CSDS 313: Assignment 2

Shaochen (Henry) ZHONG, sxz517
Ningjia HUANG, nsh239

Due and submitted on 10/19/2020
Issued by Dr. Koyutürk

Problem 1

(a)

Mean of Uniform Distribution $= \frac{a+b}{2} = \mu$
Standard Deviation of Uniform Distribution $= \sqrt{\frac{(b-a)^2}{12}} = \frac{b-a}{2\sqrt{3}} = \sigma$

$$\begin{gathered} a+b = 2\mu \\ b-a = 2\sqrt{3}\sigma \\ \Rightarrow b = \mu + \sqrt{3}\sigma \\ \Rightarrow a = \mu - \sqrt{3}\sigma \end{gathered}$$

Plugging in $\mu = 2$ and $\sigma = 5$ into a and b, we get:

$$\begin{gathered} a = 2 - 5\sqrt{3} \\ b = 2 + 5\sqrt{3} \end{gathered}$$

(b)

25th percentile of normal distribution $= \Phi^{-1}(0.25, \mu, \sigma)$
75th percentile of normal distribution $= \Phi^{-1}(0.75, \mu, \sigma)$
25th percentile of uniform distribution $\Rightarrow 0.25 = \frac{x_{25}-a}{b-a} \Rightarrow x_{25} = 0.25b + 0.75a$ 75th percentile of uniform distribution $\Rightarrow 0.75 = \frac{x_{75}-a}{b-a} \Rightarrow x_{75} = 0.75b + 0.25a$ $0.25b + 0.75a = \Phi^{-1}(0.25, \mu, \sigma)$
$0.75b + 0.25a = \Phi^{-1}(0.75, \mu, \sigma)$
$\Rightarrow a = \frac{3\Phi^{-1}(0.25,\mu,\sigma - \Phi^{-1}(0.75,\mu,\sigma))}{2}$
$\Rightarrow b = \frac{3\Phi^{-1}(0.75,\mu,\sigma) - \Phi^{-1}(0.25,\mu,\sigma)}{2}$
Plugging in $\mu = 2$ and $\sigma = 5$ into a and b and using the following code:

```
def calc_percentile():
    print(norm.ppf(0.25,2,5))
    print(norm.ppf(0.75,2,5))
```

We get the values for a and b:

$$a = -1.3724$$
$$b = 5.3724$$

Plots and Analysis

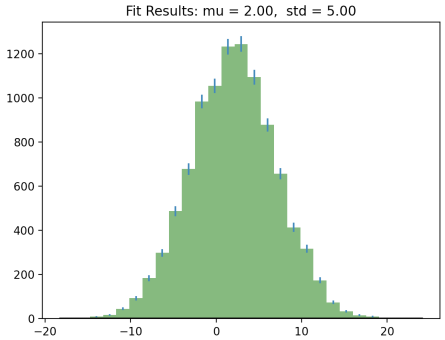

Figure 1: Normal Distribution Histogram with Error Bars

```
np.random.seed(1)
sample = norm.rvs(loc=2, scale=5, size=10000)
n, bins, _ = plt.hist(sample,bins=25,density=False,alpha=0.6,color='g')
mid = 0.5*(bins[1:] + bins[:-1])
plt.errorbar(mid, n, yerr=np.sqrt(n), fmt='none')
xmin, xmax = plt.xlim()
x = np.linspace(xmin, xmax, 100)
p = norm.pdf(x, 2, 5)
plt.plot(x, p, 'k', linewidth=2)
title = "Fit Results: mu = %.2f, std = %.2f" % (2, 5)
plt.title(title)
plt.show()
```

Figure 2: Normal Distribution Boxplot

```
np.random.seed(1)
sample = norm.rvs(loc=2, scale=5, size=10000)
df = pd.DataFrame(sample)
df.boxplot()
plt.show()
```

Figure 3: Uniform Distribution 1 Histogram with Error Bars

```
np.random.seed(1)
sample = uniform.rvs(loc=2-5*math.sqrt(3), scale=10*math.sqrt(3), size=10000)
n, bins, _ = plt.hist(sample,bins=25,density=False,alpha=0.6,color='b')
mid = 0.5*(bins[1:] + bins[:-1])
plt.errorbar(mid, n, yerr=np.sqrt(n), fmt='none')
xmin, xmax = plt.xlim()
x = np.linspace(xmin, xmax, 50)
p = uniform.pdf(x, 2-5*math.sqrt(3), 10*math.sqrt(3))
plt.plot(x, p, 'k', linewidth=2)
title = "Fit Results: a = %.2f, b = %.2f" % (2-5*math.sqrt(3), 2+5*math.sqrt(3))
plt.title(title)
plt.show()
```

