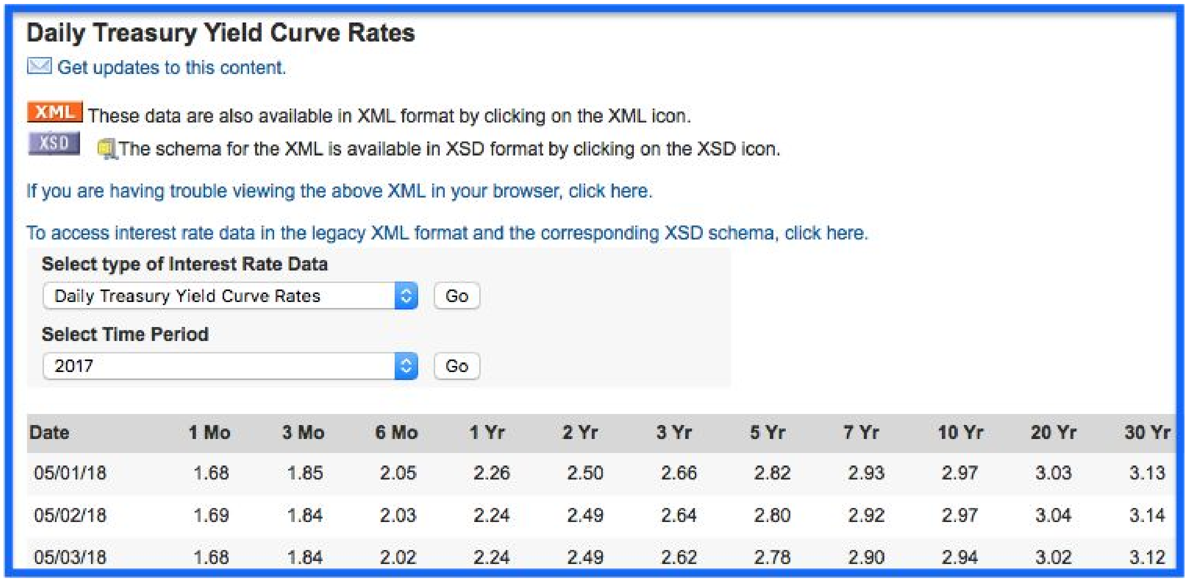
Capstone Analytic Report and Research Proposal
Unit 4 / Lesson 2 / Project 2
Analytic Report and Research Proposal by John Foxworthy

The dataset that I will be using are the Daily U.S. Treasury Yield Curve Rates and its significance is the global importance in the financial trading industry, data quality and its statistical significance. The figures below have daily dates as rows with maturity points as columns. For example, a ten year is 2.97 percent as of the 1st of May, 2018, which is the amount of the interest rate on a decade long loan. The data goes back to 1990 up until to the current date in 2018.



Daily Treasury Yield Curve Rates

Get updates to this content.

These data are also available in XML format by clicking on the XML icon.

The schema for the XML is available in XSD format by clicking on the XSD icon.

If you are having trouble viewing the above XML in your browser, click here.

To access interest rate data in the legacy XML format and the corresponding XSD schema, click here.

Select type of Interest Rate Data: Daily Treasury Yield Curve Rates

Select Time Period: 2017

Date	1 Mo	3 Mo	6 Mo	1 Yr	2 Yr	3 Yr	5 Yr	7 Yr	10 Yr	20 Yr	30 Yr
05/01/18	1.68	1.85	2.05	2.26	2.50	2.66	2.82	2.93	2.97	3.03	3.13
05/02/18	1.69	1.84	2.03	2.24	2.49	2.64	2.80	2.92	2.97	3.04	3.14
05/03/18	1.68	1.84	2.02	2.24	2.49	2.62	2.78	2.90	2.94	3.02	3.12

Daily_Treasury_Yield_Curve_Rates

Each of the figures represents hundreds of billions of U.S. dollar trades from participants from dozens of countries and have the impressive data attribute of a complete absence of stale data. Not some, but all figures, on each business day of the year, has at least one person and likely many people trying to influence the interest rate values to change. Specifically, the strong liquidity, in other words, the quantity of buyers finding sellers and vice versa is very high, in every trading day, in the year going back decades since the U.S. Dollar is the world's reserve currency. According to organizations like the International Monetary Fund (IMF), 6 out of 10 or 7 out of 10 transactions by physical hand or electronically, involve the U.S. Dollar. There are more than 200 countries in the world and each have some type of exporting and importing business that does involve the U.S. Dollar. The more a country exports, the more their currency, not the U.S. Dollar, rises in value because of increased demand, thereby making the value of their exports more expensive. To counteract, countries buy U.S. Dollars to dampen their local currency appreciation and the most common format is in the U.S. loan or fixed income market. Borrow a month at 1.68% or a year at 2.26% in U.S. dollars on the 1st of May for your exporting business from the Daily Treasury Yield Curve, for instance. Not to mention, the U.S. economy is the largest in the world so there are plenty of individuals and institutions that depend on this data to raise money for countless causes from a mortgage loan for a residential property to purchasing a new office building for an expanding company or local government institution.

