

Week 5 Deliverables

Overview: In this week, you have studied additional Python language syntax including File I/O, Exceptions and Object-Oriented Programming. The Lab for this week demonstrates your knowledge of this additional Python functionality. Be sure to use the examples in the textbook reading along with the associate libraries, functions and processes when completing the assignments for this week.

Be sure to develop and test your Python code in the AWS Cloud9 IDE provided for the class.

You should continue to use the PEP Python Style guide mentioned in the book and found here:

<https://www.python.org/dev/peps/pep-0008/>

Some examples of Python Coding Style best practices include:

- Limit all lines to a maximum of 79 characters.
- Imports are always put at the top of the file, just after any module comments and before module globals and constants.
- Use 4 spaces for indentation.

Submission requirements for this project include 4 files. (Zipping them into one file is acceptable and encouraged):

- Python Data Analysis Code
- 2 Input Files (Same files supplied to you)
- Word, Excel or PDF file containing your test results

Python Applications for Lab4: (total 100 points):

This exercise (**80 points**) allows a user to load one of two CSV files and then perform histogram analysis and plots for select variables on the datasets. The first dataset represents the population change for specific dates for U.S. regions. The second dataset represents Housing data over an extended period of time describing home age, number of bedrooms and other variables. The first row provides a column name for each dataset. The following columns should be used to perform analysis:

PopChange.csv:

- Pop Apr 1
- Pop Jul 1
- Change Pop

Housing.csv:

- AGE
- BEDRMS
- BUILT
- ROOMS
- UTILITY

Notice for the Housing CSV file, there are more columns in the file than are required to be analyzed. You can and should still load each column.

Specific statistics options should include:

- Count
- Mean
- Standard Deviation
- Min
- Max
- Histogram

Hints:

1. Use the Pandas, Numpy, Matplotlib and other Python modules when appropriate.
2. Since you are running this on a Linux box, you don't really have a graphical environment. So you will need to save the graphics for display. Notice the plt variable is assigned to fig1 and then the figure is saved. For example:

```
import numpy as np
import matplotlib.pyplot as plt

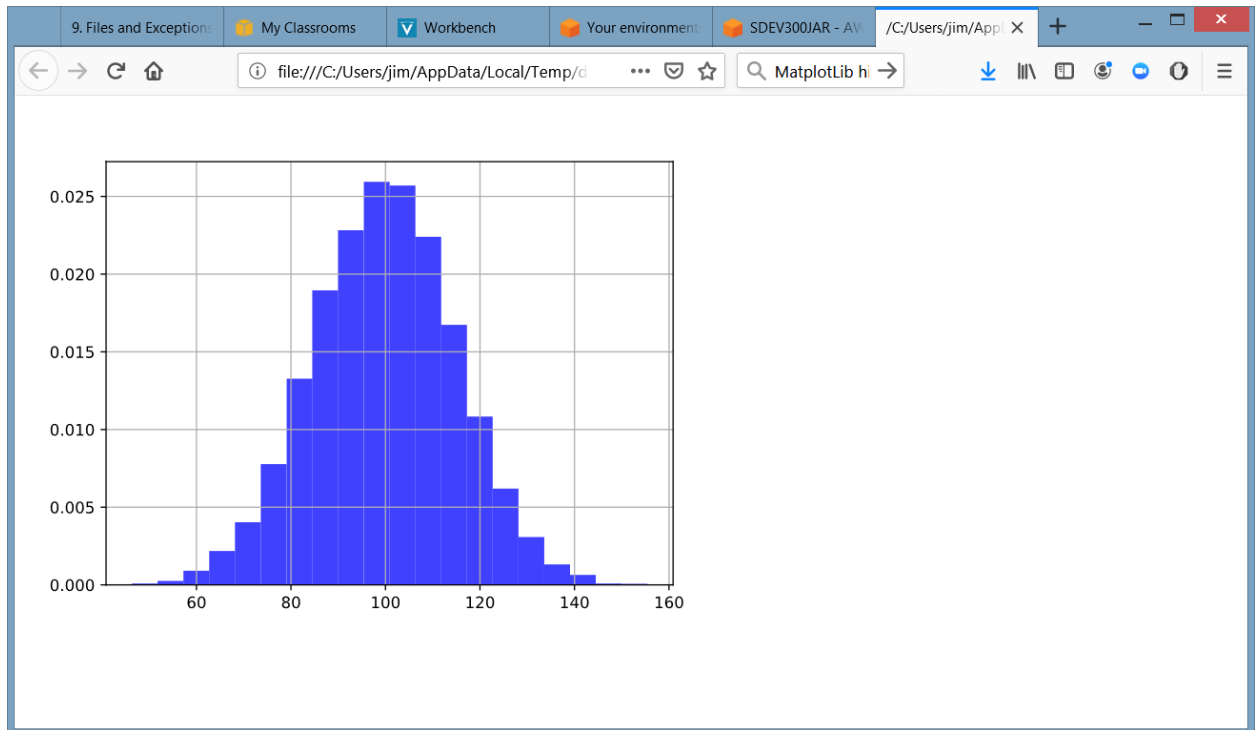
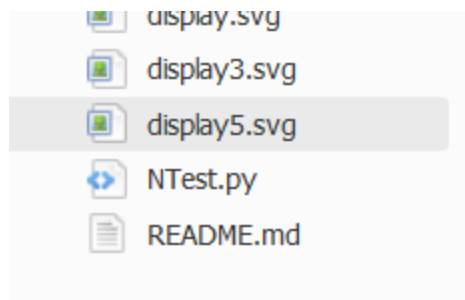
# Fixing random state for reproducibility
np.random.seed(214801)

mu, sigma = 100, 15
x = mu + sigma * np.random.randn(10000)

# the histogram of the data
n, bins, patches = plt.hist(x, 20, density=True, facecolor='b',
alpha=0.75)

plt.grid(True)
#Assign to a figure
fig1=plt
# Save Figure for Download
fig1.savefig('display5.svg')
```

