Gavin Koma
Dr. Justin Shi
Scripting for Sciences and Business
Thursday, June 2022

Lab 5: Python and Pandas

Rubric: Questions and Point Values

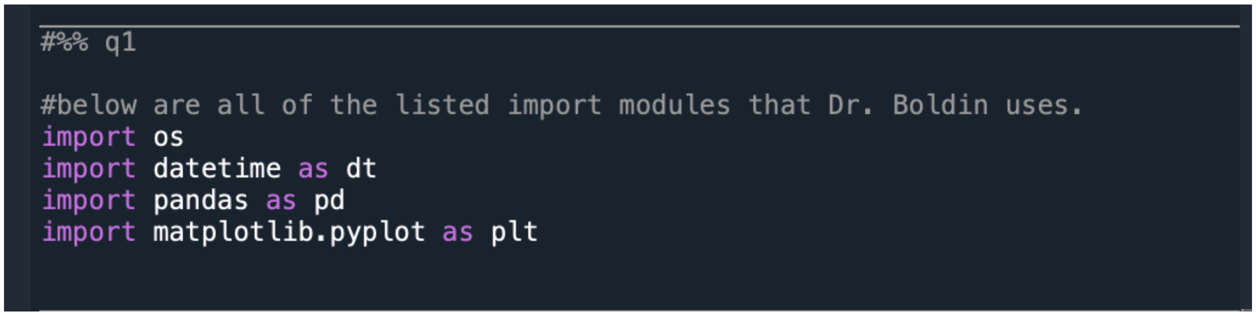
[1]	(1pt) List all **import** component names and their abbreviations, if any, in Dr. Boldin's lecture.
[2]	(1pt) Give a command to convert the list: un=[4.0,4.1,3.9,3.8,3.8] (list of 5 month unemployment rates) to become a Pandas Series. Name the series "s1"
[3]	(1pt) Name "s1" to "unrate" and define the index of s1 to be the dates from 01-01-2019 for 5 months. Picture provided clue in lab powerpoint.
[4]	(1pt) Give a command to build a DataFrame "df" from "s1". Then add two more columns: "Month" and "Laborforce Participation Rate" with k= [1,2,3,4,5] and lfp=[75,75.2,75.1,75.5,5,4]. The new "df" should look like this:
[5]	(1pt) Plot "s1" and save the plot.
[6]	(1pt) Give the scripts to shorten the labels in "df": "unrate" -> "Un","Laborforce Participation Rate" -> "Lfp". The new "df" should look like:
[7]	(1 pt) Compute "Employment Percent"(Emppct) and add it to "df". Emppct= (1-df['Un']/100)*(df['Lfp']/100). Your new "df" should look like this:
[8]	(1 pt) Give a command to query "df" the Unemployment Rate and Laborforce Participation Rate on April 30, 2019.
[9]	(1 pt) Give a command to compute the "count", "mean", "std", "min", "25,50,75%" and "max" of all data in "df".
[10]	(1 pts) Modify "df" to set the "Unemployment Rate" on April 30, 2019 to be 100% (a bad news in economy). You new "df" should look like this:
[11]	(1 pt) Give a command to query "df" the "Unemployment Rate" and "Laborforce Participation Rate" between February 20, 2019 and March 30, 2019.
[12]	(1 pt) Give a command to plot 3 charts showing "Unemployment Rate", "Laborforce Participation Rate", and "Employment Percentage" in the five months from January 1, 2019. Define the chart size to be "10x20".
[13]	(1 pt) Give a command to save "df" to a CSV file named "lab5.csv".

[14]	Give a command to read from a CSV file named “lab5.csv” to “df2”. Show “df2” to verify it is identical to “df”.
[15]	(1 pt) Give a command to “escape” from interactive Python session to the operating system. Get a screenshot of your folder contents.

Deliverables to be Included:

[1]	lab5.csv
[2]	Screenshots: s1.plot and 3 subplots for “Unemployment Rate”, “Laborparticipation Rate” and “Employment Percentage” for the 5 months in 2019
[3]	Screenshot of the escaped OS folder contents
[4]	Answers to all questions

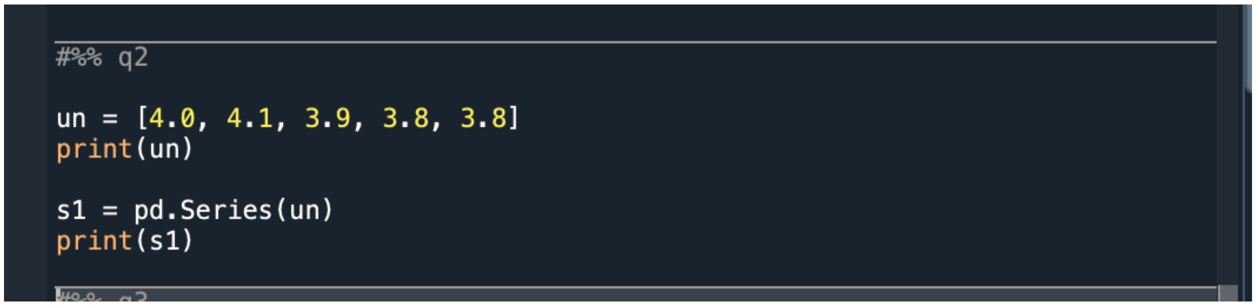
Question 1:
List all **import** component names and their abbreviations, if any, in Dr. Boldin's lecture.



```
#%% q1

#below are all of the listed import modules that Dr. Boldin uses.
import os
import datetime as dt
import pandas as pd
import matplotlib.pyplot as plt
```

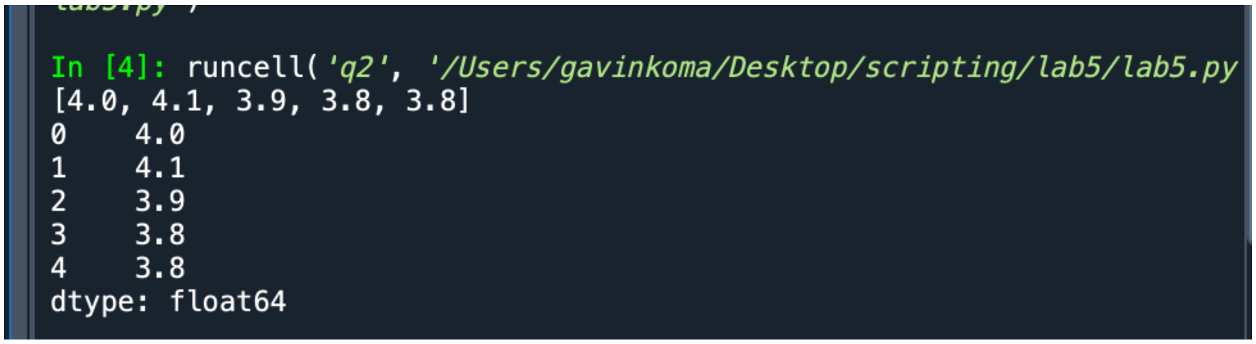
He imports os, datetime as dt, pandas as pd, and matplotlib.pyplot as plt. The abbreviations allow us to call on functions in modules without rewriting the whole name (i.e. pd instead of pandas).

Question 2:
Give a command to convert the list: un = [4.0, 4.1, 3.9, 3.8, 3.8] (list of 5 month unemployment rates) to become a Pandas Series. Name the series “s1.”



