DATA SCIENCE LAB MANUAL

1. **Consider the following data of three cricket players in 10 innings T20 Match**

Player	1	2	3	4	5	6	7	8	9	10
Cricketer1	25	10	55	45	55	78	55	0	49	10
Cricketer2	47	62	78	45	100	20	100	0	80	10
Cricketer3	80	17	7	10	45	79	75	75	80	42

a) **Find Whose average is better.**
b) **What is the middlemost value of each player?**
c) **Whose most frequent value is good.**
d) **Draw a simple plot to show performance of players.**

Solution:

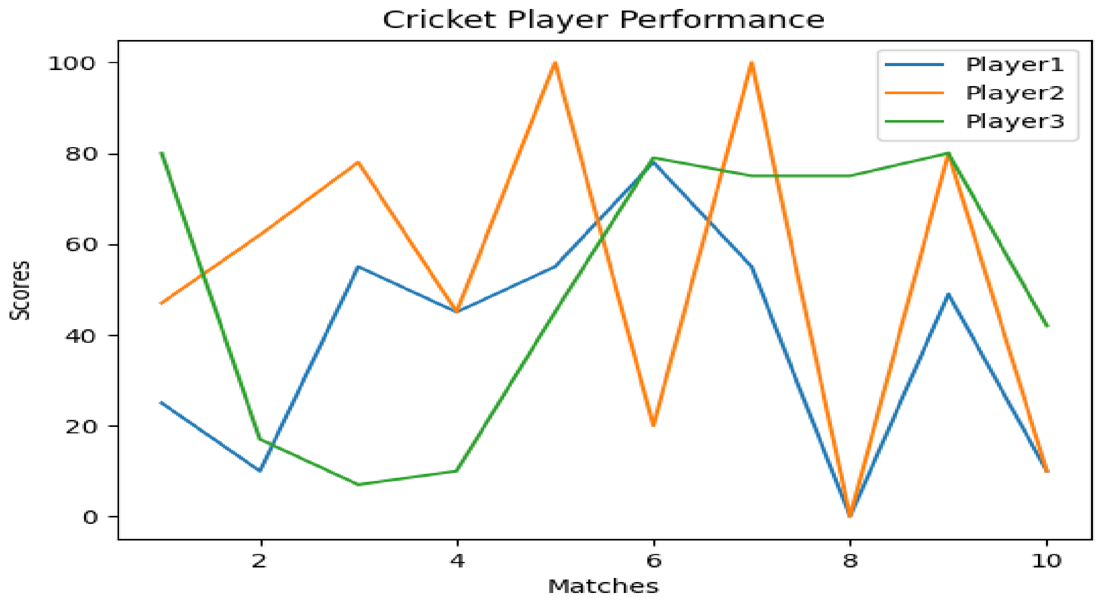
```
#Cricket Player Performance Analysis
import statistics as st
import matplotlib.pyplot as pt
import tabulate
Matches=[1,2,3,4,5,6,7,8,9,10]
Player1=[25,10,55,45,55,78,55,0,49,10]
Player2=[47,62,78,45,100,20,100,0,80,10]
Player3=[80,17,7,10,45,79,75,75,80,42]
#Player1 Summary
print("Player1 Mean = ",st.mean(Player1))
print("Player1 Median = ",st.median(Player1))
print("Player1 Mode = ",st.mode(Player1))
#Player2 Summary
print("Player2 Mean = ",st.mean(Player2))
print("Player2 Median = ",st.median(Player2))
print("Player2 Mode = ",st.mode(Player2))
#Player3 Summary
print("Player3 Mean = ",st.mean(Player3))
print("Player3 Median = ",st.median(Player3))
print("Player3 Mode = ",st.mode(Player3))
#Performance plot
pt.plot(Matches,Player1)
pt.plot(Matches,Player2)
pt.plot(Matches,Player3)
pt.title("Cricket Player Performance")
pt.xlabel("Matches")
pt.ylabel("Scores")
pt.legend(["Player1","Player2","Player3"])
pt.show()
```

OUTPUT:

```
Player1 Mean =  38.2

Player1 Median =  47.0

Player1 Mode =  55

Player2 Mean =  54.2

Player2 Median =  54.5

Player2 Mode =  100

Player3 Mean =  51

Player3 Median =  60.0

Player3 Mode =  80
```

Analysis

a) Player 2 average is better.
b) Player1 Median = 47.0, Player2 Median = 54.5, Player3 Median = 60.0
c) Player2
d) Draw a simple plot to show performance of players.



2. Consider Insurance Dataset and analyze following
    a) Count Number of Male and Female
    b) What is average age of peoples.
    c) Display simple bar plot Gender wise

Solution:

```
import pandas as pd
import openpyxl
import statistics as st
import matplotlib.pyplot as pt
data = pd.read_csv("E:\Data Science with
Python\DataSet\insurance.csv")
print(data)
#Analysis genderwise
ls=data['sex'].tolist()
y1=ls.count('female')
y2=ls.count('male')
print("female Count = ",y1)
print("male Count = ",y2)

#Aveage age of customers
avgage=data['age'].tolist()
print("Average Age=  %.2f " % st.mean(avgage))

#Display Histogram genderwise
x=["FEMALE","MALE"]
y=[y1,y2]
pt.bar(x,y)
pt.title("Genderwise Insurance Data")
pt.xlabel("Gender")
pt.ylabel("Count")
pt.show()
```

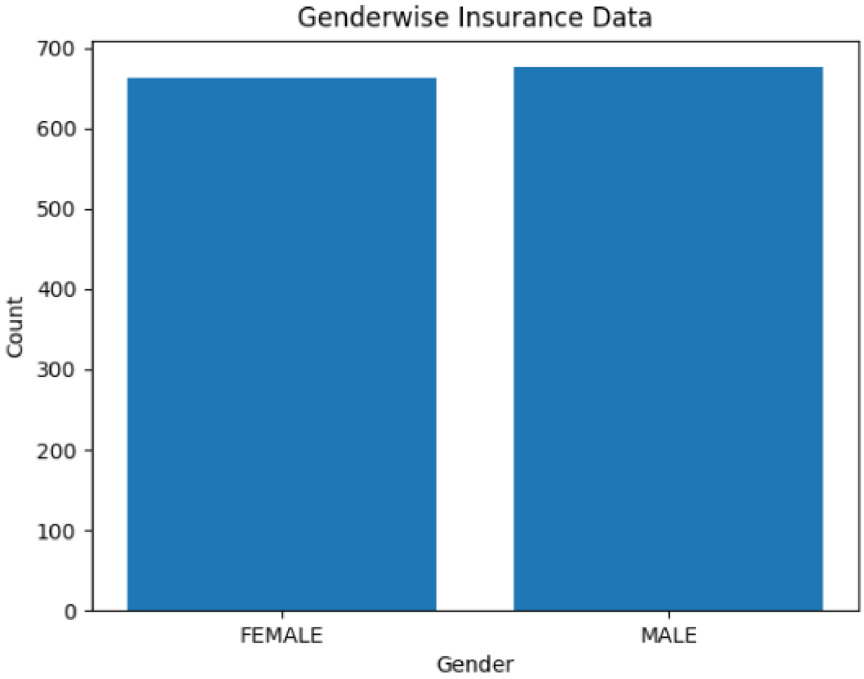
Analysis:

a)

```
female Count =  662

male Count =  676
```

b) Average Age= 39.21

c)