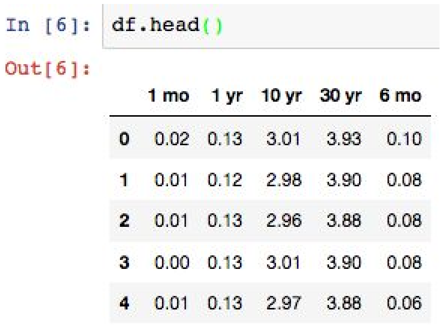
Lastly, the source of this data is the U.S. Treasury department of the U.S. government. There are no impediments for the use of this data as it public from a legal perspective and officially published by the U.S. government on every business day. Daily Treasury Yield Curve is widely used by countless people and institutions for many causes and I choose it to forward my career in Data Science.

Plotting lines, histograms and other data visualizations from csv, xml or json files will not have an issues of daily rates data, but more importantly, the evolution throughout time over the past decades, would help in advanced statistical techniques in the coming months of the Thinkful Data Science course. Below is some code for the column maturity points of 1 month, 6 month, 1 year, 10 year and 30 year from a json file. There are roughly 250 business days in a given year so beyond the demonstration below there are about 6,750 business days for this single column only from 1990 to 2017. Furthermore, the 11 columns from the 1 month to the 30 year would be more than 75,000 interest rate daily data from 1990 to the current day. For the research report, we will use daily yield curve rates from 2014 to 2017, which is 1,000 data points per column maturity point for a total of 5,000.

```
# In[1]: import json, matplotlib.pyplot as plt, pandas as pd
         %matplotlib inline
# In[2]: json.load((open('/Users/lacivert/2014_2017.json')))
# In[3]: sample_json_df = pd.read_json('/Users/lacivert/2014_2017.json')
# In[4]: #assign DataFrame
         df = pd.DataFrame(sample_json_df)
# In[5]: #list column names
         list(df.columns.values)
# Out[5]:['1 mo', '1 yr', '10 yr', '30 yr', '6 mo']
```

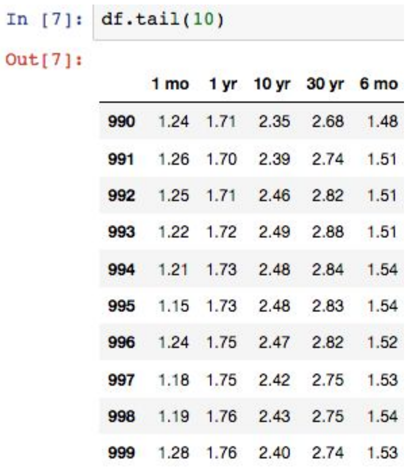
Looking forward, there are three analytical questions to answer . . .

1. How similar is the data set in terms of the minimum, maximum, average, quantiles and variation of the data set?
2. Does the data set cluster together over the 2014 to 2017 time period?
3. How much do the column maturity points vary with each other?



```
In [6]: df.head()
Out[6]:
```

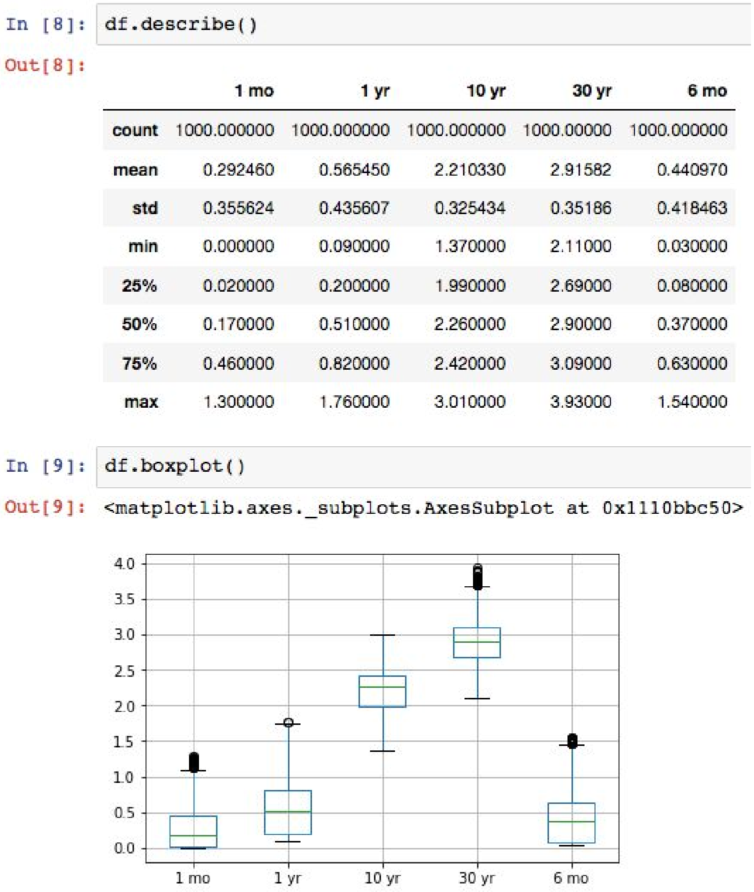
	1 mo	1 yr	10 yr	30 yr	6 mo
0	0.02	0.13	3.01	3.93	0.10
1	0.01	0.12	2.98	3.90	0.08
2	0.01	0.13	2.96	3.88	0.08
3	0.00	0.13	3.01	3.90	0.08
4	0.01	0.13	2.97	3.88	0.06



```
In [7]: df.tail(10)
Out[7]:
```