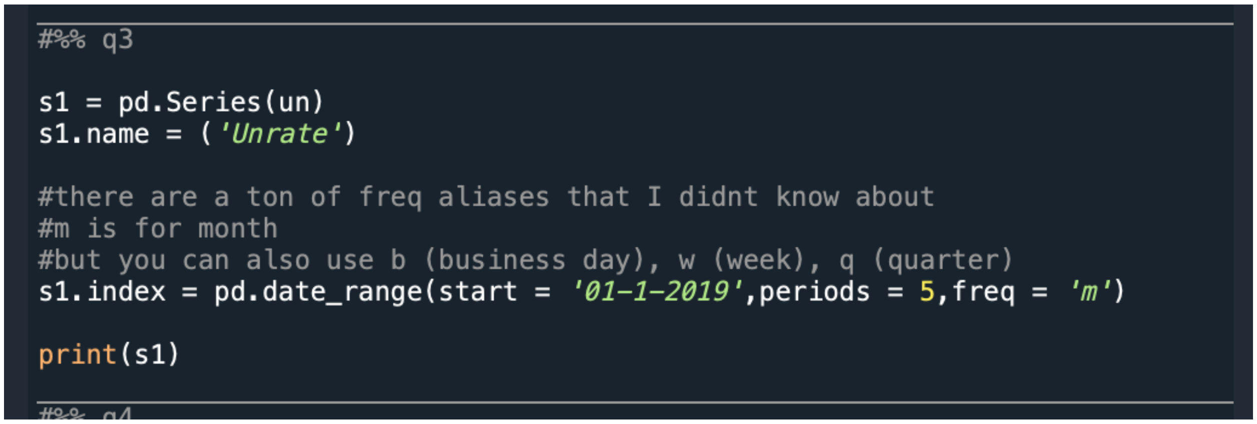
```
#%% q2

un = [4.0, 4.1, 3.9, 3.8, 3.8]
print(un)

s1 = pd.Series(un)
print(s1)
```

Terminal Output:



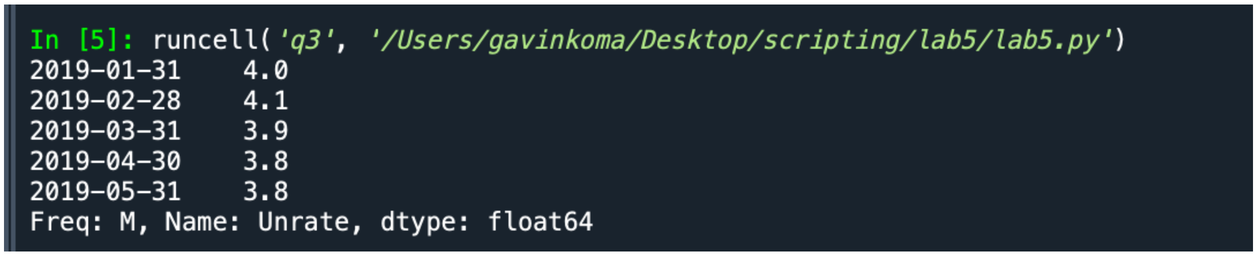
```
In [4]: runcell('q2', '/Users/gavinkoma/Desktop/scripting/lab5/lab5.py
[4.0, 4.1, 3.9, 3.8, 3.8]
0    4.0
1    4.1
2    3.9
3    3.8
4    3.8
dtype: float64
```

Question 3:
Name “s1” to “unrate” and define the index of s1 to be the dates from 01-1-2019 for 5 months.



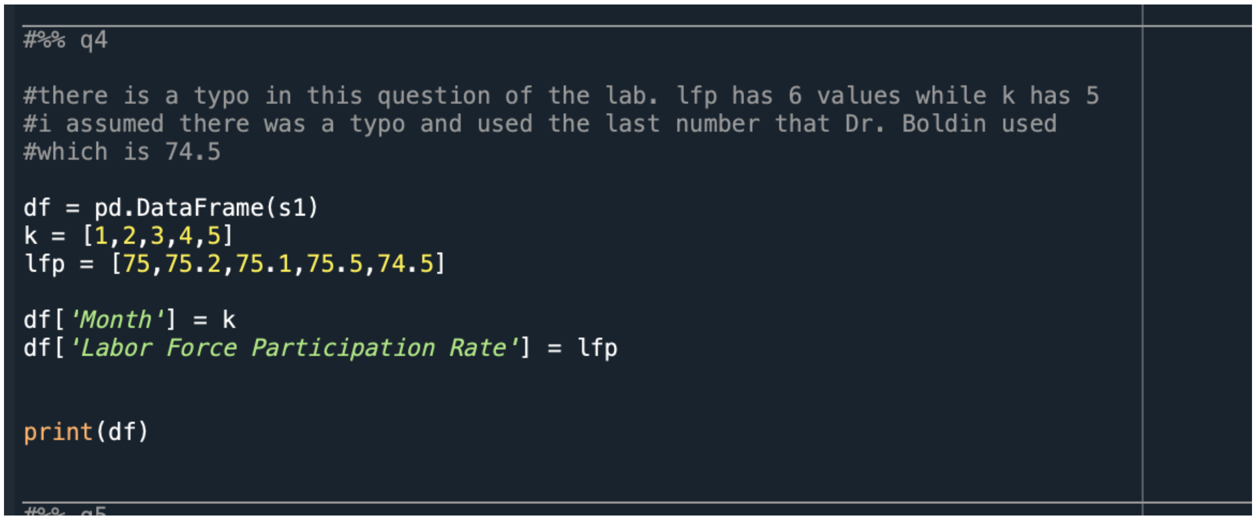
```
#%% q3

s1 = pd.Series(un)
s1.name = ('Unrate')

#there are a ton of freq aliases that I didnt know about
#m is for month
#but you can also use b (business day), w (week), q (quarter)
s1.index = pd.date_range(start = '01-1-2019',periods = 5,freq = 'm')

print(s1)
```

Terminal Output:



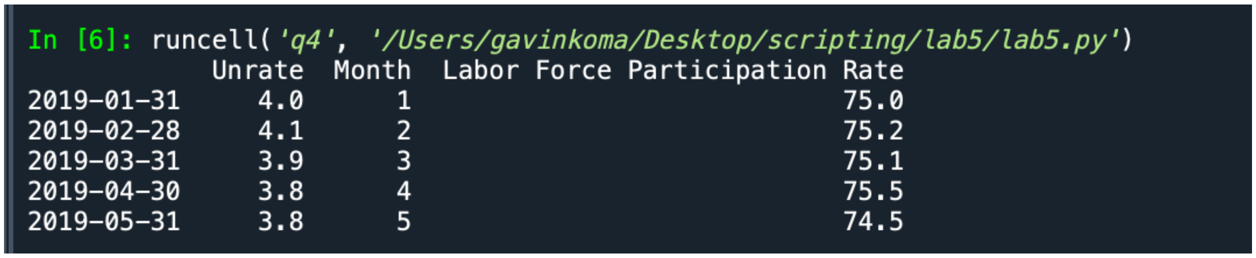
```
In [5]: runcell('q3', '/Users/gavinkoma/Desktop/scripting/lab5/lab5.py')
2019-01-31    4.0
2019-02-28    4.1
2019-03-31    3.9
2019-04-30    3.8
2019-05-31    3.8
Freq: M, Name: Unrate, dtype: float64
```

Question 4:
Give a command to build a DataFrame “df” from “s1”. Then, add two more columns: “Month” and “Laborforce Participation Rate” with k = [1,2,3,4,5] and lfp = [75,75.2,75.1,75.5,74.5]



