Programming Assignment 2

1) Import the pandas library as pd, then use the read_csv function to import the prices15 csv file, make sure that you parse_dates and also define as the column index the 'date' column. Print the first 3 records and provide more information on the dataframe with the info method

1 Point

```
           symbol        open       close         low        high      volume
date
2015-01-02      A   41.180000   40.560001   40.369999   41.310001   1529200.0
2015-01-02    AAL   54.279999   53.910000   53.070000   54.599998  10748600.0
2015-01-02    AAP  160.850006  158.559998  157.470001  162.500000    509800.0
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 126125 entries, 2015-01-02 to NaT
Data columns (total 6 columns):
symbol    124957 non-null object
open      124957 non-null float64
close     124957 non-null float64
low       124957 non-null float64
high      124957 non-null float64
volume    124957 non-null float64
dtypes: float64(5), object(1)
memory usage: 6.7+ MB
None
```

2)

Create a DataFrame called 'google' by filtering only records where the symbol is equal to 'GOOGL' and print the last 3 records of the DataFrame

Create a DataFrame called 'apple' by filtering only records where the symbol is equal to 'AAPL' and print the first 3 records of the DataFrame

1 Point

```
           symbol        open       close         low        high     volume
date
2015-12-29  GOOGL  786.989990  793.960022  786.200012  798.690002  1921500.0
2015-12-30  GOOGL  793.960022  790.299988  787.200012  796.460022  1428300.0
2015-12-31  GOOGL  787.820007  778.010010  777.320007  788.330017  1629300.0
```

```
           symbol        open       close         low        high      volume
date
2015-01-02   AAPL  111.389999  109.330002  107.349998  111.440002  53204600.0
2015-01-05   AAPL  108.290001  106.250000  105.410004  108.650002  64285500.0
2015-01-06   AAPL  106.540001  106.260002  104.629997  107.430000  65797100.0
```

3) Stack both dataframes to create a DataFrame of Apple and Google. Name this new DataFrame apple_google. ¶

Check that you performed this operations correctly by printing the unique values associated with the column symbol (['GOOGL' 'AAPL']). Print the info of the DataFrame and in a commment in your code write the size of the DataFrame (memory usage)

1 Point

```
['GOOGL' 'AAPL']
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 504 entries, 2015-01-02 to 2015-12-31
Data columns (total 6 columns):
symbol    504 non-null object
open      504 non-null float64
close     504 non-null float64
low       504 non-null float64
high      504 non-null float64
volume    504 non-null float64
dtypes: float64(5), object(1)
memory usage: 27.6+ KB
```

4) Print a DataFrame using conditonal statements where the volume > 2200000 AND the open price is > 770 AND the ticker symbol = 'GOOGL'

1 Point

date	symbol	open	close	low	high	volume
2015-11-24	GOOGL	772.960022	769.630005	758.109985	775.989990	2394000.0
2015-12-02	GOOGL	785.250000	777.849976	776.429993	793.049988	2343800.0
2015-12-09	GOOGL	771.099976	762.549988	752.010010	776.090027	2327500.0

5)

0.33 points print the average volume for the apple_google DataFrame

0.33 points print the max close price of the apple_google DataFrame

0.33 points print the median open price of apple_google filtering for Google only

```
27003624.404761903
```

```
793.960022
```

```
568.2200015
```

6) Create another column in the apple_google DataFrame named diff_daily_price by substracting the low price from the high price. Print the first 3 records of the updated DataFrame

1 Points

```
           symbol        open       close         low        high     volume  \
date
2015-01-02  GOOGL  532.599976  529.549988  527.880005  535.799988  1324000.0
2015-01-05  GOOGL  527.150024  519.460022  517.750000  527.989990  2059100.0
2015-01-06  GOOGL  520.500000  506.640015  505.549988  521.210022  2722800.0

            diff_daily_price  google_max       date  year  month  day
date
2015-01-02          7.919983  798.690002 2015-01-02  2015      1    2
2015-01-05         10.239990  798.690002 2015-01-05  2015      1    5
2015-01-06         15.660034  798.690002 2015-01-06  2015      1    6
```

7) Create another column in the apple_google DataFrame named google_max that displays the max value of the high column associated with GOOGL only (798.69). Print the 3 first three records of that column

1 point

```
date
2015-01-02    798.690002
2015-01-05    798.690002
2015-01-06    798.690002
Name: google_max, dtype: float64
```

8) Using one of the methods learned in class or in the Week_Three_Pandas notebook, create another column called 'date' that is equal to the current index in the apple_google DataFrame. Print the first 3 records of the DataFrame

1 point

date	symbol	open	close	low	high	volume	date
2015-01-02	GOOGL	532.599976	529.549988	527.880005	535.799988	1324000.0	2015-01-02
2015-01-05	GOOGL	527.150024	519.460022	517.750000	527.989990	2059100.0	2015-01-05
2015-01-06	GOOGL	520.500000	506.640015	505.549988	521.210022	2722800.0	2015-01-06

