assign3

May 4, 2020

[94]:
```
import numpy as np
import pandas as pd
from pandas_datareader import data
```

1 Assignment 3

1.1 Question 1

Create a DataFrame with the info given on the assignment

[12]:
```
data = [[155,955,66,37.10,32.0,30.31],[150,987,69,36.98,31.3,30.
 ↪56],[153,963,62,36.78,31.7,30.46],[155,1000,61,36.11,31.2,30.
 ↪11],[156,1012,66,37.07,30.0,31.00]]
columns = pd.MultiIndex.from_product([['Price','Price to earning ratio (P/
 ↪E)'],['Facebook','Google','Microsoft']])
index = pd.Index(pd.to_datetime('2017-06-05') + pd.to_timedelta(np.arange(5),␣
 ↪'D'),name='date')
ans1 = pd.DataFrame(data=data,columns=columns,index=index)
display(ans1)
```

```
              Price                     Price to earning ratio (P/E)         \
           Facebook Google Microsoft                         Facebook Google
date
2017-06-05      155    955        66                            37.10   32.0
2017-06-06      150    987        69                            36.98   31.3
2017-06-07      153    963        62                            36.78   31.7
2017-06-08      155   1000        61                            36.11   31.2
2017-06-09      156   1012        66                            37.07   30.0


           Microsoft
date
2017-06-05     30.31
2017-06-06     30.56
2017-06-07     30.46
2017-06-08     30.11
2017-06-09     31.00
```

1.2 Question 2

Set the index to be the date, then show info for each date and company

```
[14]: ans2 = ans1.stack()
      ans2
```

```
[14]:                          Price  Price to earning ratio (P/E)
      date
      2017-06-05 Facebook      155                         37.10
                 Google        955                         32.00
                 Microsoft      66                         30.31
      2017-06-06 Facebook      150                         36.98
                 Google        987                         31.30
                 Microsoft      69                         30.56
      2017-06-07 Facebook      153                         36.78
                 Google        963                         31.70
                 Microsoft      62                         30.46
      2017-06-08 Facebook      155                         36.11
                 Google       1000                         31.20
                 Microsoft      61                         30.11
      2017-06-09 Facebook      156                         37.07
                 Google       1012                         30.00
                 Microsoft      66                         31.00
```

1.3 Question 3

Find the average price and P/E per stock name

```
[36]: ans3 = ans1.unstack().groupby(level=[0,1]).mean()
      ans3
```

```
[36]: Price                         Facebook     153.800
                                    Google       983.400
                                    Microsoft     64.800
      Price to earning ratio (P/E)  Facebook      36.808
                                    Google        31.240
                                    Microsoft     30.488
      dtype: float64
```

1.4 Question 4

Consider a scenario where John is 20, Bob is 30, and Suzan is 22. Suppose that there are three courses: CS 233, CS 455, and ENGL 433. Next, suppose that John took CS 233 and got a C, took CS 455 and got a B, and Suzan took ENGL 433 and got an A. Create three data frames. The student data frame should store the student name and age. The course data frame should store the department, course number, and description. Finally, the takes data frame should store the student name, department name, course number, and grade. You can assume that each student has unique name.

[49]:
```
student = pd.DataFrame.from_dict({'John':[20],'Bob':[30],'Suzan':
↪[22]},orient='index',columns=['Age'])
student.index.name = 'Name'
student
```

[49]:
```
       Age
Name
John    20
Bob     30
Suzan   22
```

[51]:
```
course = pd.DataFrame.from_dict({'CS 233':['Comp Sci',233,'OOP'],'CS 455':
↪['Comp Sci',455,'Deep Learning'],'ENGL 433':['English',433,'British␣
↪Literature']},orient='index')
course.index.name = 'Course'
course.columns = ['Department','Course Number','Description']
course
```

[51]:
```
         Department  Course Number         Description
Course
CS 233     Comp Sci            233                 OOP
CS 455     Comp Sci            455       Deep Learning
ENGL 433    English            433  British Literature
```

[52]:
```
takes = pd.DataFrame([['John','Comp Sci',233,'C'],['John','Comp␣
↪Sci',455,'B'],['Suzan','English',433,'A']])
takes.columns = ['Name','Department','Course Number','Grade']
takes
```

[52]:
```
    Name Department  Course Number Grade
0   John   Comp Sci            233     C
1   John   Comp Sci            455     B
2  Suzan    English            433     A
```

1.5 Question 5

Find the GPA for each student, with 0 for a student that has no classes

[73]:
```
grade_number = {'A':4.0,'B':3.0,'C':2.0,'D':1.0}
gpa_totals = {person : 0.0 for person in list(student.index)}
def add_row_to_gpa(row):
    gpa_totals[row['Name']] = (gpa_totals[row['Name']] +␣
↪grade_number[row['Grade']]) / 2 if gpa_totals[row['Name']] != 0 else␣
↪grade_number[row['Grade']]
takes.apply(add_row_to_gpa,axis=1)
```

```
gpa_totals
```

```
[73]: {'John': 2.5, 'Bob': 0.0, 'Suzan': 4.0}
```

1.6 Question 6

Print the name of studnets that have taken no classes

```
[81]: for person in student.index:
          if person not in list(takes['Name']):
              print(person)
```

```
Bob
```

1.7 Question 7

Create a Series, with the index being all business days in 2018 and the numbers from 0 to 260 as values.

```
[87]: ans7 = pd.Series(data=range(261),index=pd.
       ↪date_range(start='2018',end='2019',freq='B')[:-1])
      ans7
```

```
[87]: 2018-01-01      0
      2018-01-02      1
      2018-01-03      2
      2018-01-04      3
      2018-01-05      4
                     …
      2018-12-25    256
      2018-12-26    257
      2018-12-27    258
      2018-12-28    259
      2018-12-31    260
      Freq: B, Length: 261, dtype: int64
```

1.8 Question 8

Create a DataFrame with the number of each day of the week in 2018.

```
[108]: s = pd.date_range('2018','2019',freq='D')[:-1].to_series()
       ans8 = s.dt.dayofweek.value_counts().sort_index()
       ans8.index =␣
        ↪['Monday','Tuesday','Wednesday','Thursday','Friday','Saturday','Sunday']
       ans8
```

```
[108]: Monday       53
       Tuesday      52
```

```
Wednesday    52
Thursday     52
Friday       52
Saturday     52
Sunday       52
dtype: int64
```

1.9 Question 9

Which day of the week is the most profitable for the GOOG stock in 2017? Compute the difference between the opneing and closing price for each day of the week and sum over the whole year.

[115]:
```
goog = data.DataReader('GOOG','yahoo',start='2017',end='2018')
goog = goog.groupby(goog.index.dayofweek).sum()
goog.index = ['Monday','Tuesday','Wednesday','Thursday','Friday']
goog.index.name = 'DOW'

ans9 = pd.DataFrame(data={'Profit':goog['Close'] - goog['Open']},index=goog.
 ↪index).sort_values(by='Profit',ascending=False)
ans9
```

[115]:
```
              Profit
DOW
Wednesday  77.895081
Tuesday    42.460205
Monday     38.770020
Thursday    5.105225
Friday      1.340271
```

[ ]:
```

```