Assignment -2 Name:- Ashwin Kumar Roll:-CE21BTECH11008

Data science & Analysis

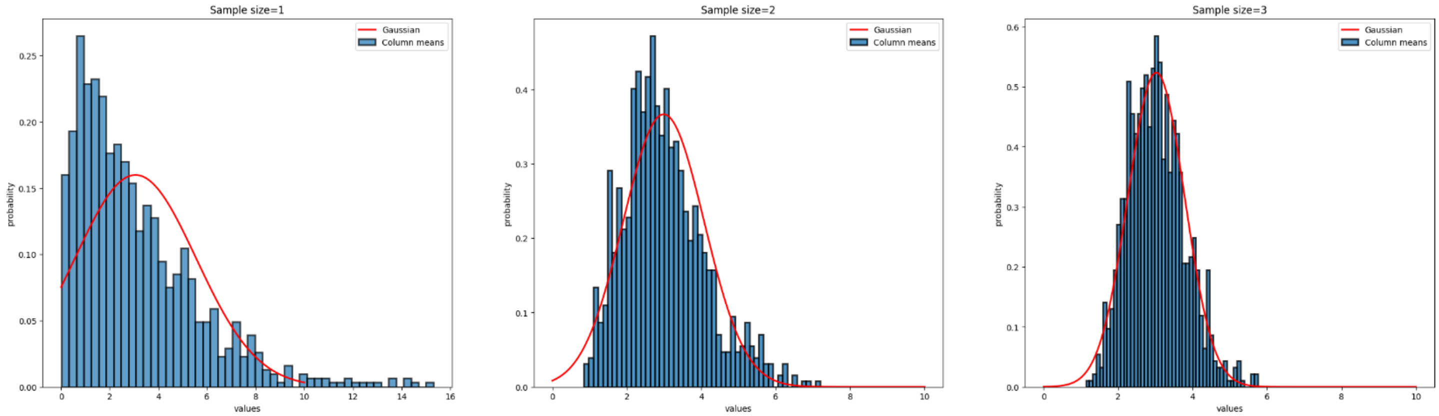
Q1.

Importing the library

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

Code

```
# taking the sample size of 100 as sample size of greater than 30 is assumed as large sample size and follow
# extarcting one sample of size 100 and degree of freedom = 3

data1= np.random.chisquare(df=3,size=(1,1000))
# print(data1)

# extarcting 5 sample of size 100 and degree of freedom = 3
data2=np.random.chisquare(df=3,size=(5,1000))
# print(data2)

# extarcting 10 sample of size 100 and degree of freedom = 3
data3=np.random.chisquare(df=3,size=(10,1000))
# print(data3)

X= np.linspace(0,10,1000)
# Now mean of the sample along the column will follow central limit theorem
# Find mean of the each column and saving into array

plt.figure(figsize=(30, 8))  #setting width to 30 inches and height to 8 inches

plt.subplot(1,3,1)
mean_data1=np.mean(data1,axis=0)

m1=np.mean(mean_data1)  #mean of mean data 1
sd1=np.std(mean_data1)  #standard deviation of mean_data1
y1 = scipy.stats.norm.pdf(X,m1,sd1) #normal distribution with same mean and standard deviation
plt.plot(X, y1,color='red', linewidth=2,label='Gaussian')
plt.hist(mean_data1, bins=50, density=True, alpha=0.7,edgecolor='k', label='Column means',lw=2)
plt.title('Sample size=1')
plt.xlabel('values')
plt.ylabel('probability')
plt.legend()

plt.subplot(1,3,2)

mean_data2=np.mean(data2,axis=0)
m2=np.mean(mean_data2)  #mean of mean data 2
sd2=np.std(mean_data2)  #standard deviation of mean_data2
y2 = scipy.stats.norm.pdf(X,m2,sd2)#normal distribution with same mean and standard deviation
plt.plot(X, y2,color='red', linewidth=2,label='Gaussian')
plt.hist(mean_data2, bins=50, density=True, alpha=0.8,edgecolor='k', label='Column means',lw=2)
plt.title('Sample size=2')
plt.xlabel('values')
plt.ylabel('probability')
plt.legend()

plt.subplot(1,3,3)
mean_data3=np.mean(data3,axis=0)
m3=np.mean(mean_data3)  #mean of mean data 3
sd3=np.std(mean_data3)  #standard deviation of mean_data3
y3 = scipy.stats.norm.pdf(X,m3,sd3)#normal distribution with same mean and standard deviation
plt.plot(X, y3,color='red', linewidth=2,label='Gaussian')

plt.hist(mean_data3, bins=50, density=True, alpha=0.8,edgecolor='k', label='Column means',lw=2)
plt.title('Sample size=3')
plt.xlabel('values')
plt.ylabel('probability')

plt.legend()
```



Q2.

