

Confidence Intervals

UCLA Soc 114

Concepts for today

Statistical concepts

- ▶ Sampling distribution
- ▶ Standard error
- ▶ Confidence interval
- ▶ Bootstrap

Coding concepts

- ▶ Writing a custom function
- ▶ Writing a for loop

Example: Mean salary of MLB players

Load data:

```
baseball <- read_csv("https://soc114.github.io/data/baseball.csv")  
# Keep only a few variables for simplicity  
select(player, team, salary)
```

```
# A tibble: 944 x 3
```

	player	team	salary
	<chr>	<chr>	<dbl>
1	Bumgarner, Madison	Arizona	21882892
2	Marte, Ketel	Arizona	11600000
3	Ahmed, Nick	Arizona	10375000

```
# i 941 more rows
```

Example: Mean salary of MLB players

True mean in population of all players

Example: Mean salary of MLB players

True mean in population of all players

```
baseball |> summarize(population_mean = mean(salary))
```

```
# A tibble: 1 x 1  
  population_mean  
      <dbl>  
1      4965481.
```

Estimate from a sample

Estimate from a sample

Draw a sample of 10 players.

Estimate from a sample

Draw a sample of 10 players.

```
sampled_players <- baseball |>  
  slice_sample(n = 10) |>  
  print(n = 3)
```

A tibble: 10 x 3

	player	team	salary
	<chr>	<chr>	<dbl>
1	Tucker, Kyle	Houston	5000000
2	Jackson, Andre	L.A. Dodgers	722500
3	Estrada, Thairo	San Francisco	2250000

i 7 more rows

Estimate from a sample

Take the mean among sampled players.

Estimate from a sample

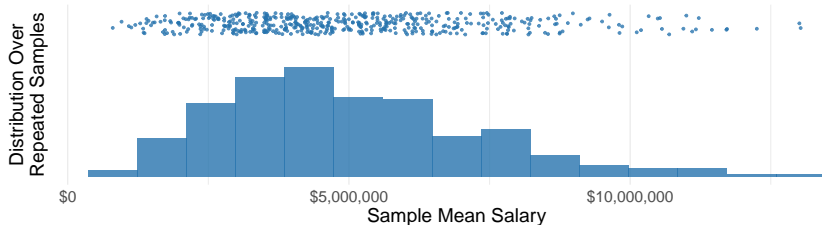
Take the mean among sampled players.

```
sampled_players <- sampled_players |>  
  summarize(sample_estimate = mean(salary)) |>  
  print()
```

```
# A tibble: 1 x 1  
  sample_estimate  
      <dbl>  
1      1536510
```

Many times

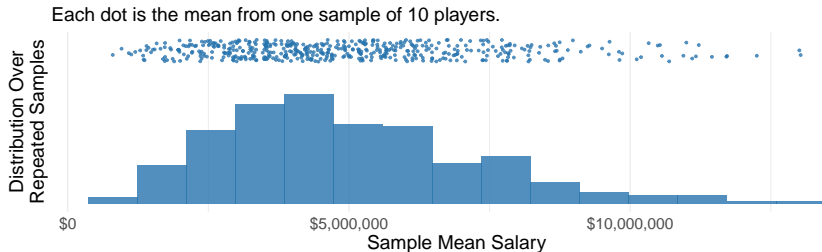
Each dot is the mean from one sample of 10 players.



If you are following, these are in `many_samples.csv`.

```
many_samples <- read_csv(many_samples, "https://soc114.gith
```

Many times

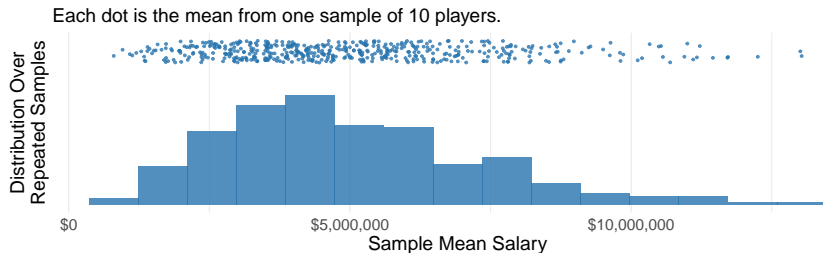


If you are following, these are in `many_samples.csv`.

```
many_samples <- read_csv(many_samples, "https://soc114.github.io/data/many_samples.csv")
```

Because each sample produces a different estimate, there is a **distribution** of different estimates across repeated samples.

