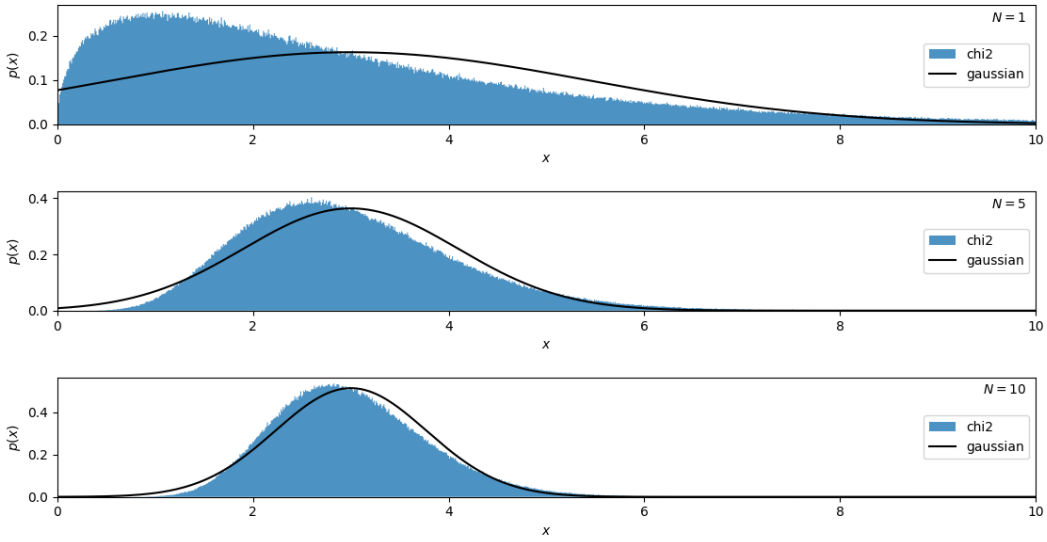
# Q1

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



N = [1, 5, 10]
dof = 3

np.random.seed(10)
x = np.zeros((max(N), int(1e6)))
for i in range(max(N)):
    x[i] = np.random.chisquare(dof, int(1E6))


fig = plt.figure(figsize=(5, 5))
fig.subplots_adjust(hspace=0.05)

for i in range(len(N)):
    ax = fig.add_subplot(3, 1, i+1)


    x_i = x[:N[i], :].mean(0)

    ax.hist(x_i, bins=np.linspace(0,10,1000), histtype='stepfilled',
alpha=0.8, density=True, label = 'chi2')


    mu = x_i.mean()
    sigma = np.sqrt(2.0*dof/N[i] )
    x_pdf = np.linspace(0, 10, 1000)
    dist = norm(mu, sigma)
    ax.plot(x_pdf, dist.pdf(x_pdf), '-k', label='gaussian')
    ax.set_xlim(0.0, 10)

    plt.legend(loc = 'center right')

    ax.text(0.99, 0.95, r"$N = %i$" % N[i], ha='right', va='top',
transform=ax.transAxes)

    ax.set_xlabel(r'$x$')
    ax.set_ylabel('$p(x)$')

plt.tight_layout()
plt.show()
```


N = 1
chi2
gaussian
p(x)
x
0.0
0.1
0.2
0
2
4
6
8
10


N = 5
chi2
gaussian
p(x)
x
0.0
0.2
0.4
0
2
4
6
8
10


N = 10
chi2
gaussian
p(x)
x
0.0
0.2
0.4
0
2
4
6
8
10

# Q2

```
import numpy as np
from matplotlib import pyplot as plt
import pandas as pd
from scipy.stats import kendalltau
from scipy.stats import spearmanr
from scipy.stats import pearsonr


a = pd.read_excel("C:\\Users\Heera Baiju\\Downloads\\Data Science\\Assignment
2\\LuminosityHeera.xlsx")
Luminosity = np.log(a["Lx"])
Redshift = np.log(a["z"])
# Resizing the figure
plt.figure(figsize=[7, 5])

# Plotting the scatter plot
plt.scatter(Luminosity, Redshift, c='g', alpha=0.6)

plt.title('Luminosity vs redshift', fontsize=15)
plt.xlabel('Luminosity', fontsize=13)
plt.ylabel('Redshift', fontsize=13)
plt.show()