Genderwise Insurance Data
700
600
500
400
300
200
100
0
Count
FEMALE
MALE
Gender

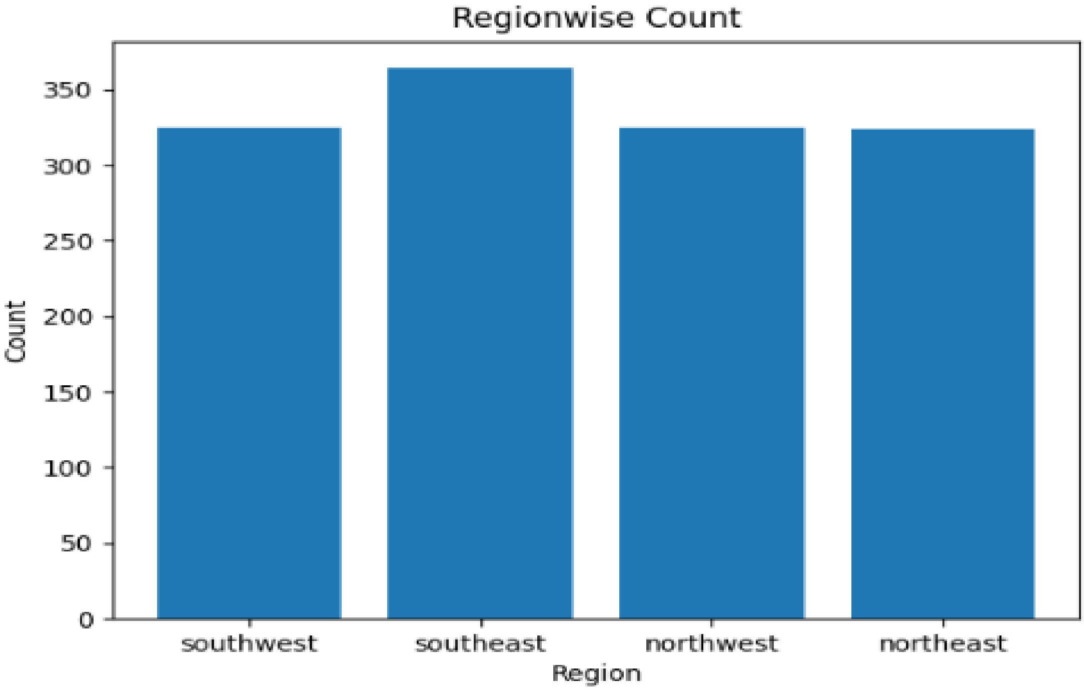
3. Consider Insurance Dataset and analyze data region wise. Also display simple bar chart region wise.

Solution:

```
import pandas as pd
import openpyxl
import matplotlib.pyplot as pt
data = pd.read_csv("E:\Data Science with
Python\DataSet\insurance.csv")
print(data)

#Regionwise count
region=data['region'].tolist()
output=[]
for x in region:
    if x not in output:
        output.append(x)
print(output)
y1=region.count('southwest')
y2=region.count('southeast')
y3=region.count('northwest')
y4=region.count('northeast')
print("Southwest count= ",y1)
print("southeast count= ",y2)
print("northwest count= ",y3)
print("northeast count= ",y4)
pt.title("Regionwise Count")
pt.xlabel("Region")
pt.ylabel("Count")
y=[y1,y2,y3,y4]
pt.bar(output,y)
pt.show()
```

Analysis:

```
Southwest count=  325
southeast count=  364
northwest count=  325
northeast count=  324
```


Regionwise Count
Count
350
300
250
200
150
100
50
0
southwest
southeast
northwest
northeast
Region

4. Consider temperature dataset and analyze average of minimum and maximum temperature, minimum temperature, maximum temperature month wise.

   Solution:

```
import pandas as pd
import openpyxl
import numpy as np
data=pd.read_excel("E:\\Data Science with
Python\\DataSet\\belgavitemp2022.xlsx")
print(data)
df1 = (data.groupby(["Year",
"Month"],sort=False).agg(Avg_of_Max_Temp=("Max", 'mean'),
    Max_temp=("Max",'max'),Avg_of_Min_Temp=("Min",
'mean'),Min_temp=("Min",'min')))
print(df1)
```

Analysis:

```
            Avg_of_Max_Temp  Max_temp  Avg_of_Min_Temp  Min_temp
Year Month
2022 January       29.290323        33        14.838710        11
     February      32.535714        35        16.928571        14
     March         35.451613        39        20.322581        17
     April         36.666667        39        22.300000        19
     May           33.838710        38        21.612903        19
     June          31.533333        36        21.033333        20
     July          28.225806        33        20.451613        19
     August        28.419355        32        20.258065        19
     September     29.533333        32        19.833333        18
     October       29.741935        32        18.677419        14
     November      30.433333        32        16.433333        11
     December      29.870968        33        17.967742        14
```

5.Consider following data and calculate Descriptive statistics using formules.

22,26,14,30,18,1135,41,12,32

Solution:

```
import numpy as np
import pandas as pd
data=[22,26,14,30,18,11,35,41,12,32]
print("Mean = %.2f"% np.mean(data))
print("Median = ",np.median(data))
print("Max = ",np.max(data))
print("Min = ",np.min(data))
print("First Quartile =",np.quantile(data,0.25))
print("Second Quartile = ",np.quantile(data,0.50))
print("Third Quartile = ",np.quantile(data,0.75))
print("20 th Percentilee = ",np.percentile(data,20))
print("99 th Percentilee = ",np.percentile(data,99))
print("Standard deviation = %.2f" % np.std(data))
print("Variance  = ",np.var(data))
```

OUTPUT:

```
Mean = 24.10
Median =  24.0
Max =  41
Min =  11
First Quartile = 15.0
Second Quartile =  24.0
Third Quartile =  31.5
20 th Percentilee =  13.6
99 th Percentilee =  40.46
Standard deviation = 9.83
Variance  =  96.69
```

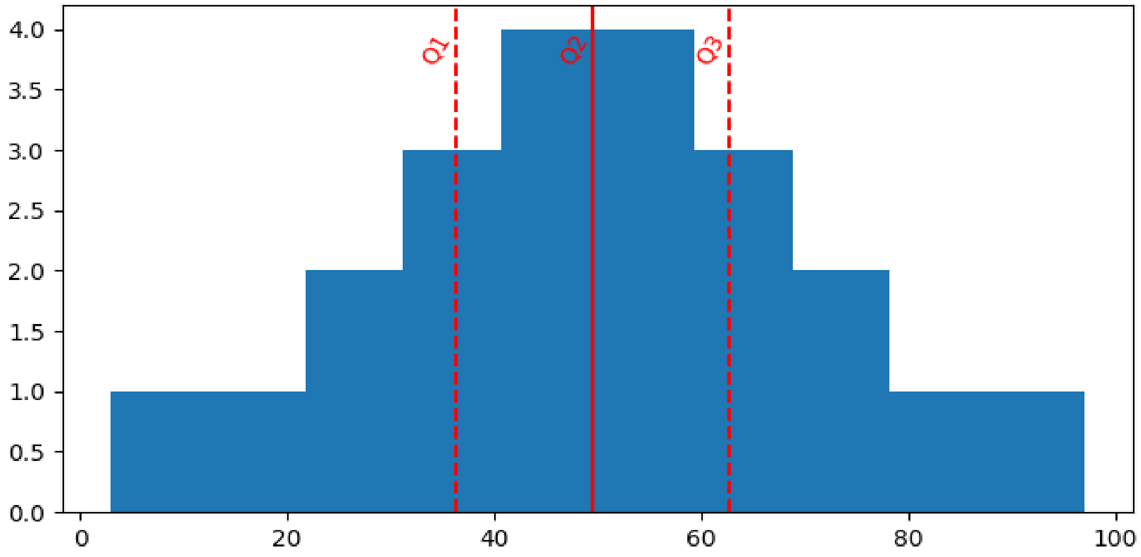
6. Find the Quartiles for the following Students Score data and visualize graphically.
   50,50,47,97,49,3,53,42,26,74,82,62,37,15,70,27,36,35,48,52,63,64.

Solution:

```
import numpy as np
import matplotlib.pyplot as pt
import numpy as np
import pandas as pd
data=[50,50,47,97,49,3,53,42,26,74,82,62,37,15,70,27,36,35,48,52,63,64]
print(data)
print("Quartile 1 = %.2f"%np.quantile(data,0.25))
print("Quartile 2 = %.2f"%np.quantile(data,0.50))
print("Quartile 3 = %.2f"%np.quantile(data,0.75))
pt.figure(figsize=(8,4))
pt.hist(data)
# Vertical lines for each percentile of interest
pt.axvline(np.quantile(data, 0.25), linestyle='--', color='red')
pt.text(np.quantile(data, 0.25), 4, 'Q1', color='r', ha='right', va='top',
rotation=60)
pt.axvline(np.quantile(data, 0.50), linestyle='-',  color='red')
pt.text(np.quantile(data, 0.50), 4, 'Q2', color='r', ha='right', va='top',
rotation=60)
pt.axvline(np.quantile(data, 0.75), linestyle='--', color='red')
pt.text(np.quantile(data, 0.75), 4, 'Q3', color='r', ha='right', va='top',
rotation=60)
pt.show()
```

