

# Practice Session 4 Answers

## Part 1: Sampling Distributions

### 1.1 Introduction:

Recall that a “sample” is a subset of a population. We can use samples to calculate “statistics”, such as the sample mean, median, and standard deviation. If we were to **repeatedly** sample from the same population, and calculate the **mean** of each sample, we would obtain what is known as a **Sampling Distribution** of the **Sample Mean**. If we were to instead calculate a **Proportion** from each sample, we would obtain the Sampling Distribution of the **Sample Proportion**.

In general, Sampling Distributions are the distributions of a **statistic**. This is in contrast to the Population, which is the distribution of all individuals or sampling units.

### 1.2 Sampling Distribution Terminology

Match each term with its definition.

#### Terms:

- (a)  $s$
- (b)  $\sigma$
- (c) Standard Error (SE)
- (d)  $x$
- (e)  $\bar{x}$

#### Definitions:

- Standard deviation of a population
- Individual unit of the population
- Standard deviation of a sample
- Sample mean

- Standard deviation of a sampling distribution

## Answers

- (a)  $s$  = Standard deviation of a sample
- (b)  $\sigma$  = Standard deviation of a population
- (c) Standard Error (SE) = Standard deviation of a sampling distribution
- (d)  $x$  = Individual unit of the population
- (e)  $\bar{x}$  = Sample Mean

## 1.3 Generating Sampling Distributions

Let's generate a sampling distribution using polling data from the president's approval rating. Make sure to load the `library(SDS1000)` to have access to the functions for this exercise.

- 1.) Use the `get_approval_sample()` function to obtain a sample of  $n = 1000$  individual approval ratings. Use the `get_proportions()` function to obtain the proportion of voters in this sample that approve of the president.

```
# Load the SDS1000 package
library(SDS1000)

# Setting the seed so that we all get the same "random sample"
set.seed(1000)

# Get a random sample of 1000 fictional voter's opinion of president Trump
approval_sample <- get_approval_sample(1000)

# Get the proportion that approves
p_hat = get_proportion(approval_sample, "approve")
p_hat
```

approve  
0.444

- 2.) Next, use the `do_it` function from class to generate 10000 samples, and repeat the above steps to obtain the proportion for each sample. Assign this vector of proportions to the variable `polling_distribution`, and create a histogram of it. Make sure to have an appropriate title.

**Challenge:** What do values on the x-axis of this histogram correspond to? How does this differ from histograms that we've previously encountered?

```

polling_distribution <- do_it(10000) * {

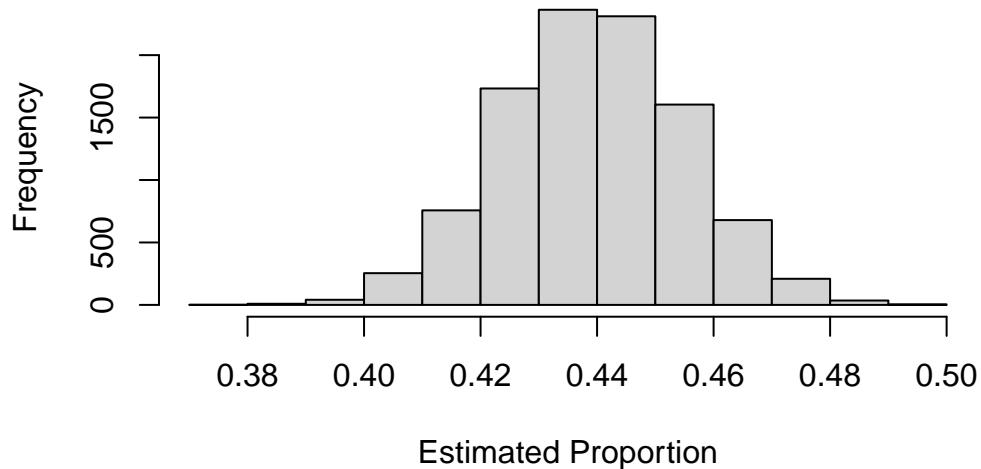
  curr_sample = get_approval_sample(1000)
  get_proportion(curr_sample, "approve")

}

hist(polling_distribution, main = "Distribution of Presidential Approval Ratings",
      xlab = "Estimated Proportion")

```

## Distribution of Presidential Approval Ratings



3.) Finally, calculate the mean and standard deviation of your collection of sample proportions.  
What is another name for the standard deviation?

```
mean(polling_distribution)
```

```
[1] 0.4398881
```

```
sd(polling_distribution)
```

```
[1] 0.01563533
```

## 1.4 Properties of Sampling Distribution

Answer the following True/False questions relating to sampling distributions.

- (a) True/False: As the sample size  $n$  decreases, the standard error of a sampling distribution also decreases.
- (b) True/False: The mean of a sampling distribution is always **greater than** the mean of population.
- (c) True/False: The mean of a sampling distribution is always **less than** the mean of the population.
- (d) True/False: Assuming that you have an unbiased sampling procedure with a sufficient sample size, the sampling distribution of the sample mean is approximately normally distributed.
- (e) True/False: It is generally better to have a larger standard error than a smaller one.

### Answers

- (a) False, the SE typically increases as  $n$  decreases.
- (b) False, assuming the sampling process is unbiased and random, the mean of the sampling distribution will be equal to the mean of the population.
- (c) False, assuming the sampling process is unbiased and random, the mean of the sampling distribution will be equal to the mean of the population.
- (d) True, sampling distributions are typically approximately normal, assuming that your sampling procedure is unbiased and has a sufficient  $n$ .
- (e) False, smaller standard errors are typically preferred.

## Part 2: Introduction to Confidence Interval

Match each term with its definition:

### Terms:

- Point Estimate
- Interval estimate
- Confidence Interval
- Confidence Level

### Definitions:

- (a) An interval that is generated with a specific method that guarantees that a certain percentage of these intervals will contain the population parameter.

- (b) The percentage of confidence intervals that will contain the population parameter of interest. For example, a 95% confidence level means that 95% of confidence intervals generated will contain the population parameter of interest.
- (c) A sample statistic that we use to estimate a population parameter. The sample mean is an example as we use it to estimate the population mean.
- (d) An interval that contains a range of plausible values for a population parameter. The point estimate sits at the center of this interval.

## Answers

- (a) Confidence Interval
- (b) Confidence Level
- (c) Point Estimate
- (d) Interval Estimate

## Part 3 : More Practices on Sampling Distributions

### Practice: Baseball Salaries

For this section, we will use the `BaseballSalaries2015` data set from the `Lock5Data` library.

1.) Using the `sample()` function, obtain a sample of  $n = 100$  baseball player salaries. See the `help` page for `sample()` for details.

```
library(Lock5Data)
set.seed(1000)

salary_sample = sample(BaseballSalaries2015$Salary, size = 100)
```

2.) Now, using the `do_it()` function, repeat this process  $n = 10,000$  times, and calculate the mean each time.

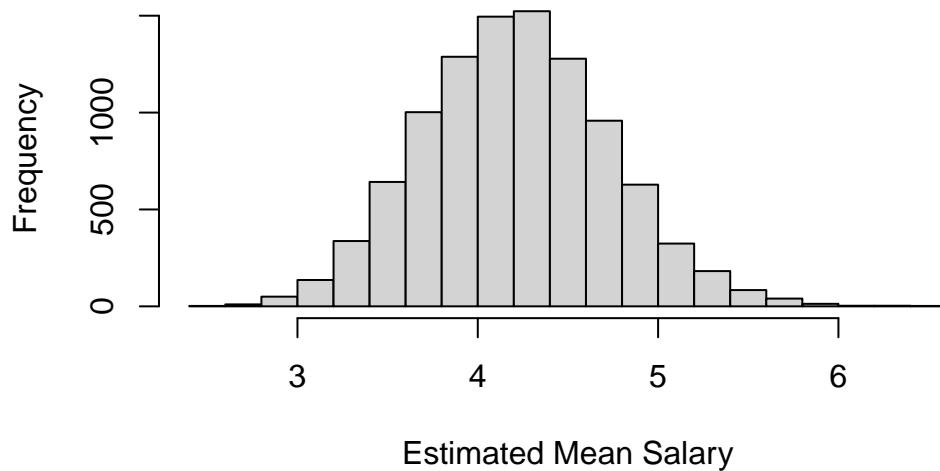
```
salary_distribution <- do_it(10000) * {

  curr_sample = sample(BaseballSalaries2015$Salary, size = 100)
  mean(curr_sample)

}

hist(salary_distribution, main = "Distribution of Salary Samples",
     xlab = "Estimated Mean Salary")
```

## Distribution of Salary Samples



- 3.) Using your original sample of 100 salaries from part 1.), calculate a point estimate of the mean salary.

```
mu_estimate = mean(salary_distribution)
```

- 4.) Calculate the standard deviation of your sampling distribution from part 2.) to estimate the standard error.

```
se = sd(salary_distribution)
```