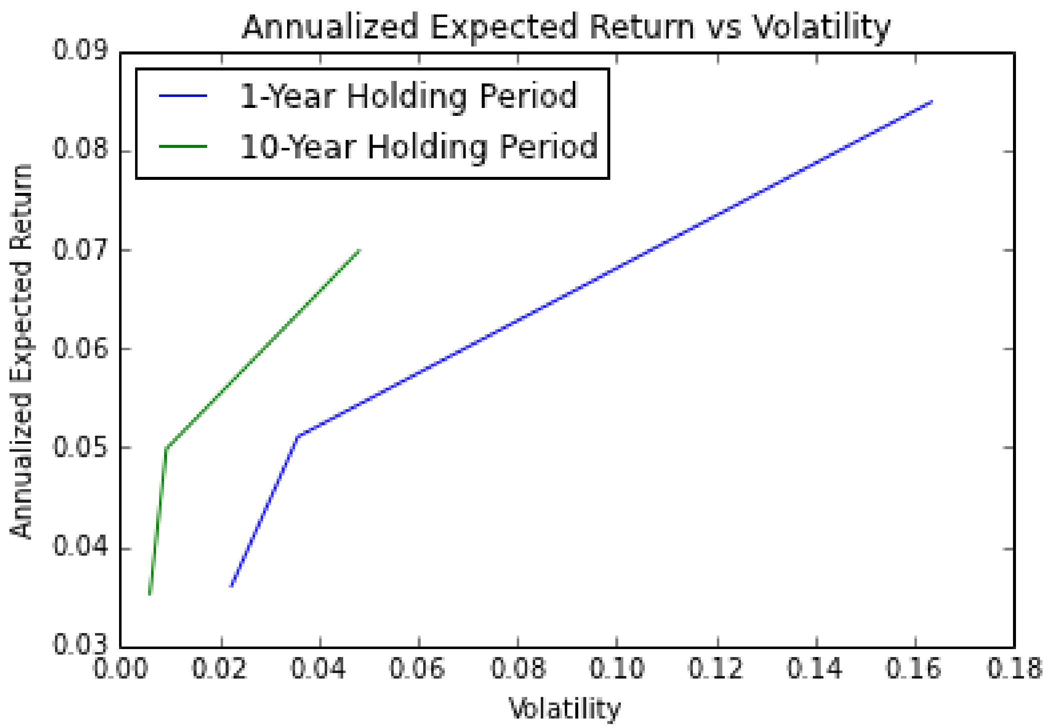
As shown in the plot, the time series for the S&P 500 annualized return is much less volatile when we increase the investment duration from 1-year to 10-year. There is no dramatic change in the times series for the EIA index strategies.

9.3 Annualized Expected Return vs Volatility

With the supporting datasets (t2.csv and t4.csv) for the annualized expected return and volatility, we can produce a plot of the annualized expected return of our chosen EIA's index strategies (Gain-Trigger and Point-to-Point) and the S&P 500 versus their corresponding volatility by investment duration with the following IPython codes.

In [15]:
```
fig, ax = plt.subplots()

ax.plot(t2.return_sd, t2.return_mean, label="1-Year Holding Period")
ax.plot(t4.return_sd, t4.return_mean, label="10-Year Holding Period")
ax.set_xlabel('Volatility')
ax.set_ylabel('Annualized Expected Return')
ax.set_title('Annualized Expected Return vs Volatility')
ax.legend(loc=2); # upper left corner
plt.savefig('f3_HenryWong.png')
```



Interpretation and Key Finding

As shown in the plot, when we increase investment duration, the annualized expected return and the volatility of our investment choices decrease. In addition, the decrease in the volatility is greater than the decrease in the expected return.

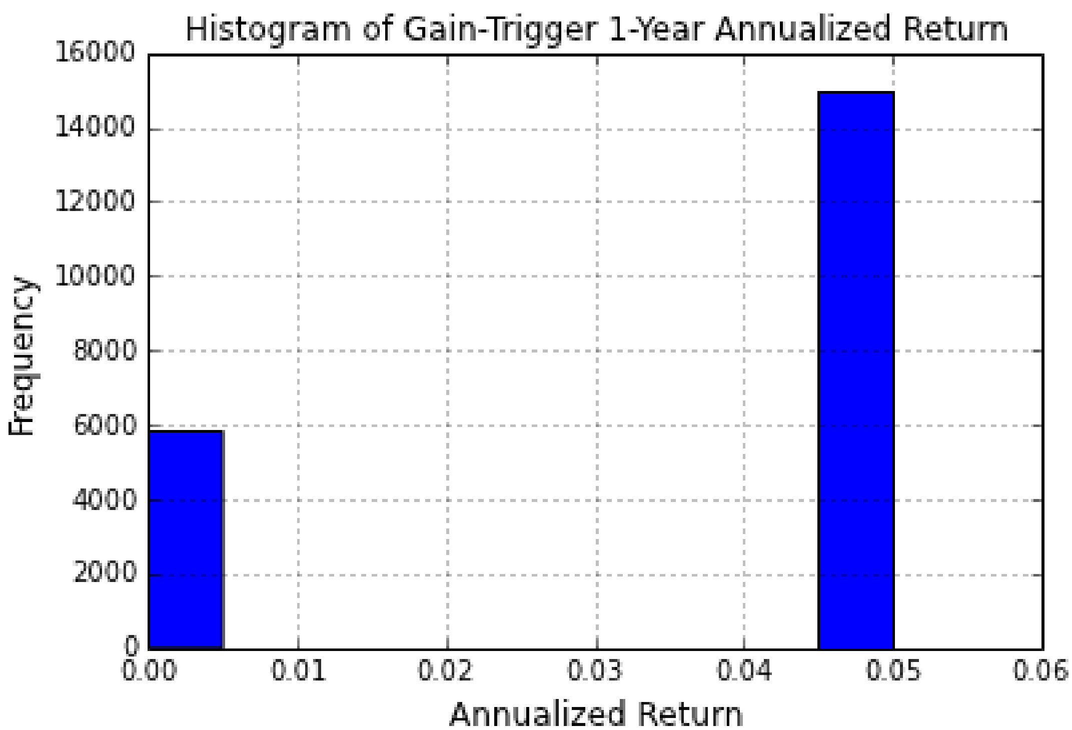
9.4 Histogram of 1-Year Annualized Return