OUTPUT:

```
Quartile 1 = 36.25

Quartile 2 = 49.50

Quartile 3 = 62.75
```



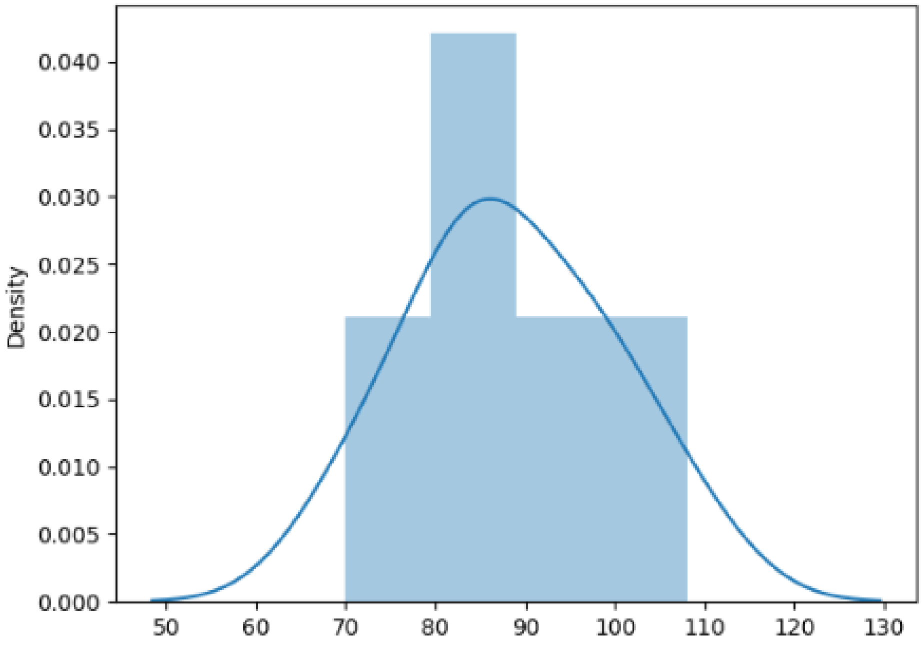
7. Calculate the skewness for the following data also conclude skewness
   85,96,76,108,84,100,86,70,95,84

   Solution

```
# Importing library
import matplotlib.pyplot as pt
import statistics as st
import seaborn as sns
# Creating a dataset
dataset =[85,96,76,108,84,100,86,70,95,84]
meandata=st.mean(dataset)
print("Mean =  %.2f"%meandata)
modedata=st.mode(dataset)
print("Mode =  %.2f"%modedata)
meddata=st.median(dataset)
print("Median =  %.2f"%meddata)
# Calculate the skewness
stddata=st.stdev(dataset)
print("Standard Deviation =%.2f" % stddata)
sk=(meandata-modedata)/stddata
print("Skewness= %.2f" % sk)
sns.distplot(dataset)
pt.show()
```

OUTPUT:

```
Mean =  88.40
Mode =  84.00
Median =  85.50
```



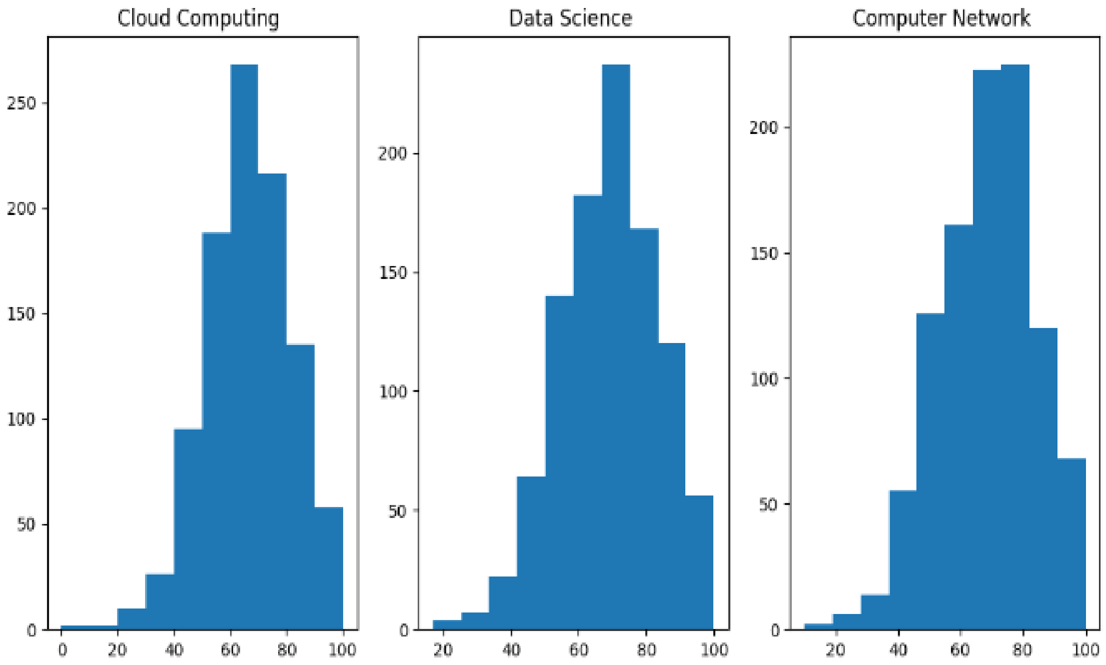
Analysis: Distribution is Positively Skewed.

8. Consider Student Performance dataset and find skewness for all subjects.

```
import pandas as pd
import matplotlib.pyplot as plt
import openpyxl
data =pd.read_csv("E:\Data Science with
Python\DataSet\StudentsPerformance.csv")
print(data)
print("Skew of Cloud Computing score:
%.2f"%data['Cloud Computing'].skew())
print("Skew of Data Science: %.2f"%data['Data
Science'].skew())
print("Skew of Computer Networks:
%.2f"%data['Computer Network'].skew())

plt.figure(figsize = (12,6))
plt.subplot(1, 3, 1)
plt.hist(data['Cloud Computing'])
plt.title('Cloud Computing ')

plt.subplot(1, 3, 2)
plt.hist(data['Data Science'])
plt.title('Data Science ')

plt.subplot(1,3,3)
plt.hist(data['Computer Network'])
plt.title('Computer Network ')

plt.show()
```

OUTPUT:

```
Skew of Cloud Computing score: -0.28
Skew of Data Science: -0.26
Skew of Computer Networks: -0.29
```



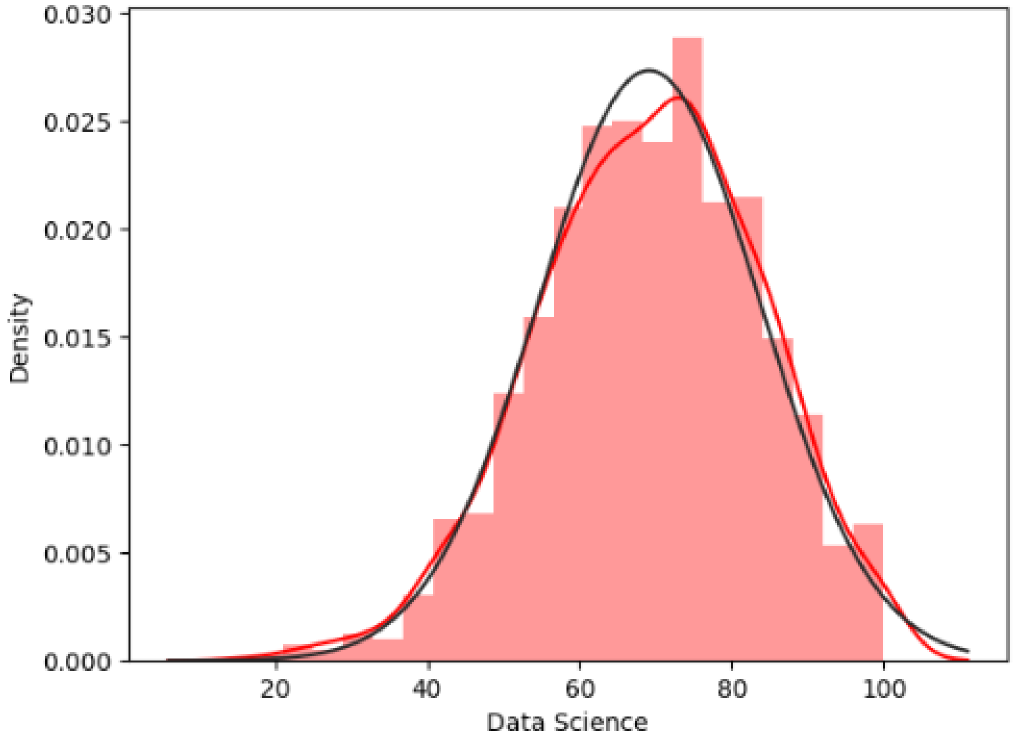
Analysis:
All subjects Distribution is negatively skewed.
Maximum students score between 60-100.