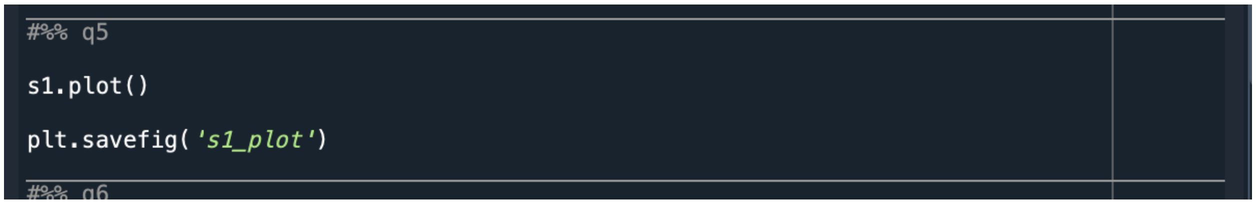
```
#%% q4

#there is a typo in this question of the lab. lfp has 6 values while k has 5
#i assumed there was a typo and used the last number that Dr. Boldin used
#which is 74.5

df = pd.DataFrame(s1)
k = [1,2,3,4,5]
lfp = [75,75.2,75.1,75.5,74.5]

df['Month'] = k
df['Labor Force Participation Rate'] = lfp


print(df)
```

Terminal Output:



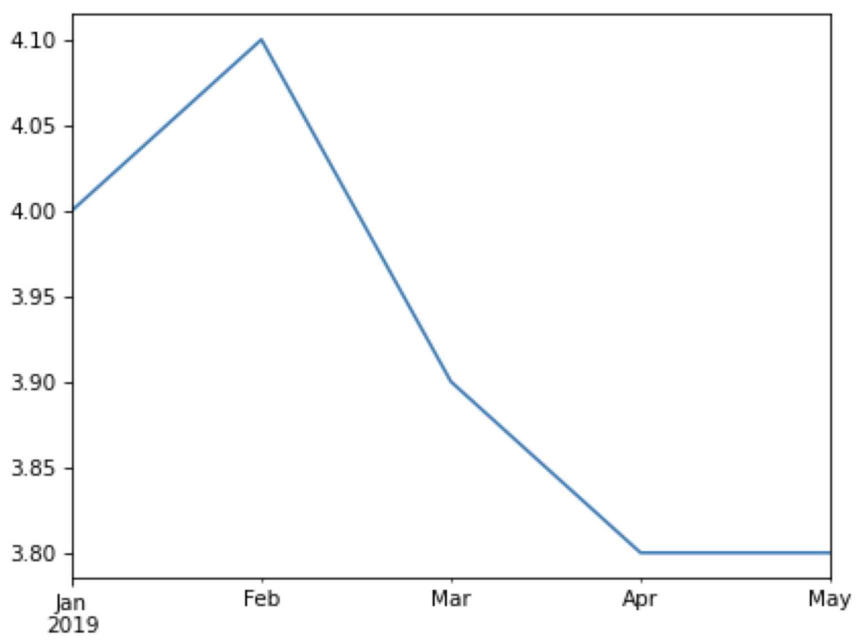
```
In [6]: runcell('q4', '/Users/gavinkoma/Desktop/scripting/lab5/lab5.py')
            Unrate  Month  Labor Force Participation Rate
2019-01-31     4.0      1                            75.0
2019-02-28     4.1      2                            75.2
2019-03-31     3.9      3                            75.1
2019-04-30     3.8      4                            75.5
2019-05-31     3.8      5                            74.5
```

Question 5:
Plot “s1” and save the plot.

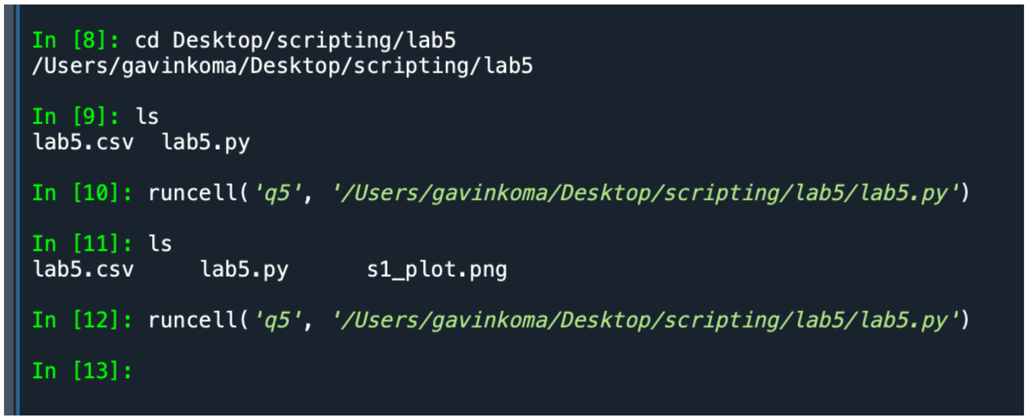


```
#%% q5

s1.plot()

plt.savefig('s1_plot')

#%% q6
```

Graph:



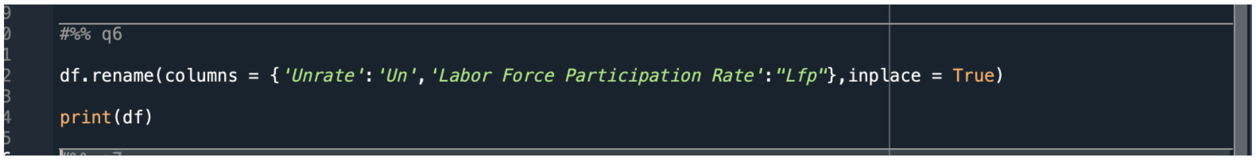
Terminal Output:



```
In [8]: cd Desktop/scripting/lab5
/Users/gavinkoma/Desktop/scripting/lab5

In [9]: ls
lab5.csv  lab5.py

In [10]: runcell('q5', '/Users/gavinkoma/Desktop/scripting/lab5/lab5.py')

In [11]: ls
lab5.csv     lab5.py      s1_plot.png

In [12]: runcell('q5', '/Users/gavinkoma/Desktop/scripting/lab5/lab5.py')

In [13]:
```

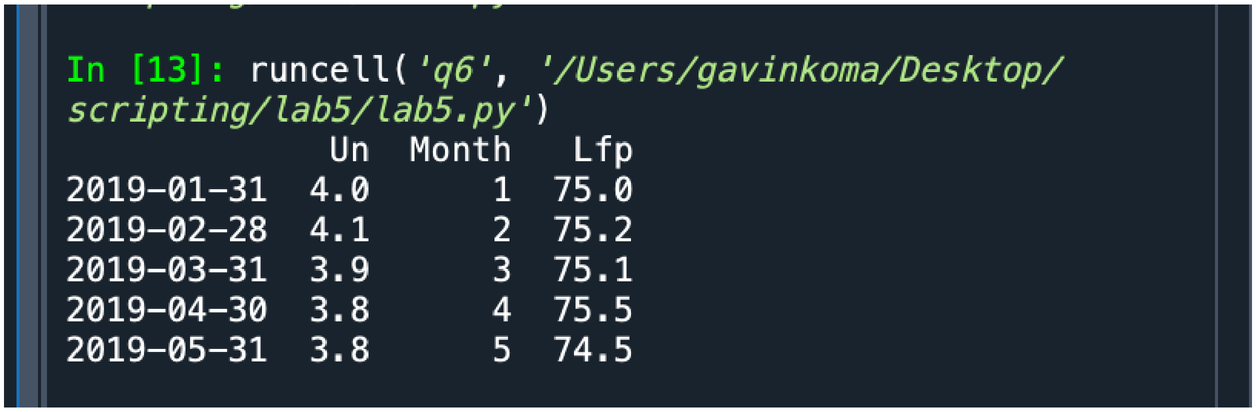
Included to show that the s1_plot.png saves from the python code.