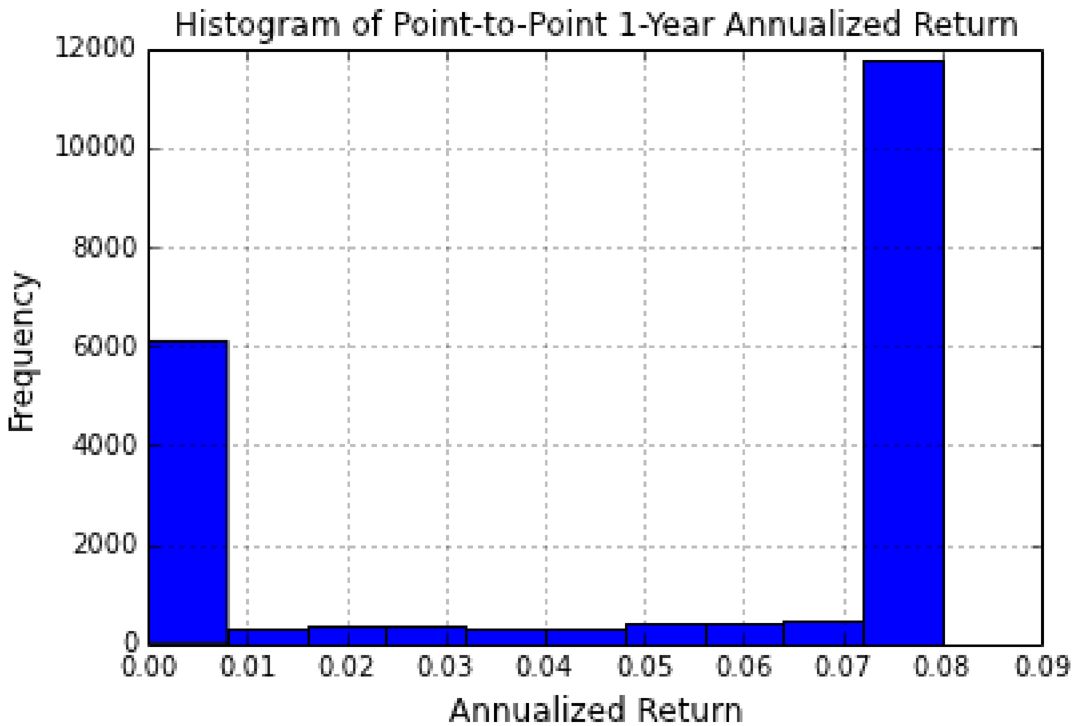
With the supporting dataset (t1.csv) for the 1-year annualized return of investment, we can produce histograms of the 1-year annualized return of our chosen EIA's index strategies (Gain-Trigger and Point-to-Point) and the S&P 500 with the following IPython codes.

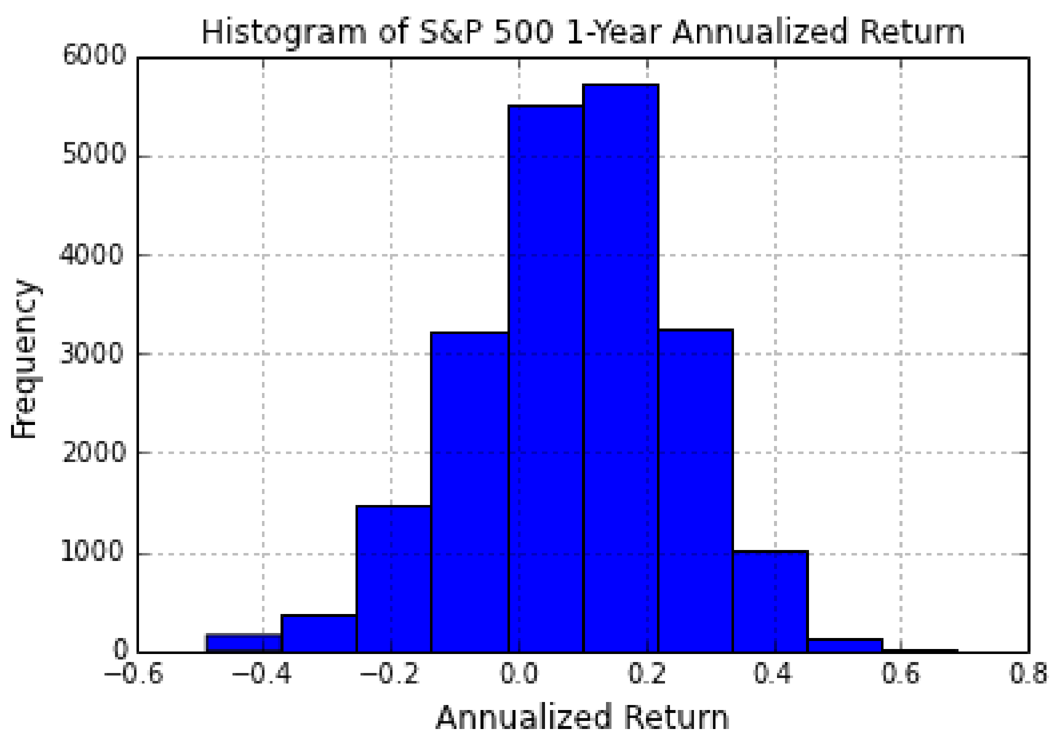
In [16]:
```
### Histogram of Gain-Trigger 1-Year Annualized Return
t1[['gt1yr', ]].hist()
plt.xlabel('Annualized Return', fontsize=12)
plt.ylabel('Frequency', fontsize=12)
plt.title('Histogram of Gain-Trigger 1-Year Annualized Return',
          fontsize=12)
plt.savefig('f4a_HenryWong.png')
```

```
### Histogram of Point-to-Point 1-Year Annualized Return
t1[['ptp1yr', ]].hist()
plt.xlabel('Annualized Return', fontsize=12)
plt.ylabel('Frequency', fontsize=12)
plt.title('Histogram of Point-to-Point 1-Year Annualized Return',
          fontsize=12)
plt.savefig('f4b_HenryWong.png')

### Histogram of S&P 500 1-Year Annualized Return
t1[['sp1yr', ]].hist()
plt.xlabel('Annualized Return', fontsize=12)
plt.ylabel('Frequency', fontsize=12)
plt.title('Histogram of S&P 500 1-Year Annualized Return',
          fontsize=12)
plt.savefig('f4c_HenryWong.png')
```




