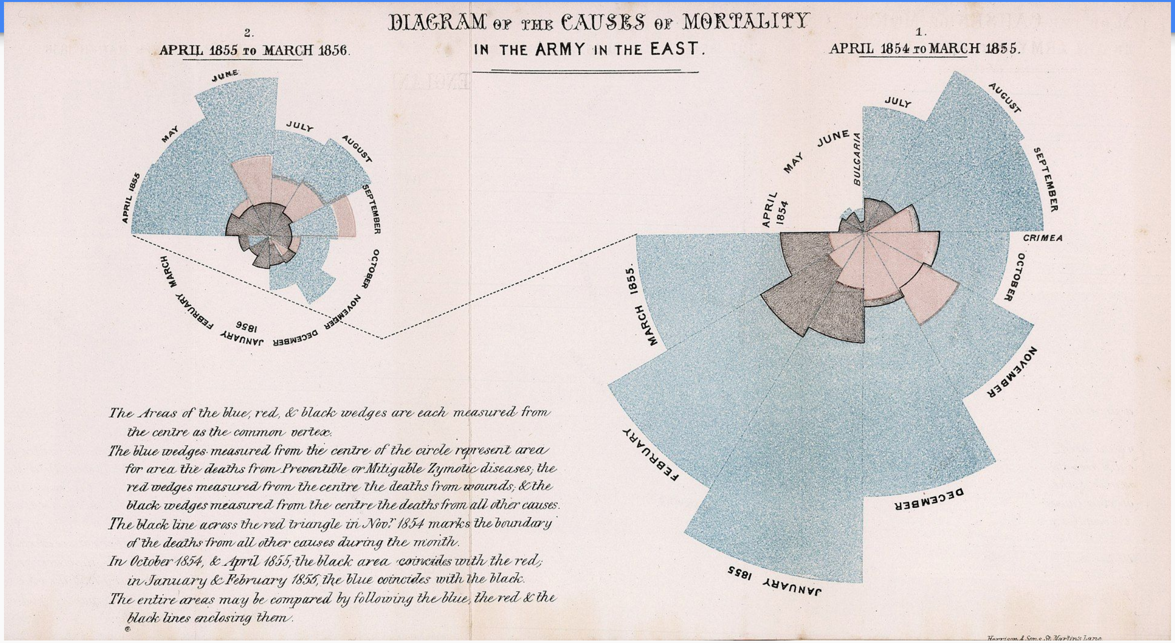
Python: descriptive statistics I

Third tutorial session

Descriptive statistics I

Rose diagram



The Areas of the blue, red, & black wedges are each measured from the centre as the common vertex.
The blue wedges measured from the centre of the circle represent area for area the deaths from Preventible or Mitigable Zymotic diseases, the red wedges measured from the centre the deaths from wounds, & the black wedges measured from the centre the deaths from all other causes.
The black line across the red triangle in Nov.r 1854 marks the boundary of the deaths from all other causes during the month.
In October 1854, & April 1855, the black area coincides with the red; in January & February 1856, the blue coincides with the black.
The entire areas may be compared by following the blue, the red & the black lines enclosing them.

Diagram of the causes of mortality in the army in the east: courtesy Wikipedia

Florence Nightingale

Florence Nightingale: courtesy Wikipedia

Grammar of graphics

1. Graphic: maps the ***data*** to the ***aesthetic attributes*** (colour, shape, size) of ***geometric objects*** (points, lines, bars)

   Wilkinson, Leland. 2005. The Grammar of Graphics. 2nd ed. Statistics and Computing. Springer.

2. Layers and scaling: notions associated with plots

Some general principles

1. Visualise data: first step in dealing with data
2. Never manipulate original data in the application: a problem faced with some excel worksheets
3. Import data: never enter manually, as far as possible
4. Visualise at every stage: for example, after regression, always plot the regression line and the data
5. What are the different ways of visualising data: descriptive statistics

Python: plotting data

Histograms and frequency tables

Consider the sampling of 10 fruits from a basket containing 100 fruits of which 15% are known to be bad. Suppose you generated data by sampling 10 times.