Pearson_Correlation_Coefficient, p_value_p = pearsonr(Luminosity, Redshift)
Kendalltau_Correlation_Coefficient, p_value_k = kendalltau(Luminosity,
Redshift)
Spearman_Correlation_Coefficient, p_value_s = spearmanr(Luminosity, Redshift)
print('Pearsons correlation: {}' .format(Pearson_Correlation_Coefficient))
print('p value of Pearson Correlation Coefficient: {}' .format(p_value_p))
print('Kendalltau correlation: {}'
.format(Kendalltau_Correlation_Coefficient))
print('p value of Kendalltau Correlation Coefficient: {}' .format(p_value_k))
print('Spearman correlation: {}' .format(Spearman_Correlation_Coefficient))
print('p value of Spearman Correlation Coefficient: {}' .format(p_value_s))
```

**OUTPUT**

Pearsons correlation: 0.599835977512611

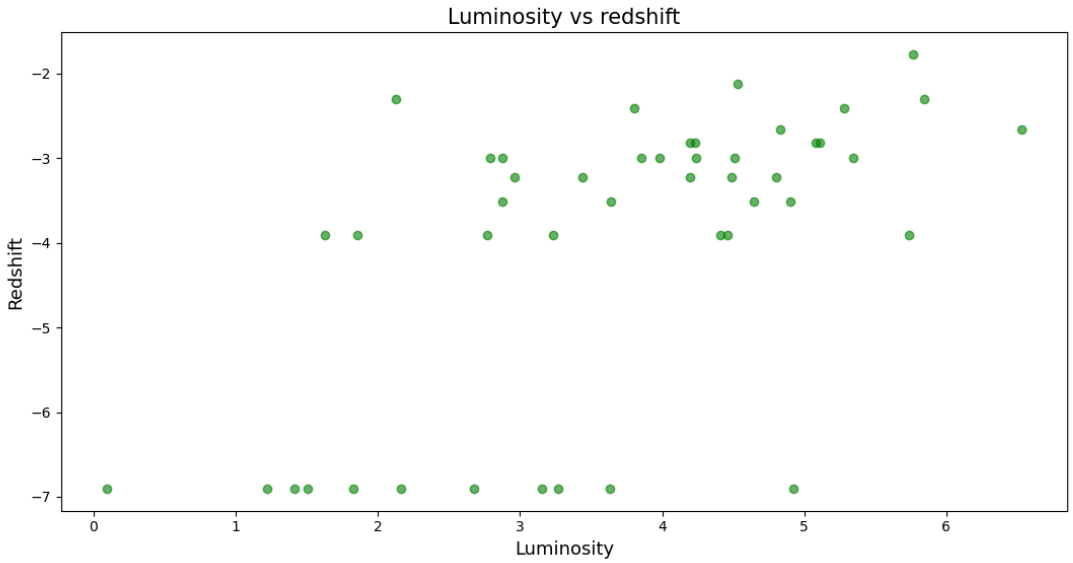
p value of Pearson Correlation Coefficient: 1.0547205916682317e-05

Kendalltau correlation: 0.44256065116630433

p value of Kendalltau Correlation Coefficient: 3.698087258609231e-05

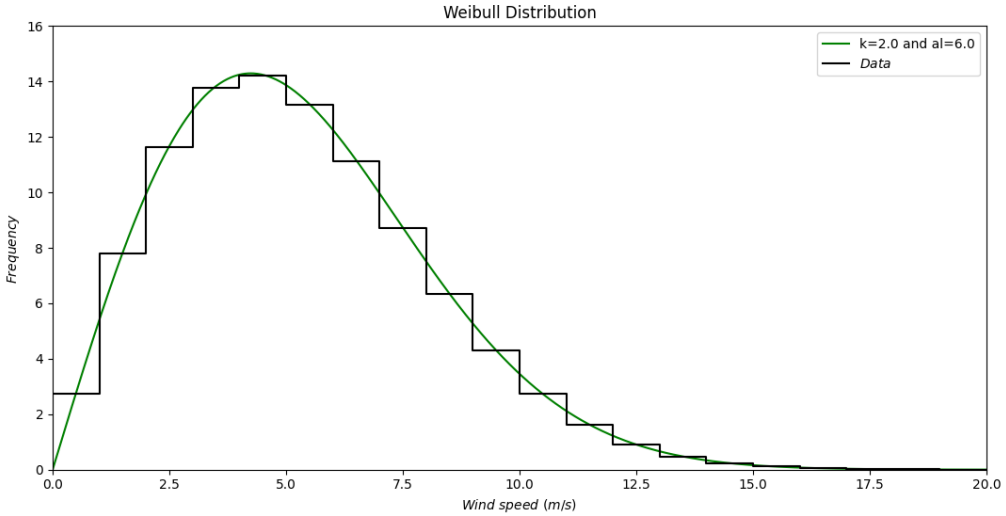
Spearman correlation: 0.5816518730305096

p value of Spearman Correlation Coefficient: 2.2411477134735858e-05



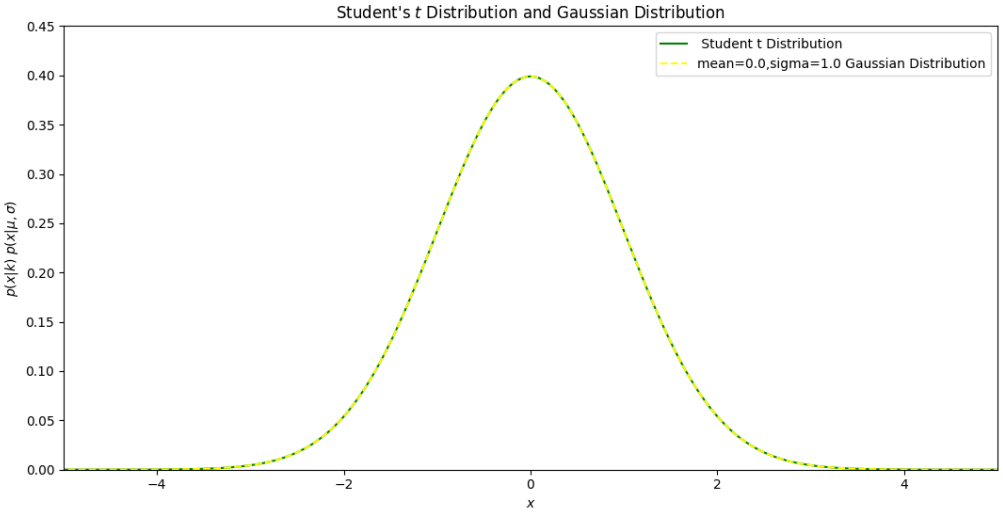
# Q3

