M50 Homework 2

Alex Craig

Exercise 1.

(Computing conditional averages): Consider a random variable $Y = (Y_1, Y_2)$ which takes values in the sample space:

$$S = \mathbb{N} \times \mathbb{N} = (i, j), i, j \in \mathbb{N}$$

That is, the sample space consists of all possible pairs of numbers (i, j). Now suppose we have some data:

$$(1, 2), (1, 2), (3, 1), (1, 4), (3, 3), (2, 2), (1, 5)$$

Give you best estimates of the following (either by hand, with Python, or a calculator)

$$E[Y_1], \quad E[Y_1 \mid Y_2 = 2], \quad E[Y_2 \mid Y_1 = 1], \quad E[Y_2 \mid Y_1 > 1]$$

Solution

$$E[Y_1] \approx \frac{1+1+3+1+3+2+1}{7} = \frac{12}{7}$$

$$E[Y_1 \mid Y_2 = 2] \approx \frac{1+1+2}{3} = \frac{4}{3}$$

$$E[Y_2 \mid Y_1 = 1] \approx \frac{2+2+4+5}{4} = \frac{13}{4}$$

$$E[Y_2 \mid Y_1 > 1] \approx \frac{1+3+2}{3} = 2$$

Exercise 2.

(Independence and conditional expectation): Let X and Y be two random variables with sample spaces S_X and S_Y.

Part A

Prove that if X and Y are independent $E[X \mid Y = y] = E[X]$ and $E[Y \mid X = x] = E[Y]$ for all $x \in S_X$ and $y \in S_Y$.

Solution

We know that conditional expectation is defined as:

$$E[X \mid Y = y] = \sum_{x \in S_X} x \cdot \mathbb{P}(X = x \mid Y = y)$$

Keep in mind that conditional probability is defined as:

$$\mathbb{P}(X = x \mid Y = y) = \frac{\mathbb{P}(X = x, Y = y)}{\mathbb{P}(Y = y)}$$

For independent random variables, we know that:

$$\mathbb{P}(X = x, Y = y) = \mathbb{P}(X = x) \cdot \mathbb{P}(Y = y)$$

Therefore, if X and Y are independent we may rewrite the conditional probability as:

$$\mathbb{P}(X = x \mid Y = y) = \frac{\mathbb{P}(X = x) \cdot \mathbb{P}(Y = y)}{\mathbb{P}(Y = y)} = \mathbb{P}(X = x)$$

Therefore, if X and Y are independent we may rewrite the conditional expectation as:

$$E[X \mid Y=y] = \sum_{x \in S_X} x \cdot \mathbb{P}(X=x \mid Y=y) = \sum_{x \in S_X} x \cdot \mathbb{P}(X=x) = E[X]$$

$$\Rightarrow E[X \mid Y=y] = E[X]$$

Part B

Prove the tower property of expectation that is stated in the class notes.

Solution

The notes state the tower property as:

$$E[E[X \mid Y]] = E[X]$$

Let's prove this in the discrete case:

$$E[E[X \mid Y]] = \sum_{y \in S_Y} E[X \mid Y=y] \cdot \mathbb{P}(Y=y)$$

We know that:

$$E[X \mid Y=y] = \sum_{x \in S_X} x \cdot \mathbb{P}(X=x \mid Y=y)$$

Therefore, we may rewrite the above as:

$$E[E[X \mid Y]] = \sum_{y \in S_Y} \sum_{x \in S_X} x \cdot \mathbb{P}(X=x \mid Y=y) \cdot \mathbb{P}(Y=y)$$

We know that:

$$\mathbb{P}(X=x \mid Y=y) \cdot \mathbb{P}(Y=y) = \mathbb{P}(X=x, Y=y)$$

Therefore, we may rewrite the above as:

$$E[E[X \mid Y]] = \sum_{y \in S_Y} \sum_{x \in S_X} x \cdot \mathbb{P}(X=x, Y=y)$$

When we sum over all y in S_Y, we marginalize out Y and are left with:

$$E[E[X \mid Y]] = \sum_{x \in S_X} x \cdot \mathbb{P}(X=x) = E[X]$$

Part C

Prove that if X and Y are independent, then $Var(X+Y) = Var(X) + Var(Y)$.

Hint: Use this formula for variance: $Var(X) = E[X^2] - E[X]^2$.