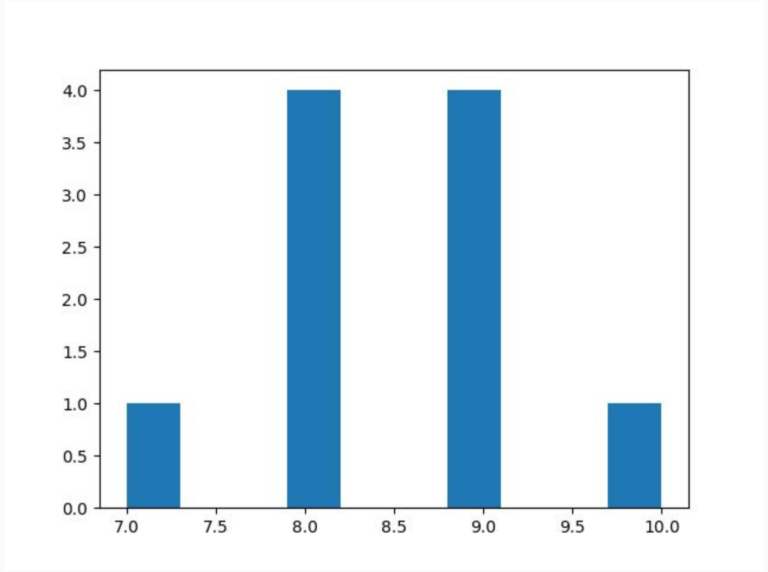
[9 9 10 8 8 10 8 7 9 6]

How do we visualise this data? Plotting the frequency of different numbers: histogram plot!

Frequency tables: 10–2; 9–3; 8–3; 7–1; 6–1

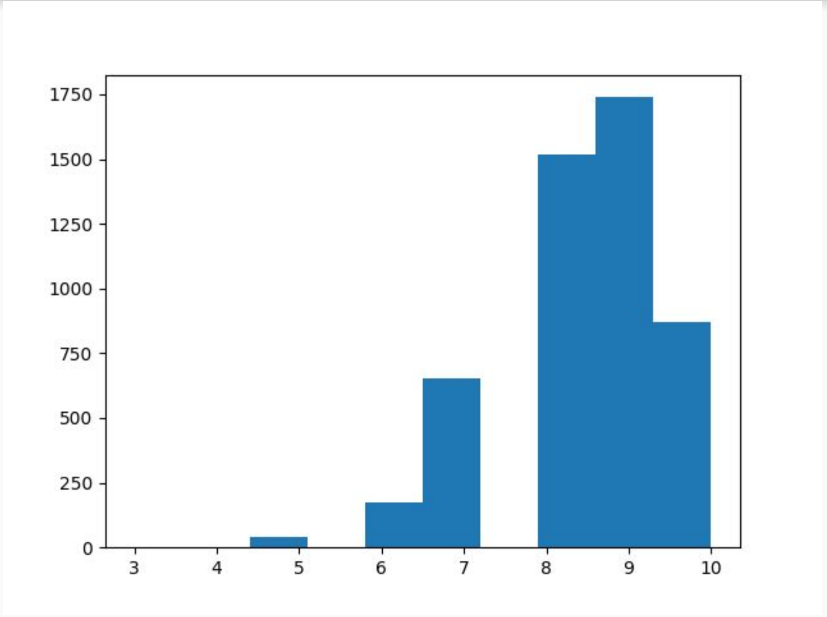
Histogram plot script

```
from numpy import random
import matplotlib.pyplot as plt

x = random.hypergeometric(85,15,10,10)

plt.hist(x)
plt.show()
```