	1 mo	1 yr	10 yr	30 yr	6 mo
990	1.24	1.71	2.35	2.68	1.48
991	1.26	1.70	2.39	2.74	1.51
992	1.25	1.71	2.46	2.82	1.51
993	1.22	1.72	2.49	2.88	1.51
994	1.21	1.73	2.48	2.84	1.54
995	1.15	1.73	2.48	2.83	1.54
996	1.24	1.75	2.47	2.82	1.52
997	1.18	1.75	2.42	2.75	1.53
998	1.19	1.76	2.43	2.75	1.54
999	1.28	1.76	2.40	2.74	1.53

Continuing, we can take a look at the top part of the json file of 2014 and the bottom data points of 2017. We can see a progression of values starting from close to zero to moving away from zero over time. This is one characteristic of all the maturity points of the U.S. Treasury Yield Curve that continues with the descriptive statistics carrying most of this theme, but not all.



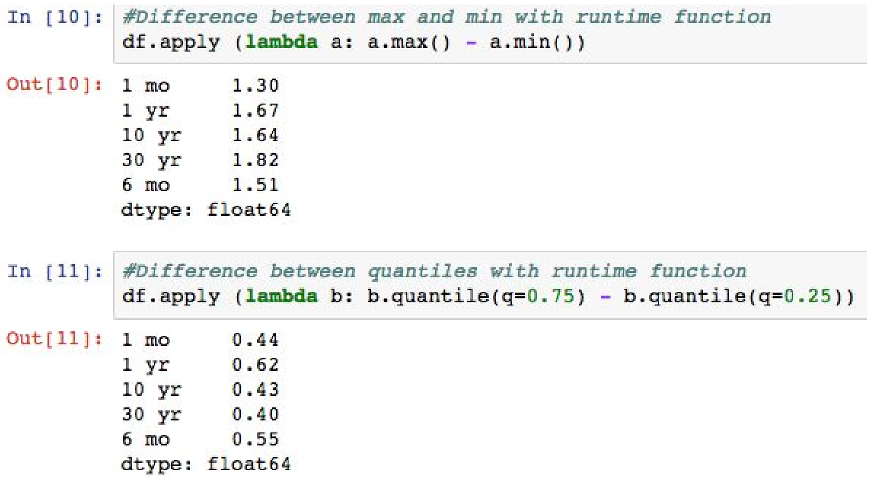
```
In [8]: df.describe()
Out[8]:
```

	1 mo	1 yr	10 yr	30 yr	6 mo
count	1000.000000	1000.000000	1000.000000	1000.00000	1000.000000
mean	0.292460	0.565450	2.210330	2.91582	0.440970
std	0.355624	0.435607	0.325434	0.35186	0.418463
min	0.000000	0.090000	1.370000	2.11000	0.030000
25%	0.020000	0.200000	1.990000	2.69000	0.080000
50%	0.170000	0.510000	2.260000	2.90000	0.370000
75%	0.460000	0.820000	2.420000	3.09000	0.630000
max	1.300000	1.760000	3.010000	3.93000	1.540000

```
In [9]: df.boxplot()
Out[9]: <matplotlib.axes._subplots.AxesSubplot at 0x1110bbc50>
```

The mean, minimum, maximum and all quantiles are in ascending order yielding a similarity, but the variation around the mean, such as captured in one way, by the standard deviation is not similar. The one month has the lowest value in terms of the mean, minimum, maximum and the three expressed quantiles above. Next, the six month is the second lowest of the same metrics followed by the 1 year, 10 year and 30 year. As you increase the length of the loan, rates ascend, but vary differently as the dissimilarity in the standard deviation demonstrates the only exception of the common theme. The standard deviation is the highest among the 1 year, followed by the 6 month, then oddly the 1 month and 30 year are similar at 0.35, followed lastly by the 10 year. The box plot provides some illustration of the similarities with the boxes in each of the maturity points. The top of each box is the third quartile, the bottom is the first quartile and the green line in between is the median. The horizontal dash line at the very top above the boxes is the maximum and the bottom dash line is the minimum.