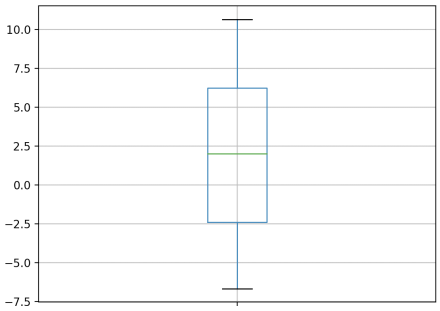

Figure 4: Uniform Distribution 1 Boxplot

```
np.random.seed(1)
sample = uniform.rvs(loc=2-5*math.sqrt(3), scale=10*math.sqrt(3), size=10000)
df = pd.DataFrame(sample)
df.boxplot()
plt.show()
```

Figure 5: Uniform Distribution 2 Histogram with Error Bars

```
np.random.seed(1)
sample = uniform.rvs(loc=-1.3724, scale=6.7448, size=10000)
n, bins, _ = plt.hist(sample,bins=25,density=False,alpha=0.6,color='y')
mid = 0.5*(bins[1:] + bins[:-1])
plt.errorbar(mid, n, yerr=np.sqrt(n), fmt='none')
xmin, xmax = plt.xlim()
x = np.linspace(xmin, xmax, 50)
p = uniform.pdf(x, -1.3723, 6.7448)
plt.plot(x, p, 'k', linewidth=2)
title = "Fit Results: a = %.2f, b = %.2f" % (-1.3724, 5.3724)
```

```
plt.title(title)
plt.show()
```

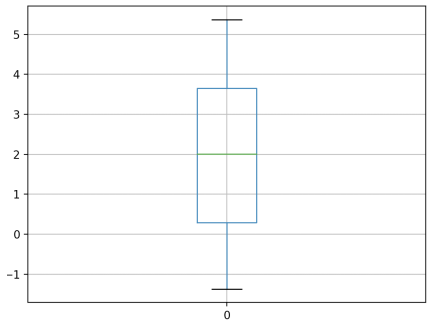

Figure 6: Uniform Distribution 2 Boxplot

```
np.random.seed(1)
sample = uniform.rvs(loc=-1.3724, scale=6.7448, size=10000)
df = pd.DataFrame(sample)
df.boxplot()
plt.show()
```

The histograms of uniform distribution 1 and 2 are quite different from the histogram of normal distribution. For uniform distribution 1 and 2, the $10,000$ sample points are more evenly distributed in the range comparing with normal distribution. For normal distribution, we can tell around $2,500$ sample points lie in the range $[0, 6]$. A few points start appearing around -18, then more appear until they reach the threshold(around 6), then the number decreases again. From the boxplots, we can tell the range of the points are larger for normal distribution than for uniform distributions. Also, for normal distribution, the distance between the minimum to the 25 percentile(around 10) is much larger than the distance between minimum and 25 percentile for uniform distributions(around 5.5 and 1.5). This is understandable because for uniform distributions, all points are equally likely to distribute in a certain range. However, for normal distribution, few points are distributed at the beginning and then the number of points increases gradually. Therefore, it takes longer time to reach the 25 percentile.
The shapes of these two uniform distributions are similar since for both of them, points are evenly distributed in the range. However, the range of the first uniform distribution is much larger than the second uniform distribution.

Problem 2:

(a)

airport:

```
df = pd.read_csv("airport_routes.csv")
array = df['NumberOfRoutes'].to_numpy()
sum = 0
for i in array:
    sum += np.log(i)
alpha = 1 + 3409 * sum**(-1)
```

Therefore, $\alpha = 1.612091630402382$.

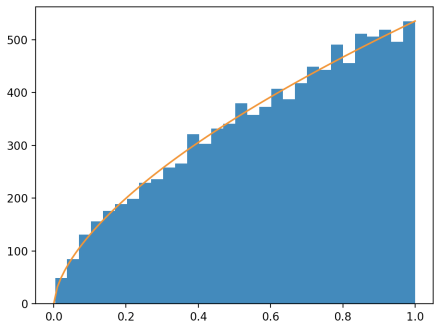

Figure 7: Power Law Distribution for Airport Routes

movie:

```
df = pd.read_csv("movie_votes.csv")
array = df['AverageVote'].to_numpy()
sum = 0
for i in array:
    sum += np.log(i/1.9)
alpha = 1 + 4392 * sum**(-1)
```

Therefore, $\alpha = 1.8505152700921788$.