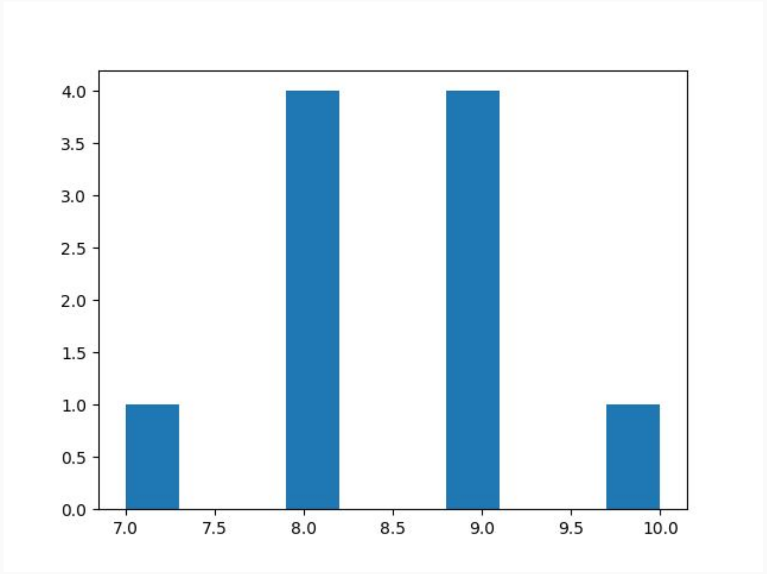


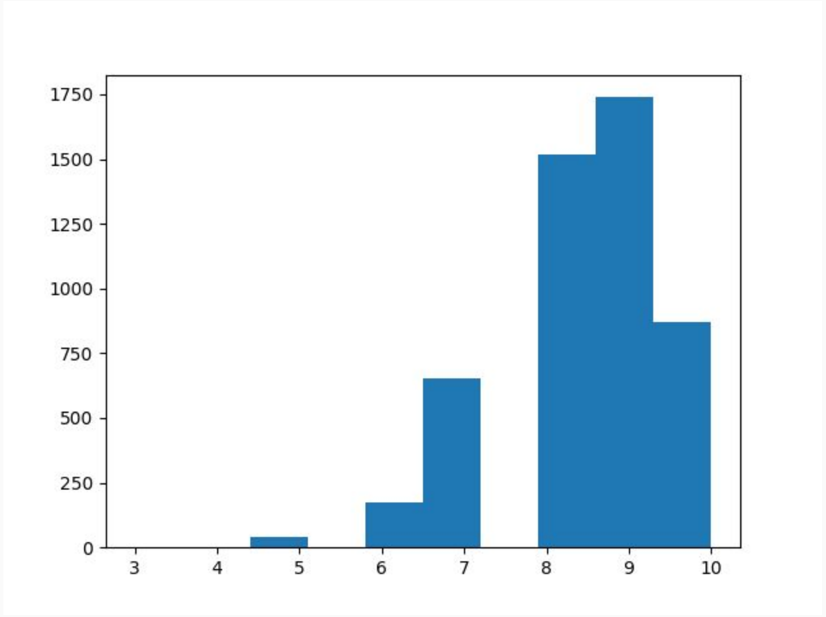
Histogram plot: makes sense when the data is large!

```
from numpy import random
import matplotlib.pyplot as plt

x = random.hypergeometric(85,15,10,5000)

plt.hist(x)
plt.show()
```



Effect of number of data points: skewness





Python: importing data, manipulations and plotting

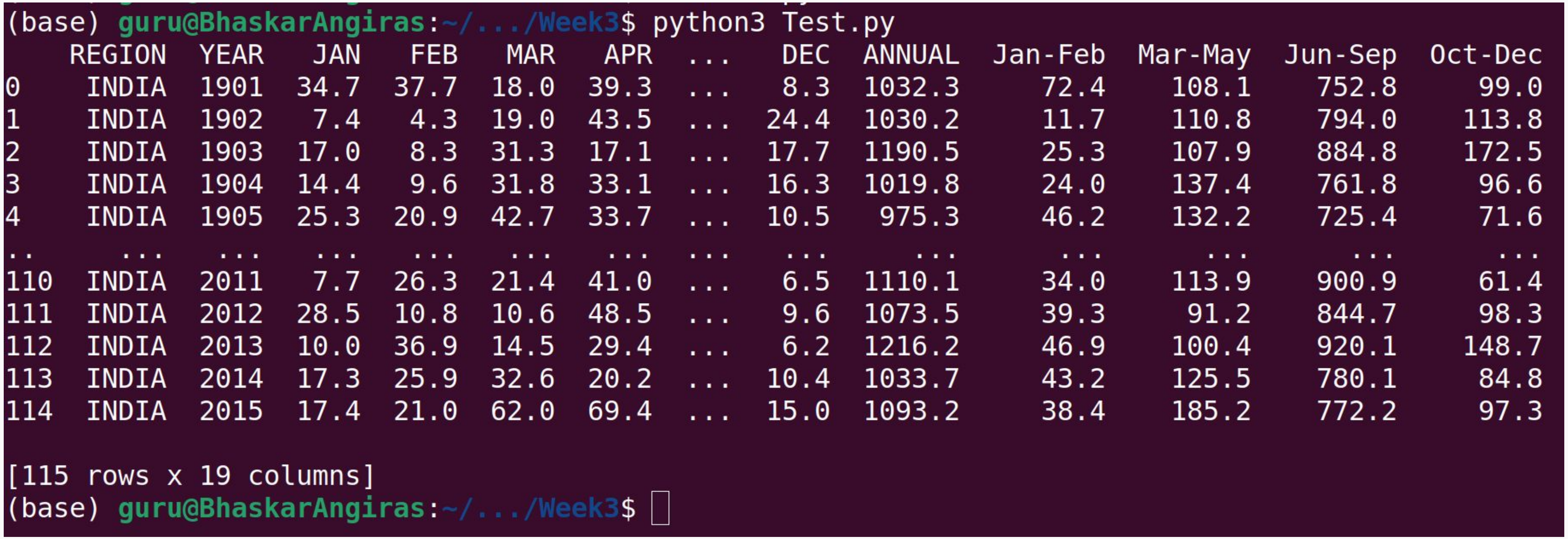
100 years of rainfall in India!