9. Consider Student Performance dataset find basic statistics of data science subject using pandas describe function, calculate skewness also visualize distribution.
   Solution:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import skew, skewtest, norm
import openpyxl
data =pd.read_csv("E:\Data Science with
Python\DataSet\StudentsPerformance.csv")
print(data)
print(data['Data Science'].describe())
print("Skewness= %.2f"%data['Data Science'].skew())
sns.distplot(data['Data Science'], fit=norm, color="r")
plt.show()
```

OUTPUT:

```
count    1000.000000
mean       69.169000
std        14.600192
min        17.000000
25%        59.000000
50%        70.000000
75%        79.000000
max       100.000000
Name: Data Science, dtype: float64
Skewness= -0.26
```


0.030
0.025
0.020
0.015
0.010
0.005
0.000
Density
20
40
60
80
100
Data Science

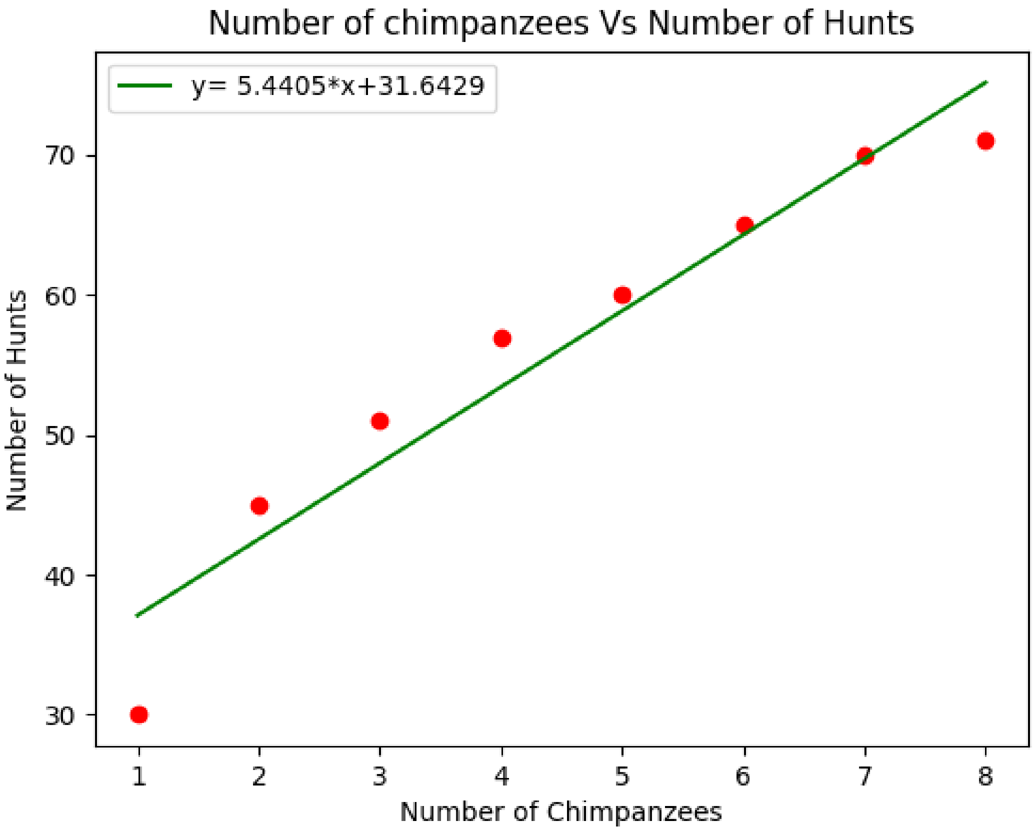
10.Draw Regression Line for the following data. Conclude your analysis.

No. of chimpanzees	1	2	3	4	5	6	7	8
No. of hunting	30	45	51	57	60	65	70	71

```
# Import packages
import numpy as np
import matplotlib.pyplot as plt
x= np.array([1,2,3,4,5,6,7,8])
# Dependent Variable - percent of successful hunts
y = np.array([30,45,51,57,60,65,70,71])
n = np.size(x)
x_mean = np.mean(x)
y_mean = np.mean(y)
b1=n * np.sum(x*y)-np.sum(x)*np.sum(y)
b2=(n * sum(x*x) - (np.sum(x)*np.sum(x)))
b=(b1/b2)
a= y_mean-b*x_mean
print("Line Slope is : %.4f"%b)
print("Line Intercept is: %.4f"%a)
y_pred=b*x+a
plt.scatter(x, y, color = 'red')
plt.plot(x, y_pred, color = 'green',label='y= 5.4405*x+31.6429')
plt.xlabel('Number of Chimpanzees')
plt.ylabel('Number of Hunts')
plt.title("Number of chimpanzees Vs Number of Hunts")
plt.legend()
plt.show()
```

OUTPUT:

```
Line Slope is : 5.4405
Line Intercept is: 31.6429
```



Analysis:

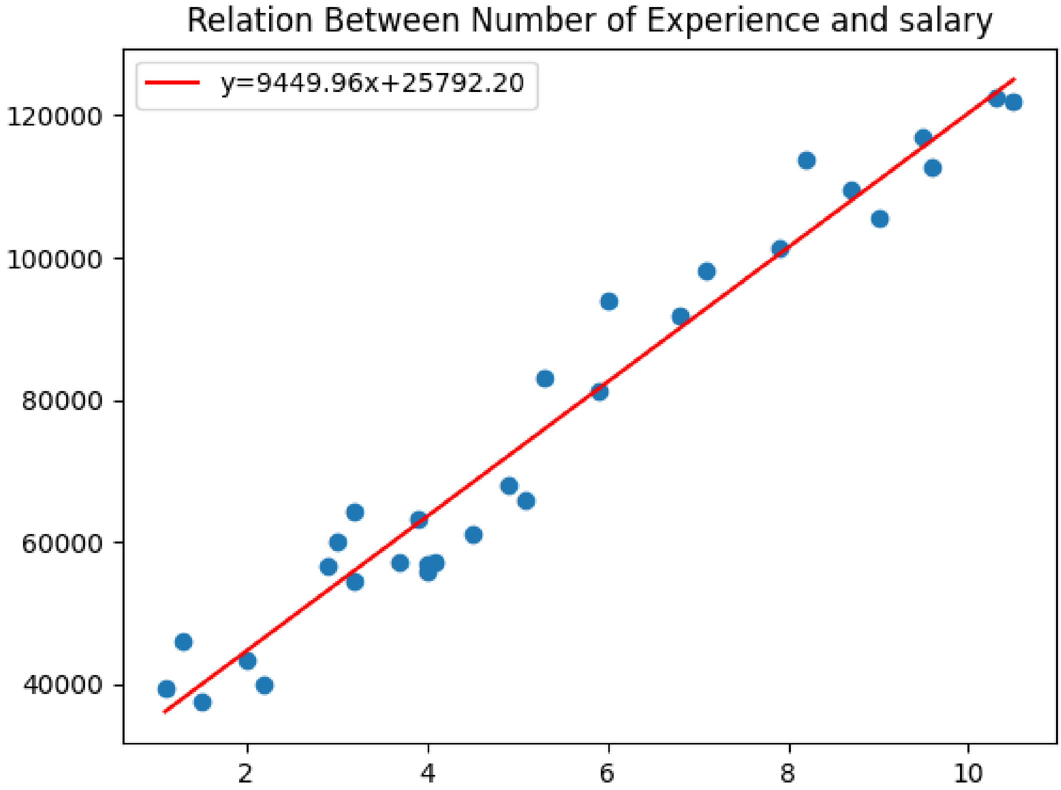
Positive Correlation exist between number of chipanzees and number of hunts.

11. Consider Salary data and draw regression line using polyfit function and visualize graph. Conclude your analysis.

