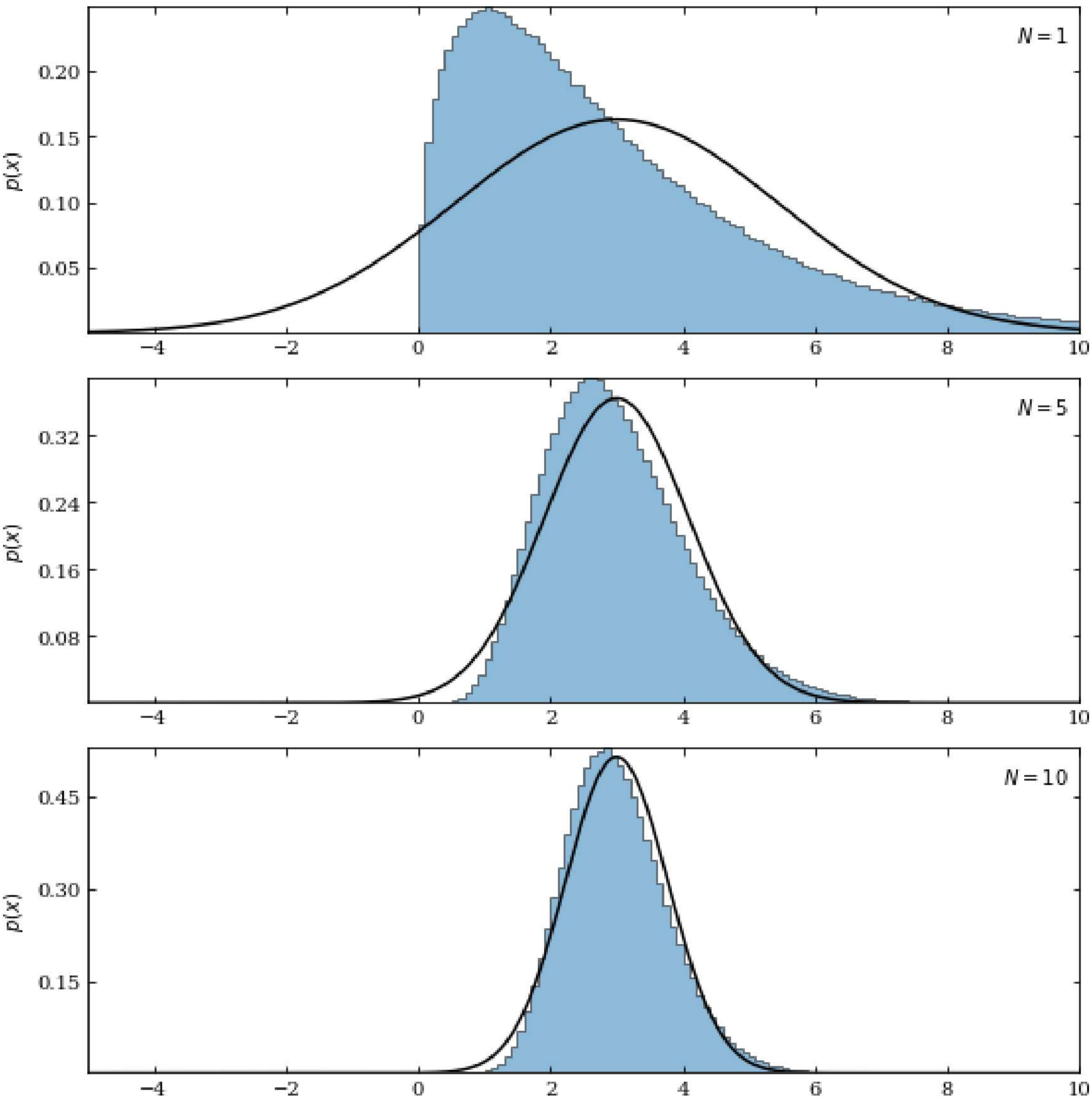
Data Science Analysis Assignment - 2

Q1. In the class, we demonstrated the Central Limit Theorem for a sample drawn from a uniform distribution. Reproduce a similar plot for a sample drawn the from chi-square distribution with degrees of freedom equal to 3, for samples drawn once, 5 times, and 10 times. Either plot all of these on one multipanel figure similar to AstroML figure 3.20.

