(5)

(a) $\hat{\theta}_{MLE} = X_{(n)} \Rightarrow PDF(\hat{\theta}_{MLE}) = PDF(X_{(n)})$

$\Rightarrow pdf(X_{(n)}) = \dfrac{d\,CDF(X_{(n)})}{d(X_{(n)})}$

$\Rightarrow CDF(X_{(n)}) = P(X_{(n)} < \alpha) = \left(\dfrac{\alpha}{\theta}\right)^n$

$\Rightarrow \dfrac{d\,CDF(X_{(n)})}{d(X_{(n)})} = n\cancel{\left(\dfrac{\alpha}{\theta}\right)^{n-1}} = pdf(X_{(n)})$

$$\boxed{\Rightarrow pdf(X_{(n)}) = \frac{n\alpha^{n-1}}{\theta^n}}$$

---

(2)(a) $\mu = \dfrac{\beta+\alpha}{2} = m_1 \approx \boxed{\hat{m}_1 = \bar{X}_n}$, $\hat{m}_2 = \displaystyle\int_\alpha^\beta x^2\left(\frac{1}{\beta-\alpha}\right)dx = \frac{\beta^2+\alpha\beta+\alpha^2}{3}$

$\sigma^2 = \displaystyle\int_\alpha^\beta x^2\left(\frac{1}{\beta-\alpha}\right)dx - \left[\int_\alpha^\beta x\left(\frac{1}{\beta-\alpha}\right)dx\right]^2 = \frac{(\beta-\alpha)^2}{12} = m_2 - m_1^2 \approx \hat{m}_2 - \hat{m}_1^2$

$\Rightarrow \beta+\alpha = 2m_1,\ \beta-\alpha = \sqrt{12(\hat{m}_2-\hat{m}_1^2)} \Rightarrow$ system of linear equations

$\Rightarrow 2\beta = 2m_1 + 2\sqrt{3(\hat{m}_2-\hat{m}_1)^2} \Rightarrow \boxed{\beta = \hat{m}_1 + \sqrt{3(\hat{m}_2-\hat{m}_1^2)}}$ $(\hat{m}_2 - \hat{m}_1^2)$

$\alpha = 2\hat{m}_1 - \beta = 2\hat{m}_1 - \hat{m}_1 - \sqrt{3(\hat{m}_2-\hat{m}_1^2)} \Rightarrow \boxed{\alpha = \hat{m}_1 - \sqrt{3(\hat{m}_2-\hat{m}_1^2)}}$

where $\hat{m}_1 = \bar{X}_n \approx \dfrac{\alpha+\beta}{2}$, $\hat{m}_2 = \dfrac{1}{n}\sum x_i^2 \approx \dfrac{\beta^2+\alpha\beta+\alpha^2}{3}$

(b) $L_n(\alpha) = \displaystyle\prod_{i=1}^{n} f(x_i;\alpha) = \begin{cases} \dfrac{1}{(\beta-\alpha)^n} & \text{if } x_{(1)} \geq \alpha \\ 0 & \text{if } x_{(1)} < \alpha \end{cases}$ | $L_n(\beta) = \begin{cases} 0 & \text{if } X_{(n)} > \beta \\ \dfrac{1}{(\beta-\alpha)^n} & \text{if } X_{(n)} \leq \beta \end{cases}$

Given $L_n(\alpha) = 0$ for $\alpha > X_{(1)}$ and $\dfrac{dL_n(\alpha)}{d(\alpha)} = n(\beta-\alpha)^{-n-1} > 0\ \forall\alpha$, $L_n(\alpha)$ maximized at $\boxed{\alpha = X_{(1)}}$

Given $L_n(\beta) = 0$ for $\beta < X_{(n)}$ and $\dfrac{dL_n(\beta)}{d\beta} = -n(\beta-\alpha)^{-n-1} < 0\ \forall\beta$, $L_n(\beta)$ maximized at $\boxed{\beta = X_{(n)}}$

(4)

(a) ψ in terms of θ: $\psi = \mathbb{E}(Y_1) = \mathbb{E}(Y_i) = 1 \cdot \mathbb{P}(Y_i = 1) + 0 \cdot \mathbb{P}(Y_i = 0)$