Histogram of Point-to-Point 1-Year Annualized Return
Frequency
Annualized Return
12000
10000
8000
6000
4000
2000
0
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09


Histogram of S&P 500 1-Year Annualized Return
Frequency
Annualized Return
6000
5000
4000
3000
2000
1000
0
−0.6
−0.4
−0.2
0.0
0.2
0.4
0.6
0.8

Interpretation and Key Finding

As shown in the histograms, the 1-year annualized return of our chosen EIA's index strategies (Gain-Trigger and Point-to-Point) are bimodal distributed, whereas the 1-year annualized return of the S&P 500 is bell-shaped and symmetrically distributed.

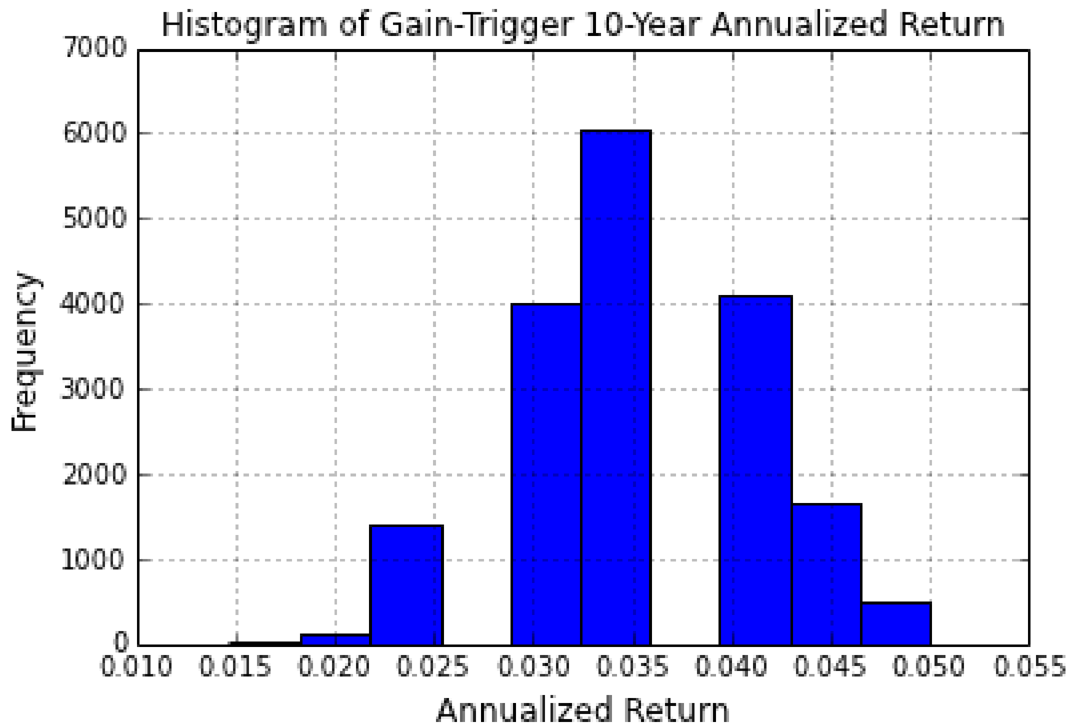
9.5 Histogram of 10-Year Annualized Return

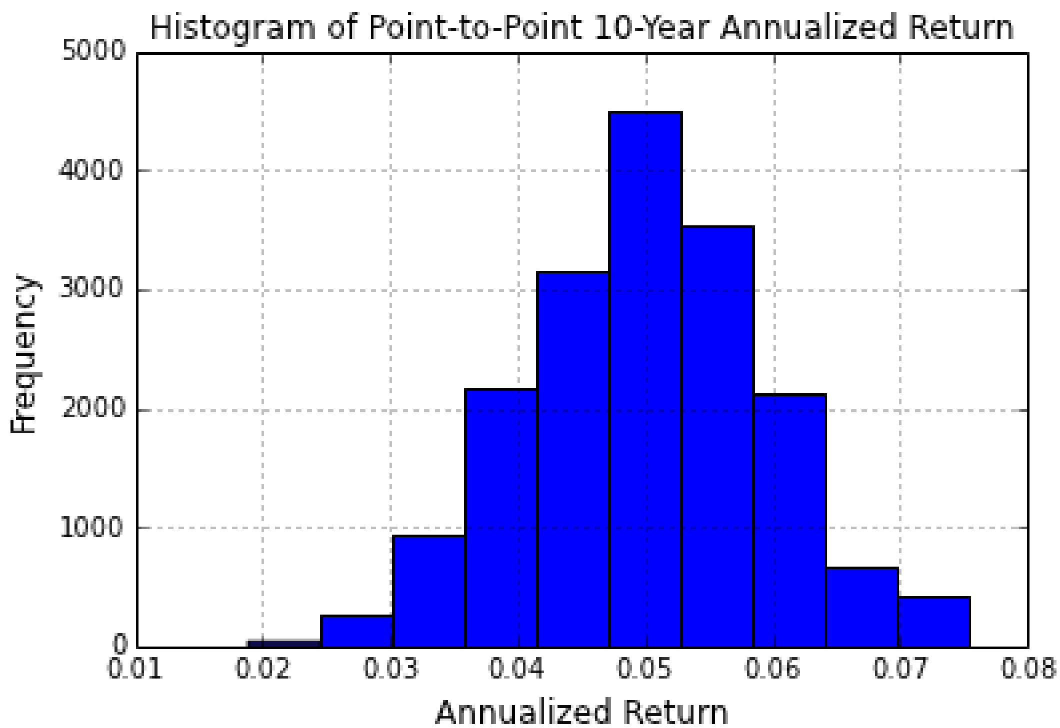
With the supporting dataset (t3.csv) for the 10-year annualized return of investment, we can produce histograms of the 10-year annualized return of our chosen EIA's index strategies (Gain-Trigger and Point-to-Point) and the S&P 500 with the following IPython codes.