1. https://www.data.gov.in/ A digital India initiative
2. Rainfall for all of India for more than 115 years in CVS format
3. Sub-division-wise rainfall for more than 115 years in CVS format
4. How to read the data?
5. How to visualise it?
6. Suppose, you want to plot a pie-chart of rain season-wise for the year 1982 in Vidarbha region. How to manipulate the data and separate out the data we want?
7. How do we plot the pie-chart?

Pandas: for data import and manipulation

```
import pandas as pd
Rainfall = pd.read_csv('IndiaRainfall.csv')
print(Rainfall)
```

Pandas: output of print command

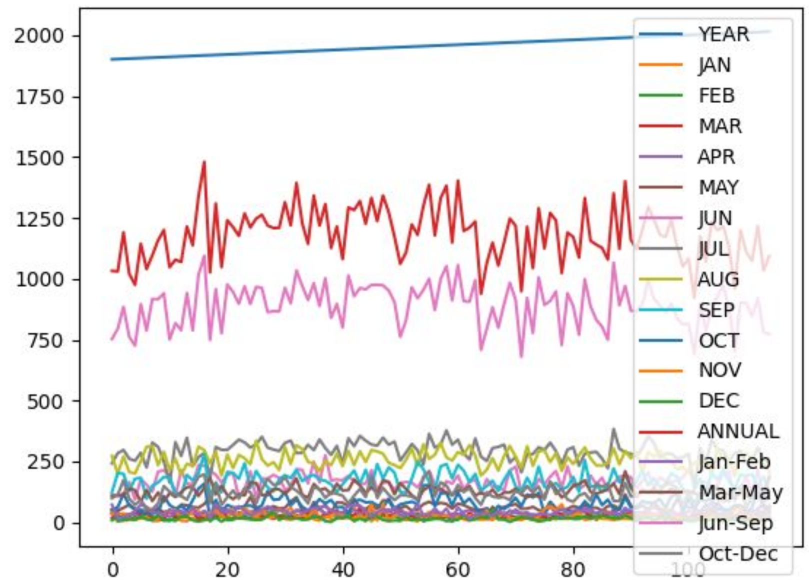


```
(base) guru@BhaskarAngiras:~/.../Week3$ python3 Test.py
    REGION  YEAR   JAN   FEB   MAR   APR  ...   DEC  ANNUAL  Jan-Feb  Mar-May  Jun-Sep  Oct-Dec
0    INDIA  1901  34.7  37.7  18.0  39.3  ...   8.3  1032.3     72.4    108.1    752.8     99.0
1    INDIA  1902   7.4   4.3  19.0  43.5  ...  24.4  1030.2     11.7    110.8    794.0    113.8
2    INDIA  1903  17.0   8.3  31.3  17.1  ...  17.7  1190.5     25.3    107.9    884.8    172.5
3    INDIA  1904  14.4   9.6  31.8  33.1  ...  16.3  1019.8     24.0    137.4    761.8     96.6
4    INDIA  1905  25.3  20.9  42.7  33.7  ...  10.5   975.3     46.2    132.2    725.4     71.6
..     ...   ...   ...   ...   ...   ...  ...   ...     ...      ...      ...      ...      ...
110  INDIA  2011   7.7  26.3  21.4  41.0  ...   6.5  1110.1     34.0    113.9    900.9     61.4
111  INDIA  2012  28.5  10.8  10.6  48.5  ...   9.6  1073.5     39.3     91.2    844.7     98.3
112  INDIA  2013  10.0  36.9  14.5  29.4  ...   6.2  1216.2     46.9    100.4    920.1    148.7
113  INDIA  2014  17.3  25.9  32.6  20.2  ...  10.4  1033.7     43.2    125.5    780.1     84.8
114  INDIA  2015  17.4  21.0  62.0  69.4  ...  15.0  1093.2     38.4    185.2    772.2     97.3

[115 rows x 19 columns]
(base) guru@BhaskarAngiras:~/.../Week3$ 
```