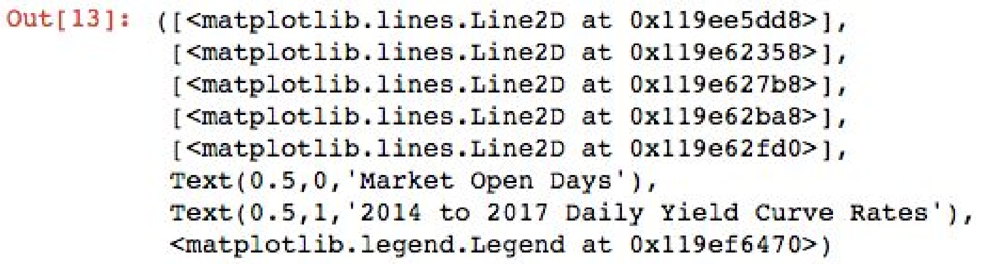
If you inspect the variation differences a bit more, then you will see there is no ascension of daily yield curve rate data when you difference the maximum and the minimum, and then separately again with the quantiles. The 1 month rates has the lowest difference in maximum and minimum rate values, followed by the 6 month, but then the 1 year of 1.67 is more than the 10 year of 1.64. There is also no ascension when you difference the 75% and the 25% quantiles because the 1 year of 0.62 is the largest difference amount, while 0.40 of the 30 year is the lowest.



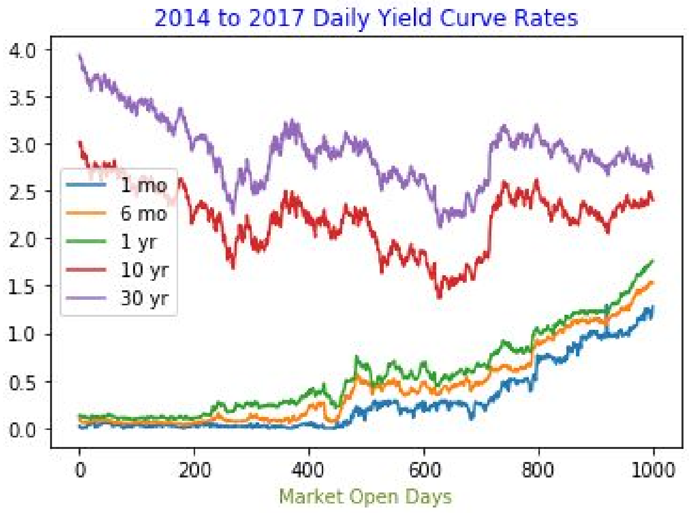
```
In [10]: #Difference between max and min with runtime function
         df.apply (lambda a: a.max() - a.min())

Out[10]: 1 mo      1.30
         1 yr      1.67
         10 yr     1.64
         30 yr     1.82
         6 mo      1.51
         dtype: float64

In [11]: #Difference between quantiles with runtime function
         df.apply (lambda b: b.quantile(q=0.75) - b.quantile(q=0.25))

Out[11]: 1 mo      0.44
         1 yr      0.62
         10 yr     0.43
         30 yr     0.40
         6 mo      0.55
         dtype: float64
```

Secondly, to address clustering, there is a common theme across the time horizon of the data set. If you go back to page of 1, then you will see eleven column maturity points from 1 month to 30 years. We will see that both ends of the daily yield curve, cluster together. To demonstrate, let's first assign the column maturity points by slicing the data frame.

```
# In[12]:  one_month, six_month, one_year
          , ten_year, thirty_year  =
          df.loc[:,'1 mo'], df.loc[:,'6 mo'], df.loc[:,'1 yr']
          , df.loc[:,'10 yr'], df.loc[:,'30 yr']
# In[13]:  plt.plot(one_month), plt.plot(six_month), plt.plot(one_year)
         ,plt.plot(ten_year), plt.plot(thirty_year)
         , plt.xlabel('Market Open Days',color='olivedrab')
         , plt.title('2014 to 2017 Daily Yield Curve Rates', color='blue')
         ,plt.legend()
```