Once the figure is saved, just download (right mouse click) to save to your desktop for viewing:



3. Be sure to read the content in this week for how to install the Python modules in your Cloud9 IDE. You need to install those modules before you can use them.

The user interface should continue to run until the user indicates they are ready to exit.

A user interface might look similar to this:

```
***** Welcome to the Python Data Analysis App*****

Select the file you want to analyze:

1. Population Data
2. Housing Data
3. Exit the Program
```

1

You have entered Population Data.

Select the Column you want to analyze:

- a. Pop Apr 1
- b. Pop Jul 1
- c. Change Pop
- d. Exit Column

a

You selected Pop Apr 1

The statistics for this column are:

Count = 10000
Mean = 32.5
Standard Deviation = 4.5
Min = 53.2
Max = 12.5

The Histogram of this column can be downloaded now.

Select the Column you want to analyze:

- a. Pop Apr 1
- b. Pop Jul 1
- c. Change Pop
- d. Exit Column

d

You selected to exit the column menu

Select the file you want to analyze:

- 1. Population Data
- 2. Housing Data
- 3. Exit the Program

3

***** Thanks for using the Data Analysis App*****

If an inappropriate entry is detected, the program should prompt for a correct value and continue to do so until a correct value is entered.

Be sure to use a Class (i.e. Object-Oriented Programming) to load your data.

For example, the following code could be part of reading each line of a file after it has been opened:

```
semesters = list()
with open(filename, 'r') as s:
    reader = csv.reader(s)
    next(reader) # skip the header
    for line in reader:
        # Append the soc record to the list
        semesters.append(objs.SEMESTER(line))
        semesters.append(objs.SEMESTER(line))

class SEMESTER:
    # Extract the critical data (maybe all of the data in a class)
    def __init__(self, line):
        self.data = line
        # Go through each element and assign value
        self.strm = self.data[0].strip()
        self.session = self.data[1].strip()
        self.startdate = self.data[2].strip()
        self.enddate = self.data[3].strip()
```

The above code snippets are part of a more comprehensive solution that would include class definitions for each column name for the files. You are welcome to use other approaches but be sure to create dedicated classes for your datasets.

2. Document your results for each application within the AWS Cloud9 classroom environment. Provide screen captures and descriptions for each test cases you provide for each of your applications. Show your Histogram graphics that were downloaded to your desktop. Be sure to go through each possible user interface combination in your test cases. **(20 points)**

Any submissions that do not represent work originating from the student will be submitted to the Dean's office and evaluated for possible academic integrity violations and sanctions.