

## HW 4.5

December 13, 2018

```
In [23]: import QuantLib as ql
import quandl
import numpy as np
import pandas as pd
import datetime

In [24]: token = "VmD_oxdaJACCTkt2Pry-"
n = range(1,22)
nms = ["CHRIS/CME_ED"+str(i) for i in n]
dfs = [quandl.get(nm, authtoken=token) for nm in nms]

In [25]: euro_dollar = pd.DataFrame()
for i in range(21):
    euro_dollar['ED'+str(i+1)] = 100-dfs[i]['Settle']
euro_dollar = euro_dollar.dropna()

In [26]: def getday(date,n=0):
    result_date = date + datetime.timedelta(days=n)
    d = result_date.strftime('%Y-%m-%d')
    d = pd.to_datetime(d)
    return d

In [27]: def ql_to_datetime(d):
    return datetime.datetime(d.year(), d.month(), d.dayOfMonth())

In [28]: # calculate maturity date
maturity_date = []
imm = ql.IMM()
last_date = getday(pd.to_datetime('2018-12-10'), 30*2)
d = ql.Date(13,9,1993)
while ql_to_datetime(d) <= last_date:
    d = imm.nextDate(d)
    maturity_date.append(d)
maturity = [ql_to_datetime(i) for i in maturity_date]

In [29]: # add the row with maturity date not in euro_dollar's index
index_add = []
for i in range(len(maturity)):
```

```

        if maturity[i] not in euro_dollar.index:
            index_add.append(maturity[i])

    for i in range(len(index_add)):
        euro_dollar.loc[index_add[i],:] = np.nan

In [30]: euro_dollar = euro_dollar.sort_index()

In [31]: # find the rows with maturity date and calculate maturity date for every contract for
maturity_date_table = pd.DataFrame(index = euro_dollar.index, columns=euro_dollar.columns)
k = 0
j = 0
while j < len(maturity_date_table):
    if maturity_date_table.index[j]<ql_to_datetime(maturity_date[k]):
        j+=1
    elif maturity_date_table.index[j]==ql_to_datetime(maturity_date[k]):
        imm = ql.IMM()
        d = maturity_date[k]
        for i in range(0,21):
            maturity_date_table.iloc[j, i] = ql_to_datetime(imm.nextDate(d))
            d = imm.nextDate(d)
        j+=1
    else:
        k+=1

In [32]: #calculate the first row of contract maturity date
maturity_date_table.iloc[0,:] = maturity[:21]

In [33]: # For the remaining row, use forward fill method to fill all the cells.
maturity_date_table = maturity_date_table.fillna(method='ffill')

In [34]: maturity_date_table.head()

```

Out [34]:

|            | ED1        | ED2        | ED3        | ED4        | ED5        | ED6        | \ |
|------------|------------|------------|------------|------------|------------|------------|---|
| Date       |            |            |            |            |            |            |   |
| 1993-09-14 | 1993-09-15 | 1993-12-15 | 1994-03-16 | 1994-06-15 | 1994-09-21 | 1994-12-21 |   |
| 1993-09-15 | 1993-12-15 | 1994-03-16 | 1994-06-15 | 1994-09-21 | 1994-12-21 | 1995-03-15 |   |
| 1993-09-16 | 1993-12-15 | 1994-03-16 | 1994-06-15 | 1994-09-21 | 1994-12-21 | 1995-03-15 |   |
| 1993-09-17 | 1993-12-15 | 1994-03-16 | 1994-06-15 | 1994-09-21 | 1994-12-21 | 1995-03-15 |   |
| 1993-09-20 | 1993-12-15 | 1994-03-16 | 1994-06-15 | 1994-09-21 | 1994-12-21 | 1995-03-15 |   |

  

|            | ED7        | ED8        | ED9        | ED10       | ... | ED12       | \ |
|------------|------------|------------|------------|------------|-----|------------|---|
| Date       |            |            |            |            | ... |            |   |
| 1993-09-14 | 1995-03-15 | 1995-06-21 | 1995-09-20 | 1995-12-20 | ... | 1996-06-19 |   |
| 1993-09-15 | 1995-06-21 | 1995-09-20 | 1995-12-20 | 1996-03-20 | ... | 1996-09-18 |   |
| 1993-09-16 | 1995-06-21 | 1995-09-20 | 1995-12-20 | 1996-03-20 | ... | 1996-09-18 |   |
| 1993-09-17 | 1995-06-21 | 1995-09-20 | 1995-12-20 | 1996-03-20 | ... | 1996-09-18 |   |
| 1993-09-20 | 1995-06-21 | 1995-09-20 | 1995-12-20 | 1996-03-20 | ... | 1996-09-18 |   |

