FINANCIAL FORECASTING IN PYTHON

Financial periods and how to work with them

Victoria Clark
CGMA Financial Analyst

The financial year

- Full reporting year for financial numbers
- Can start and end at any month of the year
    - e.g., Microsoft Financial Year is 1 July - 30 June


2017
1ST QUARTER
January
S M T W T F S
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31
February
S M T W T F S
1 2 3 4
5 6 7 8 9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28
March
S M T W T F S
1 2 3 4
5 6 7 8 9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31
2ND QUARTER
April
S M T W T F S
1
2 3 4 5 6 7 8
9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30
May
S M T W T F S
1 2 3 4 5 6
7 8 9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31
June
S M T W T F S
1 2 3
4 5 6 7 8 9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30
3RD QUARTER
July
S M T W T F S
1
2 3 4 5 6 7 8
9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30 31
August
S M T W T F S
1 2 3 4 5
6 7 8 9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31
September
S M T W T F S
1 2
3 4 5 6 7 8 9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
4TH QUARTER
October
S M T W T F S
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31
November
S M T W T F S
1 2 3 4
5 6 7 8 9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30
December
S M T W T F S
1 2
3 4 5 6 7 8 9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
31

Abbreviations

- Months
  - 01, 02 , 03 or Jan, Feb, Mar
  - Not dependent on financial year
- Quarters
  - Q1, Q2, Q3, Q4
  - Dependent on financial year
- Years
  - 2017, 2018 or 18, 17
  - Year is set based on the financial year end

FINANCIAL FORECASTING IN PYTHON

Let's practice!

FINANCIAL FORECASTING IN PYTHON

The datetime library and Split function

Victoria Clark
CGMA Financial Analyst

Types of conflicts

Date: 09/10/2018

Regional differences

- Day-Month-Year
- Month-Day-Year

Punctuation differences

- dd-mm-yy
- dd/mm/yyyy

The datetime function

```
# Import datetime module
from datetime import datetime

datetime.strptime(date_string,
                  format)
```

12-25-2000 --> %m-%d-%Y'

```
datetime.strptime('12-25-2000',
                  '%m-%d-%Y')

print(dt_object.month) = 12
```

Directive	Meaning	Example
%d	Day of the month as a zero-padded decimal number	01, 02, ..., 31
%b	Month as locale's abbreviated name	Jan, Feb, ..., Dec
%B	Month as locale's full name	January, ..., December
%m	Month as a zero-padded decimal number	01, 02, ..., 12
%y	Year without century as a zero-padded decimal number	00, 01, ..., 99
%Y	Year with century as a decimal number	1970, 1988, 2001, 2013

Using the split() function

- split() function

```
date = '14/02/2018'

# Split date string into named variables using /
day, month, year = date.split('/')

print(year) = 2018
```

FINANCIAL FORECASTING IN PYTHON

Let's practice!

FINANCIAL FORECASTING IN PYTHON

Tips and tricks when working with datasets

Victoria Clark
CGMA Financial Analyst

Common challenges when working with financial data

- Raw data is in different formats
- Date format can be different!
  - US format is Month-Day-Year, EU is Day-Month-Year
  - 09-08-2018 in US is the 8th of September, EU is 9th of August
- Can cause challenges in:
  - Interpreting
  - Combining

Using a dictionary

- **Dictionary:** associative array
- Keys are mapped to values
- un-ordered key-value-pairs

For example:

- The value 01 has a key of Jan

```
dictionary = {01: 'Jan'}
```

Remember to use the datetime library

Directive	Meaning	Example
%d	Day of the month as a zero-padded decimal number	01, 02, ..., 31
%b	Month as locale’s abbreviated name	Jan, Feb, ..., Dec
%B	Month as locale’s full name	January, ...,
%m	Month as a zero-padded decimal number	01, 02, ..., 12
%y	Year without century as a zero-padded decimal number	00, 01, ..., 99
%Y	Year with century as a decimal number	1988, 2001, 2013

```
# Example: 19-02-2018 will be written:
('19-02-2018', '%d-%m-%Y')
```

Iterate over items

- iteritems() function

```
# Create dictionary with strings as keys and ints as values
wordFrequency = {
    "Hello" : 7,
    "hi" : 10,
    "there" : 45,
    "at" : 23,
    "this" : 77
    }

# Iterate over dictionary using for loop
for key in wordFrequency:
    value = wordFrequency[key]
    print(key, " :: ", value)
```

```
Hello  ::  7
there  ::  45
at  ::  23
this  ::  77
hi  ::  10
```

FINANCIAL FORECASTING IN PYTHON

Let's practice!