Applied Statistics MATH 661 Assignment #1

September 10, 2019

Andres Castellano

---

1 Employee Application Data

a) The cases for this data set are the employee applications.

b) Employee ID is a label, Name fields are labels, Department and Education are categorical variables, years with the company, Salary and age are quantitative variables.

[1]:
```
#I will use the Pandas library to setup a dataframe or 'spreadsheet' of the␣
↪data.
import pandas as pd
from pandas import Series,DataFrame
```

[2]:
```
data = {'Employee ID':[1,2,3,4,5],
        'Last Name':['Castellano','Restrepo','Escobar','Botero','Smith'],
        'Middle Initial':['A','E','H','S','A'],
        'First Name':['Andres','Maria','George','Sophia','John'],
        'Department':
 ↪['Engineering','Operations','Production','Research','Sales'],
        'Years with the Company':[6,7,8,9,1],
        'Salary':[110000,120000,130000,140000,450000],
        'Education':['College Degree','','High School','College Degree','Some␣
 ↪College'],
        'Age':[4,5,6,7,8]}
```

[3]:
```
EmployeeRecords = DataFrame(data)
```

[4]:
```
EmployeeRecords
```

[4]:
```
   Employee ID   Last Name Middle Initial First Name   Department  \
0            1  Castellano              A     Andres  Engineering
1            2    Restrepo              E      Maria   Operations
2            3     Escobar              H     George   Production
3            4      Botero              S     Sophia     Research
4            5       Smith              A       John        Sales

   Years with the Company  Salary       Education  Age
```

```
0                        6  110000  College Degree    4
1                        7  120000                    5
2                        8  130000     High School    6
3                        9  140000  College Degree    7
4                        1  450000    Some College    8
```

---

2 Attending college in your state or another state.

a) The cases for this data set are the states.

b) The label variable is the state.

c) Number of students who attend college and number of students who attend college in their home state are both quantitative variable data.

d) Looking at these numbers it may be possible to deduce whether students from a particular state prefer in state institutions or out of the state schools. For example if for a partiucar state, there is a low percentage of students who attend in state colleges, that might indicate something about the perceived quality or cost of in-state institutions.

e) See answer d).

---

3 Alcohol-impaired driving fatalities.

The number of alcohol-impaired fatalities for each state could be divided by the population of each state. This number would give us a **per capita rate** of fatalities. However, such rates may not be ideal for comparison as they do not take into account the population make up of each state. For example, a particular state may have a large portion of citizens who cannot legally consume alcohol.

In addition, the number of alcohol-impaired fatalities can be divided by the total number of fatalities in each state. Such a **percentage** could indicate whether a state has a significant alcohol related issues.

Calculating **averages and standard deviations** for each state could help determine whether there are states or regions where alcohol-impaired fatalities are more common. A volume weighted average might help correct inaccuracies related to different population sizes.