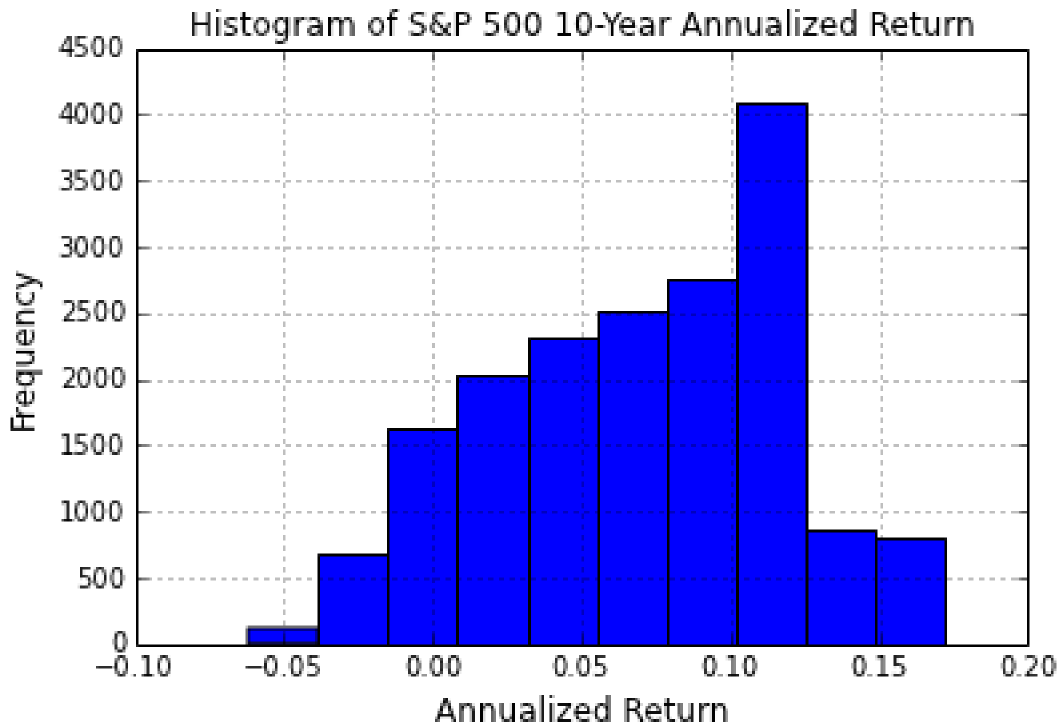
In [17]:
```
### Histogram of Gain-Trigger 10-Year Annualized Return
t3[['gt10yr', ]].hist()
plt.xlabel('Annualized Return', fontsize=12)
plt.ylabel('Frequency', fontsize=12)
plt.title('Histogram of Gain-Trigger 10-Year Annualized Return',
          fontsize=12)
plt.savefig('f5a_HenryWong.png')

### Histogram of Point-to-Point 10-Year Annualized Return
t3[['ptp10yr', ]].hist()
plt.xlabel('Annualized Return', fontsize=12)
plt.ylabel('Frequency', fontsize=12)
plt.title('Histogram of Point-to-Point 10-Year Annualized Return',
          fontsize=12)
plt.savefig('f5b_HenryWong.png')

### Histogram of S&P 500 10-Year Annualized Return
t3[['sp10yr', ]].hist()
plt.xlabel('Annualized Return', fontsize=12)
plt.ylabel('Frequency', fontsize=12)
plt.title('Histogram of S&P 500 10-Year Annualized Return',
          fontsize=12)
plt.savefig('f5c_HenryWong.png')
```


Histogram of Gain-Trigger 10-Year Annualized Return
Frequency
Annualized Return
0
1000
2000
3000
4000
5000
6000
7000
0.010
0.015
0.020
0.025
0.030
0.035
0.040
0.045
0.050
0.055


Histogram of Point-to-Point 10-Year Annualized Return
Frequency
Annualized Return
0
1000
2000
3000
4000
5000
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08