Solution

$$Var(X+Y) = E[(X+Y)^2] - E[X+Y]^2 = E[X^2 + 2XY + Y^2] - (E[X] + E[Y])^2$$

Using the linearity of expectation, we may rewrite the above as:

$$= E[X^2] + 2E[XY] + E[Y^2] - (E[X]^2 + 2E[X]E[Y] + E[Y]^2)$$

If X and Y are independent, then $E[XY] = E[X]E[Y]$. Therefore, we may rewrite the above as:

$$= E[X^2] + 2E[X]E[Y] + E[Y^2] - (E[X]^2 + 2E[X]E[Y] + E[Y]^2)$$

$$= E[X^2] - E[X]^2 + E[Y^2] - E[Y]^2 = Var(X) + Var(Y)$$

Exercise 3.

(Aspects of the binomial distribution): Suppose Y_1 and Y_2 are two independent binomial distributions:

$$Y_1 \sim Bin(n_1, p_1), \quad Y_2 \sim Bin(n_2, p_2)$$

with $n_1, n_2 \in \mathbb{N}$ and $p_1, p_2 \in (0, 1)$

Part A

If $p = p_1 = p_2$, what is the distribution of $Y_1 + Y_2$?

Solution

Given that $p = p_1 = p_2$, we may define the probability mass function of Y_k as:

$$\mathbb{P}(Y_k = y) = \binom{n_k}{y} p^y (1-p)^{n_k - y}, \quad y = 0, 1, \ldots, n_k, \quad k = 1, 2$$

Define $Y = Y_1 + Y_2$. We may then define the probability mass function of Y as:

$$\mathbb{P}(Y = y) = \mathbb{P}(Y_1 + Y_2 = y) = \sum_{i=0}^{y} \mathbb{P}(Y_1 = i, Y_2 = y - i)$$

Since Y_1 and Y_2 are independent, we may rewrite the above as:

$$\sum_{i=0}^{y} \mathbb{P}(Y_1 = i, Y_2 = y - i) = \sum_{i=0}^{y} \mathbb{P}(Y_1 = i) \cdot \mathbb{P}(Y_2 = y - i)$$

And now we may substitute in the probability mass function of Y_1 and Y_2:

$$\mathbb{P}(Y = y) = \sum_{i=0}^{y} \binom{n_1}{i} p^i (1-p)^{n_1 - i} \cdot \binom{n_2}{y-i} p^{y-i} (1-p)^{n_2 - y + i}$$

$$\mathbb{P}(Y = y) = \sum_{i=0}^{y} \binom{n_1}{i} \binom{n_2}{y-i} p^y (1-p)^{n_1 + n_2 - y}$$

$$\mathbb{P}(Y = y) = p^y (1-p)^{n_1 + n_2 - y} \sum_{i=0}^{y} \binom{n_1}{i} \binom{n_2}{y-i}$$

And, by Vandermonde's Identity:

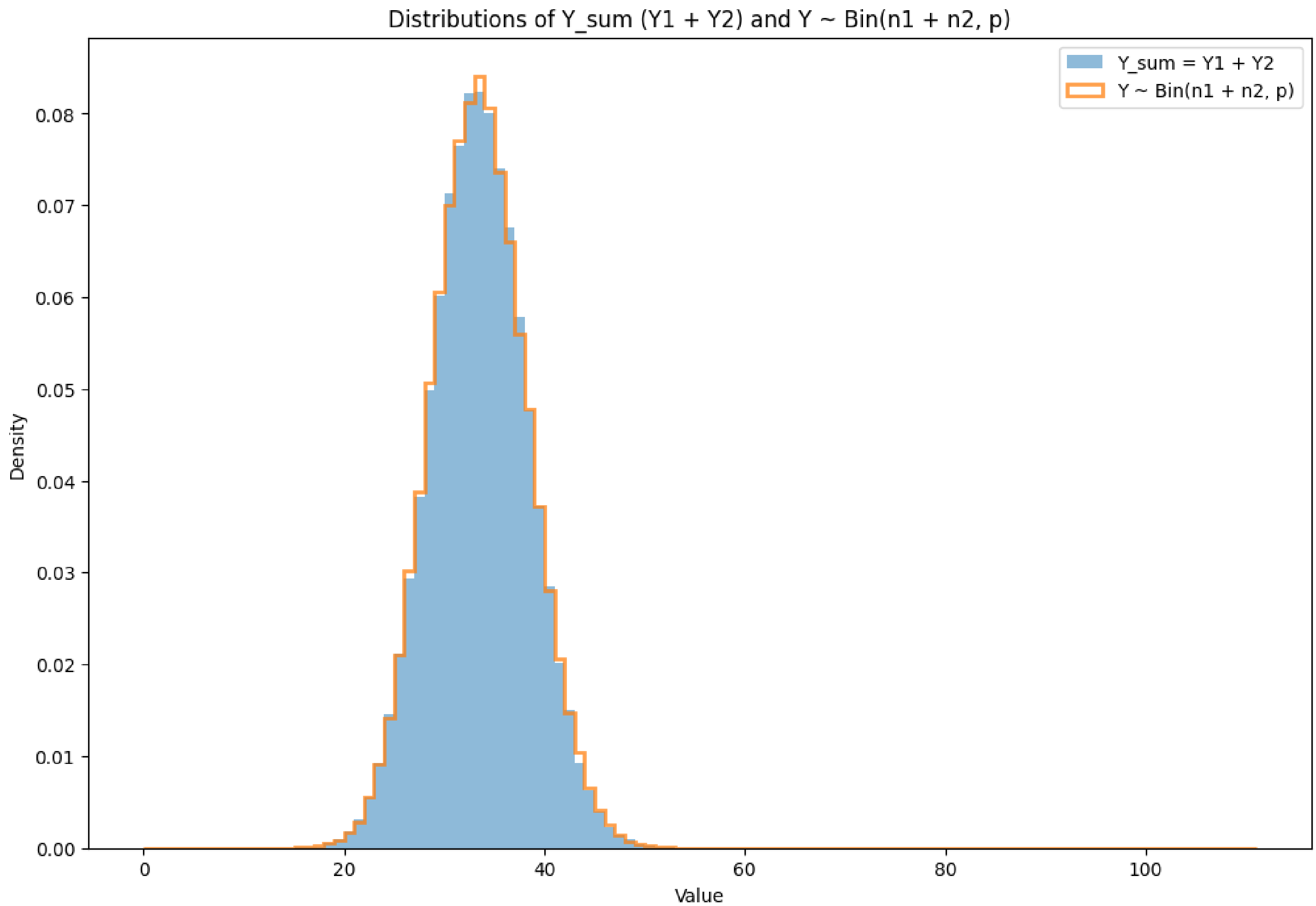
$$\mathbb{P}(Y = y) = p^y(1-p)^{n_1+n_2-y}\binom{n_1+n_2}{y}$$

