CS 249 | HW # 1 | Jason Chapman

(1.) a) $Y_1, \dots, Y_n \sim \text{Exponential}(\theta)$.

$$f(y_i) = \theta e^{-\theta y}$$

$$L(\theta) = \prod_{i=1}^{n} \theta e^{-\theta y_i}$$

$$\ell(\theta) = \sum_{i=1}^{n} \left[\log\theta - \theta y_i\right]$$

$$\ell'(\theta) = \sum_{i=1}^{n} \left[\frac{1}{\theta} - y_i\right] = 0 \Rightarrow \frac{n}{\theta} - \sum_{i=1}^{n} y_i = 0 \Rightarrow \boxed{\hat{\theta}_{MLE} = \frac{n}{\sum_{i=1}^{n} y_i}}$$

b) $Y_1, \dots, Y_n \sim \text{Uniform}(0, \theta)$

$$f(y_i) = \frac{1}{\theta}$$

$$L(\theta) = \prod_{i=1}^{n} \frac{1}{\theta} = \frac{1}{\theta^n}$$

$$\ell(\theta) = \sum_{i=1}^{n} \log\left(\frac{1}{\theta}\right) = \log\left(\frac{1}{\theta^n}\right) = -n\log\theta$$

$$\ell'(\theta) = \frac{-n}{\theta} < 0$$

$\Rightarrow$ $L(\theta)$ is a decreasing function for $\theta \geq Y_{(n)}$. Thus, $L(\theta)$ is maximized at $\theta = Y_{(n)}$

$$\Rightarrow \boxed{\hat{\theta}_{MLE} = \max(Y_i)}$$

c.) See Attached

// Need function for $E[X]$ for this function:

$$E[X] = \int_{-\infty}^{\infty} x f(x)\,dx = \int_{-\infty}^{\infty} x \frac{1}{x}\,dx = X$$

From Attached Python:

$$\boxed{\begin{aligned} \text{Bias}(\theta) &= -0.034 \\ \text{Var}(\theta) &= -1.358 \\ \text{RMSE}(\theta) &= 0.034 \end{aligned}}$$

d.) From Attached Python:

$$\boxed{\begin{aligned} \text{Bias}(\theta) &= -0.0028 \\ \text{Var}(\theta) &= -0.1108 \\ \text{RMSE}(\theta) &= 0.0028 \end{aligned}}$$

* The bias, variance, & RMSE as we increase from $n=200$ to $n=1000$. This is because as $n \to \infty$, the estimator will converge to the true value as we increase the # of samples

② a.)

$$\boxed{\begin{array}{l} SE \text{ for Median} = 0.743 \\ 95\% \text{ Confidence Interval for Median: } (2.8,\ 6.05) \end{array}}$$

b.)

$$\boxed{\begin{array}{l} SE \text{ for Median from Bootstrap} = 0.622 \\ SE \text{ for Max from Bootstrap} = 28.283 \\ SE \text{ for Median from Simulation} = 0.621 \\ SE \text{ for Max from Simulation} = 29.115 \end{array}}$$

- The standard errors from Bootstrap & from Simulation are roughly the same. Thus, bootstrap is sufficient for estimating the standard error for Median & Max.

③ a.) $\boxed{\hat{\theta} = 4.265}$

b-d) $\boxed{SE \text{ for } \theta = 0.445}$

④ a.) Prior belief: $\theta \sim N(0,3)$

Observation: $X_1, \dots X_{10} \sim N(\theta, 1)$ w/ $\bar{x} = 1.68$

$$\sigma_1^2 = \frac{\sigma_0^2 \sigma^2}{\sigma^2 + n\sigma_0^2} = \frac{3 \times 1}{1 + 10(3)} = \frac{3}{31} = 0.0967$$

$$\mu_1 = \sigma_1^2 \left(\frac{\mu_0}{\sigma_0^2} + \frac{n\bar{X}}{\sigma^2} \right) = \frac{3}{31} \left(\frac{0}{3} + \frac{10(1.68)}{1} \right) = \frac{3}{31}(16.8) = 1.6258$$