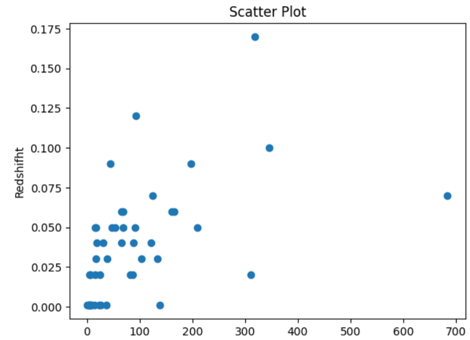
Import Library

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

Code

```
data3 = pd.read_csv("/content/test.dat",delimiter=' ')
#delimeter=' ' defines data are sepearated by space bar instead of commea as default
data3.columns  #show the name of columns
luminousity= data3['#Lx']  #first column denoting luminousity
redshift=data3['z']  #second column denoting redshift

# scatter plot to visulize and decide that pearson coeff will give good approximation or not
#as only linear correlation is detected by pearson coefficient

plt.scatter(luminousity,redshift)

#by seeing the scatter plot data is looking approximately correlated with positve coeff (a straight line barely
fits)
# lets check usign different coeffecient

corr_coeff,p_value = stats.pearsonr(luminousity,redshift)
rho,p_value = stats.spearmanr(luminousity,redshift)
tau,p_value = stats.kendalltau(luminousity,redshift)

print(f" pearson coefficient = {corr_coeff} spearman coeffecient = {rho} kendal coefficient={tau}")

#pearson coeffiicient is coming more than 0.5 that shows there is significance coorelation  and spearman
coefficient better shows
# with coefficient of 0.65 .while kendall is showing approx same as pearson coeff i.e.greater than 0.5
```

```
pearson coefficient = 0.5144497852670242

spearman coeffecient = 0.6596325957535455

kendal coefficient=0.5029584682704178
```



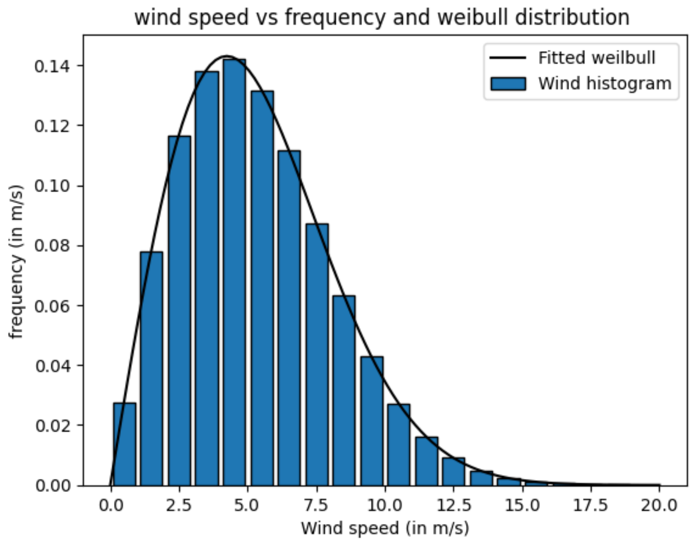
Q3.

Import Library

```
import numpy as np

import scipy.stats as stats

import matplotlib.pyplot as plt
```

Code

```
wind_speed_interval = [(0, 1),(1, 2),(2, 3),(3, 4),(4, 5),(5, 6),(6, 7),(7, 8),(8, 9),(9, 10),(10, 11),(11, 12),(12, 13),(13, 14),(14, 15),(15, 16),(16, 17),(17, 18),(18, 19),(19, 20)]

# mean will be good estimator of the each interval in wind_speed_interval
wind_speed= np.mean(wind_speed_interval,axis=1)
wind_speed

frequency =
np.array([2.75,7.80,11.64,13.79,14.20,13.15,11.14,8.72,6.34,4.30,2.73,1.62,0.91,0.48,0.24,0.11,0.05,0.02,0.01,0.00])

# frequency =np.array(frequency)
frequency=frequency*0.01   #converitng into probability
plt.bar(wind_speed,frequency,lw=1,edgecolor='k',label='Wind histogram')

# fitting the data with weilbull distribution
params, covariance = curve_fit(weibull_min.pdf, wind_speed, frequency, p0=[2, 0, 1])
k,loc,lambda_value= params
print(params)
#k= 1.99719020e+00  loc= 5.16239511e-04  lambda= 6.00786445e+00

#we can see the predicted value is nearly equal to what is provided
k=2 #shape parameter
loc=0  #location parameter
lambda_value=6  #scale parameter
r=np.linspace(0,20,100)
dist=stats.weibull_min.pdf(r,k,loc,lambda_value)

plt.plot(r,dist,color='black' , label='Fitted weilbull')

plt.xlabel('Wind speed (in m/s)')
plt.ylabel('frequency (in m/s)')
plt.title('wind speed vs frequency and weibull distribution ')
plt.legend()
plt.show()
```



Q4.

Importing Library

```
import numpy as np
import scipy.stats as stats
import matplotlib.pyplot as plt
from scipy.stats import t,pearsonr
```

Code

```
np.random.seed(1)
dist1= np.random.normal(0,1,1000)  #first 1000 draws from the gaussian distributio
dist2= np.random.normal(0,1,1000)  #first 1000 draws from the gaussian distributio

corr_coeff,p_value = stats.pearsonr(dist1,dist2)

# degree of freedom = k= N-2  //gaussian distribution have two paramete i.e. mean and standard deviation
k=1000-2

t_value=corr_coeff*np.sqrt(k/(1-corr_coeff**2))

cdf= t.cdf(t_value,k)

print(f"correlation coeffecient = {corr_coeff}, p value ={p_value},cumulative distr={cdf}\n")

print(f"P_value from pearson correlation= {p_value}")

p_value_from_t = 2*(1-cdf)  #as this is two tailed test so it wiil multiplied by 2
print(f"P_value from t statistic= {p_value_from_t}\n")

#comparing the p values from pearson correlataion and t distribution
```

```
correlation coeffecient = 0.02185695102804924, p value =0.48994551602654357,cumulative distr=0.7550272419867288

P_value from pearson correlation= 0.48994551602654357
P_value from t statistic= 0.48994551602654246
```