---

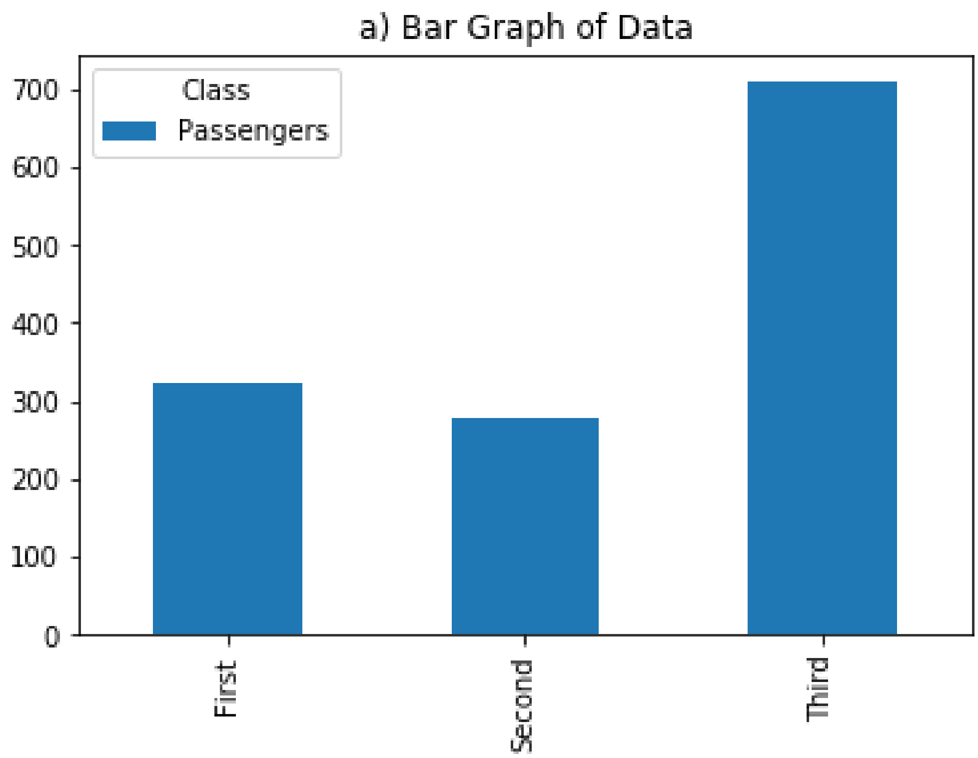
4 The *Titanic*

[5]:
```python
pass_data = {'Class':['First','Second','Third'],
             'Passengers': [323,277,709]}
PassFrame = DataFrame(pass_data,
                      index=['First','Second','Third'],
                      columns=pd.Index(['Passengers'],name='Class'))
PassFrame
```

[5]:
```
Class    Passengers
First           323
Second          277
Third           709
```

[6]:
```
import matplotlib.pyplot as plt
%matplotlib inline
PassFrame.plot(kind='bar',title='a) Bar Graph of Data')
```

[6]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1bc99ba2898>
```



b) Roughly 50% of passengers were travelling in third class and the remaining 50% were approximately evenly distriuted between first and second class

c) A bar graph of the percent of passengers in each class would look fairly similar to the one in part **b**.

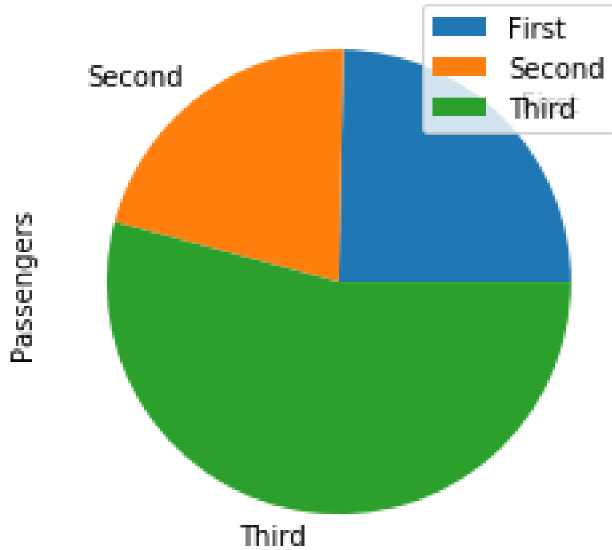
---

5 Another look at the *Titanic* and class

a) A Pie chart of passenger data

[7]:
```
%matplotlib inline
PassFrame.plot.pie(y='Passengers')
```

[7]: <matplotlib.axes._subplots.AxesSubplot at 0x1bc99c52b38>



b) I generally prefer bar graphs to pie charts. Although pie charts can sometimes make it easier to see when a proportion of the data is sgnificantly larger than the rest as in this example.

---

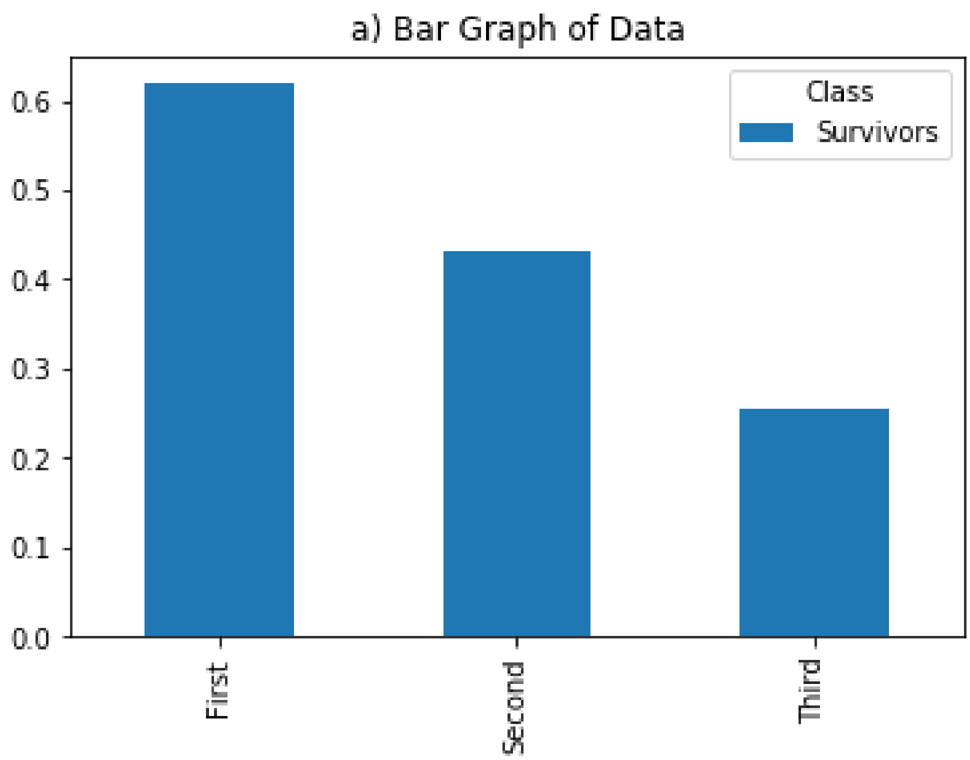
6 Who Survived?

[8]:
```
survival_data = {'Class':['First','Second','Third'],
             'Survivors': [200/323,119/277,181/709]}
SurvivalFrame = DataFrame(survival_data,
                     index=['First','Second','Third'],
                     columns=pd.Index(['Survivors'],name='Class'))