Question 6:
Give the scripts to shorten the labels in “df”: ”unrate” → “Un” and ”Laborforce Participation Rate” → “Lfp”. The new “df” should look:



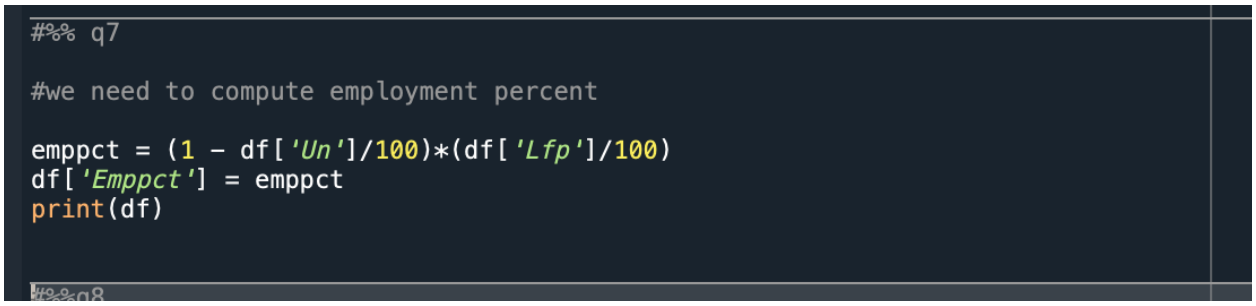
```
#%% q6

df.rename(columns = {'Unrate':'Un','Labor Force Participation Rate':"Lfp"},inplace = True)

print(df)
```

Terminal Output:



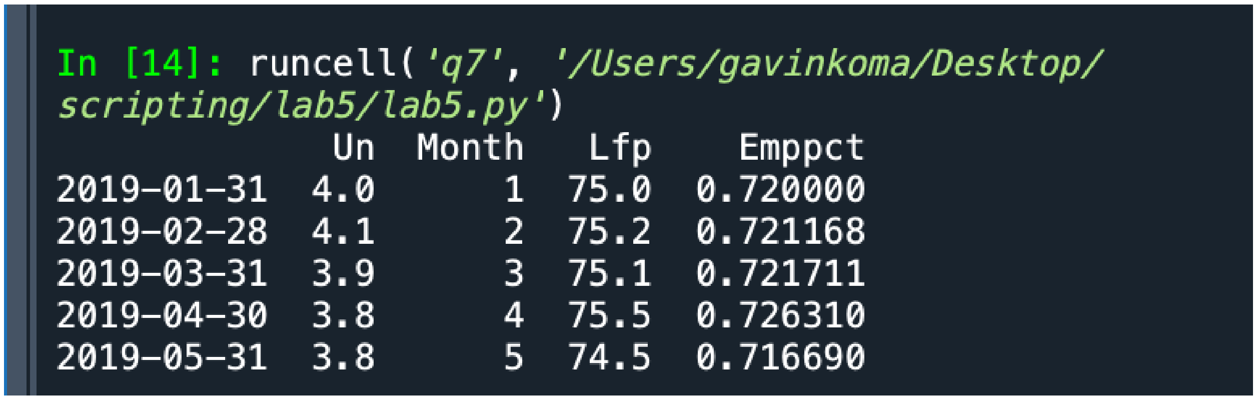
```
In [13]: runcell('q6', '/Users/gavinkoma/Desktop/
scripting/lab5/lab5.py')
             Un  Month   Lfp
2019-01-31  4.0      1  75.0
2019-02-28  4.1      2  75.2
2019-03-31  3.9      3  75.1
2019-04-30  3.8      4  75.5
2019-05-31  3.8      5  74.5
```

Question 7:
Compute “Employment Percent” (Emppct) and add it to “df”. Emppct = (10df[‘Un’]/100)*(df[‘Lfp’]/100). Your new “df”:



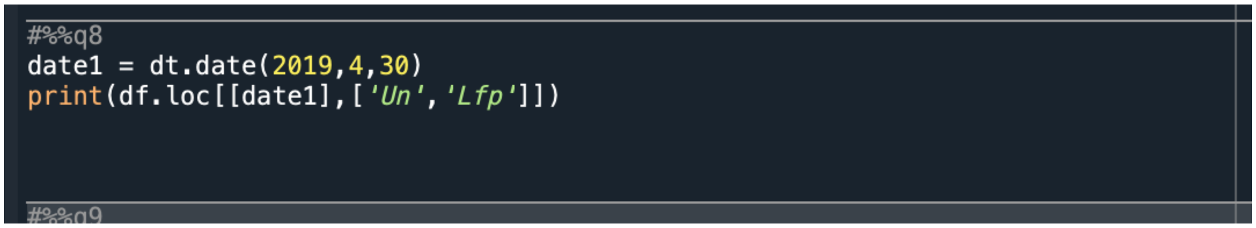
```
#%% q7

#we need to compute employment percent

emppct = (1 - df['Un']/100)*(df['Lfp']/100)
df['Emppct'] = emppct
print(df)


#%%q8
```

Terminal Output:



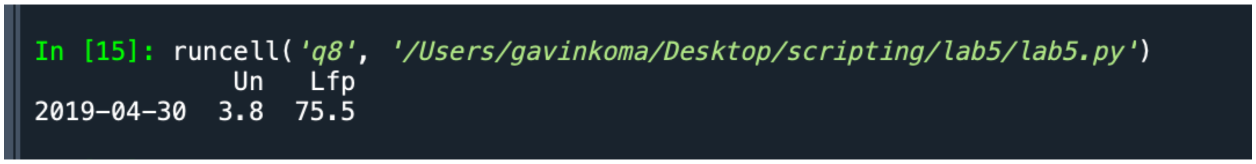
```
In [14]: runcell('q7', '/Users/gavinkoma/Desktop/
scripting/lab5/lab5.py')
             Un  Month   Lfp    Emppct
2019-01-31  4.0      1  75.0  0.720000
2019-02-28  4.1      2  75.2  0.721168
2019-03-31  3.9      3  75.1  0.721711
2019-04-30  3.8      4  75.5  0.726310
2019-05-31  3.8      5  74.5  0.716690
```

Question 8:
Give a command to query “df” the Unemployment Rate and Laborforce Participation Rate on April 30th, 2019.



