Data Science Analysis Assignment 2

Pushkal Mishra

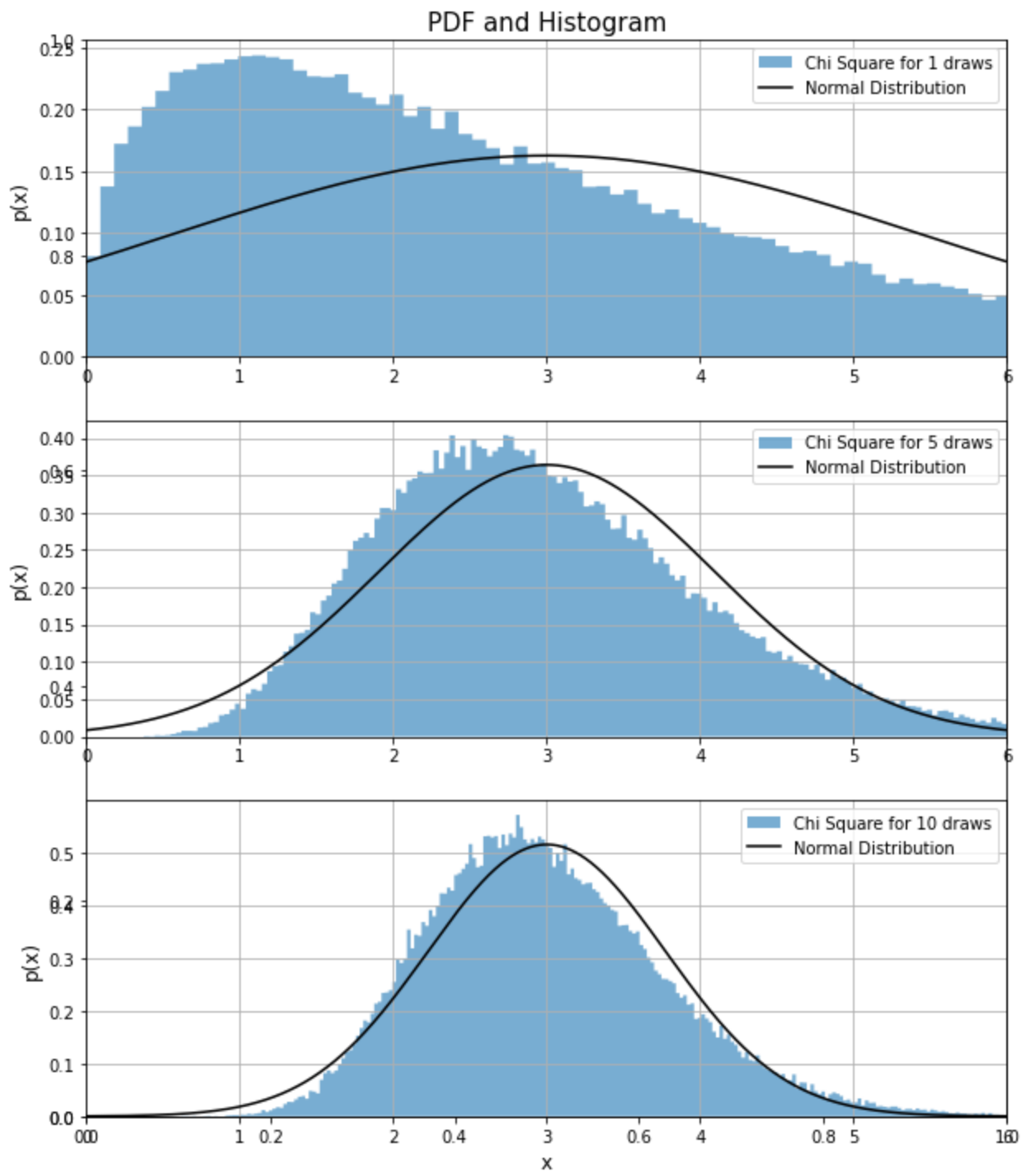
EE20BTECH11042

Importing Libraries

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats
import csv
```

Question 1

In [2]:
```
dof = 3
gaussian_mu = dof
gaussian_x = np.linspace(0, 6, 6000)