```
import pandas as pd
import matplotlib.pyplot as plt
import openpyxl
import numpy as np
data =pd.read_csv("E:\Data Science with
Python\DataSet\Salary_Data.csv")
print(data)
x=data['YearsExperience']
y=data['Salary']
plt.plot(x, y, 'o')
print("Correlation Coefficient = ",np.corrcoef(x,y))
#obtain m (slope) and b(intercept) of linear regression line
b, a = np.polyfit(x, y, 1)
print("Slope= %.2f"%b,"Intercept = %.2f"%a)
#add linear regression line to scatterplot
plt.plot(x, b*x+a,color='red',label='y=9449.96x+25792.20')
plt.legend()
plt.title("Relation Between Number of Experience and salary")
plt.legend()
plt.show()
```

OUTPUT:

```
Slope= 9449.96 Intercept = 25792.20
Correlation Coefficient =  [[1.         0.97824162]
 [0.97824162 1.        ]]
```

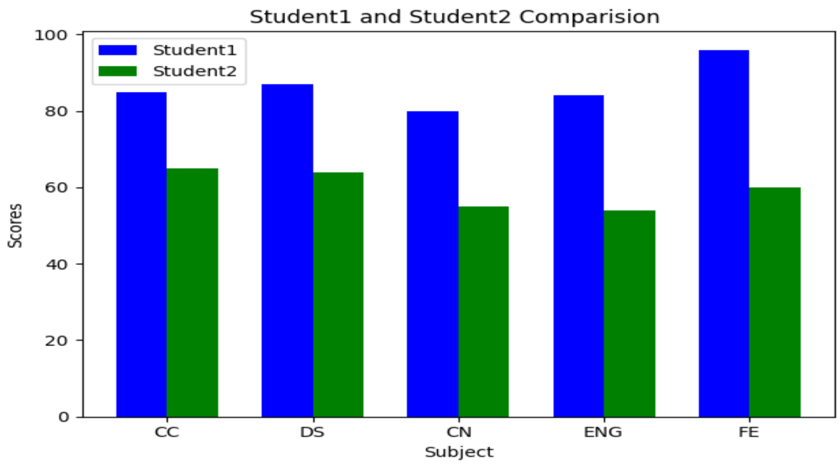

Relation Between Number of Experience and salary
y=9449.96x+25792.20
120000
100000
80000
60000
40000
2
4
6
8
10

12. Display performance of two students in different subjects using bar graph. Also Comment on analysis.

Student	CC	DS	ENG	CN	FE
Student1	85	87	80	84	96
Student2	65	64	55	54	60

```
import matplotlib.pyplot as plt
import numpy as np
Stud1=[85,87,80,84,96]
Stud2=[65,64,55,54,60]
# create plot
bar_width = 0.35
X = np.arange(5)
p1 = plt.bar(X, Stud1, bar_width, color='b',label='Student1')
# The bar of second plot starts where the first bar ends
p2 = plt.bar(X + bar_width, Stud2,
bar_width,color='g',label='Student2')
plt.xlabel('Subject')
plt.ylabel('Scores')
plt.title('Student1 and Student2 Comparision ')
plt.xticks(X + (bar_width/2) , ("CC","DS","CN","ENG","FE"))
plt.legend()
plt.tight_layout()
plt.show()
```

OUTPUT:



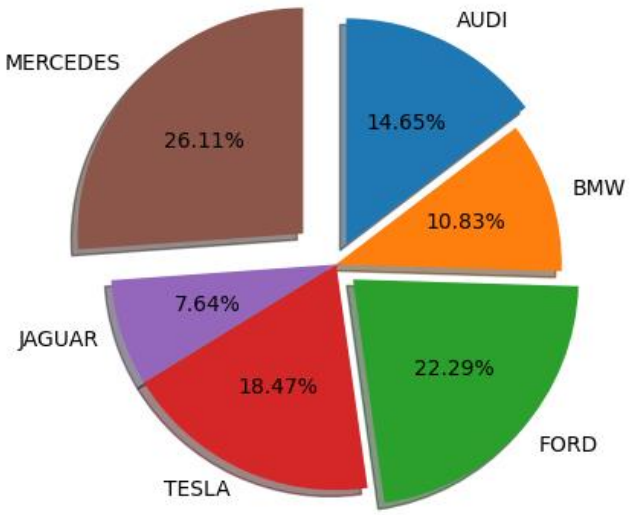
Student1 performance is good compare to student2.

13.Draw Pie chart for following data with explode, Shadow parameter.

cars	AUDI	BMW	FORD	TESLA	JAGUAR	MERCEDES
data	23	17	35	29	12	41

```
# Import libraries
from matplotlib import pyplot as plt
import numpy as np
  # Creating dataset
cars = ['AUDI', 'BMW', 'FORD','TESLA', 'JAGUAR', 'MERCEDES']
data = [23, 17, 35, 29, 12, 41]
 # Creating plot
explode = [0.1, 0, 0.1, 0, 0,0.2]
plt.pie(data, labels = cars,autopct='%1.2f%%',
        explode=explode,shadow = True,startangle =
90,counterclock=False)
 # show plot
plt.show()
```

OUTPUT:

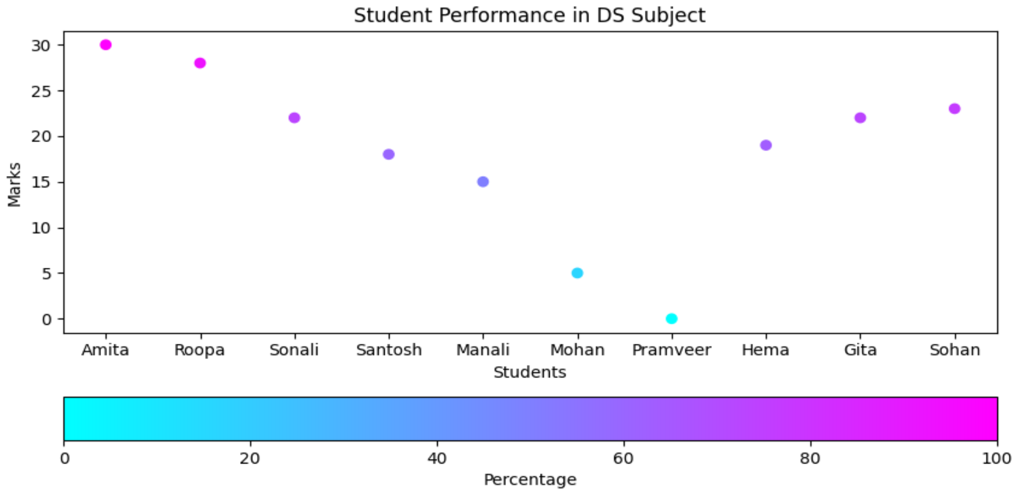


14. Consider the following Marks data of students and draw color bar for percentage. Also analyze data. Given marks is out of 30.40% and above Passing percentage.

marks= [30,28,22,18,15,5,0,19,22,23]