Interpretation and Key Finding

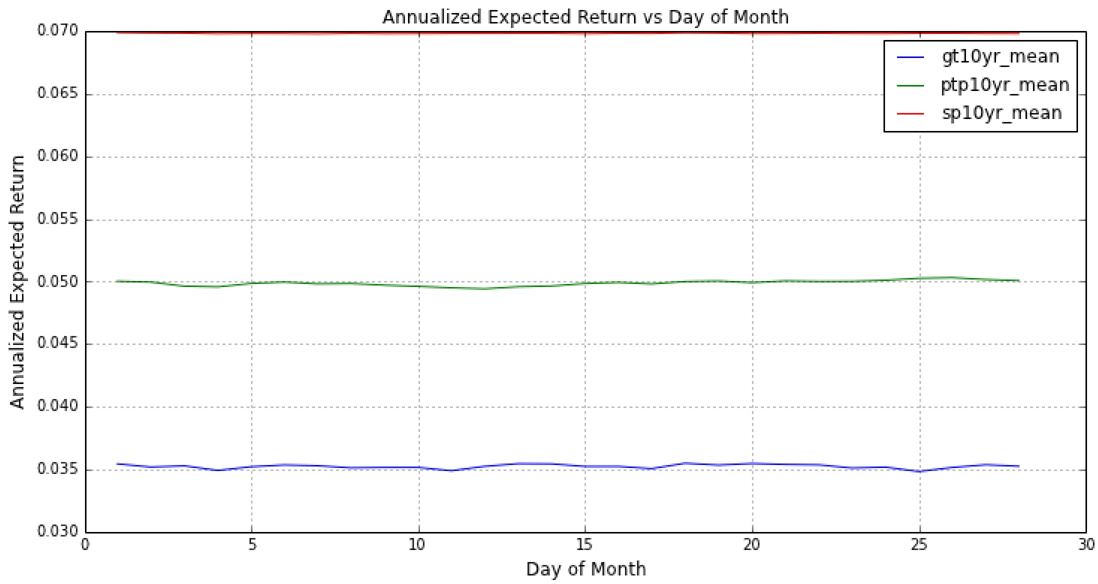
As shown in the histograms, when we increase the investment duration from 1-year to 10-year, the annualized return of our chosen EIA's index strategies (Gain-Trigger and Point-to-Point) are more bell-shaped and symmetrically distributed, whereas the corresponding distribution for the S&P 500 is more negatively-skewed.

10 Analysis on Risk and Return of Investment by Day of Month

10.1 Annualized Expected Return vs Day of Month

With the supporting dataset (t5.csv) for the 10-year annualized expected return by day of month, we can produce a plot of the annualized expected return of our chosen EIA's index strategies (Gain-Trigger and Point-to-Point) and the S&P 500 versus day of month with the following IPython codes.

In [18]:
```
f6 = t5[['gt10yr_mean', 'ptp10yr_mean',
          'sp10yr_mean', 'day']].set_index('day')
f6.plot(title = 'Annualized Expected Return vs Day of Month',
        figsize=(12,6))
plt.xlabel('Day of Month', fontsize=12)
plt.ylabel('Annualized Expected Return', fontsize=12)
plt.savefig('f6_HenryWong.png')
```



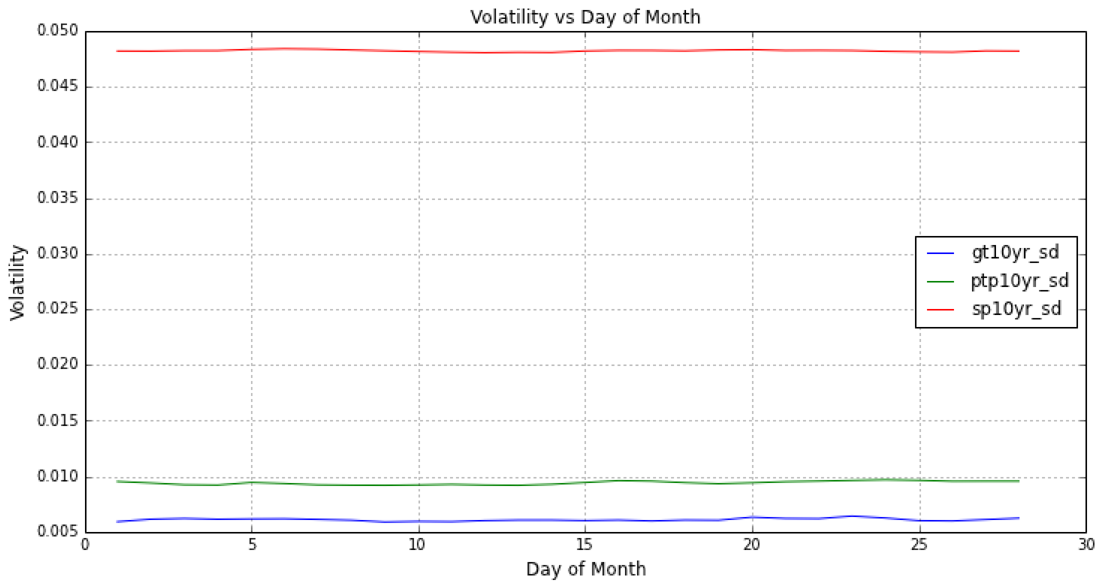
Interpretation and Key Finding

As shown in the plot, the annualized expected return are indifferent with day of month. On average, there are 3.5% and 2% of annual opportunity costs for investing in the Gain-Trigger and Point-to-Point strategy respectively.

10.2 Volatility vs Day of Month

With the supporting dataset (t6.csv) for the 10-year annualized volatility by day of month, we can produce a plot of the annualized volatility of our chosen EIA's index strategies (Gain-Trigger and Point-to-Point) and the S&P 500 versus day of month with the following IPython codes.

In [19]:
```
f7 = t6[['gt10yr_sd', 'ptp10yr_sd',
         'sp10yr_sd', 'day']].set_index('day')
f7.plot(title = 'Volatility vs Day of Month', figsize=(12,6))
plt.xlabel('Day of Month', fontsize=12)
plt.ylabel('Volatility', fontsize=12)
plt.savefig('f7_HenryWong.png')
```



Interpretation and Key Finding

As shown in the plot, the annualized volatility are indifferent with day of month. On average, the annual volatilities for investing in the Gain-Trigger and Point-to-Point are 4.2% and 3.8% respectively lower than one for the S&P 500.