|            | ED13       | ED14       | ED15       | ED16       | ED17       | ED18       | \ |
|------------|------------|------------|------------|------------|------------|------------|---|
| Date       |            |            |            |            |            |            |   |
| 1993-09-14 | 1996-09-18 | 1996-12-18 | 1997-03-19 | 1997-06-18 | 1997-09-17 | 1997-12-17 |   |
| 1993-09-15 | 1996-12-18 | 1997-03-19 | 1997-06-18 | 1997-09-17 | 1997-12-17 | 1998-03-18 |   |
| 1993-09-16 | 1996-12-18 | 1997-03-19 | 1997-06-18 | 1997-09-17 | 1997-12-17 | 1998-03-18 |   |
| 1993-09-17 | 1996-12-18 | 1997-03-19 | 1997-06-18 | 1997-09-17 | 1997-12-17 | 1998-03-18 |   |
| 1993-09-20 | 1996-12-18 | 1997-03-19 | 1997-06-18 | 1997-09-17 | 1997-12-17 | 1998-03-18 |   |

|            | ED19       | ED20       | ED21       |
|------------|------------|------------|------------|
| Date       |            |            |            |
| 1993-09-14 | 1998-03-18 | 1998-06-17 | 1998-09-16 |
| 1993-09-15 | 1998-06-17 | 1998-09-16 | 1998-12-16 |
| 1993-09-16 | 1998-06-17 | 1998-09-16 | 1998-12-16 |
| 1993-09-17 | 1998-06-17 | 1998-09-16 | 1998-12-16 |
| 1993-09-20 | 1998-06-17 | 1998-09-16 | 1998-12-16 |

[5 rows x 21 columns]

```
In [35]: # calculate each contract's time to maturity
day_diff_actual = pd.DataFrame(index = euro_dollar.index, columns=euro_dollar.columns)
for i in range(21):
    day_diff = (maturity_date_table.iloc[:,i].astype('datetime64[D]') - maturity_date)
    day_diff = [day_diff[j].days for j in range(len(day_diff))]
    day_diff_actual.iloc[:,i] = day_diff
```

```
In [36]: day_diff_actual.head()
```

```
Out [36]:
```

|            | ED1 | ED2 | ED3 | ED4 | ED5 | ED6 | ED7 | ED8 | ED9 | ED10 | ... | ED12 | \ |
|------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|-----|------|---|
| Date       |     |     |     |     |     |     |     |     |     |      | ... |      |   |
| 1993-09-14 | 1   | 92  | 183 | 274 | 372 | 463 | 547 | 645 | 736 | 827  | ... | 1009 |   |
| 1993-09-15 | 91  | 182 | 273 | 371 | 462 | 546 | 644 | 735 | 826 | 917  | ... | 1099 |   |
| 1993-09-16 | 90  | 181 | 272 | 370 | 461 | 545 | 643 | 734 | 825 | 916  | ... | 1098 |   |
| 1993-09-17 | 89  | 180 | 271 | 369 | 460 | 544 | 642 | 733 | 824 | 915  | ... | 1097 |   |
| 1993-09-20 | 86  | 177 | 268 | 366 | 457 | 541 | 639 | 730 | 821 | 912  | ... | 1094 |   |

|            | ED13 | ED14 | ED15 | ED16 | ED17 | ED18 | ED19 | ED20 | ED21 |
|------------|------|------|------|------|------|------|------|------|------|
| Date       |      |      |      |      |      |      |      |      |      |
| 1993-09-14 | 1100 | 1191 | 1282 | 1373 | 1464 | 1555 | 1646 | 1737 | 1828 |
| 1993-09-15 | 1190 | 1281 | 1372 | 1463 | 1554 | 1645 | 1736 | 1827 | 1918 |
| 1993-09-16 | 1189 | 1280 | 1371 | 1462 | 1553 | 1644 | 1735 | 1826 | 1917 |
| 1993-09-17 | 1188 | 1279 | 1370 | 1461 | 1552 | 1643 | 1734 | 1825 | 1916 |
| 1993-09-20 | 1185 | 1276 | 1367 | 1458 | 1549 | 1640 | 1731 | 1822 | 1913 |