From this probability mass function, we may conclude that $Y \sim Bin(n_1 + n_2, p)$.

Part B

Confirm you answer to part (a) with simulations with $n_1 = 100$, $n_2 = 10$ and $p = 0.3$.

```python
import numpy as np
import matplotlib.pyplot as plt

# Parameters
n1 = 100
n2 = 10
p = 0.3

# Number of simulations
num_simulations = 100000

# Simulating Y1 and Y2
Y1 = np.random.binomial(n1, p, num_simulations)
Y2 = np.random.binomial(n2, p, num_simulations)

# Y = Y1 + Y2
Y_sum = Y1 + Y2

# Simulating Y
Y = np.random.binomial(n1 + n2, p, num_simulations)

# Plotting histograms
plt.figure(figsize=(12, 8))

plt.hist(
    Y_sum, bins=range(n1 + n2 + 2), alpha=0.5, label="Y_sum = Y1 + Y2", density=True
)
plt.hist(
    Y,
    bins=range(n1 + n2 + 2),
    alpha=0.75,
    label="Y ~ Bin(n1 + n2, p)",
    density=True,
    histtype="step",
    linewidth=2,
)

plt.xlabel("Value")
plt.ylabel("Density")
plt.title("Distributions of Y_sum (Y1 + Y2) and Y ~ Bin(n1 + n2, p)")
plt.legend()

plt.show()
```



Part C

Now suppose $p_1 \neq p_2$. Let

$$Y_3 \sim Bin(n_1 + n_2, \frac{n_1}{n_1+n_2}p_1 + \frac{n_2}{n_1+n_2}p_2)$$

Here is an **erroneous** argument for why Y_3 might have the same distribution as $Y_1 + Y_2$ (it doesn't!):

$Y_1 + Y_2$ is the sum of n Bernoulli random variables. Denote these as $X_1, X_2, ..., X_n$ where $n = n_1 + n_2$. Assume they are in order, so that the first n_1 terms are the Bernoulli random variables corresponding to the first binomial distribution (the one with success probability p_1). Note that with this notation we are not specifying whether X_i comes from the first or second Bernoulli sequence. If we randomly select one of these, X_i, then the chance it is equal to 1 is:

$$\mathbb{P}(X_i = 1) = \mathbb{P}(X_i = 1 | i \leq n_1)\mathbb{P}(i \leq n_1) + \mathbb{P}(X_i = 1 | i > n_1)(i > n_1)$$

Now observe that:

$$\mathbb{P}(i \leq n_1) = \frac{n_1}{(n_1+n_2)}, \quad \mathbb{P}(i > n_1) = \frac{n_2}{(n_1+n_2)}$$
$$\mathbb{P}(X_i = 1 \mid i \leq n_1) = p_1, \quad \mathbb{P}(X_i = 1 \mid i > n_1) = p_2$$

Plugging these into the formula for $\mathbb{P}(X_i = 1)$ gives the probability of success in the definition of Y_3.

Part D

Explain why this argument above is flawed. **Hint:** Is the variable X_i and X_j independent for all $i \neq j$? If not, why is independence important?

Solution

The primary flaw in the argument is its treatment of the combined sequence of Bernoulli variables from Y_1 and Y_2 as a set of independent Bernoulli trials. However, this is misleading as within each binomial distribution, Y_1 and Y_2, the Bernoulli trials are indeed independent. But when combining trials from both Y_1 and Y_2 into one sequence, the trials are no longer independent in the context of the entire sequence. Specifically, a trial X_i originating from Y_1 has a success probability of p_1, while a trial X_j from Y_2 has a different success probability of p_2. If we randomly select a trial from the combined sequence, knowing which binomial distribution it comes from immediately informs us about its success probability, illustrating the dependence.

In essence, the argument mistakenly assumes that the trials from the two different binomial distributions are interchangeable and independent when considered in a unified sequence, which is not the case.

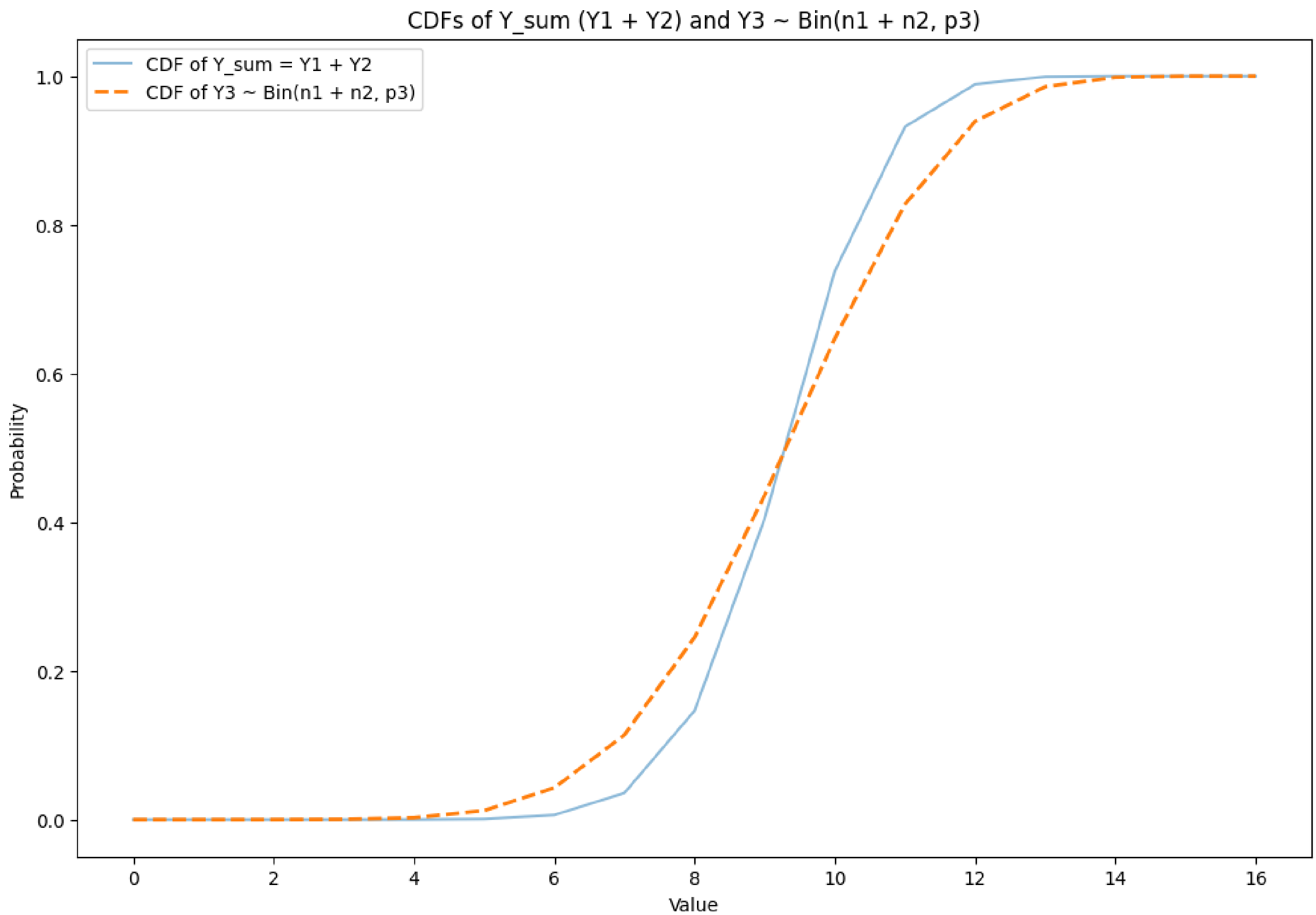
Part E

Confirm that the argument above is incorrect using simulations, that is, confirm via simulations of an example that Y_3 does not have the same probability distribution as $Y1 + Y2$. You can do this many ways, for example, by plotting $\mathbb{P}(Y_3 > k)$ as a function of k and comparing to $\mathbb{P}(Y_1 + Y_2 > k)$.

Solution

```python
import numpy as np
import matplotlib.pyplot as plt

# Parameters
n1 = 10
n2 = 5
p1 = 0.9
p2 = .15

# Number of simulations
num_simulations = 100000

# Simulating Y1 and Y2
Y1 = np.random.binomial(n1, p1, num_simulations)
Y2 = np.random.binomial(n2, p2, num_simulations)

# Y = Y1 + Y2
Y_sum = Y1 + Y2

# Weighted average probability for Y3
p3 = (n1 * p1 + n2 * p2) / (n1 + n2)

# Simulating Y3
Y3 = np.random.binomial(n1 + n2, p3, num_simulations)

# Calculate CDFs
values = range(n1 + n2 + 2)
cdf_Y_sum = [np.mean(Y_sum <= v) for v in values]
cdf_Y3 = [np.mean(Y3 <= v) for v in values]

# Plotting CDFs
plt.figure(figsize=(12, 8))

plt.plot(values, cdf_Y_sum, label="CDF of Y_sum = Y1 + Y2", alpha=0.5)
plt.plot(values, cdf_Y3, label="CDF of Y3 ~ Bin(n1 + n2, p3)", linestyle='--', linewidth=2)

plt.xlabel("Value")
plt.ylabel("Probability")
plt.title("CDFs of Y_sum (Y1 + Y2) and Y3 ~ Bin(n1 + n2, p3)")
plt.legend()
```

```
plt.show()
```



Exercise 4.

(Election modeling): Suppose again that we are interested in predicting the outcome of an election with two candidates and N voters. Based on or polling data, people's preferences are equally split between the two candidates ($q = \frac{1}{2}$). However, there is one particular person – person 1 – who is particularly influential. If person 1 votes for candidate one, then everyone else votes for candidate one, while if person 1 votes for candidate two, everyone sticks with their original preference.

The total vote for candidate two can be written as:

$$Y = \sum_{i=1}^{N} y_i \times y_1$$

where

$$y_i \sim Bernoulli(\frac{1}{2}), \quad i = 1, ..., N$$

Each y_i is 0 if they vote for candidate one and 1 if they vote for candidate two. Y_1 represents the vote of the very influential person. The code below simulates this model.

```
def sample_Y (N, n_samples):
    Y = np.zeros(n_samples)

    for i in range (n_samples):
        y_sample = np.random.choice([0,1], N)
```

```
        Y[i] = np.sum(y_sample) * y_sample[0]

    return Y
```

Part A

Let ϕ denote the fraction of votes for candidates one. How do you think the CV (coefficient of variation) of ϕ depends on N as N becomes large? Test your hypothesis with simulations.

Solution

Remember that the coefficient of variation is defined as:

$$CV = \frac{\sigma}{\mu}$$

Where σ is the standard deviation and μ is the mean.

ϕ is the fraction of votes for candidates one. We may define ϕ as:

$$\phi = \frac{N - Y}{N}$$

We know that Y is following a distribution where it had a $\frac{1}{2}$ chance of being 0 and a $\frac{1}{2}$ chance of being a binomial random variable with $N-1$ trials and a probability of success of $\frac{1}{2}$ (because the votes are independent bernoulli events) plus the additional vote from person one.

Let's then find the mean Y, denoted μ_Y.

$$\mu_Y = E[Y] = E[Y \mid y_1 = 0] \cdot \mathbb{P}(y_1 = 0) + E[Y \mid y_1 = 1] \cdot \mathbb{P}(y_1 = 1)$$

Remember that $Y \mid y_1 = 0$ is always 0, so $E[Y \mid y_1 = 0] = 0$.

$$\mu_Y = 0 \cdot \frac{1}{2} + E[Y \mid y_1 = 1] \cdot \frac{1}{2}$$

Remember that $Y \mid y_1 = 1$ is a binomial random variable with $N-1$ trials and a probability of success of $\frac{1}{2}$, plus an additional vote from person one. Therefore

$$E[Y \mid y_1 = 1] = 1 + \frac{N-1}{2} = \frac{N+1}{2}$$

Therefore:

$$\mu_Y = 0 \cdot \frac{1}{2} + \frac{N+1}{2} \cdot \frac{1}{2} = \frac{N+1}{4}$$

We may now find the standard deviation of Y, denoted σ_Y:

$$\sigma_Y = \sqrt{Var(Y)} = \sqrt{E[Y^2] - E[Y]^2}$$

We already know that $E[Y] = \frac{N+1}{4}$, so $E[Y]^2 = \frac{(N+1)^2}{16}$. We may now find $E[Y^2]$:

$$E[Y^2] = E[Y^2 \mid y_1 = 0] \cdot \mathbb{P}(y_1 = 0) + E[Y^2 \mid y_1 = 1] \cdot \mathbb{P}(y_1 = 1)$$

$E[Y^2 \mid y_1 = 0] = 0$ because $Y \mid y_1 = 0$ is always 0.

Remember $Y \mid y_1 = 1$ is a binomial random variable with $N-1$ trials and a probability of success of $\frac{1}{2}$ plus the additional first vote. We already found that such a random variable has a mean of $\frac{N+1}{2}$. Because the additional first vote doesn't add any variation as it is guaranteed, $Y \mid y_1 = 1$ will only have variance from the binomial random variable and thus it has a variance of $\frac{N-1}{4}$.

Therefore:

$$E[Y^2 \mid y_1 = 1] = Var(Y \mid y_1 = 1) + E[Y \mid y_1 = 1]^2 = \frac{N-1}{4} + \frac{(N+1)^2}{4} = \frac{N^2+3N}{4}$$

We may now find $E[Y^2]$:

$$E[Y^2] = E[Y^2 \mid y_1 = 0] \cdot \mathbb{P}(y_1 = 0) + E[Y^2 \mid y_1 = 1] \cdot \mathbb{P}(y_1 = 1) = 0 \cdot \frac{1}{2} + \frac{N^2+3N}{4} \cdot \frac{1}{2} = \frac{N^2+3N}{8}$$

We may now find σ_Y:

$$\sigma_Y = \sqrt{Var(Y)} = \sqrt{E[Y^2] - E[Y]^2} = \sqrt{\frac{N^2+3N}{8} - \frac{(N-1)^2}{16}}$$

$$\sigma_Y = \sqrt{\frac{2N^2+6N-(N^2-2N+1)}{16}} = \sqrt{\frac{N^2+4N-1}{16}} = \frac{\sqrt{N^2+4N-1}}{4}$$

Now let us find CV_ϕ, by first finding μ_ϕ and σ_ϕ:

$$\mu_\phi = E[\phi] = E[\frac{N-Y}{N}] = \frac{N-E[Y]}{N} = \frac{N - \frac{N+1}{4}}{N}$$

$$\mu_\phi = 1 - \frac{N+1}{4N} = \frac{4N-N-1}{4N} = \frac{3N-1}{4N}$$

σ_ϕ is a bit more complicated. Remember that $Var(aX+b) = a^2 Var(X)$.

$$\sigma_\phi = \sqrt{Var(\phi)} = \sqrt{Var(\frac{N-Y}{N})}$$

In this case N is fixed and Y is our random variable, so $a = -\frac{1}{N}$ and $b = \frac{N}{N} = 1$.

$$\sigma_\phi = \sqrt{(-\frac{1}{N^2})^2 Var(Y)} = \frac{1}{N}\sqrt{Var(Y)} = \frac{1}{N}\sigma_Y$$

$$\sigma_\phi = \frac{1}{N}\frac{\sqrt{N^2+4N-1}}{4} = \frac{\sqrt{N^2+4N-1}}{4N}$$