$= \mathbb{P}(Y_i = 1) = \mathbb{P}(X_i > 0) = 1 - \mathbb{P}(X_i \leq 0)$, where $X_i \in N(\theta, 1)$

notice then $X_i - \theta \in N(0,1)$, the standard normal distribution

thus: $1 - \mathbb{P}(X_i \leq 0) = 1 - \mathbb{P}(X_i - \theta \leq -\theta) = 1 - \Phi(-\theta)$ where

Φ is the standard normal CDF, thus $\Phi(-\theta) = 1 - \Phi(\theta)$

and we simplify to $1 - \Phi(-\theta) = 1 - (1 - \Phi(\theta)) = \Phi(\theta)$, which

concludes the equivalent that $\boxed{\psi = \Phi(\theta)}$, we now know

$\psi = \Phi(\theta)$ is some function solely dependent on θ so by invariance

property of MLE's $\Rightarrow \hat{\psi}_{MLE} = \Phi(\hat{\theta}_{MLE})$, which we know

for given $X_1 \dots X_n \in N(\theta, 1) \Rightarrow$ ~~MLE~~ $\hat{\theta}_{MLE} = \bar{X}_n$ (from notes)

so we also conclude by invariance property: $\boxed{\hat{\psi}_{MLE} = \Phi(\bar{X}_n)}$

(b)

Given $\psi = \Phi(\theta)$, we can build the CI for ψ from the CI of θ

given θ from $N(\theta, 1)$ and $X_1 \dots X_n \sim N(\theta, 1)$ the confidence interval

at 95% for a normal distribution is given by $\bar{X}_n \pm 1.96 \frac{\sigma}{\sqrt{n}}$ $\sigma \to \sigma = 1$

$\Rightarrow \mathbb{P}(\bar{X}_n - (1.96/\sqrt{n}) \leq \theta \leq \bar{X}_n + (1.96/\sqrt{n})) = .95$, now Φ is a

CDF and thus a monotonically increasing function, meaning applying Φ still holds the inequality:

$\mathbb{P}\left(\Phi\left(\bar{X}_n - \frac{1.96}{\sqrt{n}}\right) \leq \Phi(\theta) \leq \Phi\left(\bar{X}_n + \frac{1.96}{\sqrt{n}}\right)\right) \Rightarrow \boxed{\mathbb{P}\left(\Phi\left(\bar{X}_n - \frac{1.96}{\sqrt{n}}\right) \leq \psi \leq \Phi\left(\bar{X}_n + \frac{1.96}{\sqrt{n}}\right)\right) = .95}$

ACM 157 PS4

(3)

(a) For a uniform distribution $U[\alpha, \beta]$, the true mean μ is given by $\mu = \frac{\alpha+\beta}{2}$, from which we see μ is dependent solely on α, β, thus the MLE estimate of μ, $\hat{\mu}_{MLE}$ can be determined from its dependence on the MLE estimates of $\hat{\alpha}_{MLE}$, $\hat{\beta}_{MLE}$ as those parameters are maximum-likelihood estimates and thus any ~~that~~ linear combination / formula of those parameters also has likelihood maximized with $\hat{\alpha}_{MLE}$, $\hat{\beta}_{MLE}$ plugged in. Thus $\hat{\mu}_{MLE} = \frac{\hat{\alpha}_{MLE} + \hat{\beta}_{MLE}}{2}$, which from Question (2) we know $\hat{\alpha}_{MLE} = X_{(1)}$, $\hat{\beta}_{MLE} = X_{(n)}$ thus $\boxed{\hat{\mu}_{MLE} = \frac{X_{(1)} + X_{(n)}}{2}}$

