Data Science Analysis - Assignment 2

Varenya Upadhyaya

January 23, 2023

1. The following code plots Fig. (1) containing a description of the Central Limit Theorem using χ^2 distributions with 3 degrees of freedom.

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

r = np.random.chisquare(3, (1, 10000))
r5 = np.random.chisquare(3, (5, 10000))
r10 = np.random.chisquare(3, (10, 10000))

r = r[:1, :].mean(0)
r5 = r5[:5, :].mean(0)
r10 = r10[:10, :].mean(0)

#figure specifics
fig, ax = plt.subplots(3,1)
fig.suptitle('Central Limit Theorem')
fig.set_figheight(7)
fig.set_figwidth(6)

#plotting the chi2 distributions
ax[0].hist(r,bins=100,density=True,color='grey',label='$\chi^2$' )
ax[1].hist(r5,bins=100,density=True,color='grey',label='5 $\chi^2$ averaged')
ax[2].hist(r10,bins=100,density=True,color='grey',label='10 $\chi^2$ averaged')

#setting parameters for the normal distributions
mu = 3
sigma = np.sqrt(6/1)
sigma5 = np.sqrt(6/5)
sigma10 = np.sqrt(6/10)

dist = stats.norm(mu, sigma)
dist5 = stats.norm(mu, sigma5)
dist10 = stats.norm(mu, sigma10)

x_pdf = np.linspace(0, 10, 1000)

#plotting the normal distributions
ax[0].plot(x_pdf, dist.pdf(x_pdf), color='black', label='normal pdf')
ax[1].plot(x_pdf, dist5.pdf(x_pdf), color='black', label='normal pdf')
ax[2].plot(x_pdf, dist10.pdf(x_pdf), color='black', label='normal pdf')

#minor
ax[0].legend()
ax[1].legend()
ax[2].legend()
plt.savefig('1.png')
```

2. By eye, it looks like there might be some weak positive correlation between the luminosity and redshifts. The following code plots Fig. 2 and calculates the correlation coefficients and their corresponding p-values.

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

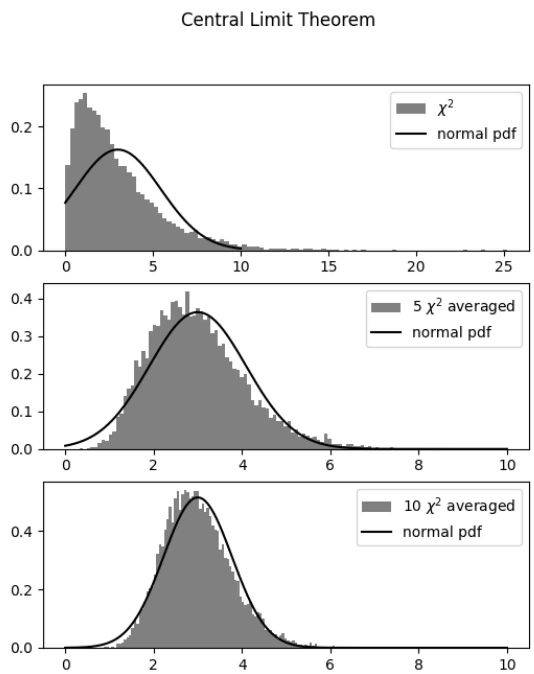

Figure 1: CLT depicted using χ^2 distributions

```

L,z = np.loadtxt('2.txt', unpack=True)

corr_coeff,p_pearson = stats.pearsonr(z,L)
rho,p_spearman = stats.spearmanr(z,L)
tau,p_kendall = stats.kendalltau(z,L)

print(corr_coeff,p_pearson)
print(rho,p_spearman)
print(tau, p_kendall)

plt.plot(np.log10(z), np.log10(L),'.',color='grey')
plt.title('Luminosity v. Redshift')
plt.xlabel('$log(z)$')
plt.ylabel('$log(L_x)$')
plt.grid()
plt.savefig('2.png')
```

Code output:

```
pearson coefficient, p-value = 0.5144497852670242 0.00025464716576124250
spearmans rho, p-value       = 0.6596325957535454 6.16648975908101e-07
kenalls tau, p-value        = 0.5029584682704178 2.9696862274734036e-06
```

3. The following code plots Fig. 2 which contains the wind speed data along with the Weibull distribution for $\lambda = 6, k = 2$

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

#generating/extracting the data
wind_data = np.loadtxt('3.txt',unpack=True)
x=np.arange(0,20,1)
```

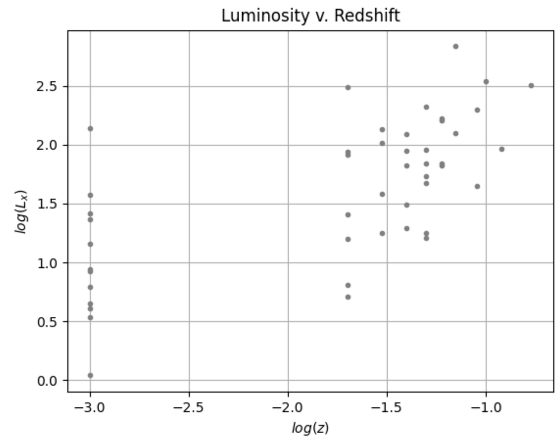

Figure 2: L_x vs. z

```
dist=stats.dweibull(2,0,6)
weibull= 200*dist.pdf(x)

#plotting
plt.bar(x,wind_data,color='lightgrey',label= 'Wind Speed Frequencies')
plt.plot(x,weibull,color='black',label='Weibull Distribution, $k=2, \lambda=6$')

#minor
plt.legend()
plt.title('Wind Speeds')
plt.xlabel('wind speeds (m/s)')
plt.ylabel('freq (%)')
plt.savefig('3.png')
```

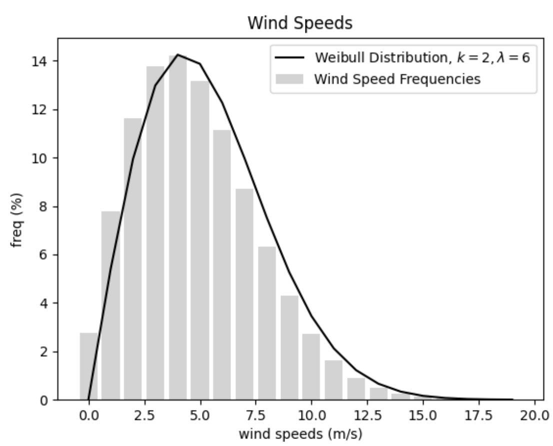

Figure 3: Wind Speeds

4. The following code computes the pearson coefficient and the two p-values (scipy and Student's t) for the two standard normal arrays.

```
import numpy as np
from scipy import stats

a=np.arange
x = stats.norm(0,1).rvs(1000)
y = stats.norm(0,1).rvs(1000)

```

```
r,p_pearson = stats.pearsonr(x,y)
t = r*np.sqrt(998/(1-r**2))

if t>0:
    p_t = 2*(1-stats.t.cdf(t,998))
else:
    p_t = 2*(stats.t.cdf(t,998))

print('pearson coefficient:',r,'\np-value:',p_pearson)
print('p-value from Students t dist:',p_t)
```

Code output:

```
pearson coefficient: -0.03357832856558897
p-value: 0.2887757751121469
p-value from Students t dist: 0.2887757751121657
```

The pearson coefficient indicates no correlation (which is expected) and the two p-values are the same.

All the codes and figures used in this assignment can be found in this repository