SurvivalFrame.plot(kind='bar',title='a) Bar Graph of Data')
SurvivalFrame['Survivors']=pd.Series(["{0:.2f}%".format(val*100) for val in␣
 ↪SurvivalFrame['Survivors']],
                                     index = SurvivalFrame.index)
SurvivalFrame
```

[8]:

Class	Survivors
First	61.92%
Second	42.96%
Third	25.53%

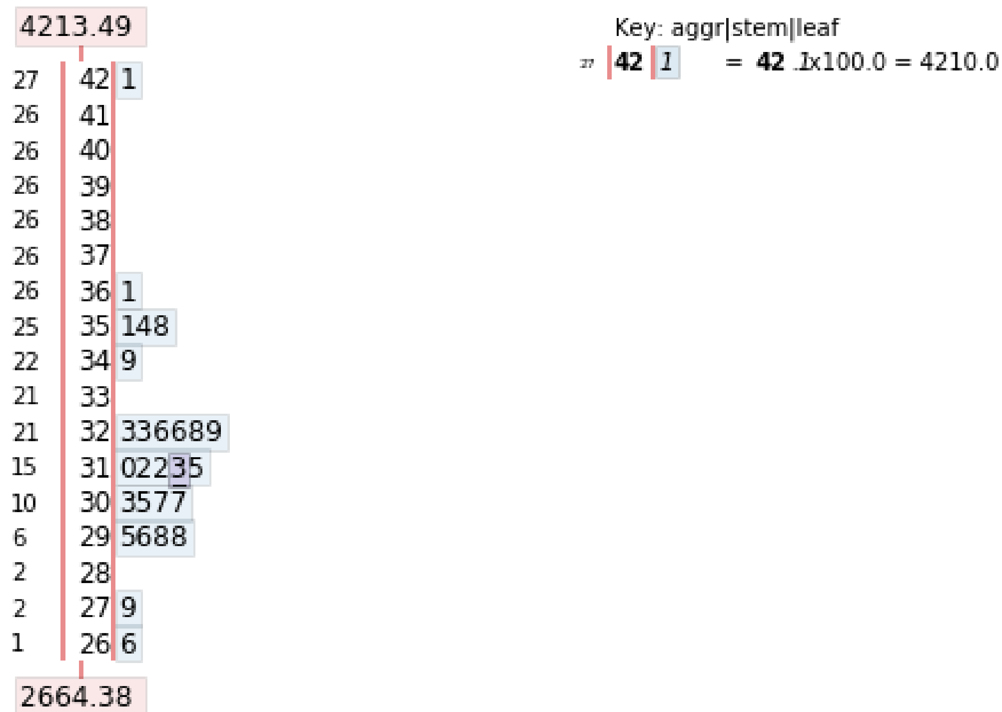


7 Potassium from Potatoes

a) Stemplot of the Data

[9]:
```
import stemgraphic