In [67]:

```
#importing required libraries
import numpy as np
from matplotlib import pyplot as plt
from scipy.stats import norm

from astroML.plotting import setup_text_plots

setup_text_plots(fontsize = 10, usetex = False)

# Generate the uniform samples
N = [1, 5, 10]

x = np.random.chisquare(3, (10, 1000000))

# Plot the results
fig = plt.figure(figsize = (8, 8))
fig.subplots_adjust(hspace = 0.1)

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

    # take the mean of the first N[i] samples
    x_i = x[:N[i], :].mean(0)

    # histogram the data
    ax.hist(x_i, bins = np.linspace(0, 10, 101),
            histtype='stepfilled', alpha=0.5, density=True)

    # plotting the expected chi-square pdf
    mu = 3 #degrees of freedom
    sigma = np.sqrt(6)/np.sqrt(N[i])
    dist = norm(mu, sigma)
    x_pdf = np.linspace(-5, 10, 1000)

    #plotting the pdf and setting the limits for the axes
    ax.plot(x_pdf, dist.pdf(x_pdf), '-k')
    ax.set_xlim(-5, 10)
    ax.set_ylim(0.001, None)

    ax.yaxis.set_major_locator(plt.MaxNLocator(5))

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

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

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

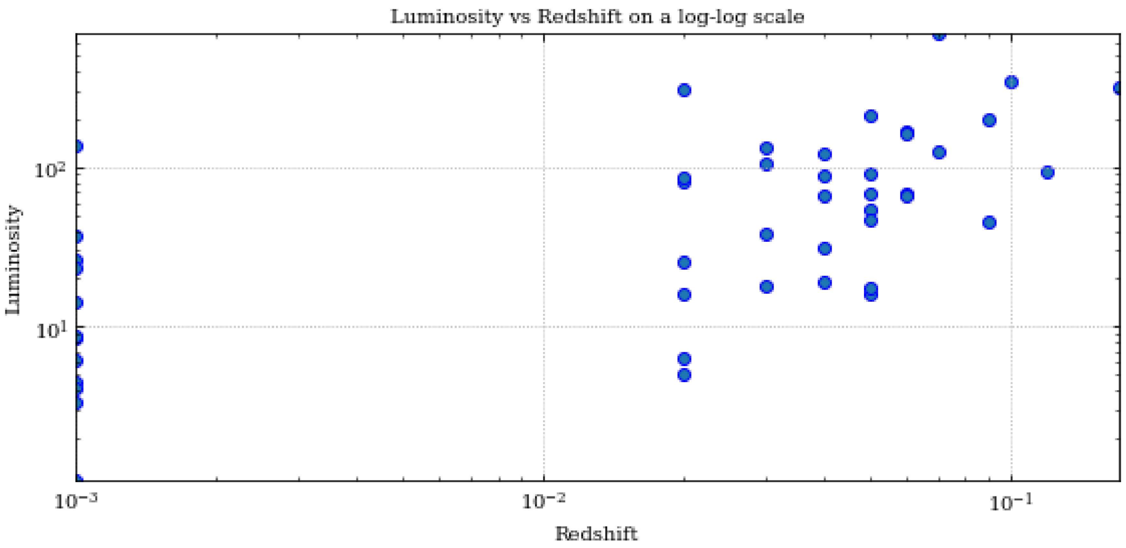


Q2. The luminosity and redshift of galaxy clusters from XMM-BCS survey (details available at arXiv:1512.01244) can be downloaded http://www.iith.ac.in/~shantanud/test.dat. Plot the luminosity as a function of redshift on a log-log scale. By eye, do you think the datasets are correlated? Calculate the Spearman, Pearson and Kendall-tau correlation coefficients and the p-value for the null hypothesis.

In [142…

```
#importing required library
import pandas as pd

#reading the file
data = pd.read_csv("D:\CLASSES\SEM 4\Data Science Analysis\A2\data.csv",
                   skipinitialspace=True, sep="\s+", usecols = ['#Lx','z'])

#rewriting the columns into arrays for easy plotting
y = data['#Lx']
x = data['z']

fig, ax = plt.subplots(1, 1, figsize =(8, 4), tight_layout = True)

#transforming the scales into log scales
plt.xscale("log")
plt.yscale("log")
```

```
#scattering the dataset
ax.scatter(x,y)

#adding title and lables
plt.title('Luminosity vs Redshift on a log-log scale')
plt.xlabel('Redshift')
plt.ylabel('Luminosity')