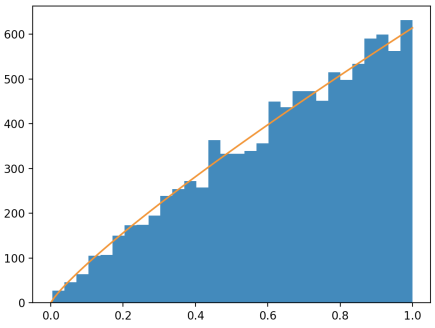

Figure 8: Power Law Distribution for Movie Votes

(b)

airport:

```
df = pd.read_csv("airport_routes.csv")
print(1/df.mean())
```

Therefore, $\lambda = 0.05039$.

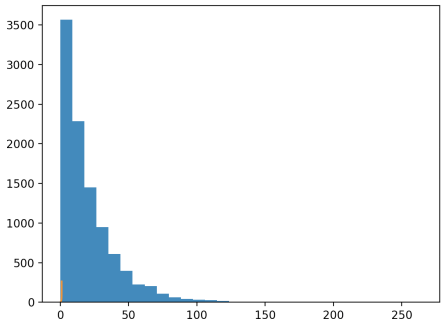

Figure 9: Exponential Distribution for Airport Routes

movie:

```
df = pd.read_csv("movie_votes.csv")
print(1/df.mean())
```

Therefore, $\lambda = 0.160593$.

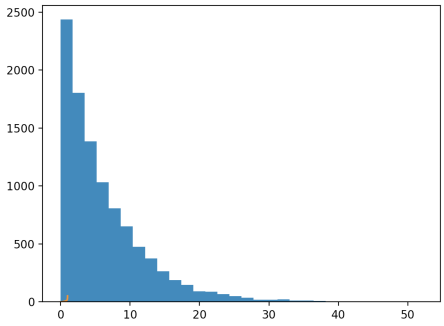

Figure 10: Exponential Distribution for Movie Votes

(c)

airport:

```
df = pd.read_csv("airport_routes.csv")
print("a = ", df.min(), ", b = ", df.max())
```

Therefore, $[a, b] = [1, 915]$.

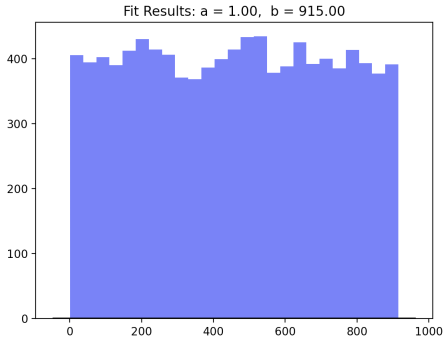

Figure 11: Uniform Distribution for Airport Routes

movie:

```
df = pd.read_csv("movie_votes.csv")
print("a = ", df.min(), ", b = ", df.max())
```

Therefore, $[a, b] = [1.9, 8.5]$.

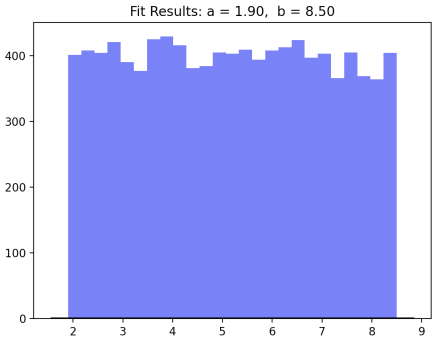

Figure 12: Uniform Distribution for Movie Ratings

(d)

airport:

```
df = pd.read_csv("airport_routes.csv")
mean = df.mean(axis=0)
std = df.std(axis=0)
```

Therefore, $(\mu, \sigma) = (19.845116, 53.506467)$.

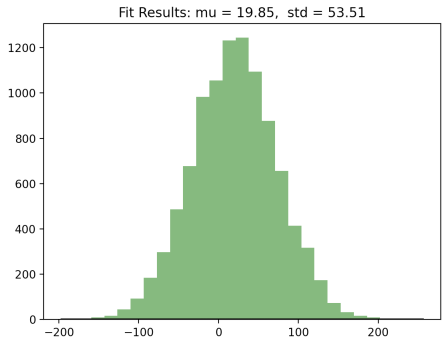

Figure 13: Normal Distribution for Airport Routes

movie:

```
df = pd.read_csv("movie_votes.csv")
mean = df.mean(axis=0)
std = df.std(axis=0)
```

Therefore, $(\mu, \sigma) = (6.226935, 0.893215)$.