$$\boxed{\theta \sim N(1.6258,\ 0.0967)}$$

b.) $Z_1, \dots, Z_{20} \sim N(\theta, 4)$ $\quad \bar{z} = 0.8$

$$\sigma_1^2 = \frac{\sigma_0^2 \sigma^2}{\sigma^2 + n\sigma_0^2} = \frac{3 \times 4}{4 + 20(3)} = \frac{12}{64} = 0.1875$$

$$\mu_1 = \sigma_1^2 \left(\frac{\mu_0}{\sigma_0^2} + \frac{n\bar{X}}{\sigma^2} \right) = 0.1875 \left(\frac{0}{3} + \frac{20(0.8)}{4} \right) = 0.75 \Rightarrow \boxed{\theta \sim N(0.75,\ 0.1875)}$$

c.) As the # of samples increase, the variance of the posterior decreases.

d.) If the priori distribution had lower variance, then the variance of the posterior also decreases.

```
import numpy as np
# CS 249 - HW #1 - Jason Chapman
# Number 1
# c
theta = 20
n = 200
for x in range(1000):
    y1 = np.random.uniform(0,theta,n)
    theta_mle = max(y1)

def exp_val(x):
    return x

bias_1c = exp_val(theta_mle) - theta
var_1c = exp_val(theta_mle**2)-(exp_val(theta))**2
RMSE_1c = np.sqrt(exp_val((theta_mle-theta)**2))
print('1c: Bias={}, Variance={}, RMSE={}'.format(bias_1c,var_1c,RMSE_1c))

# d
n = 1000
for x in range(1000):
    y1 = np.random.uniform(0,theta,n)
    theta_mle = max(y1)

bias_1d = exp_val(theta_mle) - theta
var_1d = exp_val(theta_mle**2)-(exp_val(theta))**2
RMSE_1d = np.sqrt(exp_val((theta_mle-theta)**2))
print('1d: Bias={}, Variance={}, RMSE={}'.format(bias_1d,var_1d,RMSE_1d))

# Number 2
# a
data = [3.0, 1.9, 6.4, 5.9, 4.2, 6.2, 1.4, 2.9, 2.3, 4.8, 7.8, 4.5, 0.7, 4.4, 4.4
T = np.median(data)
def t_stat(y):
    return np.median(y)

t_boot_list = []
for b in range(1000):
    data_boot = np.random.choice(data,len(data),replace=True)
    t_boot = t_stat(data_boot)
    t_boot_list.append(t_boot)

lower = np.quantile(t_boot_list,0.025)
higher = np.quantile(t_boot_list,0.975)
se = np.std(t_boot_list)
print('2a: Standard Error for Median={}, 95% Confidence Interval for Median=({},{

# b
y2 = np.random.normal(0,5,100)
def t_stats(y):
    T1 = np.median(y)
```

```
    T2 = np.argmax(y)
    return T1, T2

t1_boot_list = []
t2_boot_list = []
for b in range(1000):
    y2_boot = np.random.choice(y2,len(y2),replace=True)
    t1_boot, t2_boot = t_stats(y2_boot)
    t1_boot_list.append(t1_boot)
    t2_boot_list.append(t2_boot)

se_t1 = np.std(t1_boot_list)
se_t2 = np.std(t2_boot_list)

t1_stats = []
t2_stats = []
for i in range(10000):
    y2_sim = np.random.normal(0,5,100)
    t1_sim, t2_sim = t_stats(y2_sim)
    t1_stats.append(t1_sim)
    t2_stats.append(t2_sim)

se_t1_sim = np.std(t1_stats)
se_t2_sim = np.std(t2_stats)
print('2b: Standard Error for Median from Bootstrap={}, Standard Error for Maximu
print('Standard Error for Median from Simulation={},Standard Error for Maximum fr

# 3
# a
theta_hat = np.mean(data)
print('3a: Estimated Mean from Observed Data={}'.format(theta_hat))

# b-d
theta_sim_list = []
for j in range(10000):
    sim_data = np.random.normal(theta_hat,2,20)
    theta_sim_list.append(np.mean(sim_data))

theta_sim = np.std(theta_sim_list)
print('3d: The Estimated Standard Error for theta={}'.format(theta_sim))
```