plt.grid()
plt.show()
```



In [137…

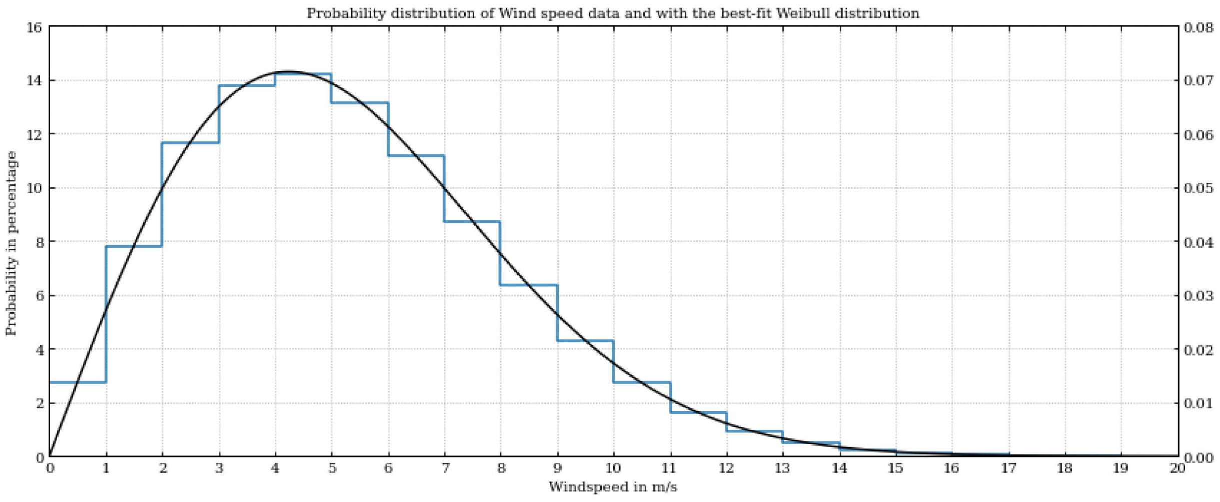
```
#importing required libraries
from scipy.stats import pearsonr, spearmanr, kendalltau

#calculating coeffs and p values
cor, p_v1 = pearsonr(x, y)
rho, p_v2 = spearmanr(x, y)
tau, p_v3 = kendalltau(x, y)

#printing the data
print("Correlation coefficients\n Pearson : %s\n Spearman: %s\n Kendall : %s"
       %(cor, rho, tau))
print("\nP-values \n Pearson : %s \n Spearman: %s \n Kendall : %s"
       %(p_v1, p_v2, p_v3))
```

```
Correlation coefficients
 Pearson : 0.5144497852670243
 Spearman: 0.6596325957535454
 Kendall : 0.5029584682704178

P-values
 Pearson : 0.00025464716576124137
 Spearman: 6.166489759081011e-07
 Kendall : 2.969686227473415e-06
```

Q3. Wind speed data from the Swiss Wind Power data website can be found at http://wind-data.ch/tools/weibull.php. Using the data provided on the website, plot the probability distribution and overlay the best-fit Weibull distribution (with the parameters shown on the website). (20 points) (Hint : A on the website is same as λ, which was used in class to parameterize the Weibull distribution.)

In [139…

```
#reading data from csv file
data2 = pd.read_csv("D:\CLASSES\SEM 4\Data Science Analysis\A2\data2.csv",
                    skipinitialspace=True, usecols = [0,1])

#making the plot more presentable
fig, ax = plt.subplots(1, 1, figsize =(12, 5), tight_layout = True)
plt.grid()

#plotting the step histogram and setting limits
ax.step(np.arange(0, 20), data2['Probability'], where = 'post')
ax.set_xlim(0, 20)
ax.set_ylim(0, 16)
ax.set_xticks(np.arange(0, 21))

#adding labels
ax.set_ylabel('Probability in percentage')
ax.set_xlabel('Windspeed in m/s')

#importing library required to plot Weibull dist.
from scipy.stats import dweibull

#making an axis on RHS for the dist.
ax1 = ax.twinx()

#making the Weibull dist.
#k=2, loc=0, lambda=6
dist = dweibull(2, 0, 6)

#generating x values for the pdf
x_pdf = np.linspace(0, 20, 1000)

#plotting
ax1.plot(x_pdf, dist.pdf(x_pdf),'-k')
ax1.set_ylim(0,0.08)

plt.title('Probability distribution of Wind speed data and with the best-fit Weibull
plt.show()
```



Q4. Generate two arrays of size 1000 drawn from a Gaussian distribution of mean of zero and standard deviation of one. Calculate Pearson correlation coefficient and its p−value using scipy module. Also check if the p− value agrees with that calculated using the Student-t distibution.

In [121…

```
#importing required libraries
```

```python
from scipy.stats import pearsonr

#creating gaussian dist with mean 0 and std 1
#mu, sigma = 0, 1
dist = norm(0, 1)

#creating two arrays of size 1000 drawn from the dist.
data1 = dist.rvs(1000)
data2 = dist.rvs(1000)

#data1 = np.random.default_rng().normal(mu, sigma, 1000)
#data2 = np.random.default_rng().normal(mu, sigma, 1000)

corr, p_v = pearsonr(data1, data2)

print('Pearson Correlation Coefficient is %s \np value is %s' %(corr, p_v))
```

```
Pearson Correlation Coefficient is 0.021066530903662433
p value is 0.5057800818556575
```

In [122…

```python
from scipy import stats

#creating t distribution
dist2 = stats.t(998)

r = corr
t = r*np.sqrt(998/(1-r**2))
p = 2*(1-dist.cdf(t))

print(f"P value obtained from students-t distribution : {p}")
```

```
P value obtained from students-t distribution : 0.505626265053148
```