****

**NAME:JASBEER SINGH**

**MOB NO:8932849535**

**PYTHON PROJECT**

**(with a sample store data)**

**Under the guidance:SHUBHAM GAUTAM SIR**

**(MAKE CAREER ACADEMY,LUCKNOW)**

**Q1.Show the 3 line graph of profit on random30-30 row sample?**

**CODE: A**

**import pandas as pd**

**import matplotlib.pyplot as plt**

**from numpy import random as rd**

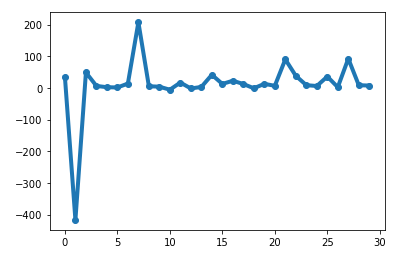
**d=pd.read\_excel(r"C:\Users\hp\Downloads\sample\_-\_superstore.xls")**

**pro1=rd.choice(d["Profit"],size=30)**

**plt.plot(pro1,marker='o', linewidth='4')**

**plt.show()**

**Output:**



**Code:B**

**import pandas as pd**

**import matplotlib.pyplot as plt**

**from numpy import random as rd**

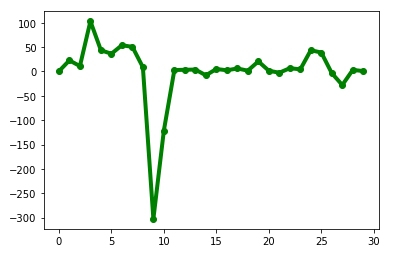
**d=pd.read\_excel(r"C:\Users\hp\Downloads\sample\_-\_superstore.xls")**

**pro1=rd.choice(d["Profit"],size=30)**

**plt.plot(pro1,marker='o', color='green', linewidth='4')**

**plt.show()**

**Output:**

****

**Code:C**

**import pandas as pd**

**import matplotlib.pyplot as plt**

**from numpy import random as rd**

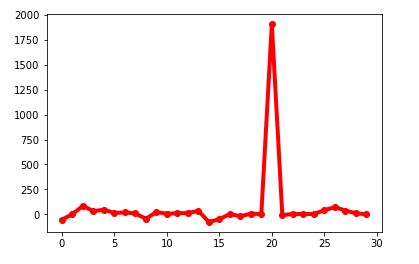
**d=pd.read\_excel(r"C:\Users\hp\Downloads\sample\_-\_superstore.xls")**