Invariance property of MLE

```
clc; clear; close all;
```

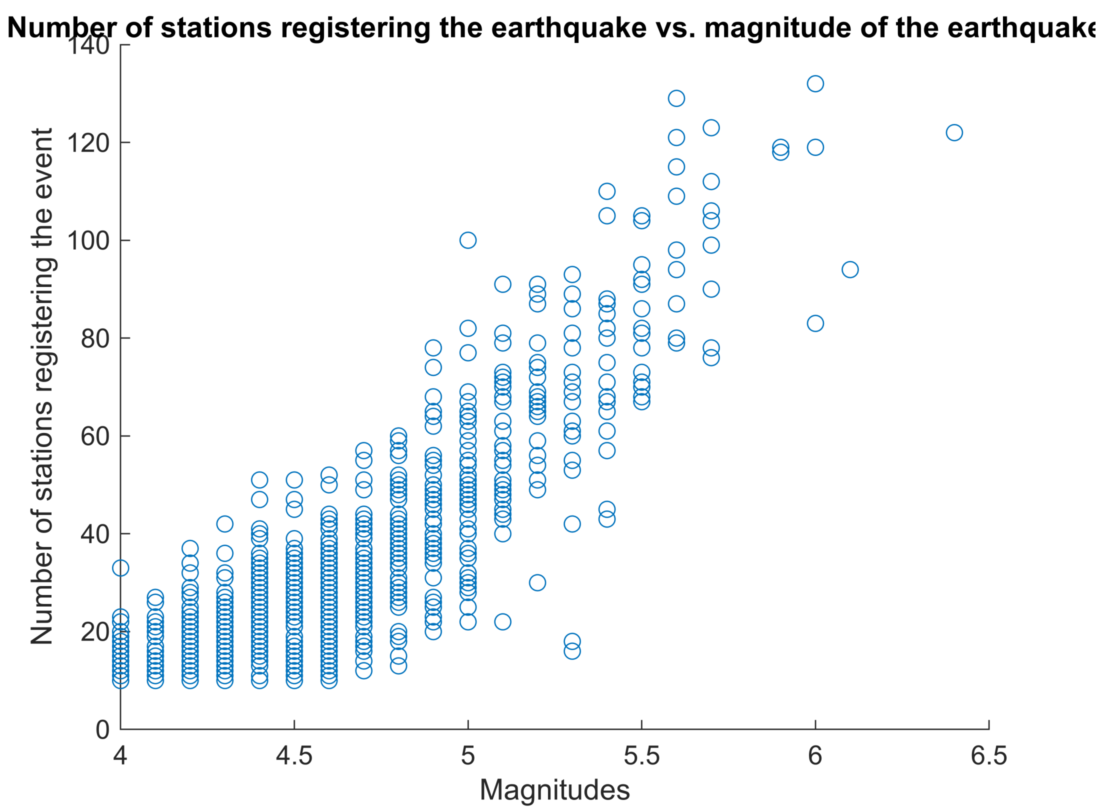
Problem Set 4 Matlab Script

Question 1

```
data = textread("C:\Users\sayuj\OneDrive - California Institute of
Technology\ACM_157\fiji.txt");
magnitudes = data(:,5);
stations = data(:, 6);
```

Part A

```
figure;
scatter(magnitudes, stations)
xlabel('Magnitudes')
ylabel('Number of stations registering the event')
title('Number of stations registering the earthquake vs. magnitude of the
earthquake')
```



Part B

```
plug_in_corr = corr(magnitudes, stations);
disp('Plug-in estimate of correlation')
```

```
Plug-in estimate of correlation
```

```
disp(plug_in_corr)
```

```
0.8512
```