Pandas: for data import and plotting

```
import pandas as pd
import matplotlib.pyplot as plt

Rainfall = pd.read_csv('IndiaRainfall.csv')

Rainfall.plot()
plt.show()
```

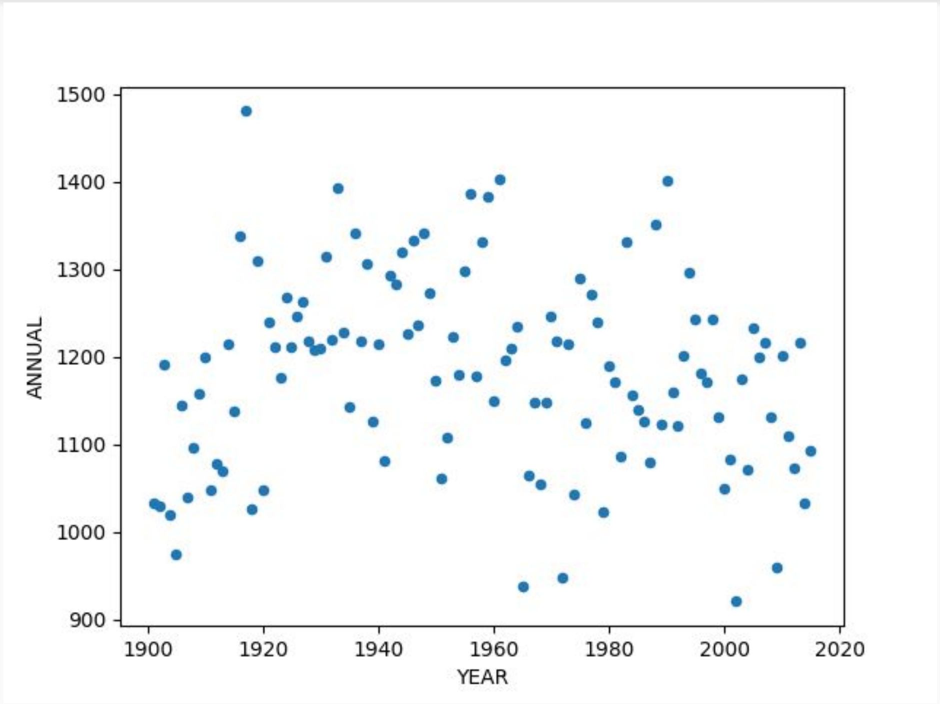
Output of plot command



Pandas: for data import and scatter plotting

```
import pandas as pd
import matplotlib.pyplot as plt

Rainfall = pd.read_csv('IndiaRainfall.csv')

Rainfall.plot(kind=’scatter’,x=’YEAR’,y=’ANNUAL’)
plt.show()
```