```
import matplotlib.pyplot as plt
rollno=
["Amita","Roopa","Sonali","Santosh","Manali","Mohan","Pramveer",
"Hema","Gita","Sohan"]
marks= [30,28,22,18,15,5,0,19,22,23]
perls=[]
for i in marks:
    per="%.2f"%(i/30*100)
    perls.append(float(per))
plt.figure(figsize=(10, 5))
plt.scatter(x=rollno, y=marks, c=perls, cmap="cool")
plt.colorbar(label="Percentage", orientation="horizontal")
plt.title("Student Performance in DS Subject")
plt.xlabel("Students")
plt.ylabel("Marks")
plt.show()
```

OUTPUT:



Analysis:

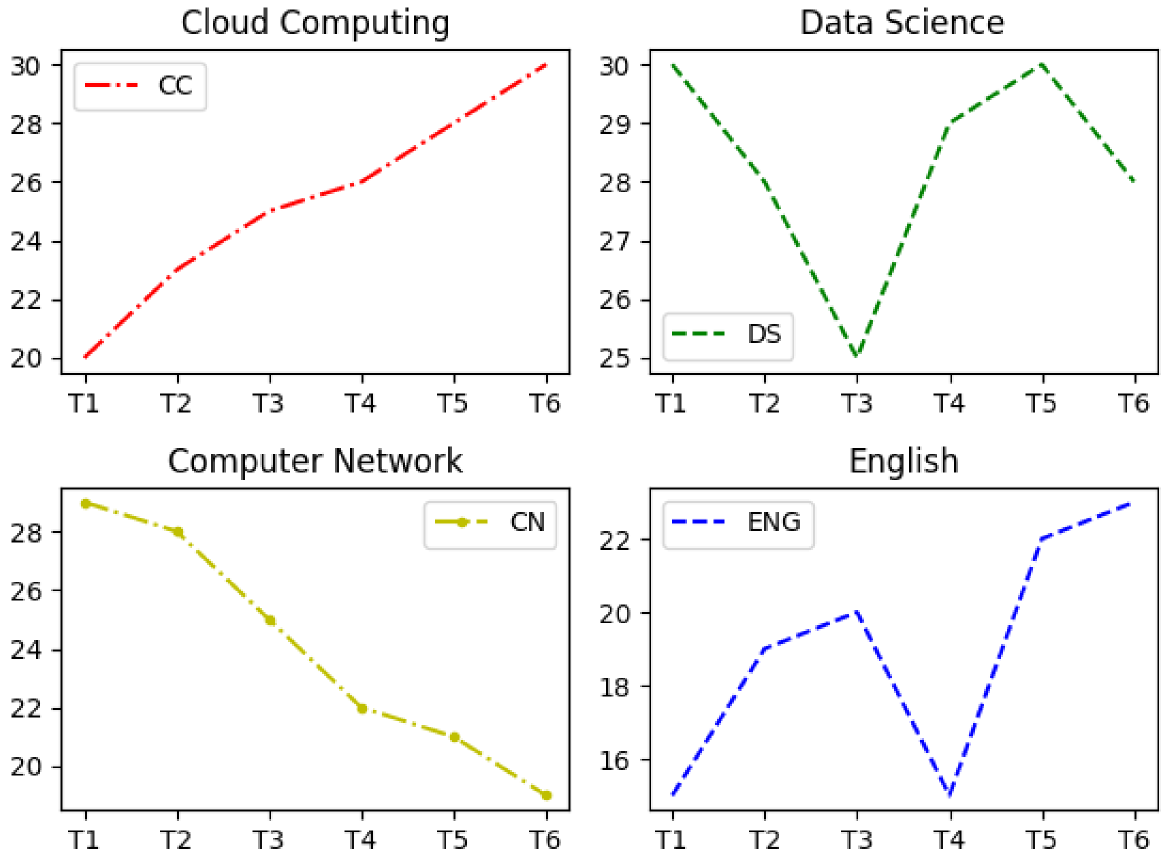
Mohan and pramveer is failed because their percentage is between0 to 20. Remaining 6 students Passed DS exam.

15.**Draw subplot 2 by 2 for the following data of student deepali in for different subjects. Comment your analysis.**

Test	T1	T2	T3	T4	T5	T6
CC	20	23	25	26	28	30
DS	30	28	25	29	30	28
CN	29	28	25	22	21	19
ENG	15	19	20	15	22	23

```
import matplotlib.pyplot as plt
Test=['T1','T2','T3','T4','T5','T6']
CC=[20,23,25,26,28,30]
DS=[30,28,25,29,30,28]
CN=[29,28,25,22,21,19]
ENG=[15,19,20,15,22,23]
plt.figure(figsize=(10,6))
fig, ax = plt.subplots(2,2)
ax[0,0].plot(Test,CC,'r-.',label='CC')
ax[0,0].legend()
ax[0,1].plot(Test,DS,'g--',label='DS')
ax[0,1].legend()
ax[1,0].plot(Test,CN,'y.-.',label='CN')
ax[1,0].legend()
ax[1,1].plot(Test,ENG,'b--',label='ENG')
ax[1,1].legend()
ax[0, 0].set_title("Cloud Computing")
ax[0, 1].set_title("Data Science")
ax[1, 0].set_title("Computer Network")
ax[1, 1].set_title("English")
# set spacing
fig.tight_layout()
plt.show()
```

OUTPUT:



Analysis:

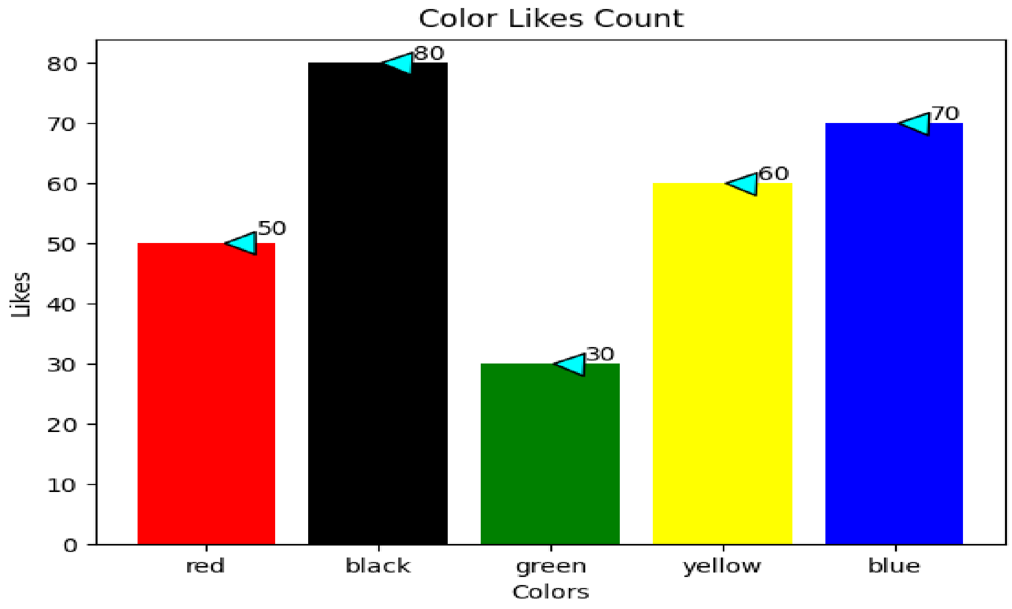
Cloud Computing performance increased whereas Computer network decreased.

16.Draw text Annotation for following data.

Color	red	black	green	yellow	blue
Likes	50	80	30	60	70

```
import matplotlib.pyplot as plt
import numpy as np
color=['red','black','green','yellow','blue']
likes=[50,80,30,60,70]
f, ax = plt.subplots()
ax.bar(color,likes,color=color)
ax.annotate(50, xy=(0.1, 50),  xytext=(0.3, 51.5),
            arrowprops=dict(facecolor='cyan',
shrink=0.05,connectionstyle="angle3"))
ax.annotate(80, xy=(1, 80),  xytext=(1.2, 80.5),
            arrowprops=dict(facecolor='cyan', shrink=0.1))
ax.annotate(30, xy=(2, 30),  xytext=(2.2, 30.5),
            arrowprops=dict(facecolor='cyan', shrink=0.1))
ax.annotate(60, xy=(3, 60),  xytext=(3.2, 60.5),
            arrowprops=dict(facecolor='cyan', shrink=0.1))
ax.annotate(70, xy=(4, 70),  xytext=(4.2, 70.5),
            arrowprops=dict(facecolor='cyan', shrink=0.1))
plt.title("Color Likes Count")
plt.xlabel("Colors")
plt.ylabel("Likes")
plt.show()
```

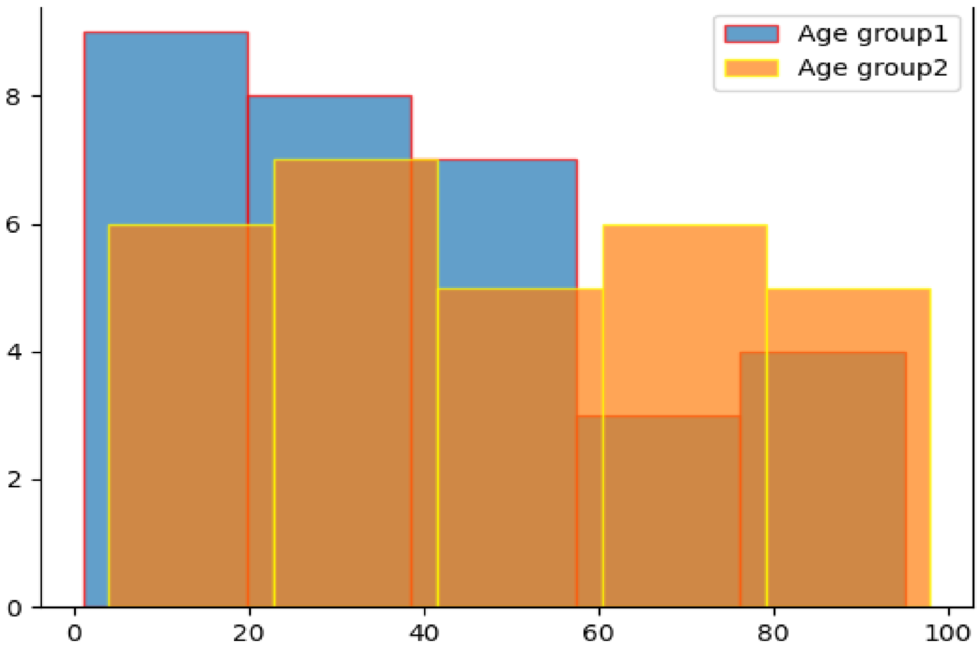
OUTPUT:



17. Display Histogram comparison for following data. Also comment your analysis.