Part C

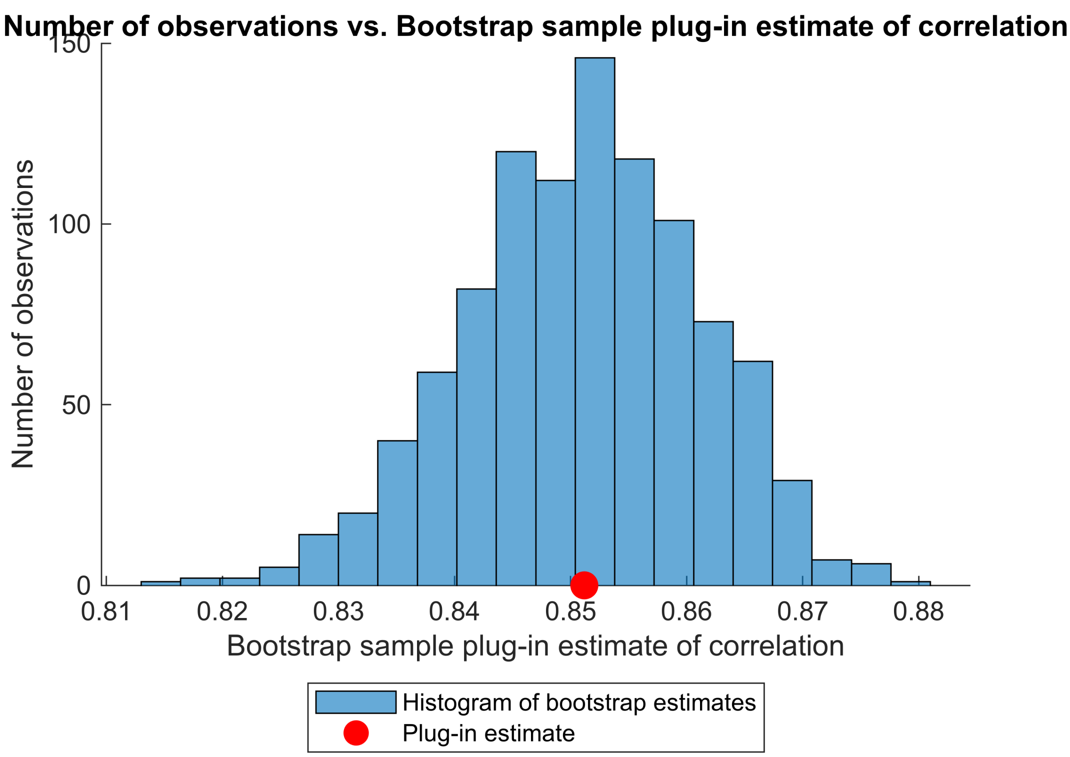
```
[estim_se, bootstraps] = bootstrap_corr_algo(magnitudes, stations, 1000);
disp('Bootstrap estimated SE of estimated correlation')
```

```
Bootstrap estimated SE of estimated correlation
```

```
disp(estim_se);
```

```
0.0100
```

```
figure;
hold on;
histogram(bootstraps, 20, 'DisplayName', 'Histogram of bootstrap estimates')
plot(plug_in_corr, 0, 'ro', 'MarkerFaceColor', 'r', 'MarkerSize', 10,
'DisplayName', 'Plug-in estimate')
xlabel('Bootstrap sample plug-in estimate of correlation')
ylabel('Number of observations')
title('Number of observations vs. Bootstrap sample plug-in estimate of correlation')
legend('Location', 'southoutside');
hold off;
```



Part D

```
norm_lower = norminv(.025, plug_in_corr, estim_se);
norm_upper = norminv(.975, plug_in_corr, estim_se);

sorted_boots = sort(bootstraps);

pivot_lower = sorted_boots(floor(1000 * .975));
pivot_upper = sorted_boots(floor(1000 * .025));

pivot_lower = 2 * plug_in_corr - pivot_lower;
pivot_upper = 2 * plug_in_corr - pivot_upper;

fprintf('Normal confidence interval (lower, upper): ');
```

```
Normal confidence interval (lower, upper):
```

```
disp(norm_lower);disp(norm_upper);
```

```
    0.8315

    0.8708
```

```

fprintf('Pivot confidence interval (lower, upper): ');
```

```
Pivot confidence interval (lower, upper):
```

```
disp(pivot_lower);disp(pivot_upper);
```

```
    0.8334

    0.8723
```

Question 3 (B)

We can analytically compute the MSE of the plug-in estimate of the mean by first knowing that the plug-in estimate is unbiased as the expected sample mean is equal to the population mean, and thus the MSE is just the variance of the sample mean which is var / n, where var = (b-a)^2 / 12. We see that the MSE for the plug-in estimate of the mean is ~.032 and the MSE for the MLE estimate of the mean is around ~.014, showing that the MSE for the MLE estimate is significantly lower than the MSE of the plug-in estimate.

```
b = 3;
a = 1;

samples = (b - a) * rand(10, 1000) + a;
```

```
mle_err = zeros(1000, 1);

for i = 1: 1000
    curr_sample = samples(:, i);
    mle_mean = (min(curr_sample) + max(curr_sample)) / 2;
    mle_err(i) = (mle_mean - 2).^2;
end

plug_in_mse = (b-a).^2 / (12 * 10);

fprintf('Plug-in estimate MSE (analytically calculated):')
```

```
Plug-in estimate MSE (analytically calculated):
```

```
disp(plug_in_mse)
```

```
    0.0333
```

```
fprintf('MLE estimate MSE (calculated via Monte Carlo simulation):')
```

```
MLE estimate MSE (calculated via Monte Carlo simulation):
```

```
disp(mean(mle_err))
```