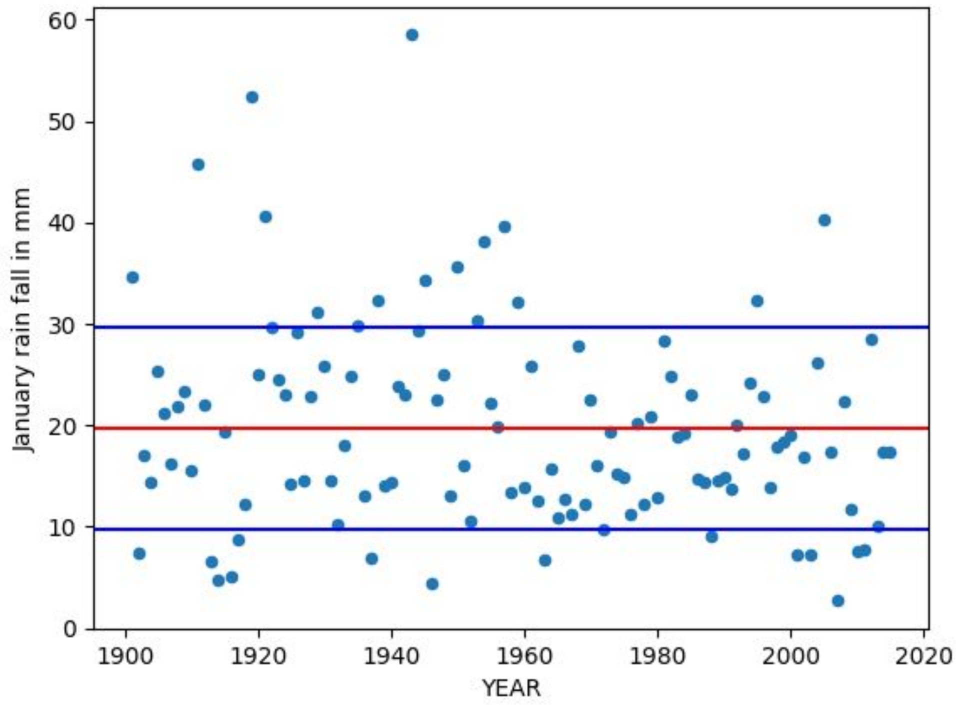
Scatter plot

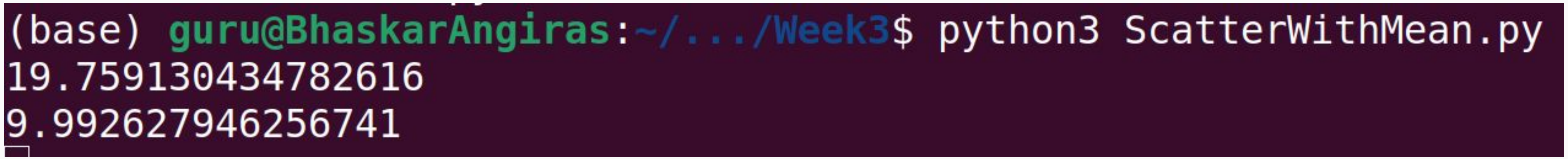


Plotting with statistics

```
import pandas as pd
import matplotlib.pyplot as plt
Rainfall = pd.read_csv(’IndiaRainfall.csv’)
average = Rainfall["JAN"].mean()
error = Rainfall["JAN"].std()
ScatPlot = Rainfall.plot(kind=’scatter’,x="YEAR",y="JAN",ylabel="January rain fall in mm")
ScatPlot.axhline(y=average, color=’r’)
ScatPlot.axhline(y=average+error,color=’b’)
ScatPlot.axhline(y=average-error,color=’b’)
print(average)
print(error)
plt.show()
```