**pro1=rd.choice(d["Profit"],size=30)**

**plt.plot(pro1,marker='o', color='red', linewidth='4')**

**plt.show()**

**Output:**

****

**Q2.Show the 2 PIE Char of sales on random 10-10 row sample with label state?**

**CODE:a**

**import pandas as pd**

**import matplotlib.pyplot as plt**

**from numpy import random as rd**

**d=pd.read\_excel(r"C:\Users\hp\Downloads\sample\_-\_superstore.xls")**

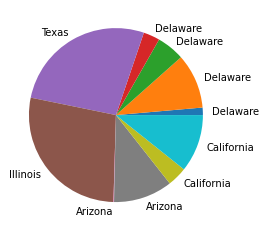
**sal=d["Sales"][120:130]**

**lab=d["State"][120:130]**

**plt.pie(sal, labels=lab)**

**plt.show()**

**Output:**

****

**b.** **import pandas as pd**

**import matplotlib.pyplot as plt**

**from numpy import random as rd**

**d=pd.read\_excel(r"C:\Users\hp\Downloads\sample\_-\_superstore.xls")**

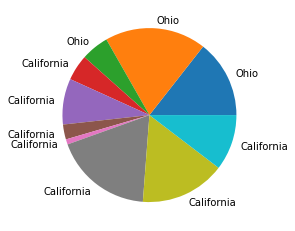
**sal=rd.choice(d["Sales"],size=30)**

**lab=rd.choice(d["State"],size=30)**

**plt.pie(sal, labels=lab)**

**plt.show()**

**Output:**

****

**Q3.Show the 2 PIE Char of profit on random 10-10 row sample with label state?**

**Code:a**

**import pandas as pd**

**import matplotlib.pyplot as plt**

**from numpy import random as rd**

**d=pd.read\_excel(r"C:\Users\hp\Downloads\sample\_-\_superstore.xls")**

**p=d["Profit"][122:132]**

**l=[]**

**a=1**

**for i in d["Profit"][122:]:**

**while a<=10:**

**if i>0:**

**l.append(i)**

**a=a+1**

**lab=d["State"][122:132]**

**plt.pie(l,labels=lab)**

**plt.show()**

**Output:**

****

**Code:b**

**import pandas as pd**

**import matplotlib.pyplot as plt**

**from numpy import random as rd**

**d=pd.read\_excel(r"C:\Users\hp\Downloads\sample\_-\_superstore.xls")**

**p=d["Profit"][132:142]**

**l=[]**

**a=1**

**for i in d["Profit"][132:]:**

**while a<=10:**

**if i>0:**

**l.append(i)**

**a=a+1**

**lab=d["State"][132:142]**

**plt.pie(l,labels=lab)**

**plt.show()**

**Output:**

****

**Q4.Show the 3 line graph comparison between profit and sales on 30-30 rows sample ?**

**Code:a**

**import pandas as pd**

**import matplotlib.pyplot as plt**

**d=pd.read\_excel(r"C:\Users\hp\Downloads\sample\_-\_superstore.xls")**

**pro=d["Profit"][50:80]**

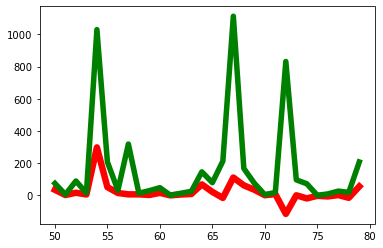
**sal=d["Sales"][50:80]**

**plt.plot(pro, color='red', linewidth='6.5')**

**plt.plot(sal, color='green', linewidth='5.5')**

**plt.show()**

**Output:**

****

**Code:b**

**import pandas as pd**

**import matplotlib.pyplot as plt**

**d=pd.read\_excel(r"C:\Users\hp\Downloads\sample\_-\_superstore.xls")**

**pro=d["Profit"][160:190]**

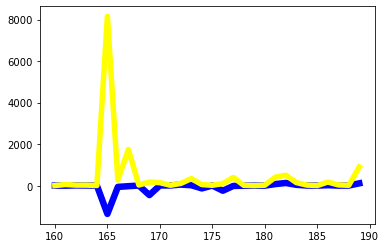
**sal=d["Sales"][160:190]**

**plt.plot(pro, color='blue', linewidth='6.5')**

**plt.plot(sal, color='yellow', linewidth='5.5')**

**plt.show()**

**Output:**

****

**Code:C**

**import pandas as pd**

**import matplotlib.pyplot as plt**

**d=pd.read\_excel(r"C:\Users\hp\Downloads\sample\_-\_superstore.xls")**

**pro=d["Profit"][1500:1530]**

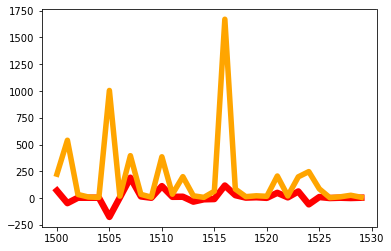
**sal=d["Sales"][1500:1530]**

**plt.plot(pro, color='red', linewidth='6.5')**

**plt.plot(sal, color='orange', linewidth='5.5')**

**plt.show()**

**Output:**

****

**Q5. Show the 2 Bar charts on profit and sales random 30-30 rows sample with label?**

**Code:a**

**import pandas as pd**

**import matplotlib.pyplot as plt**

**from numpy import random as rd**

**d=pd.read\_excel(r"C:\Users\hp\Downloads\sample\_-\_superstore.xls")**

**pro=rd.choice(d["Profit"],size=30)**

**sal=rd.choice(d["Sales"],size=30)**

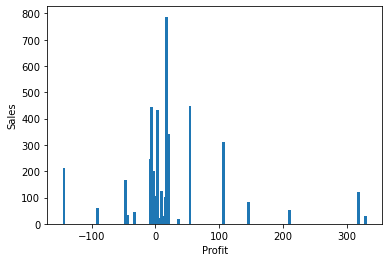