potdata = pd.read_csv(r'c:
↪\Users\Castellano\Documents\Fall2019\Statistics\Potatoes.csv')
potdata
y = pd.Series(potdata['Potassium_mg'])
fig, ax = stemgraphic.stem_graphic(y)
```



b) The data seems to be normally distributed around 3260mg.

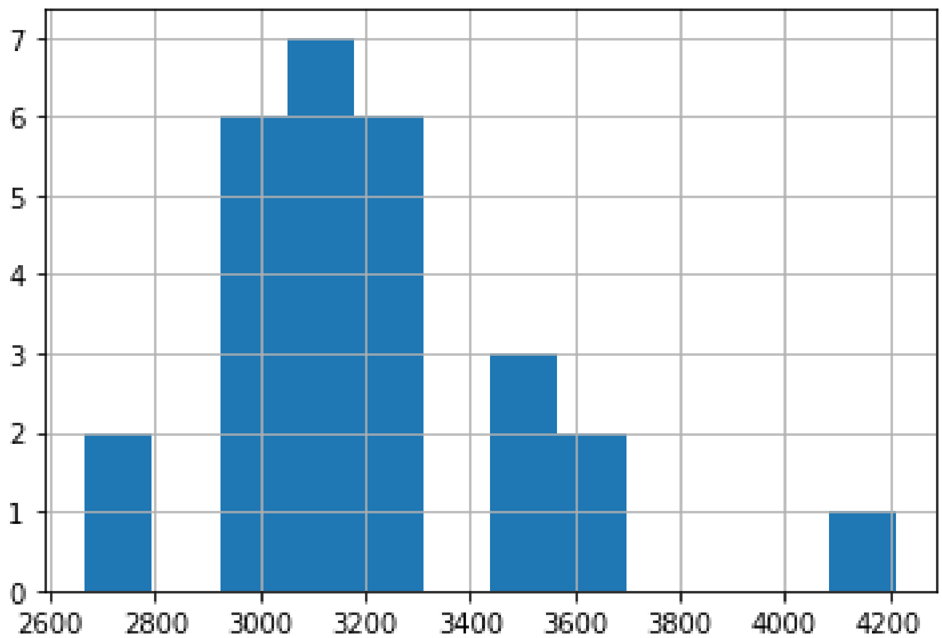
c) 4200mg seems to be an outlier. From the graphical representation this appears to be an outlier but further tests i.e (1.5IQR) need to be conducted to be sure.

d) The distribution is centered around 3200mg and has an asymetrical bell shape spreading from 2600mg to 3600mg

Histogram

[10]:
```
potFrame = DataFrame(potdata,
                     index=['ID'],
                     columns=pd.Index(['Potassium_mg','Dose','Source'],
                                      name='ID'))
y.hist(bins=12,)
```

[10]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1bc9c114470>
```



Comparison This histogram does a better job of representing the distribution of the data. Although, this may be a technology issue. The pandas method for creating stem and leaf plots is not the greatest.

7.1 Energy Consumption

[17]:
```
EnergyData = {'Month':['January','February','March','April','May','June',␣
↪'July','August','September','October','November','December'],
              'Energy (Quadrillion BTU)':[9.58,8.46,8.56,7.56,7.66,7.79,8.23,8.
↪21,7.64,7.78,8.19,8.82]}
EnergyFrame = DataFrame(EnergyData,␣
↪index=['January','February','March','April','May','June',␣
↪'July','August','September','October','November','December'],
                        columns=pd.Index(['Energy (Quadrillion␣
↪BTU)'],name='Month'))
EnergyFrame
```

[17]:
```
Month       Energy (Quadrillion BTU)
January                         9.58
February                        8.46
March                           8.56
April                           7.56
May                             7.66
```

```
June                        7.79
July                        8.23
August                      8.21
September                   7.64
October                     7.78
November                    8.19
December                    8.82
```

a) It appears the highest energy consumption takes place during the colder months of the year as can be expected due to increase utilization of heating appliances.

[21]:
```
dataE = [go.Scatter( x=EnergyFrame['Month'], y=EnergyFrame['Energy (Quadrillion␣
↪BTU)'] )]

py.iplot(dataE, filename='pandas-time-series')
```

```
      ␣
↪---------------------------------------------------------------------------

    NameError                                 Traceback (most recent call␣
↪last)

    <ipython-input-21-f8997def36b9> in <module>
  ----> 1 dataE = [go.Scatter( x=EnergyFrame['Month'], y=EnergyFrame['Energy␣
↪(Quadrillion BTU)'] )]
        2
        3 py.iplot(dataE, filename='pandas-time-series')


    NameError: name 'go' is not defined
```

[ ]:
```

```