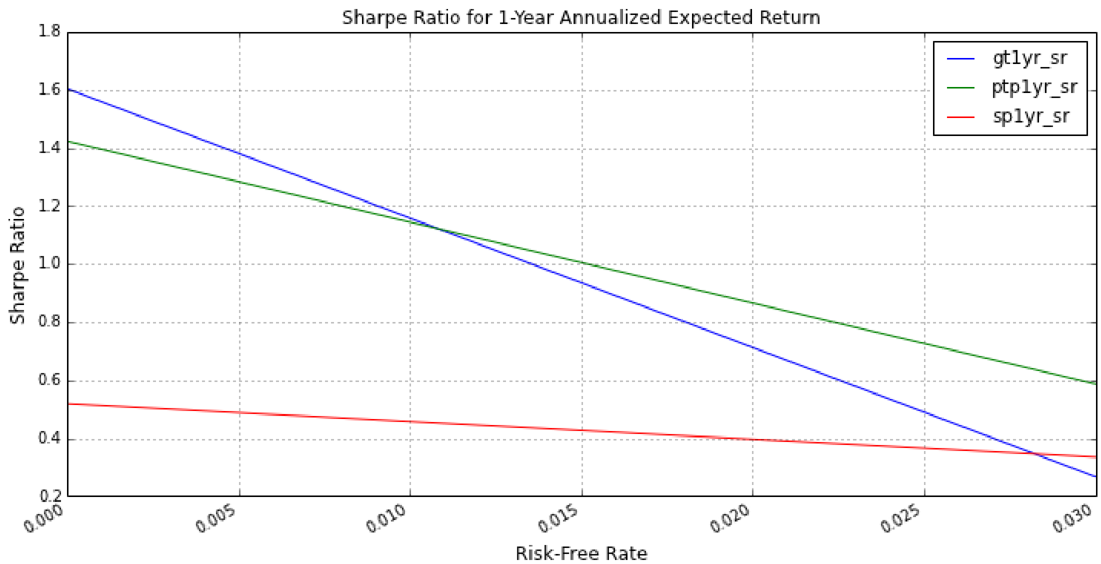
11 Analysis on Sharpe Ratio

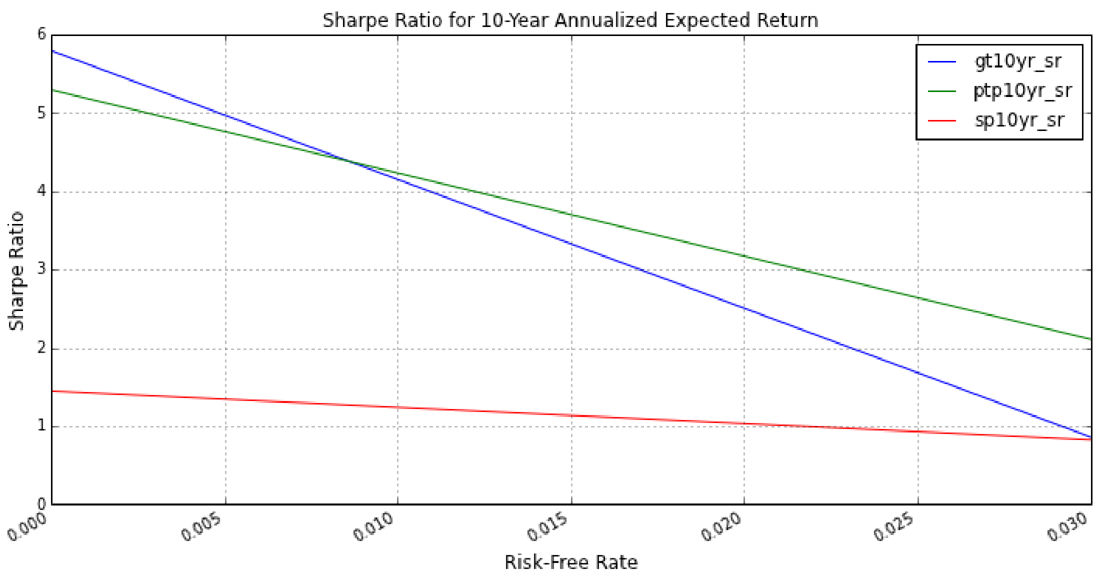
11.1 Sharpe Ratio by Investment Duration

With the supporting dataset (t7.csv) for Shape ratio, we can produce plots of Shape ratio of our chosen EIA's index strategies (Gain-Trigger and Point-to-Point) and S&P 500 by investment duration with the following IPython codes.

In [20]:
```
### Sharpe Ratio for 1-Year Annualized Expected Return
f8a = t7[['gt1yr_sr', 'ptp1yr_sr', 'sp1yr_sr', 'rf']].set_index('rf')
f8a.plot(title = 'Sharpe Ratio for 1-Year Annualized Expected Return',
        figsize=(12,6))
plt.gcf().autofmt_xdate()
plt.xlabel('Risk-Free Rate', fontsize=12)
plt.ylabel('Sharpe Ratio', fontsize=12)
plt.savefig('f8a_HenryWong.png')

### Sharpe Ratio for 10-Year Annualized Expected Return
f8b = t7[['gt10yr_sr', 'ptp10yr_sr', 'sp10yr_sr', 'rf']].set_index('rf')
f8b.plot(title = 'Sharpe Ratio for 10-Year Annualized Expected Return',
        figsize=(12,6))
plt.gcf().autofmt_xdate()
plt.xlabel('Risk-Free Rate', fontsize=12)
plt.ylabel('Sharpe Ratio', fontsize=12)
plt.savefig('f8b_HenryWong.png')
```





Interpretation and Key Finding

As shown in the plots, when annual risk-free rate is less than 3%, our chosen EIA's index strategies (Gain-Trigger and Point-to-Point) have higher Sharpe ratios than the S&P 500. Also, when annual risk-free rate is less than 1%, Gain-Trigger has higher Sharpe ratios than Point-to-Point and the relationship are reverse otherwise. Note that Sharpe ratio can be considered as a function of risk-free rate.