**plt.bar(pro,sal, width=4)**

**plt.xlabel("Profit")**

**plt.ylabel("Sales")**

**plt.show()**

**Output:**

****

**Code:b**

**import pandas as pd**

**import matplotlib.pyplot as plt**

**from numpy import random as rd**

**d=pd.read\_excel(r"C:\Users\hp\Downloads\sample\_-\_superstore.xls")**

**pro=rd.choice(d["Profit"],size=30)**

**sal=rd.choice(d["Sales"],size=30)**

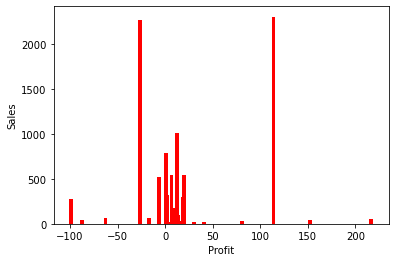
**plt.bar(pro,sal, width=4, color="red")**

**plt.xlabel("Profit")**

**plt.ylabel("Sales")**

**plt.show()**

**Output:**

****

**Q6.What is Data Cleaning and need of data cleaning?**

**Ans. Data Cleaning is a technique which helps to convert improper data into meaningful data.**

**Data cleaning means fixing bad data in your data set.**

**Bad data could be:**

* **Empty cells**
* **Data in wrong format**
* **Wrong data**
* **Duplicates**

**We will carry out the below steps for Data Cleaning.**

1. **Remove Repeated Row**
2. **Missing value treatment**
3. **Removal of Irrelevant Data**
4. **Manual error while typing**
5. **Renaming Columns**

**Need Of Data Cleaning**

* **Empty cell**

Empty cells can potentially give you a wrong result when you analyze data

* **Data in wrong format**

Cells with data of wrong format can make it difficult, or even impossible, to analyze data.

To fix it, you have two options: remove the rows, or convert all cells in the columns into the same format.

* **Wrong data**

"Wrong data" does not have to be "empty cells" or "wrong format", it can just be wrong, like if someone registered "199" instead of "1.99".

* **Duplicates**

Duplicate rows are rows that have been registered more than one time.

**Q7.Define Mean, Media, Mode, variance, Standard Deviation?**

**Ans.**

## Mean:

## The mean value is the average value.

**To calculate the mean, find the sum of all values, and divide the sum by the number of values**

### **Example**

**Use the NumPy mean() method to find the average speed:**

**import numpy  
  
speed = [99,86,87,88,111,86,103,87,94,78,77,85,86]  
  
x = numpy.mean(speed)  
  
print(x)**

89.76923076923077

## Median

**The median value is the value in the middle, after you have sorted all the values**

### **Example**

**Use the NumPy median() method to find the middle value:**

**import numpy  
  
speed = [99,86,87,88,111,86,103,87,94,78,77,85,86]  
  
x = numpy.median(speed)  
  
print(x)**

87.0

## Mode:

**The Mode value is the value that appears the most number of times**

### **Example**

**Use the SciPy mode() method to find the number that appears the most:**

**from scipy import stats  
  
speed = [99,86,87,88,111,86,103,87,94,78,77,85,86]  
  
x = stats.mode(speed)  
  
print(x)**

ModeResult(mode=array([86]), count=array([3]))

## Variance

## 1. Find the mean:

## 2. For each value: find the difference from the mean:

## 3. For each difference: find the square value:

## 4. The variance is the average number of these squared differences:

## Example:

## import numpy

## speed=[32,111,138,28,59,77,97]

## x=numpy.var(speed)

## print(x)

1432.2448979591834

## Standard Deviation

Standard deviation is a number that describes how spread out the values are.