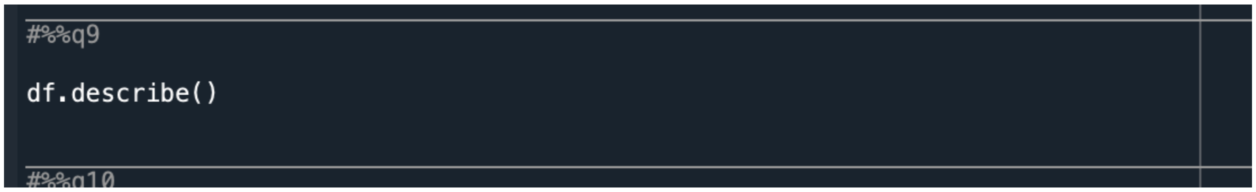
```
#%%q8
date1 = dt.date(2019,4,30)
print(df.loc[[date1],['Un','Lfp']])


#%%q9
```

Terminal Output:



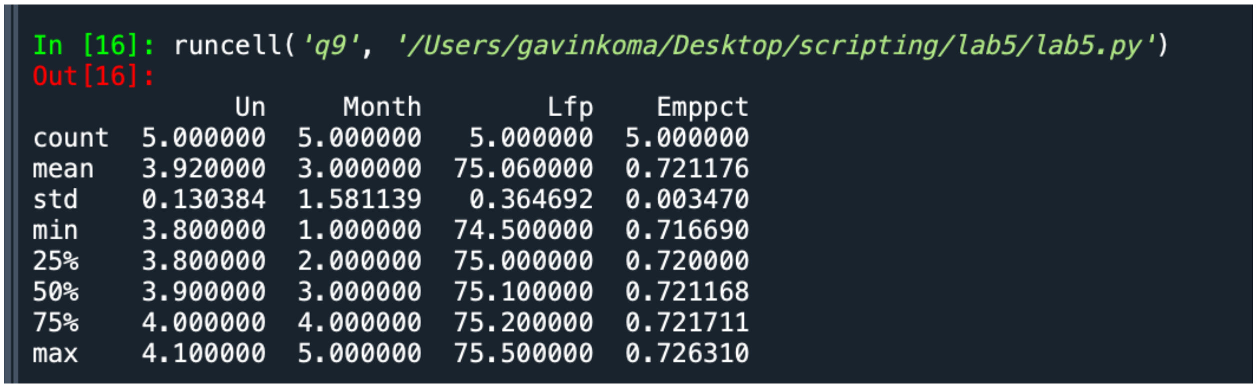
```
In [15]: runcell('q8', '/Users/gavinkoma/Desktop/scripting/lab5/lab5.py')
             Un   Lfp
2019-04-30  3.8  75.5
```

Question 9:
Give a command to compute the “count”, “mean”, “std”, “min”, “25, 50, 75%”, and “max” of all the data in “df”.



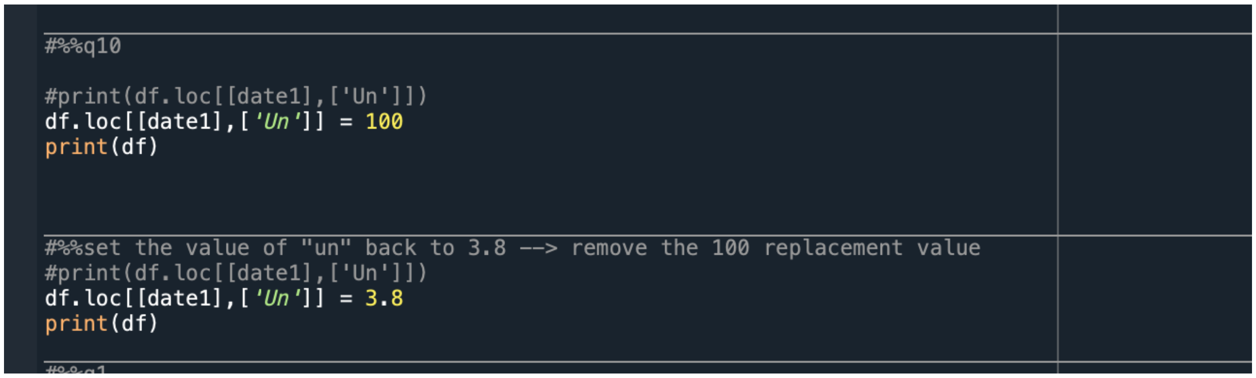
```
#%%q9

df.describe()

#%%q10
```

Terminal Output:



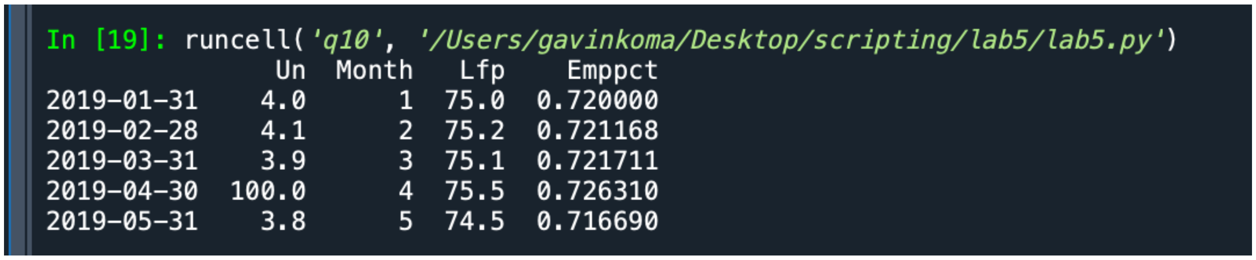
```
In [16]: runcell('q9', '/Users/gavinkoma/Desktop/scripting/lab5/lab5.py')
Out[16]:
             Un     Month        Lfp    Emppct
count  5.000000  5.000000   5.000000  5.000000
mean   3.920000  3.000000  75.060000  0.721176
std    0.130384  1.581139   0.364692  0.003470
min    3.800000  1.000000  74.500000  0.716690
25%    3.800000  2.000000  75.000000  0.720000
50%    3.900000  3.000000  75.100000  0.721168
75%    4.000000  4.000000  75.200000  0.721711
max    4.100000  5.000000  75.500000  0.726310
```

Question 10:
Modify “df” to set the “Unemployment Rate” on April 30th, 2019 to be 100% (a bad news economy). Your new “df”:



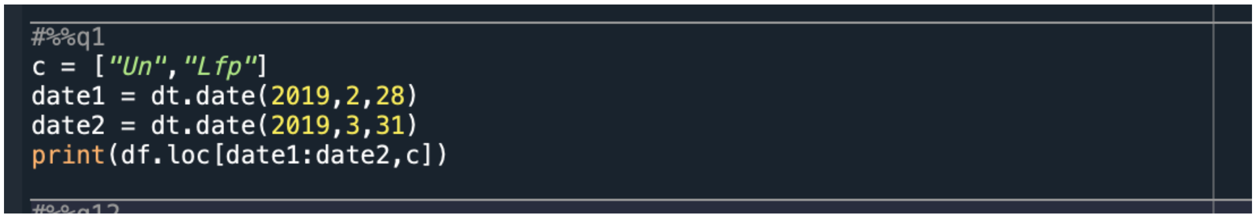
```
#%%q10

#print(df.loc[[date1],['Un']])
df.loc[[date1],['Un']] = 100
print(df)



#%%set the value of "un" back to 3.8 --> remove the 100 replacement value
#print(df.loc[[date1],['Un']])
df.loc[[date1],['Un']] = 3.8
print(df)
```

Terminal Output:



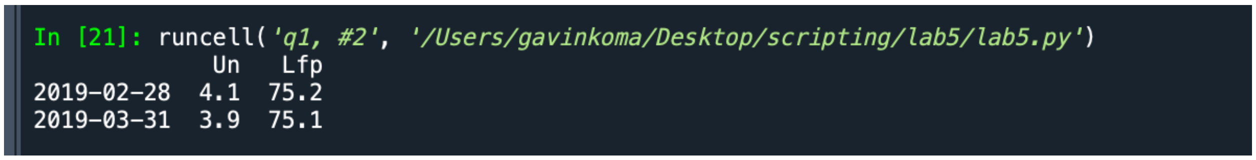
```
In [19]: runcell('q10', '/Users/gavinkoma/Desktop/scripting/lab5/lab5.py')
               Un  Month   Lfp    Emppct
2019-01-31    4.0      1  75.0  0.720000
2019-02-28    4.1      2  75.2  0.721168
2019-03-31    3.9      3  75.1  0.721711
2019-04-30  100.0      4  75.5  0.726310
2019-05-31    3.8      5  74.5  0.716690
```

Question 11:
Give a command to query “df” the “Unemployment Rate” and “Laborforce Participation Rate” between February 20th, 2019 and March 30th, 2019.