```
import numpy as np
from matplotlib import pyplot as plt
import pandas as pd
from scipy import stats

a = pd.read_excel("C:\\Users\\Heera Baiju\\Downloads\\Data Science\\Assignment
2\\Swiss Wind Power data.xlsx")

freq = a["frequency"]
windsp = a["class"]
x = np.linspace(0, 20, 1000)

k = 2
al = 6
v = np.mean(x)

def weib(x, k, al):
    return (k / al) * (x / al)**(k - 1) * np.exp(-(x / al)**k)


fig, ax = plt.subplots(figsize=(5, 3.75))
plt.plot(x, 100*weib(x, k, al), ls = '-', c='g', label = r'k=%.1f and al=%.1f'
% (k, al))
plt.step(windsp, freq , ls = '-', c='black', label = r'$Data$')
plt.xlabel('$Wind\ speed\ (m/s)$')
plt.ylabel('$Frequency$')
plt.title('Weibull Distribution')
plt.xlim(0, 20)
plt.ylim(0.0, 16)
plt.legend()
plt.show()
```


Weibull Distribution
k=2.0 and al=6.0
Data
Frequency
Wind speed (m/s)
0
2
4
6
8
10
12
14
16
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0

# Q4

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


mu = 0
sigma = 1
gau_x = np.linspace(-10, 10, 1000)
k = 1e10
stu_x = np.linspace(-10, 10, 1000)


fig, ax = plt.subplots(figsize=(5, 3.75))

studist = student_t(k)
plt.plot(stu_x, studist.pdf(stu_x), ls = '-', c = 'green', label = r' Student
t Distribution')

dist_g = norm(mu, sigma)
plt.plot(gau_x, dist_g.pdf(gau_x), ls = '--', c = 'yellow', label =
r'mean=%.1f,sigma=%.1f Gaussian Distribution' % (mu, sigma))

plt.xlim(-5, 5)
plt.ylim(0.0, 0.45)
plt.xlabel('$x$')
plt.ylabel(r'$p(x|k)\ p(x|\mu,\sigma)$')
plt.title("Student's $t$ Distribution and Gaussian Distribution")
plt.legend()
plt.show()
corr_s, p_value_s = pearsonr(stu_x, studist.pdf(stu_x))
corr_g, p_value_g = pearsonr(gau_x, dist_g.pdf(gau_x))


print('p value of Student\'s t Distribution: {}' .format(p_value_s))
print('p value of Gaussian Distribution: {}' .format(p_value_g))
```




Output

p value of Student's t Distribution: 1.0000000000000662
p value of Gaussian Distribution: 1.0000000000000662