9) Create three more columns ('year'), ('month') and ('day') using a pandas to extract the year, month and day from the date column. Print the first three records of the updated DataFrame

1 point

date	symbol	open	close	low	high	volume	date	year	month	day
2015-01-02	GOOGL	532.599976	529.549988	527.880005	535.799988	1324000.0	2015-01-02	2015	1	2
2015-01-05	GOOGL	527.150024	519.460022	517.750000	527.989990	2059100.0	2015-01-05	2015	1	5
2015-01-06	GOOGL	520.500000	506.640015	505.549988	521.210022	2722800.0	2015-01-06	2015	1	6

10) Using the groupby method, calculate and display the mean value of the open price of both Apple and Google by month

1 point

```
month
1     313.822003
2     332.100262
3     346.410679
4     338.715953
5     338.375000
6     339.608860
7     370.623410
8     391.851665
9     380.950954
10    401.722502
11    438.387500
12    441.830228
Name: open, dtype: float64
```

11) Using the groupby method, calculate and display the max value of the open column by day from 1 to 31 for Apple only.

1 point

```
day
1     130.279999
2     129.860001
3     130.660004
4     129.580002
5     129.500000
6     128.399994
7     127.639999
8     128.899994
9     127.959999
10    127.919998
11    129.179993
12    128.190002
13    128.369995
14    127.410004
15    129.070007
```

12) Create a DataFrame called securities by importing the securities.csv file using the read_csv function. Do not customize the import. Print the first three records of the securities DataFrame

1 point

```
  Ticker symbol             Security SEC filings  GICS Sector  \
0           MMM           3M Company     reports  Industrials
1           ABT  Abbott Laboratories     reports  Health Care
2          ABBV               AbbVie     reports  Health Care

          GICS Sub Industry  Address of Headquarters Date first added      CIK
0  Industrial Conglomerates      St. Paul, Minnesota              NaN    66740
1     Health Care Equipment  North Chicago, Illinois       1964-03-31     1800
2           Pharmaceuticals  North Chicago, Illinois       2012-12-31  1551152
```

13) Merge the prices15 DataFrame you created in question one with the securities DataFrame you just imported in question 12. Analyze both DataFrames to identify the column from each DataFrame to do the join on. Create a new DataFrame called prices15_securities by merging (inner join) prices15 and securities. Print the first 3 records

1 point

```
  symbol       open      close        low       high     volume Ticker symbol  \
0      A  41.180000  40.560001  40.369999  41.310001  1529200.0             A
1      A  40.320000  39.799999  39.700001  40.459999  2041800.0             A
2      A  39.810001  39.180000  39.020000  40.020000  2080600.0             A

                  Security SEC filings  GICS Sector      GICS Sub Industry  \
0  Agilent Technologies Inc     reports  Health Care  Health Care Equipment
1  Agilent Technologies Inc     reports  Health Care  Health Care Equipment
2  Agilent Technologies Inc     reports  Health Care  Health Care Equipment

   Address of Headquarters Date first added      CIK
0  Santa Clara, California              NaN  1090872
1  Santa Clara, California              NaN  1090872
2  Santa Clara, California              NaN  1090872
```

14) You will notice that once the merged occured, the symbol and the Ticker symbol columns have the exact same data. Drop the Ticker symbol column and also change the name of the column symbol to company_symbol. Print the first 3 records

```
  company_symbol       open      close        low       high     volume  \
0              A  41.180000  40.560001  40.369999  41.310001  1529200.0
1              A  40.320000  39.799999  39.700001  40.459999  2041800.0
2              A  39.810001  39.180000  39.020000  40.020000  2080600.0

  Ticker symbol                  Security SEC filings  GICS Sector  \
0             A  Agilent Technologies Inc     reports  Health Care
1             A  Agilent Technologies Inc     reports  Health Care
2             A  Agilent Technologies Inc     reports  Health Care

       GICS Sub Industry  Address of Headquarters Date first added      CIK
0  Health Care Equipment  Santa Clara, California              NaN  1090872
1  Health Care Equipment  Santa Clara, California              NaN  1090872
2  Health Care Equipment  Santa Clara, California              NaN  1090872
```

15) Using the groupby method, print the average close price by GICS Sector of the new DataFrame prices15_securities

1 point

```
GICS Sector
Consumer Discretionary         106.307607
Consumer Staples                75.587241
Energy                          58.935638
Financials                      66.812489
Health Care                    124.758577
Industrials                     84.580768
Information Technology          84.199695
Materials                       89.017597
Real Estate                     89.991233
Telecommunications Services     33.890730
Utilities                       51.893776
Name: close, dtype: float64
```