Plot with statistics



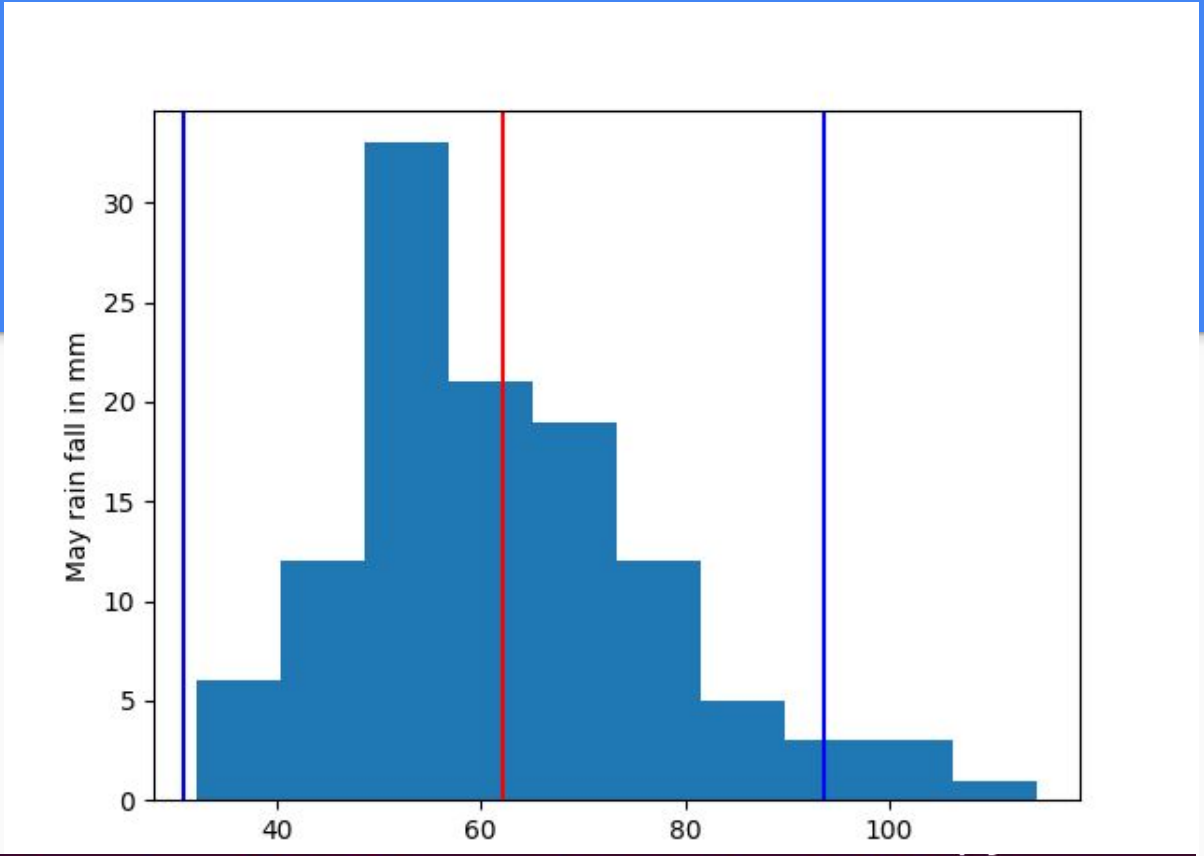


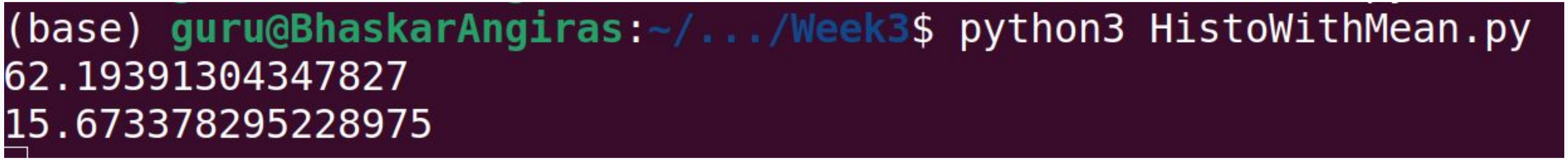
```
(base) guru@BhaskarAngiras:~/.../Week3$ python3 ScatterWithMean.py
19.759130434782616
9.992627946256741
```

What does this code do?

```
import pandas as pd
import matplotlib.pyplot as plt
Rainfall = pd.read_csv('IndiaRainfall.csv')
average = Rainfall["MAY"].mean()
error = Rainfall["MAY"].std()
HistPlot = Rainfall["MAY"].plot(kind='hist',ylabel="May rain fall in mm")
HistPlot.axvline(x=average, color='r')
HistPlot.axvline(x=average+2*error,color='b')
HistPlot.axvline(x=average-2*error,color='b')
print(average)
print(error)
plt.show()
```

Histogram





```
(base) guru@BhaskarAngiras:~/.../Week3$ python3 HistoWithMean.py
62.19391304347827
15.673378295228975
```

Summary

1. Pandas : a very powerful module
2. Pandas: can import csv and excel data files
3. Pandas: to manipulate and extract data – in the next session
4. Pandas + matplotlib : a very powerful tool

Thank you!!

ALL THE BEST!