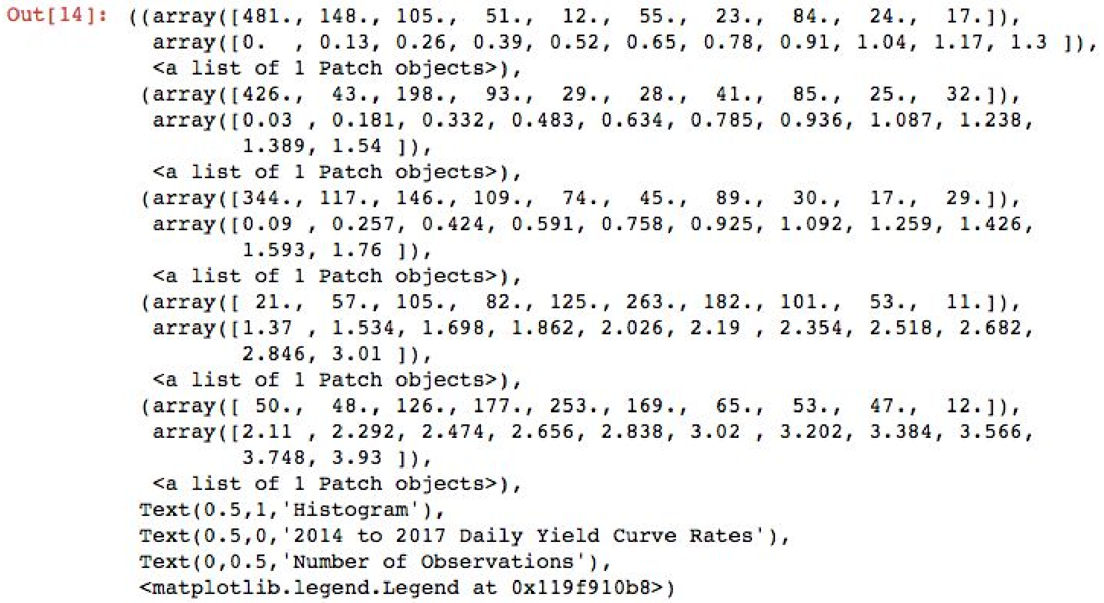


```
Out[13]: ([<matplotlib.lines.Line2D at 0x119ee5dd8>],
          [<matplotlib.lines.Line2D at 0x119e62358>],
          [<matplotlib.lines.Line2D at 0x119e627b8>],
          [<matplotlib.lines.Line2D at 0x119e62ba8>],
          [<matplotlib.lines.Line2D at 0x119e62fd0>],
          Text(0.5,0,'Market Open Days'),
          Text(0.5,1,'2014 to 2017 Daily Yield Curve Rates'),
          <matplotlib.legend.Legend at 0x119ef6470>)
```

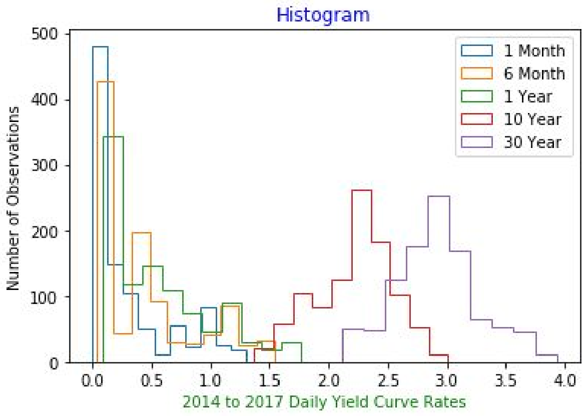


From the line plot above we can see that the shorter maturity points of 1 month, 6 month and 1 year cluster together and differently from the longer maturity points of the 10 year and the 30 year. Vice versa on the longer maturity points, i.e. the 10 year and the 30 year clustering together and differing from the shorter maturity points. To illustrate another way, we can use a histogram that shows the overlapping of the dual clustering theme.

```
# In[14]: plt.hist(one_month, histtype='step', label='1 Month')
          ,plt.hist(six_month, histtype='step',label='6 Month')
          ,plt.hist(one_year, histtype='step',label='1 Year')
          ,plt.hist(ten_year, histtype='step',label='10 Year')
          , plt.hist(thirty_year, histtype='step',label='30  Year')
          ,plt.title('Histogram',color='blue')
          , plt.xlabel('2014 to 2017 Daily Yield Curve Rates',color='green')
          ,plt.ylabel('Number of Observations')
          ,plt.legend()
```



```
Out[14]: ((array([481., 148., 105.,  51.,  12.,  55.,  23.,  84.,  24.,  17.]),
           array([0.  , 0.13, 0.26, 0.39, 0.52, 0.65, 0.78, 0.91, 1.04, 1.17, 1.3 ]),
           <a list of 1 Patch objects>),
          (array([426.,  43., 198.,  93.,  29.,  28.,  41.,  85.,  25.,  32.]),
           array([0.03 , 0.181, 0.332, 0.483, 0.634, 0.785, 0.936, 1.087, 1.238,
                  1.389, 1.54 ]),
           <a list of 1 Patch objects>),
          (array([344., 117., 146., 109.,  74.,  45.,  89.,  30.,  17.,  29.]),
           array([0.09 , 0.257, 0.424, 0.591, 0.758, 0.925, 1.092, 1.259, 1.426,
                  1.593, 1.76 ]),
           <a list of 1 Patch objects>),
          (array([ 21.,  57., 105.,  82., 125., 263., 182., 101.,  53.,  11.]),
           array([1.37 , 1.534, 1.698, 1.862, 2.026, 2.19 , 2.354, 2.518, 2.682,
                  2.846, 3.01 ]),
           <a list of 1 Patch objects>),
          (array([ 50.,  48., 126., 177., 253., 169.,  65.,  53.,  47.,  12.]),
           array([2.11 , 2.292, 2.474, 2.656, 2.838, 3.02 , 3.202, 3.384, 3.566,
                  3.748, 3.93 ]),
           <a list of 1 Patch objects>),
          Text(0.5,1,'Histogram'),
          Text(0.5,0,'2014 to 2017 Daily Yield Curve Rates'),
          Text(0,0.5,'Number of Observations'),
          <matplotlib.legend.Legend at 0x119f910b8>)
```



