discount_factor

January 22, 2018

```
In [253]: import csv
          import datetime

          class discount_factor:
              def __init__(self, ir_list_name):
                  ## コンストラクタの順番に注意．　先に_ir_listを定義し，　_load_ir_listの中で_base_
                  ## を呼び出すと，　_base_dateがまだ定義されていないのでエラーがでる．
                  self._base_date = ir_list_name[0:4] + '/' + ir_list_name[4:6] + '/' + ir_list_
                  self._ir_list = self._load_ir_list(ir_list_name)

              def _load_ir_list(self, ir_list_name):
                  with open(ir_list_name, 'r') as csvfile:
                      reader_obj = csv.reader(csvfile)
                      # rewritten header_obj by using next method(???)
                      header_obj = next(reader_obj)
                      ir_list = []
                      for row in reader_obj:
                          ir_list.append(row)
                      temp_num = [[] for i in range(len(ir_list))] # comprehension expression fo
                      for i in range(len(ir_list)):
                          if (ir_list[i][0][0].isdigit()):
                              num_tenor = ir_list[i][0][0: len(ir_list[i][0])-1]
                              unit_tenor = ir_list[i][0][-1]
                              temp_num[i] = "{:.1f}".format(int(num_tenor))
                              ir_list[i][0] = temp_num[i] + unit_tenor
                  ir_list_with_cf = self._add_cash_flow(ir_list)
                  return ir_list_with_cf

              def _add_cash_flow(self, ir_list):
                  obj_trade_date = datetime.datetime.strptime(self._base_date, '%Y/%m/%d')
                  over_night_date = (obj_trade_date + datetime.timedelta(days=1)).strftime('%Y/%
                  spot_date = (obj_trade_date + datetime.timedelta(days=2)).strftime('%Y/%m/%d')
                  for i in range(len(ir_list)):
                      if (ir_list[i][0] == 'O/N'):
                          ir_list[i].append(self._base_date)
                          ir_list[i].append(over_night_date)
                      elif (ir_list[i][0] == 'T/N'):
                          ir_list[i].append(over_night_date)
```

```
                ir_list[i].append(spot_date)
            else:
                ir_list[i].append(spot_date)
                ir_list[i].append(self._calc_end_date(self._base_date, ir_list[i][0]))
        return ir_list

    def _calc_end_date(self, start_day, str_maturity):
        datetime_obj_start = datetime.datetime.strptime(start_day, '%Y/%m/%d')
        unit = str_maturity[-1]
        int_num = int(str_maturity[0:len(str_maturity)-3])
        if (unit == 'W'):
            trade_days = int_num * 7
        elif (unit == 'M'):
            trade_days = int_num * 30
        elif (unit == 'Y'):
            trade_days = int_num * 365
        end_day = datetime_obj_start + datetime.timedelta(days=trade_days)
        return end_day.strftime('%Y/%m/%d')

    def get_ir_list(self):
        return self._ir_list

    def get_base_date(self):
        return self._base_date

#TODO implement make list for interpolation and interpolate swap rate
#TODO implement calc DF for Money Market
```

```
In [252]: discount_factor('20180118_IR.csv').get_ir_list()
```

```
Out[252]: [['O/N', 'Money Market', 'USD', '0.014375', '', '2018/01/18', '2018/01/19'],
           ['1.0W', 'Money Market', 'USD', '0.0146533', '', '2018/01/20', '2018/01/25'],
           ['1.0M', 'Money Market', 'USD', '0.0156118', '', '2018/01/20', '2018/02/17'],
           ['2.0M', 'Money Market', 'USD', '0.0163482', '', '2018/01/20', '2018/03/19'],
           ['3.0M', 'Money Market', 'USD', '0.017447', '', '2018/01/20', '2018/04/18'],
           ['6.0M', 'Money Market', 'USD', '0.019255', '', '2018/01/20', '2018/07/17'],
           ['1.0Y', 'Swap', 'USD', '0.02045', '6M', '2018/01/20', '2019/01/18'],
           ['2.0Y', 'Swap', 'USD', '0.02257', '6M', '2018/01/20', '2020/01/18'],
           ['3.0Y', 'Swap', 'USD', '0.02366', '6M', '2018/01/20', '2021/01/17'],
           ['4.0Y', 'Swap', 'USD', '0.02427', '6M', '2018/01/20', '2022/01/17'],
           ['5.0Y', 'Swap', 'USD', '0.02468', '6M', '2018/01/20', '2023/01/17'],
           ['6.0Y', 'Swap', 'USD', '0.02504', '6M', '2018/01/20', '2024/01/17'],
           ['7.0Y', 'Swap', 'USD', '0.02536', '6M', '2018/01/20', '2025/01/16'],
           ['8.0Y', 'Swap', 'USD', '0.02565', '6M', '2018/01/20', '2026/01/16'],
           ['9.0Y', 'Swap', 'USD', '0.02591', '6M', '2018/01/20', '2027/01/16'],
           ['10.0Y', 'Swap', 'USD', '0.02615', '6M', '2018/01/20', '2028/01/16'],
           ['15.0Y', 'Swap', 'USD', '0.0269', '6M', '2018/01/20', '2033/01/14'],
           ['20.0Y', 'Swap', 'USD', '0.02722', '6M', '2018/01/20', '2038/01/13'],
           ['30.0Y', 'Swap', 'USD', '0.02715', '6M', '2018/01/20', '2048/01/11']]
```