N = [1, 5, 10]
np.random.seed(11042)
draws = np.random.chisquare(dof, size = (max(N), int(1e5)))

fig = plt.figure(figsize = (10, 12))
plt.title("PDF and Histogram", size = 15)

for i in range(len(N)):
    ax = fig.add_subplot(len(N), 1, i + 1)
    x = draws[ : N[i], : ].mean(0)
    ax.hist(x, bins = 300, histtype = 'stepfilled',
            label = f'Chi Square for {N[i]} draws',
            density = True, alpha = 0.6)

    gaussian_sigma = np.sqrt(2 * dof / N[i])
    gaussian_dist = stats.norm(gaussian_mu, gaussian_sigma)
    ax.plot(gaussian_x, gaussian_dist.pdf(gaussian_x),
            color='black', label='Normal Distribution')

    ax.set_xlim(0.0, 6.0)
    ax.set_ylabel('p(x)', size = 12)
    ax.legend(loc = 1)
    ax.grid()
plt.xlabel('x', size = 12)
plt.show()
```

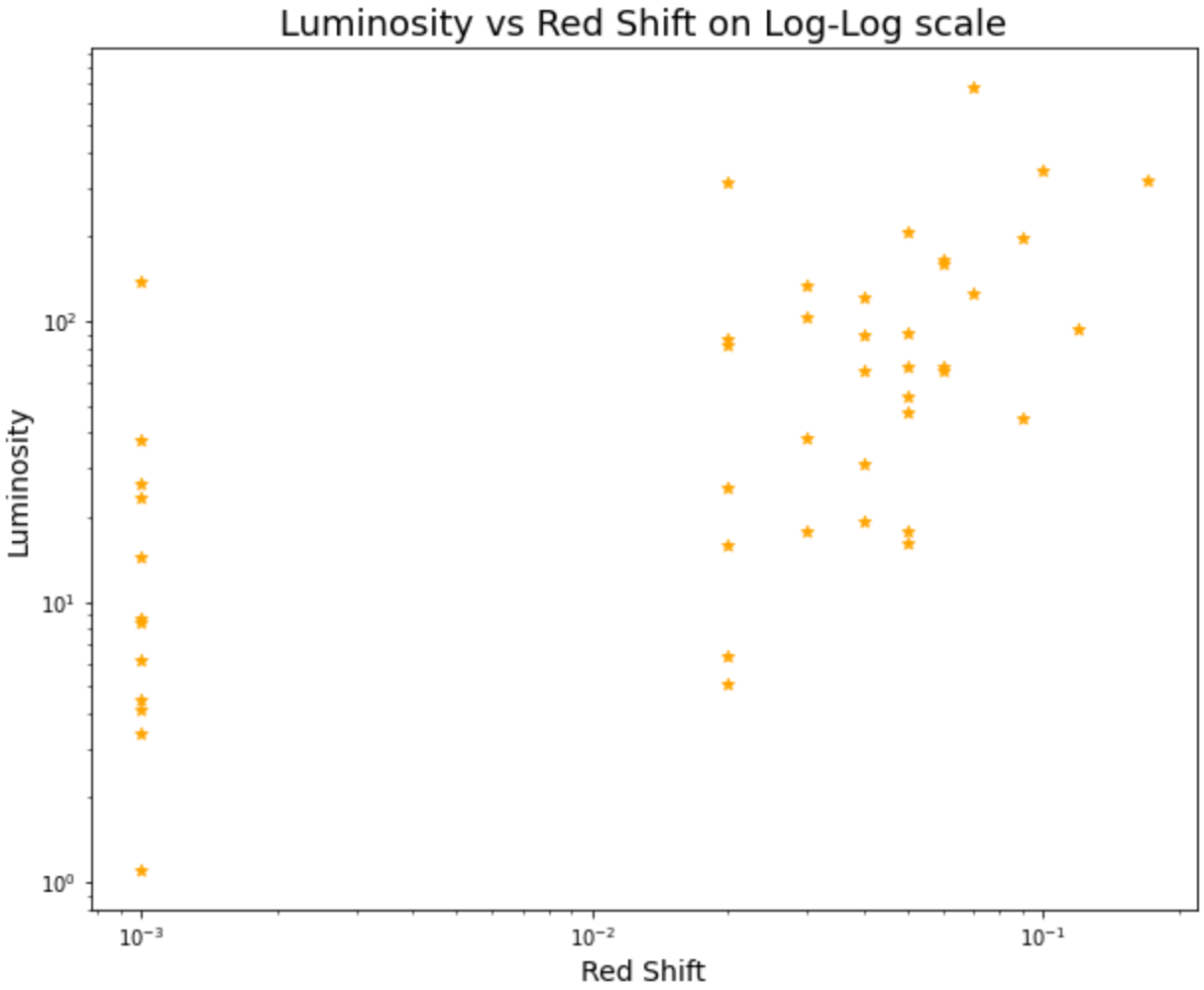


Question 2

In [3]:

```
luminosity = []
red_shift = []

with open('luminosity.csv', mode ='r') as file:
    csvFile = csv.reader(file)
    for lines in csvFile:
        l, r = lines[0].split(" ")
        luminosity.append(l)
        red_shift.append(r)

luminosity.pop(0)
red_shift.pop(0)

luminosity = [float(l) for l in luminosity]
red_shift = [float(r) for r in red_shift]
```

In [4]:
```
figure = plt.figure(figsize = (10, 8))
plt.xscale('log')
plt.yscale('log')
plt.scatter(red_shift, luminosity, marker = '*', color = 'orange')
plt.title("Luminosity vs Red Shift on Log-Log scale", size = 18)
plt.xlabel("Red Shift", size = 14)
plt.ylabel("Luminosity", size = 14)
plt.show()
```



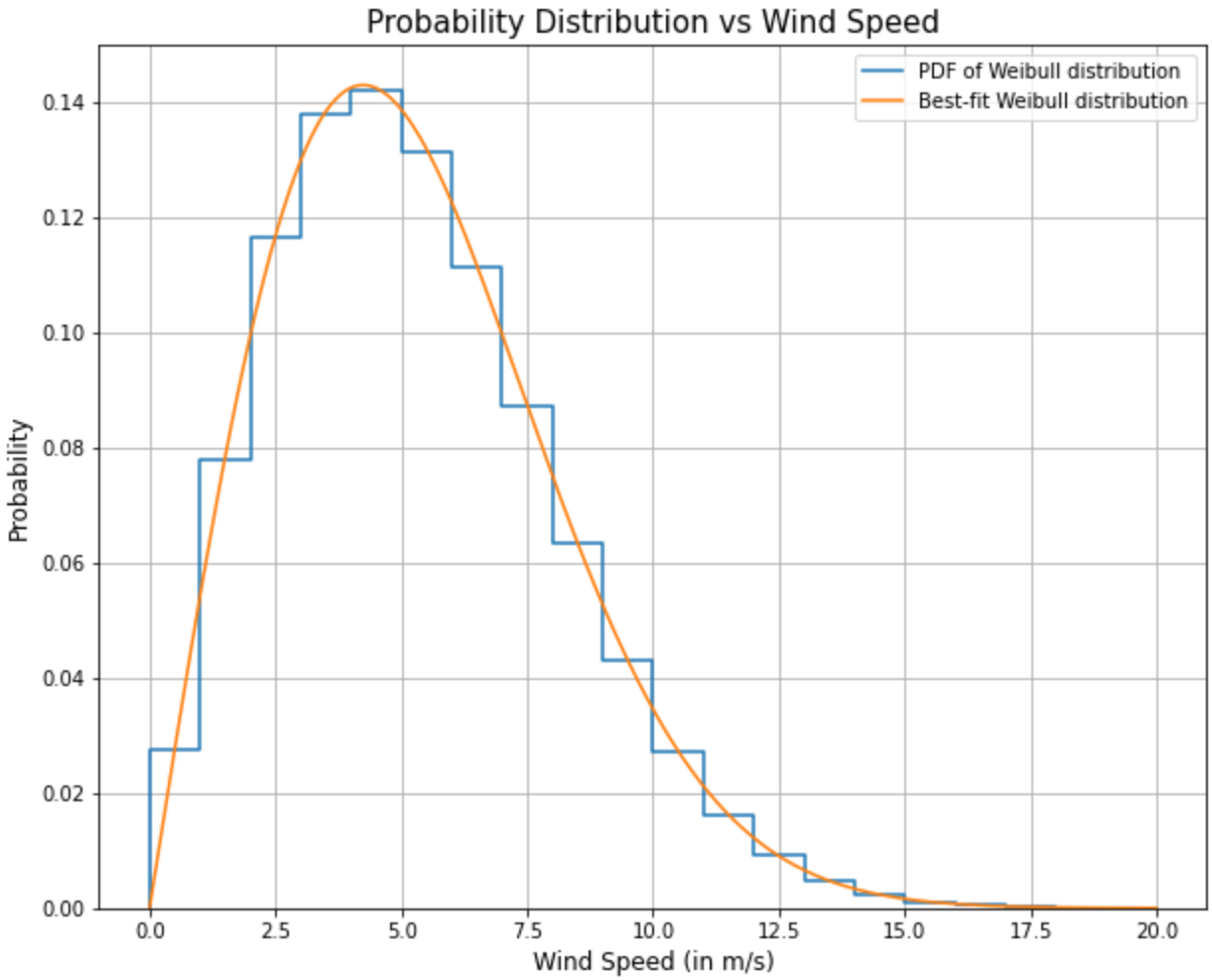
In [5]:
```
rho, p_value1 = stats.spearmanr(red_shift, luminosity)
corr_coeff, p_value2 = stats.pearsonr(red_shift, luminosity)
tau, p_value3 = stats.kendalltau(red_shift, luminosity)