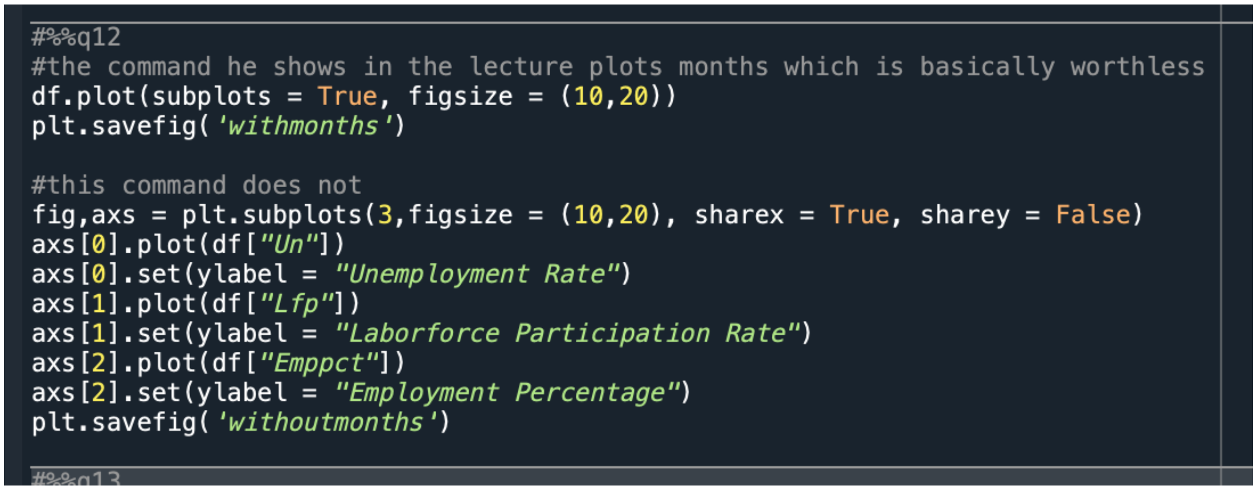
```
#%%q1
c = ["Un","Lfp"]
date1 = dt.date(2019,2,28)
date2 = dt.date(2019,3,31)
print(df.loc[date1:date2,c])

#%%q12
```

Terminal Output:



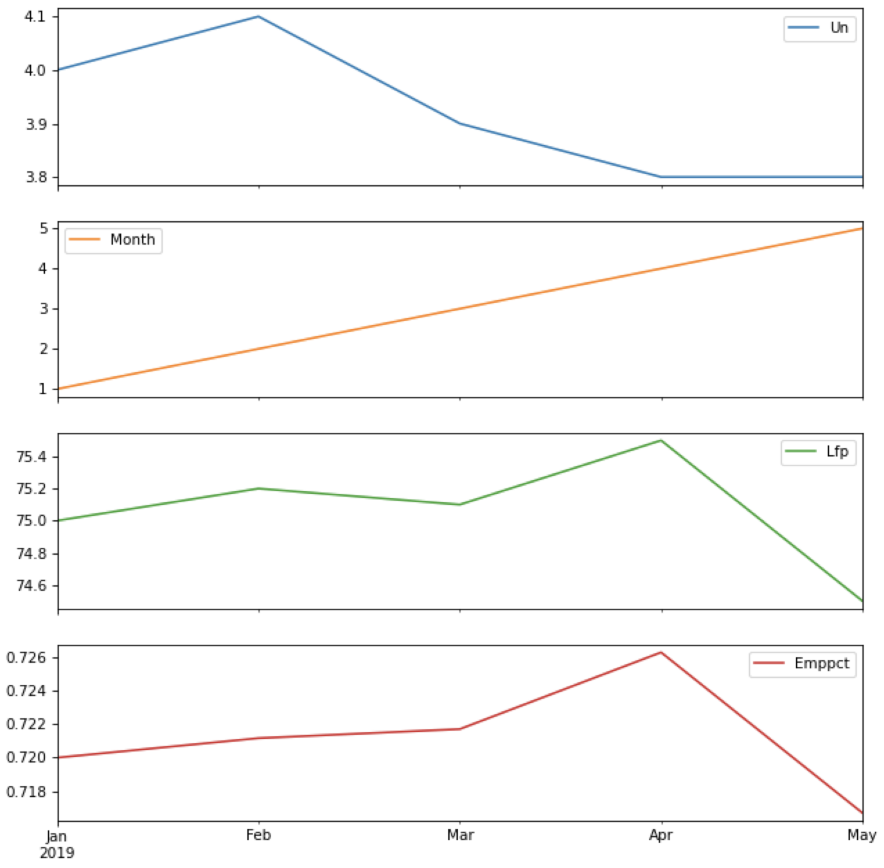
```
In [21]: runcell('q1, #2', '/Users/gavinkoma/Desktop/scripting/lab5/lab5.py')
             Un   Lfp
2019-02-28  4.1  75.2
2019-03-31  3.9  75.1
```

Question 12:
Give a command to plot 3 charts showing “Unemployment Rate”, “Laborforce Participation Rate”, and “Employment Percentage” in the five months from January 1st, 2019. Define the chart size to be “10x20”:

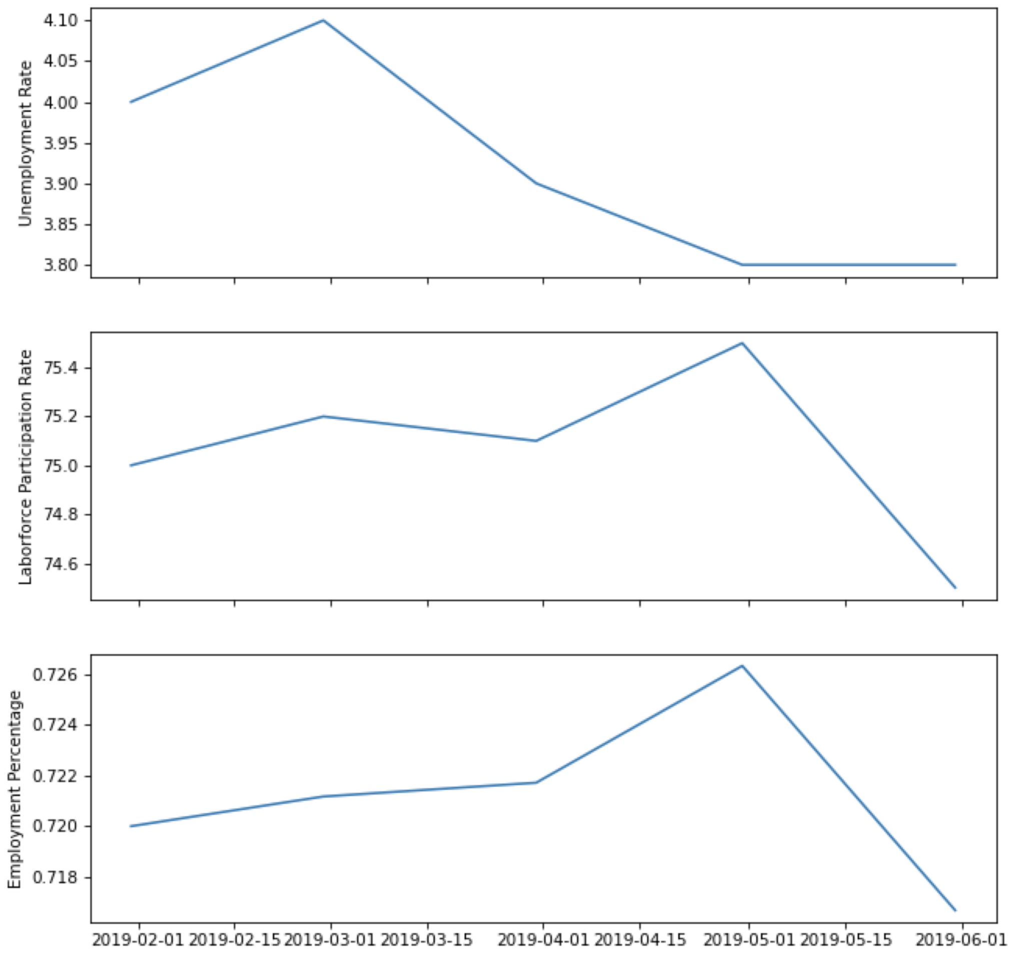


```
#%%q12
#the command he shows in the lecture plots months which is basically worthless
df.plot(subplots = True, figsize = (10,20))
plt.savefig('withmonths')

#this command does not
fig,axs = plt.subplots(3,figsize = (10,20), sharex = True, sharey = False)
axs[0].plot(df["Un"])
axs[0].set(ylabel = "Unemployment Rate")
axs[1].plot(df["Lfp"])
axs[1].set(ylabel = "Laborforce Participation Rate")
axs[2].plot(df["Emppct"])
axs[2].set(ylabel = "Employment Percentage")
plt.savefig('withoutmonths')

#%%q13
```

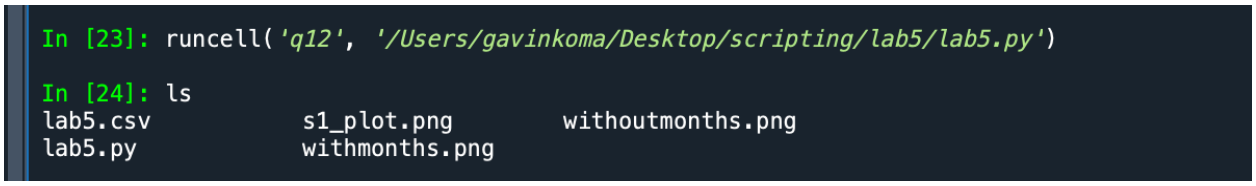
Graphs similar to Professor Boldin:



Graphs that I coded differently to not include months as a graph:

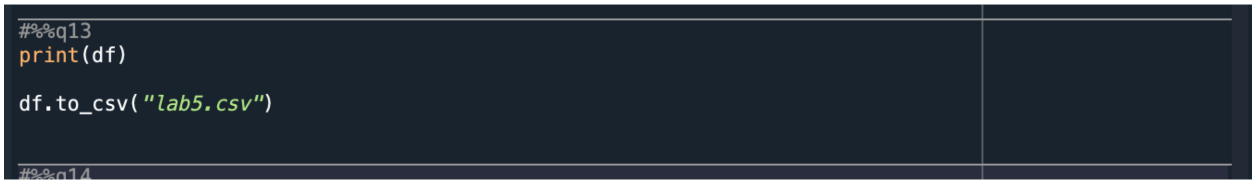


Terminal Output:



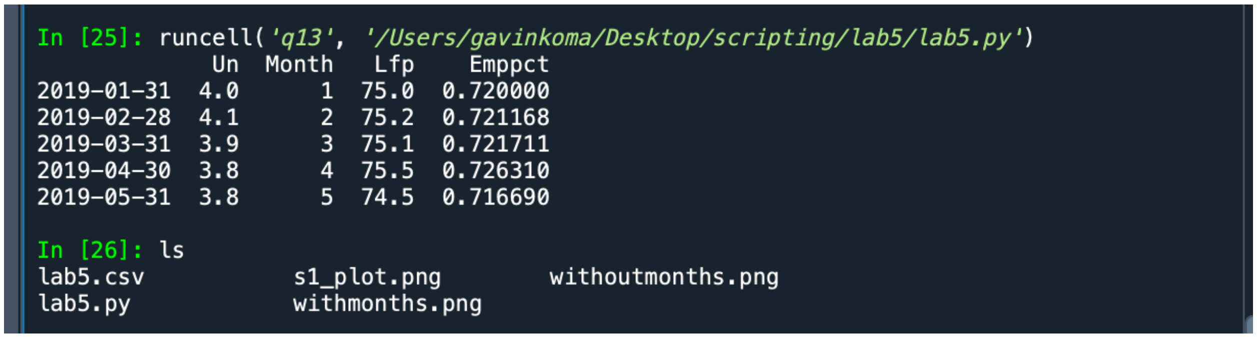
```
In [23]: runcell('q12', '/Users/gavinkoma/Desktop/scripting/lab5/lab5.py')

In [24]: ls
lab5.csv            s1_plot.png         withoutmonths.png
lab5.py             withmonths.png
```

Question 13:
Give a command to save “df” to a csv file names “lab5.csv”:



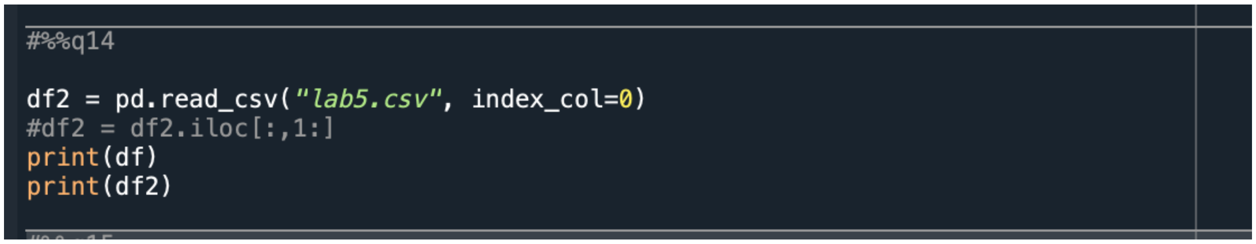
```
#%%q13
print(df)

df.to_csv("lab5.csv")

#%%q14
```

Terminal Output:



