Practice Session 4

Introduction to Sampling Distributions

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.

Sampling Distribution Terminology

Match each term with its definition.

Terms:

- s
- σ
- Standard Error (SE)
- x
- $\bar{x}$

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) σ = 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

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
  0.586
```

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.

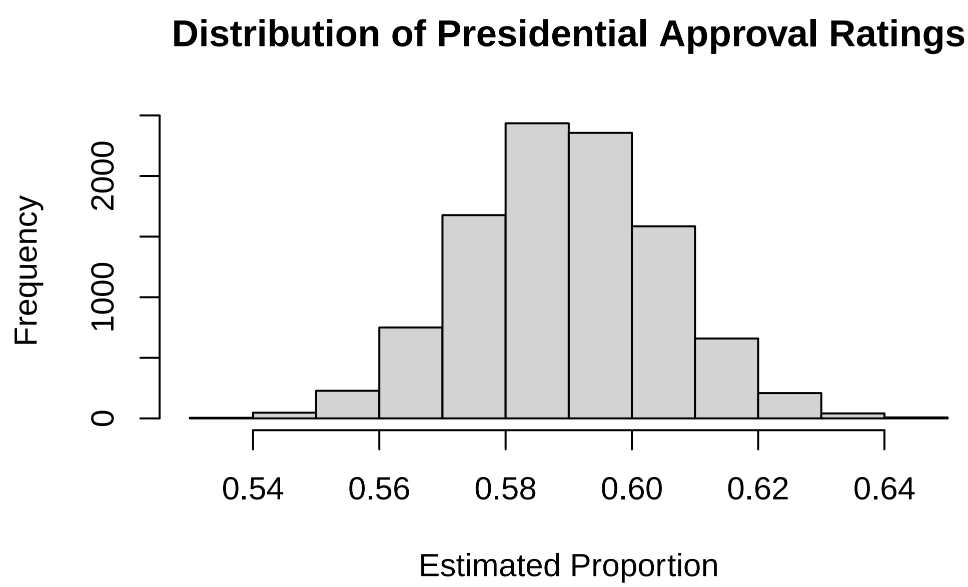
Challenge: What do values on the x-axis of this histogram correspond to? How does this differ from histograms that we've previously encountered?

```r
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")
```



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

```r
mean(polling_distribution)
```

```
[1] 0.5900361
```

```r
sd(polling_distribution)
```

```
[1] 0.01550708
```

Properties of Sampling Distribution

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

- True/False: As the sample size n decreases, the standard error of a sampling distribution also decreases.
- True/False: The mean of a sampling distribution is always **greater than** the mean of population.
- True/False: The mean of a sampling distribution is always is always **less than** the mean of the population.
- 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.
- True/False: It is generally better to have a larger standard error than a smaller one.

Answers

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

Introduction to Confidence Intervals

Match each term with its definition:

Terms:

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

Definitions:

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

- 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.
- 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.
- 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

Confidence Interval for 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. Make sure to sample **without replacement**. See the `help` page for `sample()` for details.

```
library(Lock5Data)
set.seed(1000)
salary_sample = sample(BaseballSalaries2015$Salary, size = 100, replace = F)
```

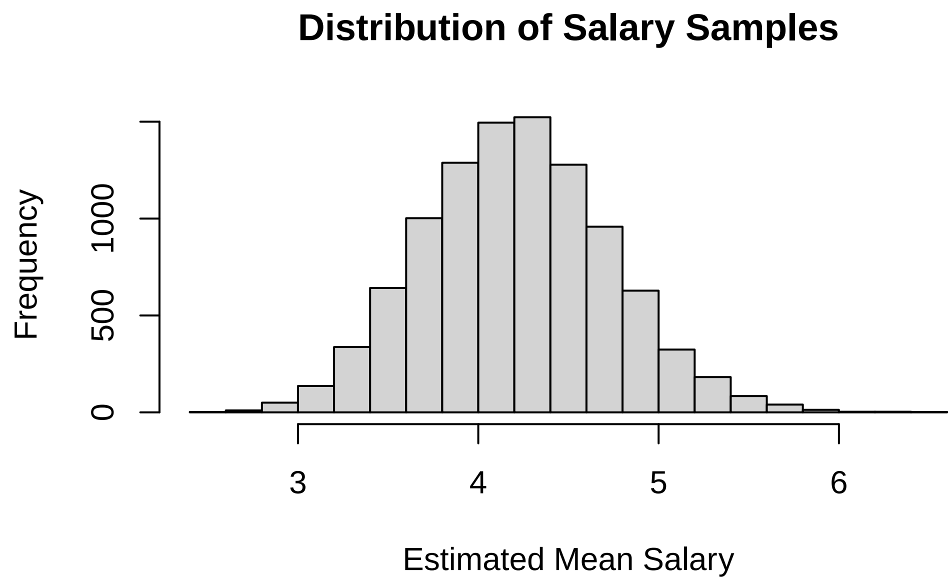
2.) Now, using the `do_it()` function, repeat this process $n = 10,000$ times, and calculate the mean each time.

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

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

}

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



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)
```

5.) Using your answers from part 3.) and 4.), generate a 95% interval estimate for the mean salary of baseball players. The interval estimate is of the form:

Point Estimate $\pm 1.96 \cdot$ Standard Error.

```
int_estimate = c(mu_estimate - 1.96 * se, mu_estimate + 1.96 * se)
int_estimate
```

```
[1] 3.199773 5.224964
```

6.) Interpret your interval estimate.

NOTE: Answers may vary due to randomness in sampling

"The true average salary of baseball players for the 2015 season is likely to lie in the range [Lower-Bound, Upper-Bound], since 95% of the time we use this method the parameter is in such intervals."

7.) Compare your interval estimates to the actual mean salary from the entire data set. Comment on what you notice.

```
mu = mean(BaseballSalaries2015$Salary)
sigma = sd(BaseballSalaries2015$Salary)
mu
```

```
[1] 4.214702
```

We see that our interval contains the true mean salary.