In [201]:
```
int('{:.0f}'.format('1.0'))
```

```
  ---------------------------------------------------------------------------

  ValueError                                Traceback (most recent call last)

  <ipython-input-201-6fdf4566b10c> in <module>()
----> 1 int('{:.0f}'.format('1.0'))


  ValueError: Unknown format code 'f' for object of type 'str'
```

In [70]:
```
  def get_ir_list(ir_list_name):
    with open(ir_list_name, 'r') as csvfile:
      reader_obj = csv.reader(csvfile)
      # rewritten header_obj by using next method(???)
      header_obj = next(reader_obj)
      ir_list = []
      for row in reader_obj:
        ir_list.append(row)
      temp_num = [[] for i in range(len(ir_list))] # comprehension expression for
      for i in range(len(ir_list)):
        if (ir_list[i][0][0].isdigit()):
          num_tenor = ir_list[i][0][0: len(ir_list[i][0])-1]
          unit_tenor = ir_list[i][0][-1]
          temp_num[i] = "{:.1f}".format(int(num_tenor))
          ir_list[i][0] = temp_num[i] + unit_tenor
    return ir_list
```

In [71]:
```
get_ir_list('20180118_IR.csv')
```

Out[71]:
```
[['O/N', 'Money Market', 'USD', '0.014375', ''],
 ['1.0W', 'Money Market', 'USD', '0.0146533', ''],
 ['1.0M', 'Money Market', 'USD', '0.0156118', ''],
 ['2.0M', 'Money Market', 'USD', '0.0163482', ''],
 ['3.0M', 'Money Market', 'USD', '0.017447', ''],
 ['6.0M', 'Money Market', 'USD', '0.019255', ''],
 ['1.0Y', 'Swap', 'USD', '0.02045', '6M'],
 ['2.0Y', 'Swap', 'USD', '0.02257', '6M'],
 ['3.0Y', 'Swap', 'USD', '0.02366', '6M'],
 ['4.0Y', 'Swap', 'USD', '0.02427', '6M'],
 ['5.0Y', 'Swap', 'USD', '0.02468', '6M'],
 ['6.0Y', 'Swap', 'USD', '0.02504', '6M'],
 ['7.0Y', 'Swap', 'USD', '0.02536', '6M'],
 ['8.0Y', 'Swap', 'USD', '0.02565', '6M'],
 ['9.0Y', 'Swap', 'USD', '0.02591', '6M'],
 ['10.0Y', 'Swap', 'USD', '0.02615', '6M'],
 ['15.0Y', 'Swap', 'USD', '0.0269', '6M'],
 ['20.0Y', 'Swap', 'USD', '0.02722', '6M'],
 ['30.0Y', 'Swap', 'USD', '0.02715', '6M']]
```

```
In [55]: import csv

         with open('20180118_IR.csv', 'r') as csvfile:
             reader_obj = csv.reader(csvfile)
             # rewritten header_obj by using next method(???)
             header_obj = next(reader_obj)
             ir_list = []
             for row in reader_obj:
                 ir_list.append(row)
             temp_num = [[] for i in range(len(ir_list))] # comprehension expression for making
             for i in range(len(ir_list)):
                 if (ir_list[i][0][0].isdigit()):
                     num_tenor = ir_list[i][0][0: len(ir_list[i][0])-1]
                     unit_tenor = ir_list[i][0][-1]
                     temp_num[i] = "{:.1f}".format(int(num_tenor))
                     ir_list[i][0] = temp_num[i] + unit_tenor

         ir_list
```

```
Out[55]: [['O/N', 'Money Market', 'USD', '0.014375', ''],
          ['1.0W', 'Money Market', 'USD', '0.0146533', ''],
          ['1.0M', 'Money Market', 'USD', '0.0156118', ''],
          ['2.0M', 'Money Market', 'USD', '0.0163482', ''],
          ['3.0M', 'Money Market', 'USD', '0.017447', ''],
          ['6.0M', 'Money Market', 'USD', '0.019255', ''],
          ['1.0Y', 'Swap', 'USD', '0.02045', '6M'],
          ['2.0Y', 'Swap', 'USD', '0.02257', '6M'],
          ['3.0Y', 'Swap', 'USD', '0.02366', '6M'],
          ['4.0Y', 'Swap', 'USD', '0.02427', '6M'],
          ['5.0Y', 'Swap', 'USD', '0.02468', '6M'],
          ['6.0Y', 'Swap', 'USD', '0.02504', '6M'],
          ['7.0Y', 'Swap', 'USD', '0.02536', '6M'],
          ['8.0Y', 'Swap', 'USD', '0.02565', '6M'],
          ['9.0Y', 'Swap', 'USD', '0.02591', '6M'],
          ['10.0Y', 'Swap', 'USD', '0.02615', '6M'],
          ['15.0Y', 'Swap', 'USD', '0.0269', '6M'],
          ['20.0Y', 'Swap', 'USD', '0.02722', '6M'],
          ['30.0Y', 'Swap', 'USD', '0.02715', '6M']]
```