The shorter maturities of 1 month, 6 month and 1 year are all near zero on the left of the overlapping histograms as the longer maturities are away from daily zero yield curve rates.

Lastly, to address the last analytical question of how each of the maturites vary from each other, we look into covariance. To build a case, we begin with expected value leading to mean average, variance and then covariance. If we approach the U.S. Daily Yield Curve Rates as random values, then buyers and sellers in the rate market have an expectation expressed as an expected value, i.e $E(X)$. All bids and offers in the rate market are equally likely as if it is a flip of a coin so a probability, p_i, can altogether be summed to be equal to one. If we take the expected value as the mean like on page 4, then the arithmetic mean can be calculated by the numpy mean function.

$$Expected\ Value\ =\ x_1 \bullet p_1 + x_2 \bullet p_2 + \ldots + x_n \bullet p_n = \sum_{i=1}^{n} x_i \bullet p_i,\ where\ \sum_{i=1}^{n} p_i = 1$$

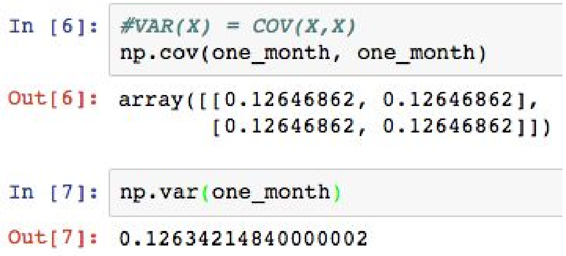
$$Expected\ Value\ =\ E[X] = sum(x_i) \bullet \frac{1}{n} = Arithmetic\ Mean = \mu,\ where\ n\ is\ the\ number\ of\ observations$$

In turn, the values that vary around the mean such as variance are the calculated average squared difference of each value from the expected value or mean. To put it another way, the expected squared difference from the expected value or μ. Since individual probabilities are equal to each assigned individual value, then we can drop the probabilities of an equally weighted distribution to calculate the variance.

$$Variance\ =\ E[(X_i - \mu)^2]$$

To understand how the column maturity point vectors vary with each other, the simplest approach to calculate covariance is comparing to itself, such as with a 1 dimensional array of the one month maturity point. The covariance equals 0.126, just like the variance.

$$Variance\ =\ E[(X_i - \mu)^2] = Covariance\ =\ (E[(X_i - \mu)^2],\ E[(X_i - \mu)^2])$$



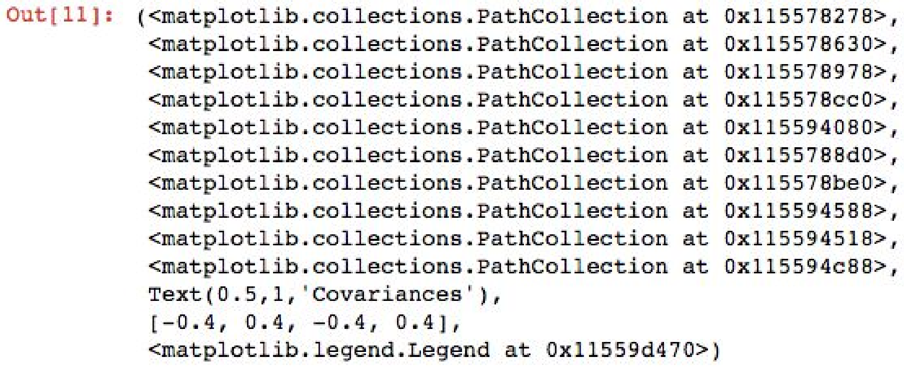
```
In [6]: #VAR(X) = COV(X,X)
        np.cov(one_month, one_month)

Out[6]: array([[0.12646862, 0.12646862],
               [0.12646862, 0.12646862]])

In [7]: np.var(one_month)

Out[7]: 0.12634214840000002
```