print(f"Spearman Coefficient: {rho}")
print(f"Pearson Coefficient: {corr_coeff}")
print(f"Kendall-tau Coefficient: {tau}\n")
print(f"p-value for Spearman: {p_value1}")
print(f"p-value for Pearson: {p_value2}")
print(f"p-value for Kendall-tau: {p_value3}")
```

```
Spearman Coefficient: 0.6596325957535454
Pearson Coefficient: 0.5144497852670242
Kendall-tau Coefficient: 0.5029584682704178

p-value for Spearman: 6.166489759081011e-07
p-value for Pearson: 0.0002546471657612425
p-value for Kendall-tau: 2.969686227473415e-06
```

They do have some positive correlation which is also evident from the correlation coefficients and p-values

Question 3

In [6]:
```
speed = np.arange(0, 21, 1)
frequency = np.array([0.0, 2.75, 7.8, 11.64, 13.79, 14.2, 13.15, 11.14, 8.72, 6.34, 4.3,
                      2.73, 1.62, 0.91, 0.48, 0.24, 0.11, 0.05, 0.02, 0.01, 0.0]) / 100

x = np.linspace(0, 20, 2000)
k, avg, lam = 2.0, 0.0, 6.0
weibull_dist = stats.weibull_min(k, avg, lam)
weibull_pdf = weibull_dist.pdf(x)

figure = plt.figure(figsize = (10, 8))
plt.step(speed, frequency, label = 'PDF of Weibull distribution')
plt.plot(x, weibull_pdf, label = 'Best-fit Weibull distribution')
plt.title("Probability Distribution vs Wind Speed", size = 15)
plt.xlabel("Wind Speed (in m/s)", size = 12)
plt.ylabel("Probability", size = 12)
plt.ylim(0.0, 0.15)
plt.legend(loc = 1)
plt.grid()
plt.show()
```



Question 4

In [7]:
```
N = 1000
gaussian_dist = stats.norm(loc = 0, scale = 1)
arr1 = gaussian_dist.rvs(size = N)
arr2 = gaussian_dist.rvs(size = N)
```

```
corr_coeff, p_value = stats.pearsonr(arr1, arr2)
t_for_student_t = corr_coeff * np.sqrt((N - 2) / (1 - (corr_coeff ** 2)))

if t_for_student_t >= 0:
    p_value_student_t = 2 * (1 - stats.t.cdf(t_for_student_t, N - 2))
else:
    p_value_student_t = 2 * stats.t.cdf(t_for_student_t, N - 2)

print(f"Pearson Correlation Coefficient: {corr_coeff}")
print(f"t value of Student-t distribution: {t_for_student_t}")
print(f"p-value from Pearson: {p_value}")
print(f"p-value from Student-t: {p_value_student_t}")

if round(p_value, 3) == round(p_value_student_t, 3):
    print(f"\nThe p-value agrees with that calculated using the Student-t distribution."
```

```
Pearson Correlation Coefficient: 0.027230789448261853
t value of Student-t distribution: 0.8605707498648328
p-value from Pearson: 0.38968119091539943
p-value from Student-t: 0.38968119091541964

The p-value agrees with that calculated using the Student-t distribution.
```