[5 rows x 21 columns]

```
In [37]: #c = [91,90,92,92]
#a = 0
#constant = []
#for i in range(21):
```

```

    #a+=c[(i)%4]
    #constant.append(a)
constant = [90*(i+1) for i in range(21)]
constant = np.array(constant)
day_constant = pd.DataFrame(index = euro_dollar.index, columns=euro_dollar.columns)
for i in range(len(day_constant)):
    day_constant.iloc[i,:] = constant
day_constant.head()

```

```

Out[37]:
           ED1  ED2  ED3  ED4  ED5  ED6  ED7  ED8  ED9  ED10  ...  ED12  \
Date
1993-09-14   90  180  270  360  450  540  630  720  810   900  ...  1080
1993-09-15   90  180  270  360  450  540  630  720  810   900  ...  1080
1993-09-16   90  180  270  360  450  540  630  720  810   900  ...  1080
1993-09-17   90  180  270  360  450  540  630  720  810   900  ...  1080
1993-09-20   90  180  270  360  450  540  630  720  810   900  ...  1080

```

```

           ED13  ED14  ED15  ED16  ED17  ED18  ED19  ED20  ED21
Date
1993-09-14  1170  1260  1350  1440  1530  1620  1710  1800  1890
1993-09-15  1170  1260  1350  1440  1530  1620  1710  1800  1890
1993-09-16  1170  1260  1350  1440  1530  1620  1710  1800  1890
1993-09-17  1170  1260  1350  1440  1530  1620  1710  1800  1890
1993-09-20  1170  1260  1350  1440  1530  1620  1710  1800  1890

```

[5 rows x 21 columns]

```

In [38]: yield_curve = pd.DataFrame(index= euro_dollar.index, columns = euro_dollar.columns)
        for i in range(len(euro_dollar)):
            yield_curve.iloc[i,:] = np.interp(day_constant.iloc[i,:].values, day_diff_actual.

```

```

In [39]: yield_curve = yield_curve.dropna()

```

```

In [40]: yield_curve.head()

```

```

Out[40]:
           ED1      ED2      ED3      ED4      ED5      ED6      ED7  \
Date
1993-09-14  3.58868  3.79308  4.04857  4.38469      4.49  4.68333  4.84398
1993-09-15      3.48  3.5289  3.73308  3.97082  4.32121      4.435      4.62
1993-09-16      3.49  3.53945  3.75516  3.98449  4.32648  4.43583  4.62214
1993-09-17  3.49066      3.55  3.77747  4.00704  4.35044  4.45667  4.64429
1993-09-20  3.49308  3.56791  3.80531  4.04408  4.40154  4.49917  4.68163

           ED8      ED9      ED10  ...      ED12      ED13      ED14  \
Date
1993-09-14  5.10077  5.18253  5.31033  ...      5.62945  5.67231  5.76099
1993-09-15  4.80033  5.06901  5.15253  ...      5.39703  5.60725      5.66
1993-09-16  4.80231  5.0722  5.15297  ...      5.40824  5.61989      5.67
1993-09-17  4.81571  5.08538  5.16341  ...      5.41945  5.63253      5.68

```

|            |         |         |         |         |         |         |         |
|------------|---------|---------|---------|---------|---------|---------|---------|
| 1993-09-20 | 4.84242 | 5.11495 | 5.18473 | ...     | 5.44308 | 5.66044 | 5.7     |
|            | ED15    | ED16    | ED17    | ED18    | ED19    | ED20    | ED21    |
| Date       |         |         |         |         |         |         |         |
| 1993-09-14 | 5.84231 | 5.99989 | 6.02824 | 6.07    | 6.1111  | 6.23077 | 6.28    |
| 1993-09-15 | 5.75099 | 5.83231 | 5.98989 | 6.01824 | 6.06    | 6.1011  | 6.22077 |
| 1993-09-16 | 5.76231 | 5.84308 | 6.00945 | 6.03791 | 6.08077 | 6.12143 | 6.24956 |
| 1993-09-17 | 5.77363 | 5.85385 | 6.02165 | 6.04758 | 6.09154 | 6.13176 | 6.26143 |
| 1993-09-20 | 5.79758 | 5.87615 | 6.04824 | 6.06659 | 6.11385 | 6.16033 | 6.29703 |

[5 rows x 21 columns]

In [ ]: