EI035 Econometrics I
The Graduate Institute, Fall 2024, Marko Milkota

PS1 Solutions

Jingle Fu

Solution (a).

For the Least Squares estimation, we need to solve the following problem:

$$\min \sum_{i=1}^{n} (x_i - \theta)^2$$

Denote $F = \sum\limits_{i=1}^{n} (x_i - \theta)^2$ and take the first order derivative with respect to θ, we have the FOC:

$$\frac{\partial F}{\partial \theta} = \sum -2(x_i - \theta) = 0$$
$$\Rightarrow \hat{\theta} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

Solution (b).

The mean of $\hat{\theta}$ is its expectation, so we calculate:

$$\mathbb{E}[\hat{\theta}] = \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n} x_i\right] = \frac{1}{n}\mathbb{E}\left[\sum_{i=1}^{n} x_i\right] \stackrel{1}{=} \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}[x_i] \stackrel{2}{=} \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}[u_i] = \frac{1}{n} \cdot n\theta = \theta$$

1. We are using the property of $\mathbb{E}$: $\mathbb{E}\left[\sum\limits_{i=1}^{n} x_i\right] = \sum\limits_{i=1}^{n} \mathbb{E}[x_i]$, no matter x_i are independent or not.

2. We are using the property of $\mathbb{E}$: $\mathbb{E}[\theta + u_i] = \mathbb{E}[u_i] + \theta$ if θ is a constant number.

Thus, $\hat{\theta}$ is unbiased and we make no other assumptions on pdf of $x_i|\theta$ and the sample $\{x_i\}_{i=1}^{n}$.

Solution (c).

$$\mathbb{V}[\hat{\theta}] = \mathbb{V}\left[\frac{1}{n}\sum_{i=1}^{n} x_i\right] = \frac{1}{n^2}\mathbb{V}\left[\sum_{i=1}^{n} x_i\right] = \frac{1}{n^2}\sum_{i=1}^{n} \mathbb{V}[x_i] = \frac{\sigma^2}{n}$$

By having this result, we make the following 2 assumptions:

1. For sample set $\{x_i\}_{i=1}^n$, we assume that for all samples x_i, they are mutually independent, which gives that $\mathbb{V}\left[\sum_{i=1}^n u_i\right] = \sum_{i=1}^n \mathbb{V}[u_i]$.
2. For the pdf of $x_i|\theta$, we assume that x_i are independently distributed, which means that $\mathbb{V}[x_i] = \mathbb{V}[u_i] = \sigma^2$

Solution (d).

```python
import numpy as np
theta = 175
n = 10
u_i = np.random.uniform(-5, 5, n)
x_i = theta + u_i
theta_hat = np.mean(x_i)

print("Estimated theta = ", theta_hat)

```

For $n = 10$, estimated $\hat{\theta} = 175.27620432553718$, it is close to the true value of 175. Since $u_i \sim \mathcal{U}[-5, 5]$, we would have: $\mathbb{V}[\hat{\theta}] = \frac{10^2}{12} = \frac{25}{3}$.

```python
import numpy as np
# Parameters
theta = 175
n_values = [100, 1000]

# Simulation
theta_hat = {n: float(np.mean(theta + np.random.uniform(-5, 5, n))) for n in n_values}

print("Estimated theta =", theta_hat)

```

For $n = 100$, estimated $\hat{\theta} = 175.09859744383778$, it is closer to $\theta = 175$ than $n = 10$.
For $n = 1000$, estimated $\hat{\theta} = 175.03889315916487$, it is closer to $\theta = 175$ than $n = 100$.

Solution (e).

To analyze the performance of the estimator as the sample size changes, we simulate the process for different values of n and plot the distributions of the estimator:

```python
import numpy as np
import matplotlib.pyplot as plt
# Function to simulate
def simulate_theta_hat(n, theta, M):
    theta_hats = []
    for _ in range(M):
        u_i = np.random.uniform(-5, 5, n)
        x_i = theta + u_i
        # Estimate theta_hat
        theta_hat = np.mean(x_i)
        theta_hats.append(theta_hat)
    return theta_hats
# Parameters
theta = 175
M = 100
# Simulate and plot for n = 10
n = 10
theta_hats_10 = simulate_theta_hat(n, theta, M)
plt.figure(figsize=(6, 4))
plt.hist(theta_hats_10, bins=20, edgecolor='black')
plt.title(f'n = {n}')
plt.xlabel(r'$\hat{\theta}$ values')
plt.ylabel('Frequency')
plt.show()
```

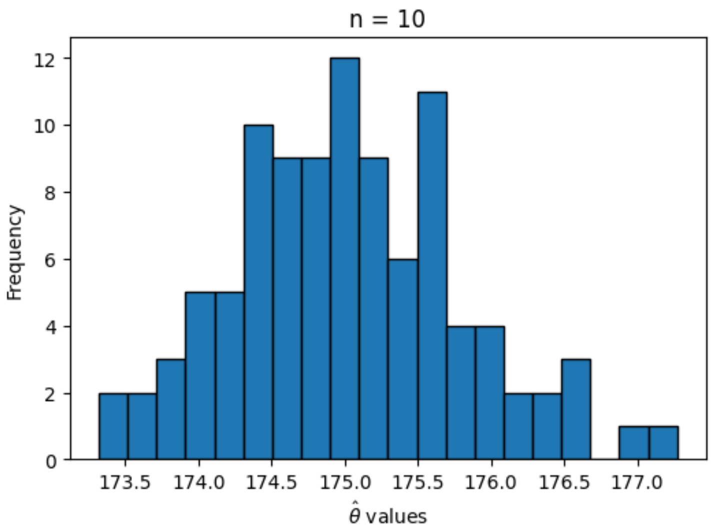


Figure 1: Simulation $n = 10$

The distribution of $\hat{\theta}$ appears to be centered around 175, but given that $n = 10$ is relatively small, the wider spread (ranging roughly from 173.5 to 177.0) reflects higher variability or higher standard error in the estimation. It is in line with my expectations.

Solution (f).

```
# Simulate_theta_hat Function is the same with previous question
# Simulate and plot for n=100, 1000
n_values = [100, 1000]
results = {}
for n in n_values:
    results[n] = simulate_theta_hat(n, theta, M)
fig, axes = plt.subplots(1, 2, figsize=(18, 6))
for ax, n in zip(axes, n_values):
    ax.hist(results[n], bins=20, edgecolor='black')
    ax.set_title(f'n = {n}')
    ax.set_xlabel(r'$\hat{\theta}$ values')
    ax.set_ylabel('Frequency')
plt.tight_layout()
plt.show()

```

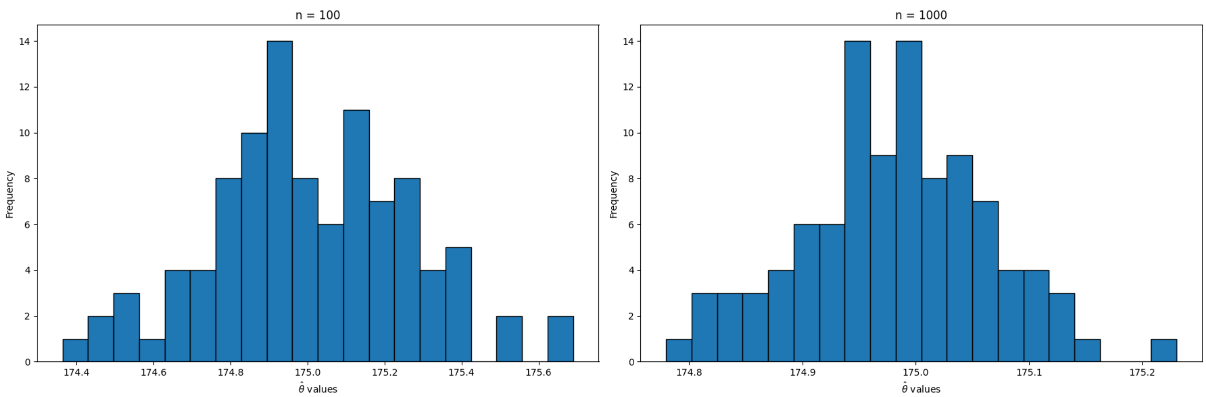


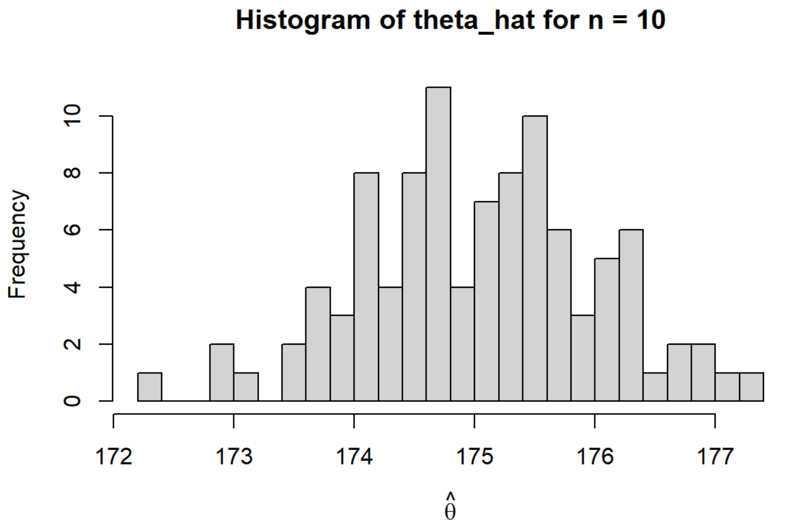
Figure 2: Simulation $n = 100, 1000$

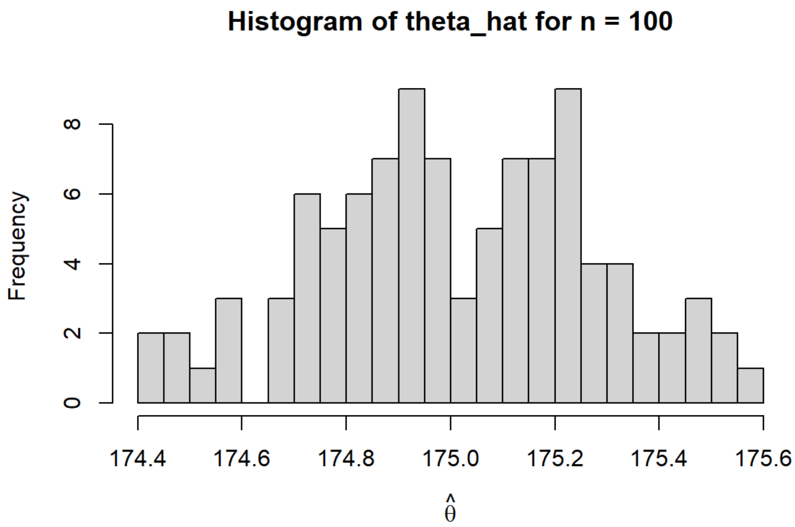
For $n = 100$, the histogram is more centralized around 175 than $n = 10$. It appears more symmetric and bell-shaped than the histogram for $n = 10$, although it still exhibits some irregularities and skewness. This starts to hint at the **Central Limit Theorem**'s effect. For $n = 1000$, the histogram centers very closely around 175, the increase of sample size further reduced variability. It is the most symmetric and bell-shaped among the three, very closely resembling a normal distribution. As the mean of the simulation seems closer to 175, it also shows the **Law of Large Numbers**.

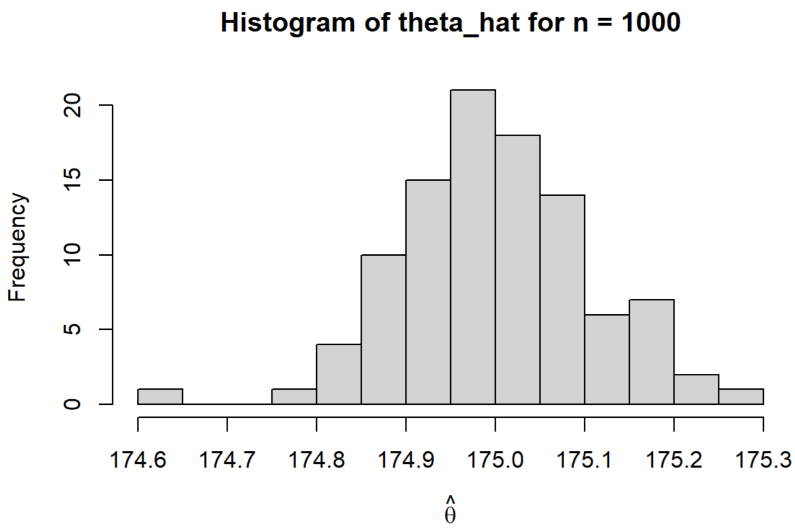
Appendix

My *R* code for the problem set:

```
# Parameters
theta <- 175
n <- 10
# Simulation
u_i <- runif(n, min = -5, max = 5)  # ui ~ U(-5, 5)
x_i <- theta + u_i
theta_hat <- mean(x_i)
print(paste("Estimated theta:", theta_hat))

***Then, we only need to change n from 10 to 100 and 1000 to estimate
    the theta_hat for theses two conditions.

# Function to simulate LS estimator
simulate_theta_hat_10 <- function(theta, M) {
  theta_hats <- numeric(M)
  for (i in 1:M) {
    u_i <- runif(10, min = -5, max = 5)  # ui ~ U(-5, 5)
    x_i <- theta + u_i
    theta_hats[i] <- mean(x_i)
  }
  return(theta_hats)
}
# Parameters
M <- 100
# Simulate and plot for n = 10
theta_hats_10 <- simulate_theta_hat_10(theta, M)
hist(theta_hats_10, breaks = 20, main = "Histogram of theta_hat for n =
    10", xlab = expression(hat(theta)), ylab = "Frequency")

***Then, we just need to change n=10 to n=100 and n=1000, other codes
    stay the same.
```


Histogram of theta_hat for n = 10
Frequency
0
2
4
6
8
10
172
173
174
175
176
177
$\hat{\theta}$


Histogram of theta_hat for n = 100
Frequency
0
2
4
6
8
174.4
174.6
174.8
175.0
175.2
175.4
175.6
$\hat{\theta}$


Histogram of theta_hat for n = 1000
Frequency
0
5
10
15
20
174.6
174.7
174.8
174.9
175.0
175.1
175.2
175.3
$\hat{\theta}$