$$Covariance\ =\ (E[(X_i - \mu)^2],\ E[(Y_i - \mu)^2])$$

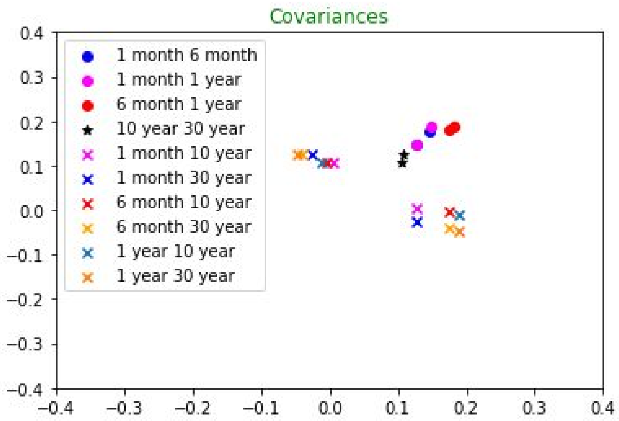
As we apply covariances to different maturity point vectors, then we will be able to see a pattern of clustering data versus non - clustering.

```
#assigning covariances to short maturity cluster
In[8]:  j, k, l = np.cov(one_month,six_month), np.cov(one_month,one_year),
np.cov(six_month,one_year)
#assigning covariance to longer maturity cluster
In[9]:  m = np.cov(ten_year,thirty_year)
```

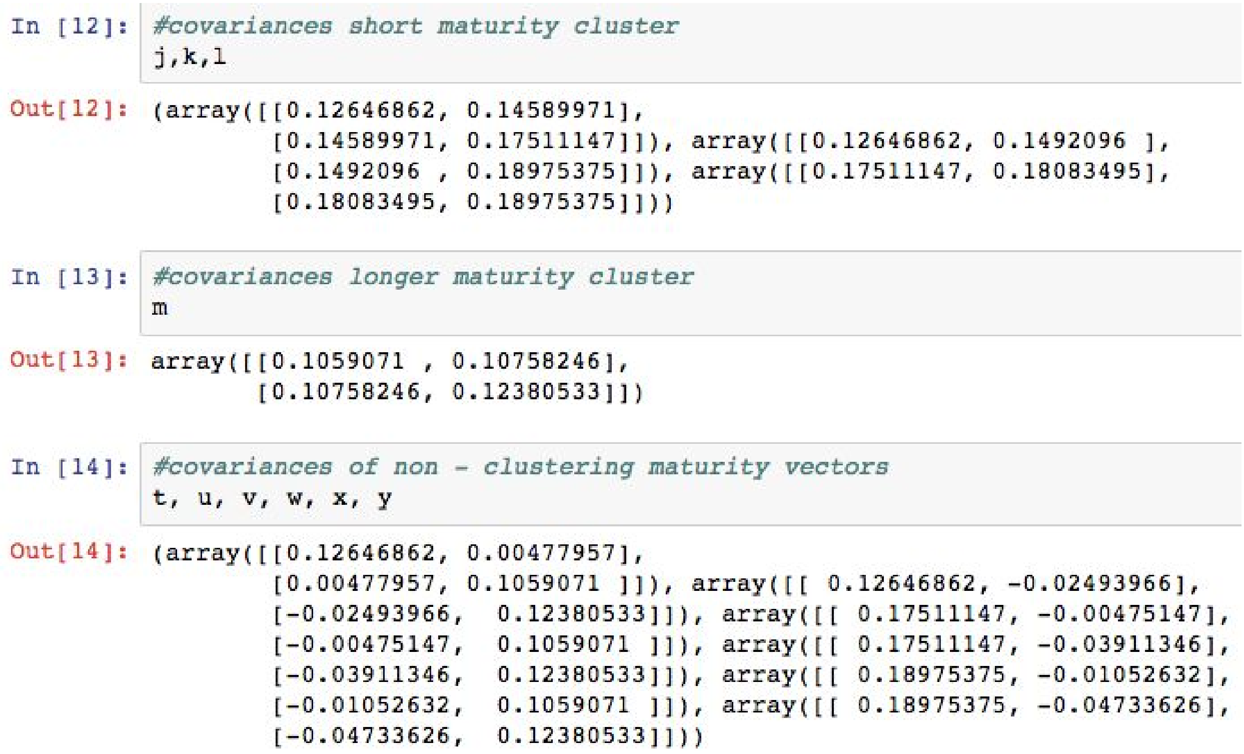
```
#all maturity vectors that vary differently, i.e. not clustering
In[10]:  t, u, v, w, x, y = np.cov(one_month, ten_year), np.cov(one_month,
thirty_year),np.cov(six_month, ten_year), np.cov(six_month, thirty_year),
np.cov(one_year, ten_year), np.cov(one_year, thirty_year)
In[11]:  plt.scatter(j[:1], j[-1:],label='1 month 6 month', color='blue')
       , plt.scatter(k[:1], k[-1:],label='1 month 1 year', color='magenta')
       , plt.scatter(l[:1], l[-1:],label='6 month 1 year', color='red')
       , plt.scatter(m[:1],m[-1:], label='10 year 30 year', color= 'black',marker='*')
       ,plt.scatter(t[:1],t[-1:], label ='1 month 10 year',
color='magenta',marker='x')
       , plt.scatter(u[:1],u[-1:], label = '1 month 30 year', color='blue',
marker='x')
       ,plt.scatter(v[:1],v[-1:], label = '6 month 10 year', color='red', marker='x')
       ,plt.scatter(w[:1],w[-1:], label = '6 month 30 year',
color='orange',marker='x')
       ,plt.scatter(x[:1],x[-1:], label = '1 year 10 year', marker='x')
       ,plt.scatter(y[:1],y[-1:], label='1 year 30 year', marker = 'x')
       ,plt.title('Covariances',color='green')
       ,plt.axis([-0.40,0.40,-0.40,0.40]),plt.legend()
```



```
Out[11]: (<matplotlib.collections.PathCollection at 0x115578278>,
          <matplotlib.collections.PathCollection at 0x115578630>,
          <matplotlib.collections.PathCollection at 0x115578978>,
          <matplotlib.collections.PathCollection at 0x115578cc0>,
          <matplotlib.collections.PathCollection at 0x115594080>,
          <matplotlib.collections.PathCollection at 0x1155788d0>,
          <matplotlib.collections.PathCollection at 0x115578be0>,
          <matplotlib.collections.PathCollection at 0x115594588>,
          <matplotlib.collections.PathCollection at 0x115594518>,
          <matplotlib.collections.PathCollection at 0x115594c88>,
          Text(0.5,1,'Covariances'),
          [-0.4, 0.4, -0.4, 0.4],
          <matplotlib.legend.Legend at 0x11559d470>)
```