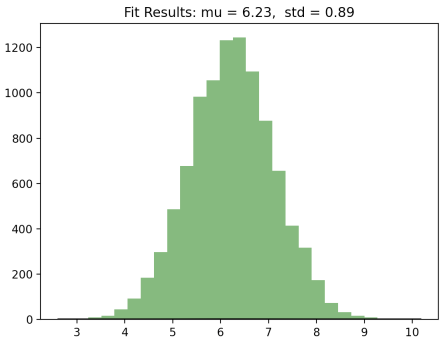

Figure 14: Normal Distribution for Movie Ratings

Analysis:

The actual histogram plot for airport routes is:

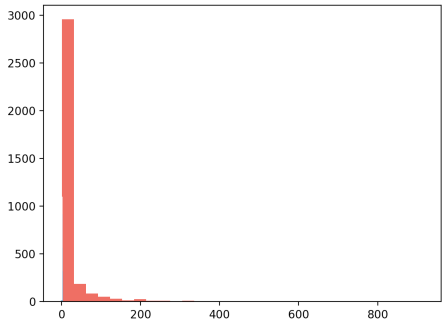

Figure 15: Actual Histogram for Airport Routes

We can tell it fits the exponential distribution the most according to its shape. Also, exponential distribution seems to be a reasonable simulation since only a few of airports have a huge number of routes. Most of airports aggregate on the side with relatively small number of routes.
The actual histogram plot for movie votes is:

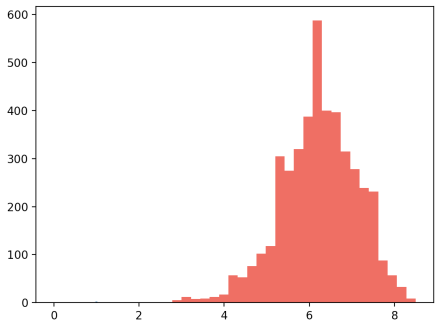

Figure 16: Actual Histogram for Movie Ratings

We can tell it fits the normal distribution the most since the histogram seems to have a bell shape. The threshold appears around 6, which agrees with the statistic we calculated before(around 6.22).

Problem 3

(i)

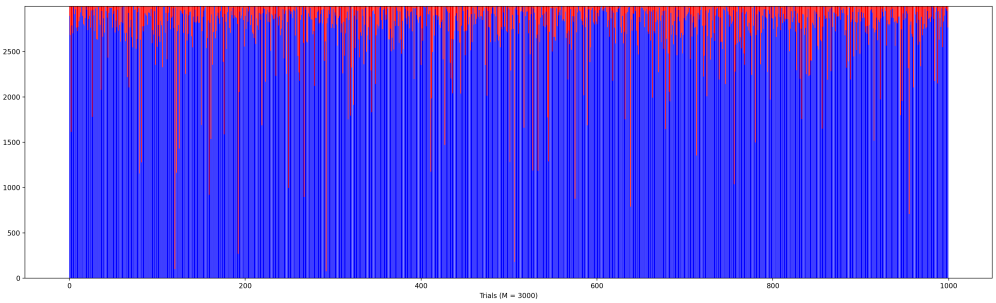

Figure 17: $E(MLE_M)$ v. $E((M))$

Note the MLE for the population is just M itself, thus $E(M) = M$. We simulated `1000` trials with $M = 3000$, where $E(MLE_M)$ is in blue, and its distance to $E((M))$ is in red.

Note most of the trials there is a noticiable red bar between them, which shows the M_{MLE} on sample is not an unbiased estimator of M.

(ii)

M_{MVU} showed a lower variance over our `1000` trials simulation. The avergae finding are below:

- $Var(M_{MEAN})$: `2000277.10133029`
- $Var(M_{MVU})$: `2854104731.1510997`

We also picked `20` trials to visualize, not there the difference of the two estimators' variances are huge, as we can't even see the variance of M_{MVU} without some extreme zooming.

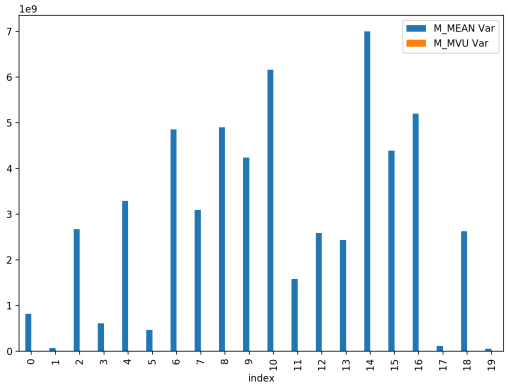

Figure 18: $Var(M_{MEAN})$ v. $Var(M_{MVU})$