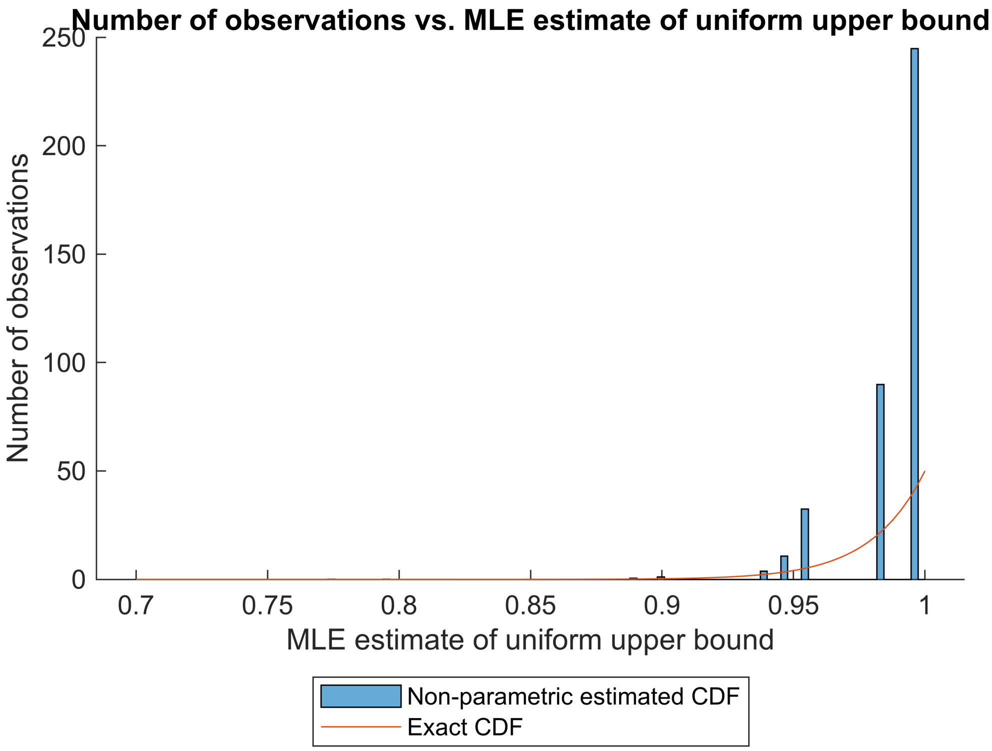
```
    0.0125
```

Question 5 (B)

We notice that the histogram of the non-parametric and parametric bootstraps of the MLE estimate have very similar shapes to the exact CDF of the MLE which is the n-th order statistic of the sample. We notice that both bootstraps have a maximum value dependent on the MLE estimate of the sample which is the max of the sample, so both histograms don't necessarily reach 1 but do approach very close to it (~.99). The parametric bootstrap appears to be more monotonically increasing and thus CDF-like compared to the non-parametric method which makes sense as the parametric bootstrap is estimating a pdf and thus the cdf histogram sampled from that pdf should appear to be more similar to the real CDF than the non-parametric method which can have outliers due to the cdf histogram generated from the bootstrap sample eCDF which can vary more and not appear as continuous.