The shorter maturity cluster of 1 month, 6 month and 1 year have covariances with round dots, followed by the longer maturity cluster covariance of 10 year and 30 year with black stars. The clusters have a pattern of covariances from the bottom left to the top right on the scatter plot, which is the opposite of the non - clustering covariances. All of the covariances with an x shaped marker do not cluster and have a trend of upper left to lower right. The first variable of the non - clustering covariance are the shorter maturities of 1 month, 6 month and 1 year, followed by the second variable in the covariance function of the longer maturities of 10 year and 30 year.



```
In [12]: #covariances short maturity cluster
         j,k,l

Out[12]: (array([[0.12646862, 0.14589971],
                 [0.14589971, 0.17511147]]), array([[0.12646862, 0.1492096 ],
                 [0.1492096 , 0.18975375]]), array([[0.17511147, 0.18083495],
                 [0.18083495, 0.18975375]]))

In [13]: #covariances longer maturity cluster
         m

Out[13]: array([[0.1059071 , 0.10758246],
                [0.10758246, 0.12380533]])

In [14]: #covariances of non - clustering maturity vectors
         t, u, v, w, x, y

Out[14]: (array([[0.12646862, 0.00477957],
                 [0.00477957, 0.1059071 ]]), array([[ 0.12646862, -0.02493966],
                 [-0.02493966,  0.12380533]]), array([[ 0.17511147, -0.00475147],
                 [-0.00475147,  0.1059071 ]]), array([[ 0.17511147, -0.03911346],
                 [-0.03911346,  0.12380533]]), array([[ 0.18975375, -0.01052632],
                 [-0.01052632,  0.1059071 ]]), array([[ 0.18975375, -0.04733626],
                 [-0.04733626,  0.12380533]]))
```

All of the clustering maturities have covariances greater than zero because of a positive, linear relationship where one maturity increases, then the other maturity also increases. This is also true when a maturity rate decreases in value, the other maturity rate also decreases. The shorter maturity cluster of 1 month, 6 month and 1 year rates tend to increase and decrease together just like the longer maturity rate cluster of the 10 year and the 30 year. Conversely, the non - clustering have some covariances less than zero because of a negative linear relationship. If the one month rate increase, then it is likely, but not absolutely certain, the 30 year rate will decrease. This is also true of the 6 month versus the 10 year and so on.

Altogether, the selection of the Daily Yield Curve Data have both similarities and differences along with common themes and patterns. Looking ahead, the strong data quality of the U.S. Treasury Yield Curve, the daily public availability for use and its representation in the global financial markets would be applicable for data science techniques in the Thinkful course. To propose a future research, then I would be interested in principal component analysis or PCA.

Some the questions, PCA may attempt to address is how we can identify the dominant ways the various points on the yield curve move together. What are the characteristics of yield curve changing over time? For example, how much does the level of yield curve rates, in this case, close to zero contribute to yield curve rate change over time?   Or how about the quantity of the rate change and the quantity of the time change, i.e. the slope of the yield curve contribute to the evolution of the U.S. yield curve changes? There are several related questions, but principal component analysis would be a good technique to answer the evolution of yield curve. In conclusion, I look forward to demonstrate my net worth in the coming months.