```
# importing libraries
import matplotlib.pyplot as plt
# giving two age groups data
age_g1 = [1, 3, 5, 10, 15, 17, 18, 16, 19,
        21, 23, 28, 30, 31, 33, 38, 32,
        40, 45, 43, 49, 55, 53, 63, 66,
        85, 80, 57, 75, 93, 95]

age_g2 = [6, 4, 15, 17, 19, 21, 28, 23, 31,
        36, 39, 32, 50, 56, 59, 74, 79, 34,
        98, 97, 95, 67, 69, 92, 45, 55, 77,
        76, 85]
# plotting first histogram
plt.hist(age_g1, label='Age group1', bins=5, alpha=.7, edgecolor='red')
# plotting second histogram
plt.hist(age_g2, label="Age group2", bins=5, alpha=.7,
edgecolor='yellow')
plt.legend()
# Showing the plot using plt.show()
plt.show()
```

OUTPUT:



In Age group1 0-20 years' people are more where as in age group2 60-80 peoples are more.

18. Display 2D ndarray basic operation accessing, inserting, deleting, updating elements operations also show additional functions of numpy array.

```
import numpy as np
#Create 2 D Array
arr=np.array([[1,2,3],[4,5,6],[7,8,9]])
arr1=np.array([[10,11,12],[13,14,15],[16,17,18]])
#print array
print("Array = ",arr)
#Display Dimesion of array
print("Dimesion of array = ",arr.ndim)
#Display Shape of Array
print("Dimesion of array = ",arr.shape)
# Access element 5
print("Accessed Element= ",arr[1,1])
#Insert new value at position 1 rowwise
arr=np.insert(arr,1,[9,4,7],axis=0)
print("After Insertion = ",arr)
#Modification 8 with 88
arr[3,1]=88
print("After Modification = ",arr)
#Deleting elemnts
print(arr)
arr = np.delete(arr, 1, axis=0)
print("After Deletion = ",arr)
#Addtional numpy array functions
print("Transpose of matrix= ",np.transpose(arr))
print("After Concatnation Columnwise of arr and arr1= ",
np.concatenate((arr,arr1),axis=1))
print("After Vetical stack operation on arr and arr1=
",np.vstack((arr,arr1)))
print("After Horizontal stack operation on arr and arr1=
",np.hstack((arr,arr1)))
```

OUTPUT:

```
Array =  [[1 2 3]
 [4 5 6]
 [7 8 9]]
Dimesion of array =  2
Dimesion of array =  (3, 3)
Accessed Element=  5
After Insertion =  [[1 2 3]
```

```
 [9 4 7]
 [4 5 6]
 [7 8 9]]
After Modification =  [[ 1  2  3]
 [ 9  4  7]
 [ 4  5  6]
 [ 7 88  9]]
[[ 1  2  3]
 [ 9  4  7]
 [ 4  5  6]
 [ 7 88  9]]
After Deletion =  [[ 1  2  3]
 [ 4  5  6]
 [ 7 88  9]]
Transpose of matrix=  [[ 1  4  7]
 [ 2  5 88]
 [ 3  6  9]]
After Concatnation Columnwise of arr and arr1=  [[ 1  2  3 10 11 12]
 [ 4  5  6 13 14 15]
 [ 7 88  9 16 17 18]]
After Vetical stack operation on arr and arr1=  [[ 1  2  3]
 [ 4  5  6]
 [ 7 88  9]
 [10 11 12]
 [13 14 15]
 [16 17 18]]
After Horizontal stack operation on arr and arr1=  [[ 1  2  3 10 11 12]
 [ 4  5  6 13 14 15]
 [ 7 88  9 16 17 18]]
```

19. Display 3D ndarray basic operation accessing, inserting, deleting, updating elements.

```
import numpy as np
arr=np.array([[[1,2,3],[4,5,6]],
             [[7,8,9],[10,11,12]],
            [[13,14,15],[16,17,18]]])
arr1=np.array([[[19,20,21],[22,23,24]],
             [[25,26,27],[28,29,30]],
            [[31,32,33],[34,35,36]]])
#Print Dimnsion and shape
print("Dimension= ",arr.ndim,"Shape = ",arr.shape)
#Access 5
print("Accessing Element 5 =",arr[0,1,1])
#Access 10,11,12
print("Accessing Element [10,11,12] =",arr[1,1,:])
#Insert new row [[19,20,21],[22,23,24]]
arr=np.insert(arr,3,[[19,20,21],[22,23,24]],axis=0)
print("After Insertion",arr)
#Modify 8 to 18
arr[1,0,1]=18
print("After Modifying 8 to 18 = ",arr)
#Delete row 2
arr=np.delete(arr,2,axis=0)
print("After deleting 2 row = ",arr)
#Addtional Functions
print("Transpose of matrix= ",np.transpose(arr))
print("After Concatnation Columnwise of arr and arr1= ",
np.concatenate((arr,arr1),axis=1))
print("After Vetical stack operation on arr and arr1=
",np.vstack((arr,arr1)))
print("After Horizontal stack operation on arr and arr1=
",np.hstack((arr,arr1)))
```

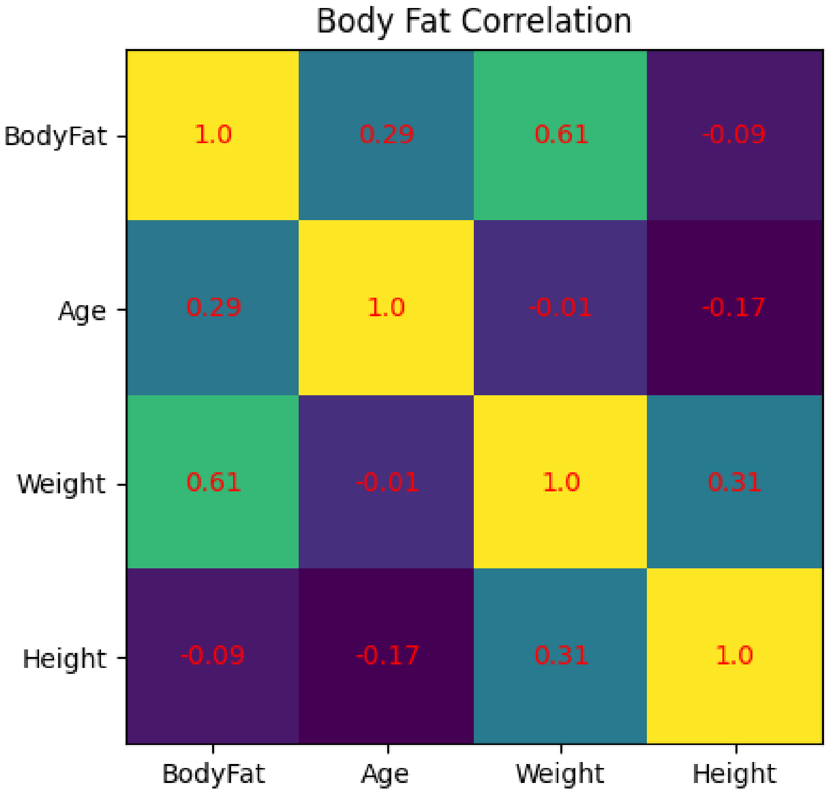
OUTPUT:

```
Dimension=  3 Shape =  (3, 2, 3)
Accessing Element 5 = 5
Accessing Element [10,11,12] = [10 11 12]
After Insertion [[[ 1  2  3]
  [ 4  5  6]]

 [[ 7  8  9]
  [10 11 12]]
 [[13 14 15]
  [16 17 18]]