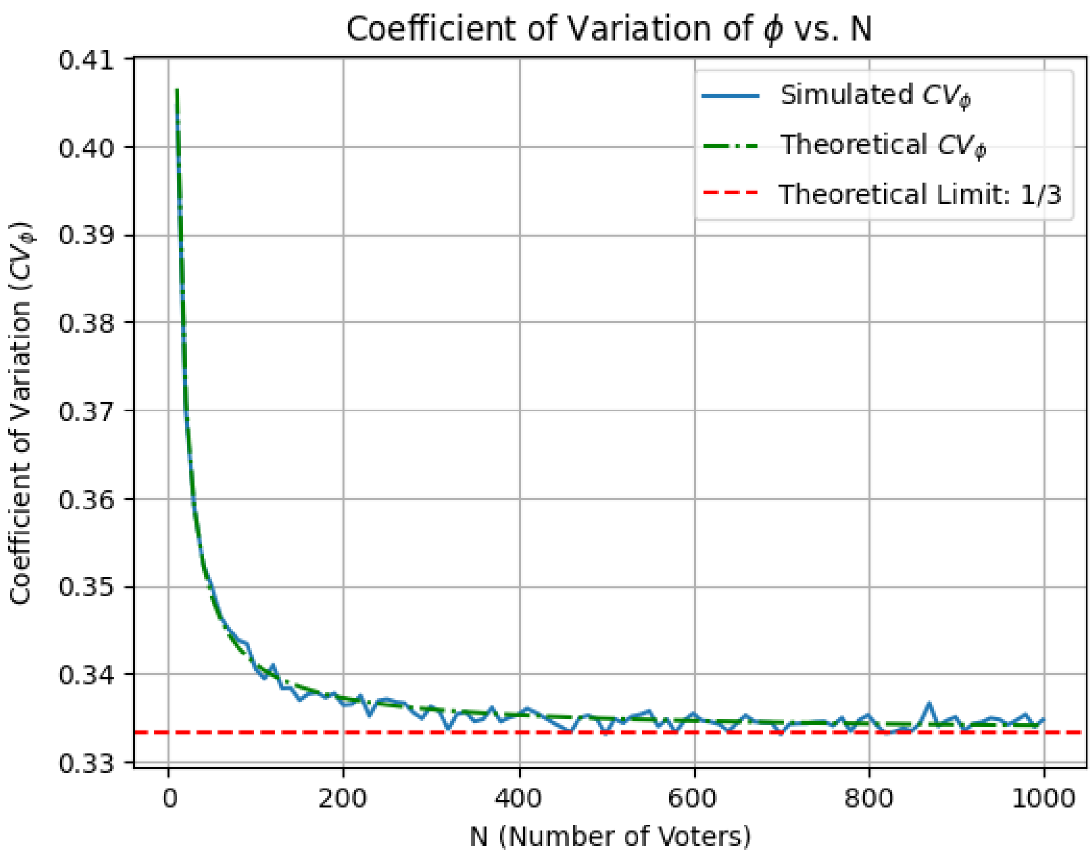
We may now find CV_ϕ:

$$CV_\phi = \frac{\sigma_\phi}{\mu_\phi} = \frac{\frac{\sqrt{N^2+4N-1}}{4N}}{\frac{3N-1}{4N}} = \frac{\sqrt{N^2+4N-1}}{3N-1}$$

We see that as N becomes large, CV_ϕ approaches $\frac{1}{3}$. Let's confirm this with simulations.

```
def coefficient_of_variation_phi(samples, N):
    phi_values = (N - samples) / N
    mean_phi = np.mean(phi_values)
    std_phi = np.std(phi_values)
    return std_phi / mean_phi

def test_hypothesis():
    N_values = np.arange(10, 1001, 10)
    CV_values = []

    n_samples = 30000  # Number of samples for each N
```

```
    for N in N_values:
        samples = sample_Y(N, n_samples)
        CV_phi = coefficient_of_variation_phi(samples, N)
        CV_values.append(CV_phi)

    # Theoretical CV_phi values for the N_values
    theoretical_values = np.sqrt(N_values**2 + 4*N_values - 1) / (3*N_values - 1)

    # Plotting both simulated CV, theoretical line, and theoretical values
    plt.plot(N_values, CV_values, label='Simulated $CV_{\phi}$')
    plt.plot(N_values, theoretical_values, label='Theoretical $CV_{\phi}$', linestyle='-.', color='g')
    plt.axhline(y=1/3, color='r', linestyle='--', label='Theoretical Limit: 1/3')
    plt.xlabel('N (Number of Voters)')
    plt.ylabel('Coefficient of Variation ($CV_{\phi}$)')
    plt.title('Coefficient of Variation of $\phi$ vs. N')
    plt.legend()
    plt.grid(True)
    plt.show()

test_hypothesis()
```



Part B

What are $E[\phi]$ and $E[\phi \mid y_1 = 0]$? Confirm your answers with simulations of the model.