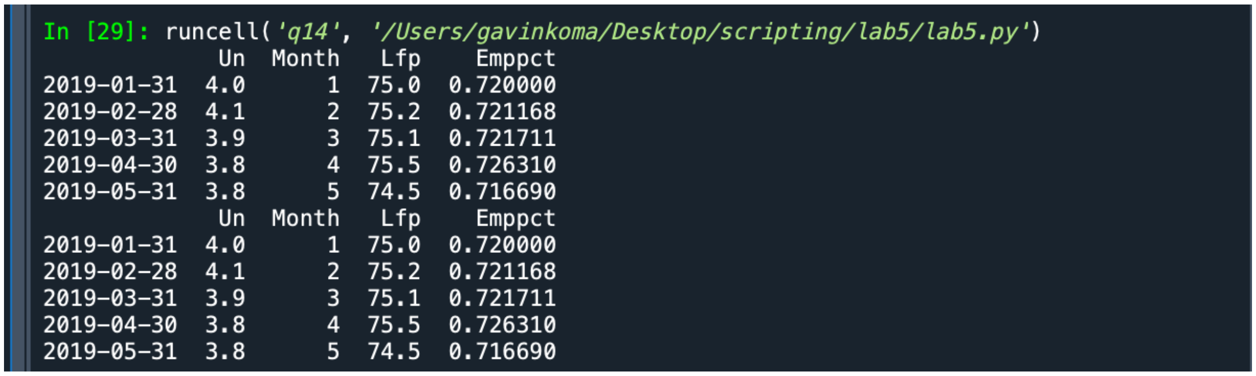
```
In [25]: runcell('q13', '/Users/gavinkoma/Desktop/scripting/lab5/lab5.py')
             Un  Month   Lfp    Emppct
2019-01-31  4.0      1  75.0  0.720000
2019-02-28  4.1      2  75.2  0.721168
2019-03-31  3.9      3  75.1  0.721711
2019-04-30  3.8      4  75.5  0.726310
2019-05-31  3.8      5  74.5  0.716690

In [26]: ls
lab5.csv           s1_plot.png        withoutmonths.png
lab5.py            withmonths.png
```

Question 14:
Give a command to read from a csv file names “lab5.csv” to “df2”. Show “df2” to verify that it is identical to “df”



```
#%%q14

df2 = pd.read_csv("lab5.csv", index_col=0)
#df2 = df2.iloc[:,1:]
print(df)
print(df2)
```

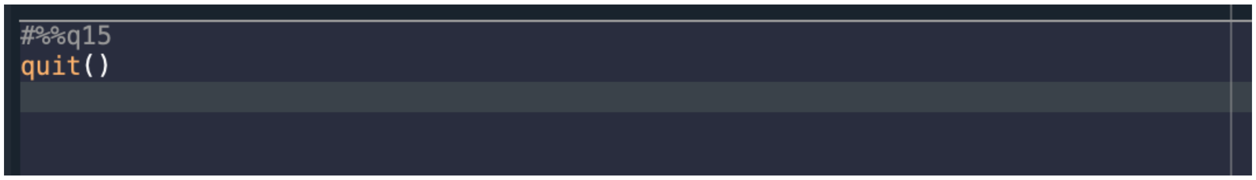
Terminal Output:



```
In [29]: runcell('q14', '/Users/gavinkoma/Desktop/scripting/lab5/lab5.py')
             Un  Month   Lfp    Emppct
2019-01-31  4.0      1  75.0  0.720000
2019-02-28  4.1      2  75.2  0.721168
2019-03-31  3.9      3  75.1  0.721711
2019-04-30  3.8      4  75.5  0.726310
2019-05-31  3.8      5  74.5  0.716690
             Un  Month   Lfp    Emppct
2019-01-31  4.0      1  75.0  0.720000
2019-02-28  4.1      2  75.2  0.721168
2019-03-31  3.9      3  75.1  0.721711
2019-04-30  3.8      4  75.5  0.726310
2019-05-31  3.8      5  74.5  0.716690
```

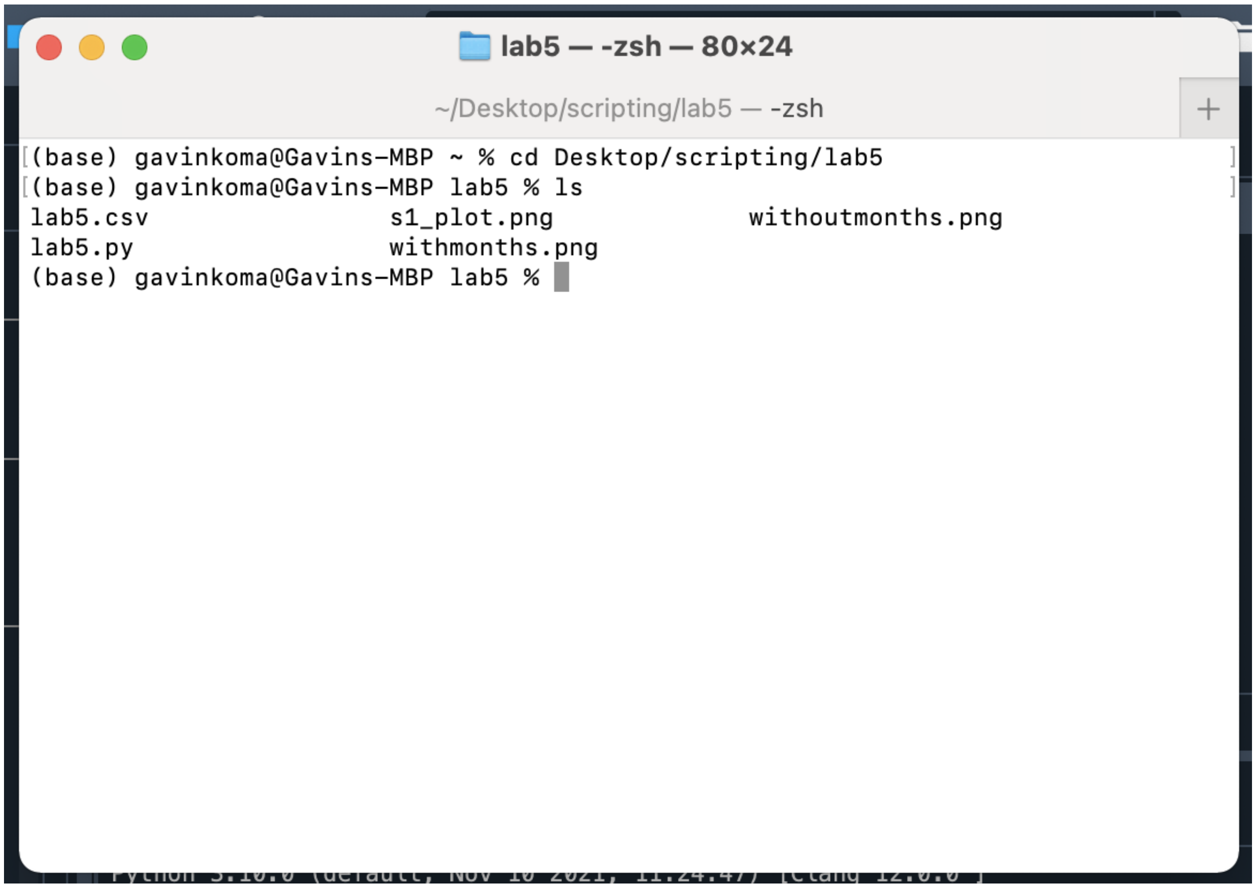
df is the first one, df2 is the second one; I printed them both to facilitate comparison between them. They are the same.

Question 15:
Give a command to “escape” from interactive python session to the operating system. Get a screenshot of your folder contents:



```
#%%q15
quit()
```

Screenshot of folder contents:



```
(base) gavinkoma@Gavins-MBP ~ % cd Desktop/scripting/lab5
(base) gavinkoma@Gavins-MBP lab5 % ls
lab5.csv                s1_plot.png             withoutmonths.png
lab5.py                 withmonths.png
(base) gavinkoma@Gavins-MBP lab5 % 
```

Debugging:
The only debugging I did can be seen in Question 12. I decided that having a graph that was linear for “Months” didnt make any sense and seemed irrelevant to have as well as distracting to the other data. I created a second portion of code that scripts three other subplots of (10x20) size and uploaded that as well. The differences in code can be seen above.