11.2 Sharpe Ratio by Investment Strategy

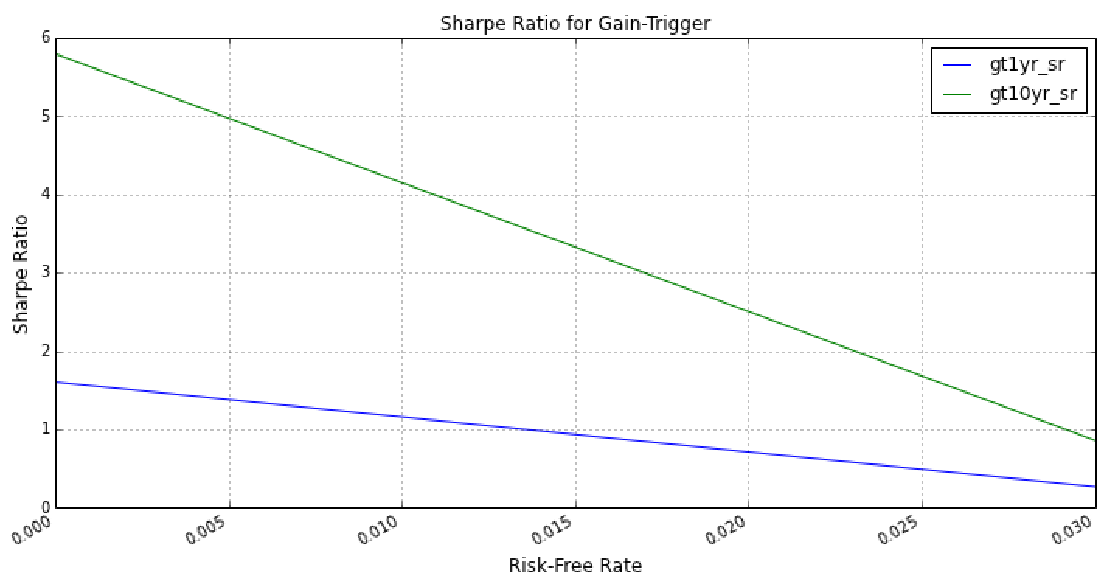
With the supporting dataset (t7.csv) for Shape ratio, we can produce three plots of Shape ratio, each corresponds to our chosen EIA's index strategies (Gain-Trigger and Point-to-Point) and the S&P 500, with the following IPython codes.

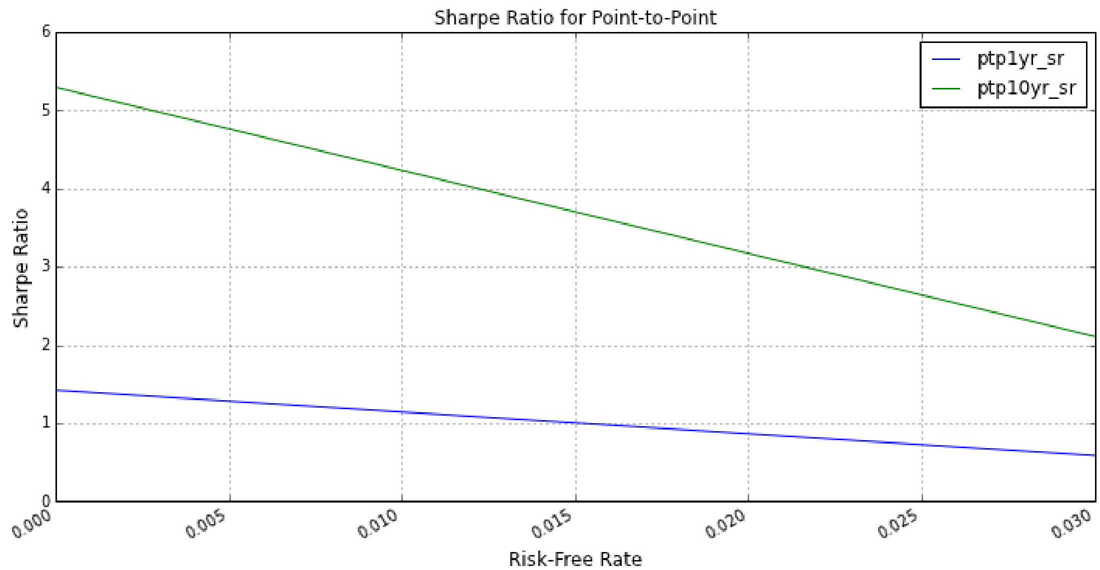
In [21]:

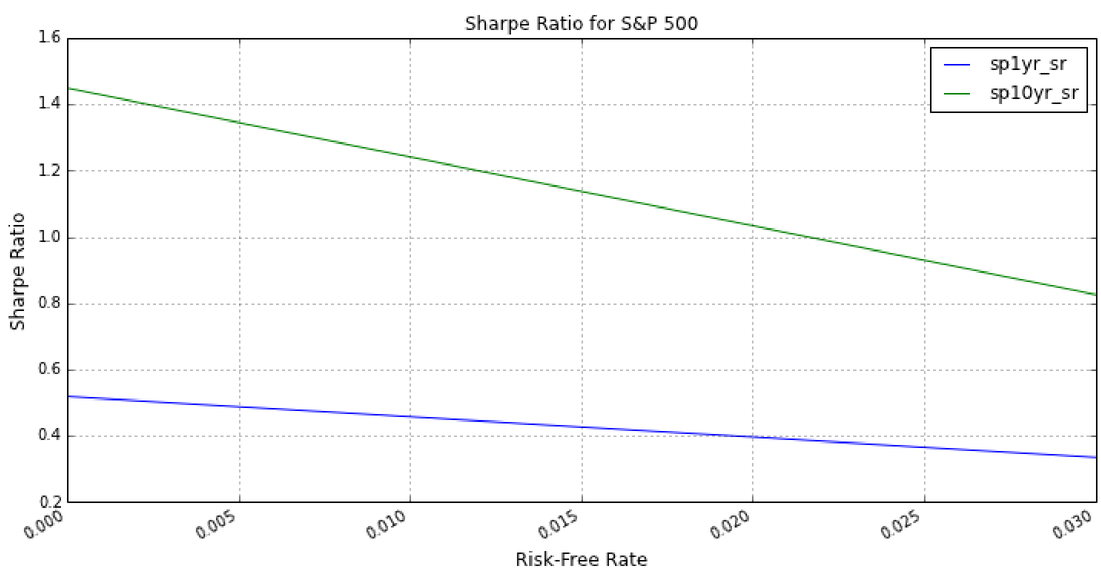
```
### Sharpe Ratio for Gain-Trigger
f9a = t7[['gt1yr_sr', 'gt10yr_sr', 'rf']].set_index('rf')
f9a.plot(title = 'Sharpe Ratio for Gain-Trigger', figsize=(12,6))
plt.gcf().autofmt_xdate()
plt.xlabel('Risk-Free Rate', fontsize=12)
plt.ylabel('Sharpe Ratio', fontsize=12)
plt.savefig('f9a_HenryWong.png')

### Sharpe Ratio for Point-to-Point
f9b = t7[['ptp1yr_sr', 'ptp10yr_sr', 'rf']].set_index('rf')
f9b.plot(title = 'Sharpe Ratio for Point-to-Point', figsize=(12,6))
plt.gcf().autofmt_xdate()
plt.xlabel('Risk-Free Rate', fontsize=12)
plt.ylabel('Sharpe Ratio', fontsize=12)
plt.savefig('f9b_HenryWong.png')

### Sharpe Ratio for S&P 500
f9c = t7[['sp1yr_sr', 'sp10yr_sr', 'rf']].set_index('rf')
f9c.plot(title = 'Sharpe Ratio for S&P 500', figsize=(12,6))
plt.gcf().autofmt_xdate()
plt.xlabel('Risk-Free Rate', fontsize=12)
plt.ylabel('Sharpe Ratio', fontsize=12)
plt.savefig('f9c_HenryWong.png')
```