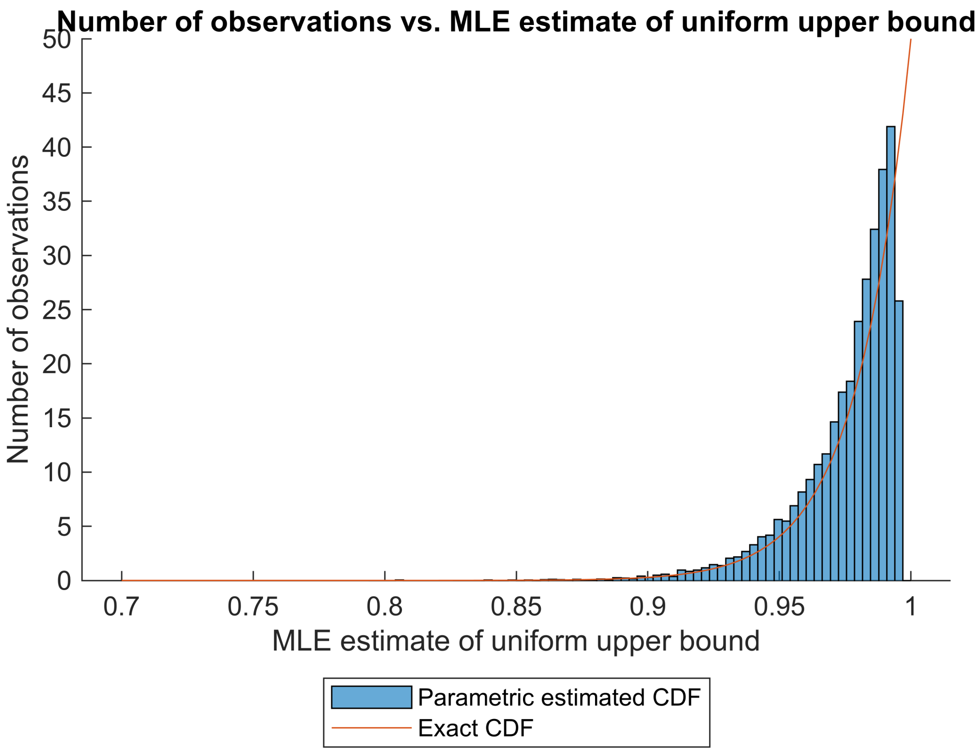
```
% Exact function

n = 50;

x = linspace(.7, 1, 100);
y = n*x.^(n-1);

sample = rand(1, n);

% Non-parametric plot
```

```
figure;
hold on;
bootstraps_ecdf = bootstrap_ecdf_algo(50, sample);
histogram(bootstraps_ecdf, 'BinLimits', [.7 1], 'Normalization', 'pdf',
'DisplayName', 'Non-parametric estimated CDF');
plot(x, y, 'DisplayName', 'Exact CDF');
xlabel('MLE estimate of uniform upper bound');
ylabel('Number of observations');
title('Number of observations vs. MLE estimate of uniform upper bound');
legend('location', 'southoutside');
hold off;
```




```
% Parametric plot

figure;
hold on;
bootstraps_func = bootstrap_func_algo(50, sample);
histogram(bootstraps_func, 'BinLimits', [.7 1], 'Normalization', 'pdf',
'DisplayName', 'Parametric estimated CDF');
plot(x, y, 'DisplayName', 'Exact CDF');
xlabel('MLE estimate of uniform upper bound');
ylabel('Number of observations');
title('Number of observations vs. MLE estimate of uniform upper bound');
legend('location', 'southoutside');
```

```
hold off;
```



Question 1 Functions

```
function [bootstrap_se, bootstraps] = bootstrap_corr_algo(sample1, sample2, n)

    bootstrap_samples = 1000;
    sample_corrs = zeros(bootstrap_samples, 1);

    merged_sample = [sample1, sample2];

    for i = 1:bootstrap_samples
        sample_data = datasample(merged_sample, n, 'Replace', true);


        sample_corr = corr(sample_data(:, 1), sample_data(:, 2));

        sample_corrs(i) = sample_corr;
    end

    bootstrap_se = sqrt(sum((sample_corrs - (mean(sample_corrs))).^2) /
bootstrap_samples);
    bootstraps = sample_corrs;
end
```

Question 5 Functions

```
function bootstraps = bootstrap_ecdf_algo(n, sample)
    num_bootstraps = 10000;
    bootstraps = zeros(num_bootstraps, 1);
    for i = 1: num_bootstraps
        sample_data = datasample(sample, n);
        bootstraps(i) = max(sample_data);
    end
end

function bootstraps = bootstrap_func_algo(n, sample)
    num_bootstraps = 10000;
    bootstraps = zeros(num_bootstraps, 1);
    for i = 1: num_bootstraps
        sample_data = rand(1, n) * max(sample);

        bootstraps(i) = max(sample_data);
    end
end
```