Assignment 01

The Treasury Yield Curve, which is also known as the term structure of interest rates, draws out a line chart to demonstrate a relationship between yields and maturities of on-the-run treasury fixed income securities. It illustrates the yields of Treasury securities at fixed maturities, viz. 1, 3 and 6 months and 1, 2, 3, 5, 7, 10, 20 and 30 years. Therefore, they are commonly referred to as "Constant Maturity Treasury" Rates or CMTs.

Yields on Treasury securities are in theory *free of credit risk* and are often used as a benchmark to evaluate the relative worth of US Non-Treasury securities.

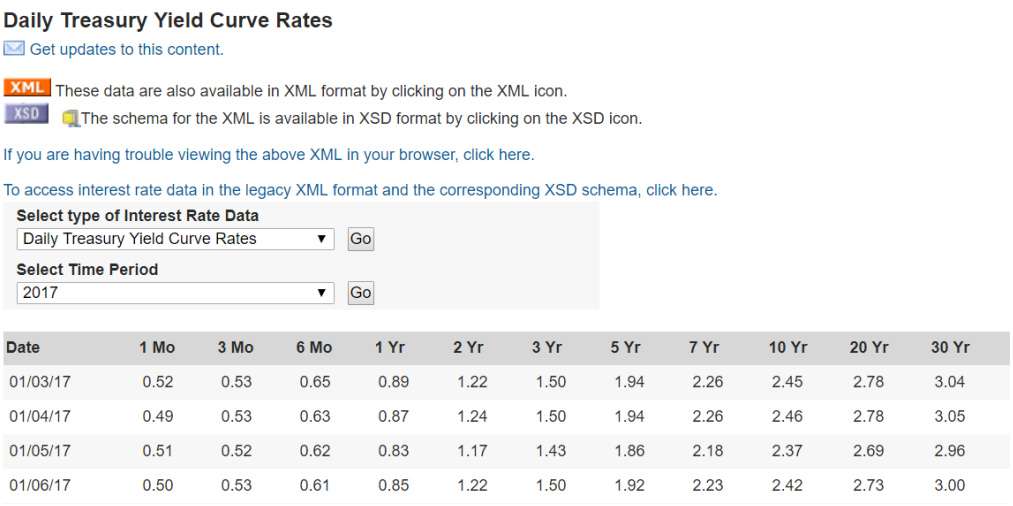
(https://www.investopedia.com)

Code the following in Python

In this assignment you learn how to extract data from US Treasury website.

This URL shows the yield rates for 2017

https://www.treasury.gov/resource-center/data-chart-center/interest-rates/Pages/TextView.aspx?data=yieldYear&year=2017



Daily Treasury Yield Curve Rates

Get updates to this content.

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

XSD 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 ▼ Go

Select Time Period

2017 ▼ Go

Date	1 Mo	3 Mo	6 Mo	1 Yr	2 Yr	3 Yr	5 Yr	7 Yr	10 Yr	20 Yr	30 Yr
01/03/17	0.52	0.53	0.65	0.89	1.22	1.50	1.94	2.26	2.45	2.78	3.04
01/04/17	0.49	0.53	0.63	0.87	1.24	1.50	1.94	2.26	2.46	2.78	3.05
01/05/17	0.51	0.52	0.62	0.83	1.17	1.43	1.86	2.18	2.37	2.69	2.96
01/06/17	0.50	0.53	0.61	0.85	1.22	1.50	1.92	2.23	2.42	2.73	3.00

Notice the red-marked parameter on the URL above that specifies the time period of the data. The parameter can be changed directly on the URL or on the text input on the website.

Clicking the XML button generates the data in XML format:

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<feed xml:base="http://data.treasury.gov/Feed.svc/"
xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices"
xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata"
xmlns="http://www.w3.org/2005/Atom">
  <title type="text">DailyTreasuryYieldCurveRateData</title>
  <id>http://data.treasury.gov/feed.svc/DailyTreasuryYieldCurveRateData</id>
  <updated>2018-01-29T21:17:51Z</updated>
  <link rel="self" title="DailyTreasuryYieldCurveRateData"
href="DailyTreasuryYieldCurveRateData" />
  <entry>
    <id>http://data.treasury.gov/Feed.svc/DailyTreasuryYieldCurveRateData(6758)</id>
    <title type="text"></title>
    <updated>2018-01-29T21:17:51Z</updated>
    <author>
      <name />
    </author>
    <link rel="edit" title="DailyTreasuryYieldCurveRateDatum"
href="DailyTreasuryYieldCurveRateData(6758)" />
    <category term="TreasuryDataWarehouseModel.DailyTreasuryYieldCurveRateDatum"
scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme" />
    <content type="application/xml">
      <m:properties>
        <d:Id m:type="Edm.Int32">6758</d:Id>
        <d:NEW_DATE m:type="Edm.DateTime">2017-01-03T00:00:00</d:NEW_DATE>
        <d:BC_1MONTH m:type="Edm.Double">0.52</d:BC_1MONTH>
        <d:BC_3MONTH m:type="Edm.Double">0.53</d:BC_3MONTH>
        <d:BC_6MONTH m:type="Edm.Double">0.65</d:BC_6MONTH>
        <d:BC_1YEAR m:type="Edm.Double">0.89</d:BC_1YEAR>
        <d:BC_2YEAR m:type="Edm.Double">1.22</d:BC_2YEAR>
        <d:BC_3YEAR m:type="Edm.Double">1.5</d:BC_3YEAR>
        <d:BC_5YEAR m:type="Edm.Double">1.94</d:BC_5YEAR>
        <d:BC_7YEAR m:type="Edm.Double">2.26</d:BC_7YEAR>
        <d:BC_10YEAR m:type="Edm.Double">2.45</d:BC_10YEAR>
        <d:BC_20YEAR m:type="Edm.Double">2.78</d:BC_20YEAR>
        <d:BC_30YEAR m:type="Edm.Double">3.04</d:BC_30YEAR>
        <d:BC_30YEARDISPLAY m:type="Edm.Double">3.04</d:BC_30YEARDISPLAY>
      </m:properties>
    </content>
  </entry>
```

…

In the data, there are multiple “entry” blocks, of which each is for 1 line of rates in the US Treasury website above. The red-marked area is where the rates are.

In order to handle XML, you need to install urllib and lxml by run the following commands on the Cmd prompt in Windows or terminal in MacOS/Linux:

pip install lxml or conda install lxml

pip install urllib or conda install urllib (or pip install urllib3 or conda install urllib3)

Use an IDE of your choice (Spyder, Jupyter, etc.) to work. First needs to import some libraries:

```
import urllib.request

from lxml import objectify, etree
```

Then send a request to US Treasury for 2017 data, store data in the variable namely “xml”:

```
xml = urllib.request.urlopen('http://data.treasury.gov/feed.svc/DailyTreasuryYieldCurveRateData?$filter=year(NEW_DATE)%20eq%202017').read()
```

The fromstring() method is to parse an XML string:

```
tree = etree.fromstring(xml)
```

xpath() is to show all the elements in the XML:

```
tree.xpath("//*")
```

To extract all the “BC_3MONTH” elements, use findall() method, and store data in a list namely tbill3month:

```
tbill3month =tree.findall(".//{http://schemas.microsoft.com/ado/2007/08/dataservices}BC_3MONTH")
```

To access to the price in the first element of the list:

```
tbill3month[0].text
```

would show the rate. Other elements can be accessed in the same way using their indices (e.g. tbill3month[1].text, tbill3month[2].text, etc.)

The same can be done for 1 month, 6 month, etc. up to 30 year yield rates

BC_1MONTH

BC_6MONTH

BC_1YEAR

BC_2YEAR

BC_3YEAR

BC_5YEAR

BC_7YEAR

BC_10YEAR

BC_20YEAR

BC_30YEAR

More information on how to use the lxml package can be found at

http://lxml.de/

However, in the scope of the assignment, the information provided is sufficient.

Students are encouraged to read more about XML format for better understandings (not required for the assignment):

https://www.w3schools.com/xml/

(6 points) Write a function that takes a period (e.g. 2017), sends requests to US Treasury website, retrieves the response, and returns the yield rates in a 2-dimensional list/array (each column is a yield rate type, each row is a date).

(4 point) Write code that, given periods spanning over several years (e.g. 2010 to 2018), calls the function above for each year, and saves the data to an Excel file in 1 single sheet, of which each column stores 1 type of yield rate (e.g. 1 month rate, 3 month rate, etc.). Save the file to your Google Drive, and download it to your computer to submit.

Submission includes:

a. A notebook with name as “group_xxx_assignment_yyy.ipynb” with detailed explanation of the work as markdown text
b. Link to your GitHub for the shared notebook
c. All the other files
d. For (a) and (b), if your group has more than 1 notebook (each per student), please include all of them but select one for being graded