Solution

We have already calculated $E[\phi]$ above.

$$E[\phi] = \mu_\phi = \frac{3N-1}{4N}$$

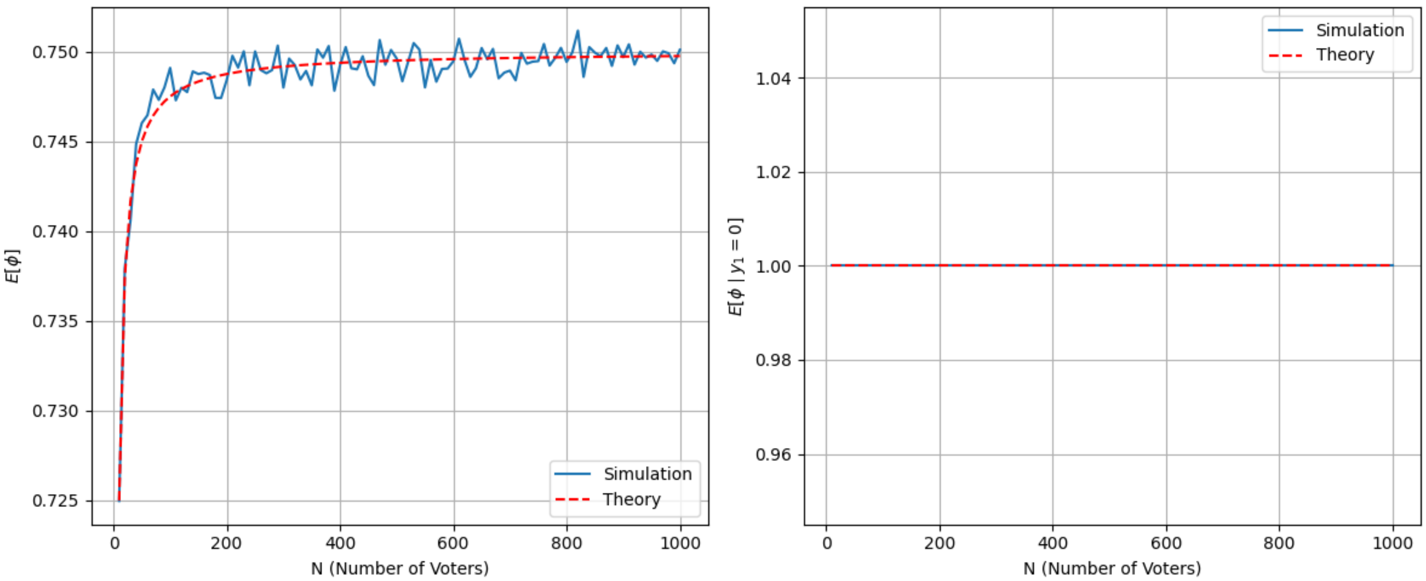
Now let's find $E[\phi \mid y_1 = 0]$:

$$E[\phi \mid y_1 = 0] = E[\frac{N-Y}{N} \mid y_1 = 0] = \frac{N - E[Y \mid y_1 = 0]}{N}$$

We have already calculated $E[Y \mid y_1 = 0] = 0$, above, so:

$$E[\phi \mid y_1 = 0] = \frac{N-0}{N} = 1$$

```python
def compute_phi(samples, N):
    return (N - samples) / N

def simulate_expected_phi():
    N_values = np.arange(10, 1001, 10)
    expected_phi_values = []
    conditional_phi_values = []

    n_samples = 100000  # Number of samples for each N

    for N in N_values:
        samples = sample_Y(N, n_samples)

        # E[phi]
        phi_values = compute_phi(samples, N)
        expected_phi = np.mean(phi_values)
        expected_phi_values.append(expected_phi)

        # E[phi | y_1 = 0]
        conditional_samples = samples[samples == 0]
        conditional_phi = compute_phi(conditional_samples, N)
        conditional_phi_mean = np.mean(conditional_phi) if len(conditional_phi) > 0 else 0
        conditional_phi_values.append(conditional_phi_mean)

    plt.figure(figsize=(12, 5))

    plt.subplot(1, 2, 1)
    plt.plot(N_values, expected_phi_values, label="Simulation")
    plt.plot(N_values, [(3*N - 1)/(4*N) for N in N_values], 'r--', label="Theory")
    plt.xlabel('N (Number of Voters)')
    plt.ylabel('$E[\phi]$')
    plt.legend()
    plt.grid(True)

    plt.subplot(1, 2, 2)
    plt.plot(N_values, conditional_phi_values, label="Simulation")
    plt.plot(N_values, [1 for _ in N_values], 'r--', label="Theory")
    plt.xlabel('N (Number of Voters)')
    plt.ylabel('$E[\phi \mid y_1 = 0]$')
    plt.legend()
    plt.grid(True)

    plt.tight_layout()
    plt.show()

simulate_expected_phi()
```



$E[\phi]$
N (Number of Voters)
Simulation
Theory
$E[\phi \mid y_1 = 0]$
N (Number of Voters)
Simulation
Theory