A low standard deviation means that most of the numbers are close to the mean (average) value.

A high standard deviation means that the values are spread out over a wider range.

### **Example**

Use the NumPy std() method to find the standard deviation:

import numpy  
  
speed = [86,87,88,86,87,85,86]  
  
x = numpy.std(speed)  
  
print(x)

0.9035079029052513

**Q8.Define supervise learning and unsupervised learning?**

**Ans.**[**Supervised learning**](https://www.ibm.com/cloud/learn/supervised-learning) **is a machine learning approach that’s defined by its use of labeled datasets. These datasets are designed to train or “supervise” algorithms into classifying data or predicting outcomes accurately. Using labeled inputs and outputs, the model can measure its accuracy and learn over time.**

**Supervised learning can be separated into two types of problems when**[**data mining**](https://www.ibm.com/cloud/learn/data-mining)**: classification and regression:**

* **Classification problems use an algorithm to accurately assign test data into specific categories, such as separating apples from oranges. Or, in the real world, supervised learning algorithms can be used to classify spam in a separate folder from your inbox. Linear classifiers, support vector machines, decision trees and**[**random forest**](https://www.ibm.com/cloud/learn/random-forest)**are all common types of classification algorithms.**
* **Regression is another type of supervised learning method that uses an algorithm to understand the relationship between dependent and independent variables. Regression models are helpful for predicting numerical values based on different data points, such as sales revenue projections for a given business. Some popular regression algorithms are linear regression, logistic regression and polynomial regression.**

[**Unsupervised learning**](https://www.ibm.com/cloud/learn/unsupervised-learning)**uses machine learning algorithms to analyze and cluster unlabeled data sets. These algorithms discover hidden patterns in data without the need for human intervention (hence, they are “unsupervised”).**

**Unsupervised learning models are used for three main tasks: clustering, association and dimensionality reduction:**

* **Clustering is a data mining technique for grouping unlabeled data based on their similarities or differences. For example, K-means clustering algorithms assign similar data points into groups, where the K value represents the size of the grouping and granularity. This technique is helpful for market segmentation, image compression, etc.**
* **Association is another type of unsupervised learning method that uses different rules to find relationships between variables in a given dataset. These methods are frequently used for market basket analysis and recommendation engines, along the lines of “Customers Who Bought This Item Also Bought” recommendations.**
* **Dimensionality reduction is a learning technique used when the number of features  (or dimensions) in a given dataset is too high. It reduces the number of data inputs to a manageable size while also preserving the data integrity. Often, this technique is used in the preprocessing data stage, such as when autoencoders remove noise from visual data to improve picture quality.**

## The main difference between supervised and unsupervised learning: Labeled data

**The main distinction between the two approaches is the use of labeled datasets. To put it simply, supervised learning uses labeled input and output data, while an unsupervised learning algorithm does not.**

**In supervised learning, the algorithm “learns” from the training dataset by iteratively making predictions on the data and adjusting for the correct answer. While supervised learning models tend to be more accurate than unsupervised learning models, they require upfront human intervention to label the data appropriately. For example, a supervised learning model can predict how long your commute will be based on the time of day, weather conditions and so on. But first, you’ll have to train it to know that rainy weather extends the driving time.**

**Unsupervised learning models, in contrast, work on their own to discover the inherent structure of unlabeled data. Note that they still require some human intervention for validating output variables. For example, an unsupervised learning model can identify that online shoppers often purchase groups of products at the same time. However, a data analyst would need to validate that it makes sense for a recommendation engine to group baby clothes with an order of diapers, applesauce and sippy cups.**