Interpretation and Key Finding

As shown in the plots, the Sharpe ratios for our investment choices increase when we increase the investment duration.

12 Key Findings

Our key findings for the four research questions can be summarized as follows.

1. How do the distribution, the expected return, and the volatility of the investment performances change by increasing the investment holding period from 1-year to 10-year?
    - Distribution: More bell-sharped for the EIA index strategies and more negatively-skewed for the S&P 500 index.

- Expected return and volatility of the investments decrease across the board.
- The decreases in the expected returns of the investments are less than their volatilities.

2. Is there any particular day of month has higher or lower expected return and volatility than the other days?
    - No, the results are indifferent with day of month.
3. Does the statistically analysis provide a good benchmark for measuring opportunity cost and volatility for investing in the EIA index strategies?
    - Yes, on average, there are 3.5% and 2% of annual opportunity costs for investing in the Gain-Trigger and Point-to-Point strategy respectively.
    - The annual volatilities for investing in the Gain-Trigger and Point-to-Point are 4.2% and 3.8% respectively lower than one for the S&P 500.
4. Can the chosen index strategies of the S&P 500 EIA outperform the S&P 500 index with 10-year holding period?
    - Yes for the risk-averse investors if annual risk-free rate is less than 3% (Sharpe ratio).
    - No for the risk-neutral and risk-loving investors (Positive opportunity costs for both EIA strategies).

13 How have We Changed the Minds?

13.1 EIA Investors

Our findings for the research quesitons 1, 2, and 4 have changed the mind of the EIA investors. If investors increase their investment duration, the expected return and volatility of the investments decrease across the board. Also, the decreases in the expected returns of the investments are less than their volatilities. In addition, there is no particular day of month has higher or lower expected return and volatility than other days of month. Therefore, for the similar EIA products with different sets of issue days available, investors can randomly pick one to purchase because the investment performances would be about the same. Finally, if annual risk-free rate is less than 3%, risk-averse investors should invest on the S&P 500 EIA products, otherwise, all investors should not consider buying EIA products because they can be better off by investing on the S&P 500 index on their own.

13.2 EIA Industry

Our findings for the research quesitons 1 and 4 have changed the mind of the EIA industry. As we have mentioned previously, if investors increase their investment duration, the expected return and volatility of the investments decrease across the board. Also, the decreases in the expected returns of the investments are less than their volatilities. Therefore, increasing investment duration would have positive impact on the risk and return of the EIA investments. Thus, the EIA industry can retain their clients and drive their sales with marketing on the positive impact. With our finding for research question 4, the EIA industry should put their foucs on the risk-averse investors instead of other investors with different risk appetites.

13.3 Academia

Our findings for the research quesiton 1 have changed the mind of the academia. Based on our findings, when we increase our investment duration, the expected return and volatility of the investments decrease across the board. In addition, the decreases in the expected returns of the investments are less than their volatilities. Furthermore, the annualized return of our chosen EIA's index strategies (Gain-Trigger and Point-to-Point) are more bell-shaped and symmetrically distributed, whereas the corresponding distribution for the S&P 500 is more negatively-skewed. These findings are helpful for scholars in Economics, Finance, and Statistics to identify their possible distributions, therefore, they can validate the distributions by Hypothesis Testing, in turn, they can estimate the parameters with methods such as MLE and they can also estimate its standard error with bootstrap.

13.4 Acturaries

Our findings for the research quesiton 3 have changed the mind of actuaries, in particular, the actuaries who do pricing and hedging work for the EIA products. Based on our findings, on average, there are 3.5% and 2% of annual opportunity costs for investing in the Gain-Trigger and Point-to-Point strategy respectively. In addition, the annual volatilities for investing in the Gain-Trigger and Point-to-Point index strategies are 4.2% and 3.8% respectively lower than one for the S&P 500. The opportunity cost and volatility are the required parameters for pricing and hedging EIA products. Therefore, company who have the good estimates of the parameters would have the comparative advantage than others companies in making profits from designing and selling EIA products.

Part II

References

[1] David C. M. Dickson, Mary R. Hardy, and Howard R. Waters. Actuarial Mathematics for Life Contingent Risk. Cambridge University Press, 2009.

[2] David G. Luenberger. Investment Science. Oxford University Press, 1998.

[3] Fernando Pérez, Brian E. Granger, IPython: A System for Interactive Scientific Computing, Computing in Science and Engineering, vol. 9, no. 3, pp. 21-29, May/June 2007, doi:10.1109/MCSE.2007.53. URL: http://ipython.org