 [[19 20 21]
  [22 23 24]]]
After Modifying 8 to 18 =  [[[ 1  2  3]
  [ 4  5  6]]

 [[ 7 18  9]
  [10 11 12]]

 [[13 14 15]
  [16 17 18]]

 [[19 20 21]
  [22 23 24]]]
After deleting 2 row =  [[[ 1  2  3]
  [ 4  5  6]]

 [[ 7 18  9]
  [10 11 12]]

 [[19 20 21]
  [22 23 24]]]
Transpose of matrix=  [[[ 1  7 19]
  [ 4 10 22]]

 [[ 2 18 20]
  [ 5 11 23]]

 [[ 3  9 21]
  [ 6 12 24]]]
After Concatnation Columnwise of arr and arr1=  [[[ 1  2  3]
  [ 4  5  6]
  [19 20 21]
  [22 23 24]]

 [[ 7 18  9]
  [10 11 12]
  [25 26 27]
  [28 29 30]]

 [[19 20 21]
  [22 23 24]
  [31 32 33]
  [34 35 36]]]
After Vetical stack operation on arr and arr1=  [[[ 1  2  3]
  [ 4  5  6]]

 [[ 7 18  9]
  [10 11 12]]

 [[19 20 21]
  [22 23 24]]

 [[19 20 21]
  [22 23 24]]

 [[25 26 27]
  [28 29 30]]

 [[31 32 33]
  [34 35 36]]]
After Horizontal stack operation on arr and arr1=  [[[ 1  2  3]
  [ 4  5  6]
  [19 20 21]
  [22 23 24]]

 [[ 7 18  9]
  [10 11 12]
  [25 26 27]
  [28 29 30]]

 [[19 20 21]
  [22 23 24]
  [31 32 33]
  [34 35 36]]]
```

20.For bodyfat dataset calculate Correlation and Visualize Using Hitmap.

```
import matplotlib.pyplot as plt
import pandas as pd
import openpyxl
import numpy as np
import seaborn as sns
data=pd.read_csv("E:\\Data Science with
Python\\DataSet\\bodyfat.csv")
print(data)
corr=data.corr()
print(corr)
fig ,ax=plt.subplots()
plt.title("Body Fat Correlation")
im= ax.imshow(corr.values)
# set labels
ax.set_xticks(np.arange(len(corr.columns)))
ax.set_yticks(np.arange(len(corr.columns)))
ax.set_xticklabels(corr.columns)
ax.set_yticklabels(corr.columns)
#Adding values
for i in range(len(corr.columns)):
    for j in range(len(corr.columns)):
        text = ax.text(j, i, np.around(corr.iloc[i, j], decimals=2),
                       ha="center", va="center", color="red")
plt.show()
```

OUTPUT:


Body Fat Correlation
BodyFat
Age
Weight
Height
1.0
0.29
0.61
-0.09
0.29
1.0
-0.01
-0.17
0.61
-0.01
1.0
0.31
-0.09
-0.17
0.31
1.0
BodyFat
Age
Weight
Height

21.Create dataframe in python for IPL Data and apply some basic operation on dataframe.

Team	MI	CSK	Devils	MI	CSK	RCB	CSK	CSK	KKR	KKR	KKR
Year	2014	2015	2014	2015	2014	2015	2016	2017	2016	2014	2015
Points	876	789	863	673	741	812	756	788	694	701	804

```
import pandas as pd
df=pd.DataFrame({"Team":["MI","CSK","Devils","MI","CSK","RCB","
CSK",
                    "CSK","KKR","KKR","KKR"],
                "Rank":[1,2,2,3,3,4,1,1,2,4,1],
                "Year":[2014,2015,2014,2015,2014,2015,2016,2017,
                    2016,2014,2015],
                "Points":[876,789,863,673,741,812,756,788,694,
                     701,804]},
               index=["R1","R2","R3","R4","R5","R6","R7","R8",
                   "R9","R10","R11"])
print("DataFrame = ")
print(df)
#Access Rows 2,4,6,8 using index and using labels
print("After Accessing Rows 2,4,6,8 Using Labels = ")
print(df.loc[["R2","R4","R6","R8"]])
print("After Accessing Rows 2,4,6,8 Using Index = ")
print(df.iloc[1:8:2])
#Access top 3 Rows and also bottom 3 rowa
print("Top 3 Rows = ")
print(df.head(3))
print("Bottom 3 Rows= ")
print(df.tail(3))
#Access columns team and points
print("After Accessing 2 Columns  Team and Points= ")
print(df[['Team','Points']])
#Access Row 3 and column 1,3,4 using index
print("After Accessing row 3 and Columns  1,3,4 using index= ")
print(df.iloc[2,[0,2,3]])
#Access Row 3 and column 1,3,4 using labels
print("After Accessing row 3 and Columns  1,3,4 using labels= ")
print(df.loc["R3",['Team','Year','Points']])
#Update last record with values 'RCB',3,2016,800
df.iloc[10]=['RCB',3,2016,800]
print("After Updating Last Row = ")
print(df)
#Insert new record in dataframe
df.loc[len(df.index)] = ['MI',2,2017,800]
print("After Inserting Last Row= ")
print(df)
#Delete row from dataframe
df=df.drop([11])
print("After Deleting Last Row = ")
print(df)
```

```
OUTPUT: DataFrame =
       Team  Rank  Year  Points
R1       MI     1  2014     876
R2      CSK     2  2015     789
R3   Devils     2  2014     863
R4       MI     3  2015     673
R5      CSK     3  2014     741
R6      RCB     4  2015     812
R7      CSK     1  2016     756
R8      CSK     1  2017     788
R9      KKR     2  2016     694
R10     KKR     4  2014     701
R11     KKR     1  2015     804
After Accessing Rows 2,4,6,8 Using Labels =
   Team  Rank  Year  Points
R2  CSK     2  2015     789
R4   MI     3  2015     673
R6  RCB     4  2015     812
R8  CSK     1  2017     788
After Accessing Rows 2,4,6,8 Using Index =
   Team  Rank  Year  Points
R2  CSK     2  2015     789
R4   MI     3  2015     673
R6  RCB     4  2015     812
R8  CSK     1  2017     788
Top 3 Rows =
      Team  Rank  Year  Points
R1      MI     1  2014     876
R2     CSK     2  2015     789
R3  Devils     2  2014     863
```

```
Bottom 3 Rows=
    Team  Rank  Year  Points
R9   KKR     2  2016     694
R10  KKR     4  2014     701
R11  KKR     1  2015     804
After Accessing 2 Columns  Team and Points=
       Team  Points
R1       MI     876
R2      CSK     789
R3   Devils     863
R4       MI     673
R5      CSK     741
R6      RCB     812
R7      CSK     756
R8      CSK     788
R9      KKR     694
R10     KKR     701
R11     KKR     804
After Accessing row 3 and Columns  1,3,4 using index=
Team      Devils
Year        2014
Points       863
Name: R3, dtype: object
After Accessing row 3 and Columns  1,3,4 using labels=
Team      Devils
Year        2014
Points       863
Name: R3, dtype: object
After Updating Last Row =
       Team  Rank  Year  Points
R1       MI     1  2014     876
R2      CSK     2  2015     789
R3   Devils     2  2014     863
R4       MI     3  2015     673
R5      CSK     3  2014     741
R6      RCB     4  2015     812
R7      CSK     1  2016     756
R8      CSK     1  2017     788
R9      KKR     2  2016     694
R10     KKR     4  2014     701
```

```
R11     RCB     3  2016     800
After Inserting Last Roe=
       Team  Rank  Year  Points
R1       MI     1  2014     876
R2      CSK     2  2015     789
R3   Devils     2  2014     863
R4       MI     3  2015     673
R5      CSK     3  2014     741
R6      RCB     4  2015     812
R7      CSK     1  2016     756
R8      CSK     1  2017     788
R9      KKR     2  2016     694
R10     KKR     4  2014     701
R11     RCB     3  2016     800
11       MI     2  2017     800
After Deleting Last Row =
       Team  Rank  Year  Points
R1       MI     1  2014     876
R2      CSK     2  2015     789
R3   Devils     2  2014     863
R4       MI     3  2015     673
R5      CSK     3  2014     741
R6      RCB     4  2015     812
R7      CSK     1  2016     756
R8      CSK     1  2017     788
R9      KKR     2  2016     694
R10     KKR     4  2014     701
R11     RCB     3  2016     800
```

22.