**Q9.Define Liner regression and Polynomial regression?**

## Ans. Regression

**The term regression is used when you try to find the relationship between variables.**

**In Machine Learning, and in statistical modeling, that relationship is used to predict the outcome of future events.**

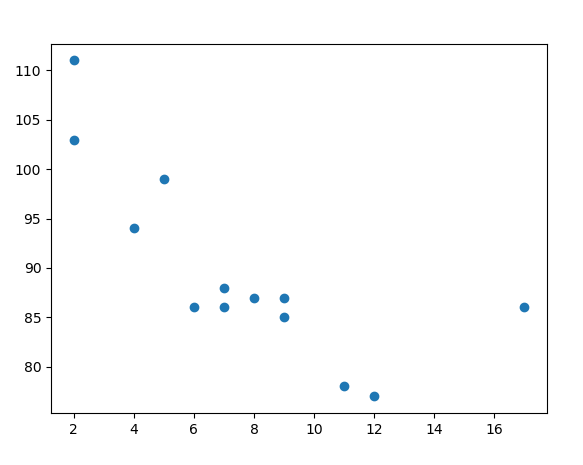
## Linear Regression

**Linear regression uses the relationship between the data-points to draw a straight line through all them.**

### **Example**

Start by drawing a scatter plot:

import matplotlib.pyplot as plt  
  
x = [5,7,8,7,2,17,2,9,4,11,12,9,6]  
y = [99,86,87,88,111,86,103,87,94,78,77,85,86]  
  
plt.scatter(x, y)  
plt.show()

****

## Polynomial Regression

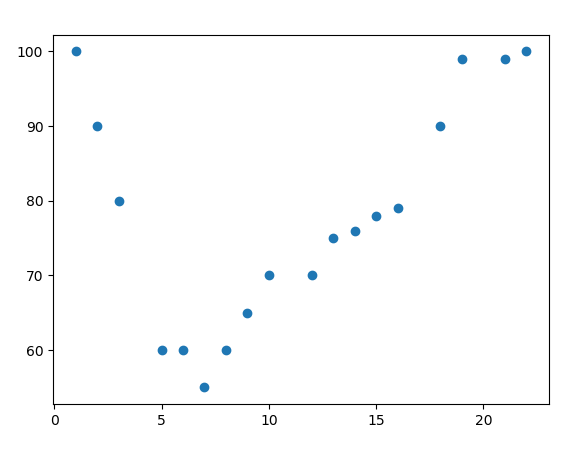
If your data points clearly will not fit a linear regression (a straight line through all data points), it might be ideal for polynomial regression.

Polynomial regression, like linear regression, uses the relationship between the variables x and y to find the best way to draw a line through the data points

### **Example**

**Start by drawing a scatter plot:**

**import matplotlib.pyplot as plt  
  
x = [1,2,3,5,6,7,8,9,10,12,13,14,15,16,18,19,21,22]  
y = [100,90,80,60,60,55,60,65,70,70,75,76,78,79,90,99,99,100]  
  
plt.scatter(x, y)  
plt.show()**

****

**Q10.Define the processs of data fetching from excel and csv with full explanation?**

**Ans:One cruicial feature of Pandas is its ability to write and read Excel,CSV and many other types of files. Function likes the Pandas read\_csv() method enable you to work with files effectively.**

**You can use use them to save the and labels fromPandas objects to a file and load them later as Pandas series or Dataframe.**

**CSV stands for “Comma Separated Values.” It is the simplest form of storing data in tabular form as plain text. It is important to know to work with CSV because we mostly rely on CSV data in our day-to-day lives as data scientists.**

## Read CSV Files

**A simple way to store big data sets is to use CSV files (comma separated files).**

**CSV files contains plain text and is a well know format that can be read by everyone including Pandas.**

**Example:**

**In our examples we will be using a CSV file called 'data.csv'.**

**import pandas as pd  
  
df = pd.read\_csv('data.csv')  
  
print(df)**

### **Reading an excel file using Python using Pandas**

**In this method, We will first import the Pandas module then we will use Pandas to read our excel file. You can read more operations using the excel file using Pandas**

**Example:**

**# import pandas lib as pd**

**import pandas as pd**

**# read by default 1st sheet of an excel file**

**dataframe1 = pd.read\_excel('book2.xlsx')**

**print(dataframe1)**