Many times



If you are following, these are in `many_samples.csv`.

```
many_samples <- read_csv(many_samples, "https://soc114.github.io/data/many_samples.csv")
```

Because each sample produces a different estimate, there is a **distribution** of different estimates across repeated samples.

Can you propose a summary statistic for this distribution?

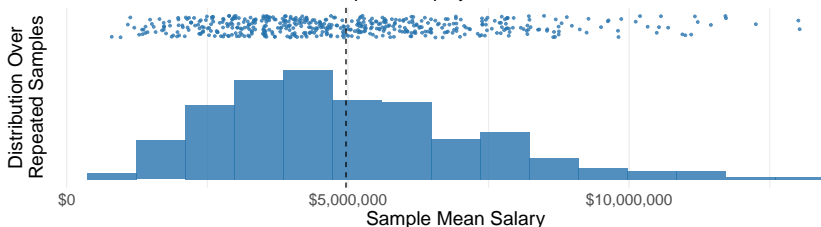
Mean of the distribution

Also called the **expected value**.

```
many_samples |>  
  summarize(estimator_mean = mean(sample_estimate))
```

```
# A tibble: 1 x 1  
  estimator_mean  
      <dbl>  
1    4999932.
```

Each dot is the mean from one sample of 10 players.



(In practice, the mean of the distribution is unknown)

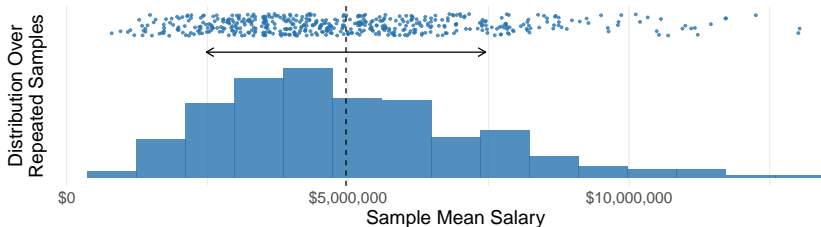
Standard Error

A measure of dispersion for the distribution of sample mean estimates.

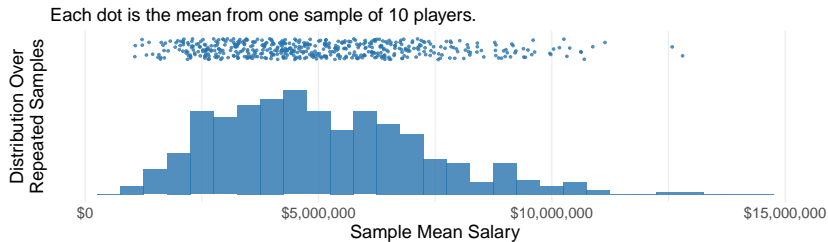
```
many_samples |>  
  summarize(standard_error = sd(sample_estimate))
```

```
# A tibble: 1 x 1  
  standard_error  
    <dbl>  
1      2285513.
```

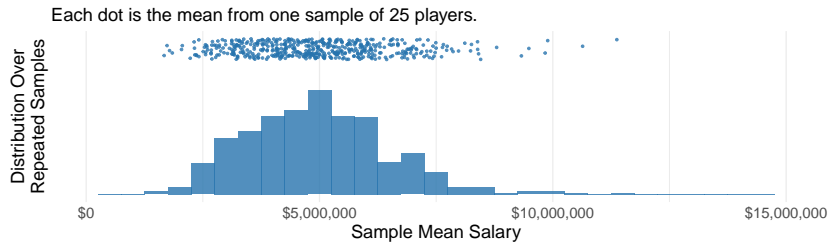
Each dot is the mean from one sample of 10 players.



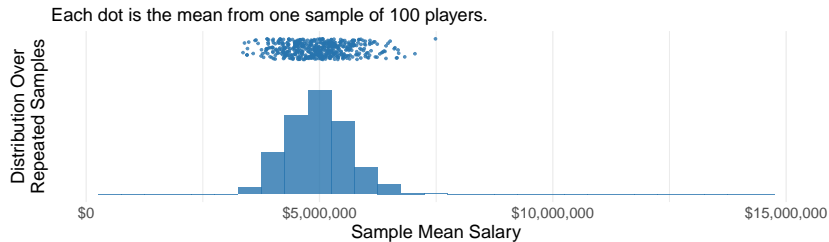
As the sample size grows



As the sample size grows



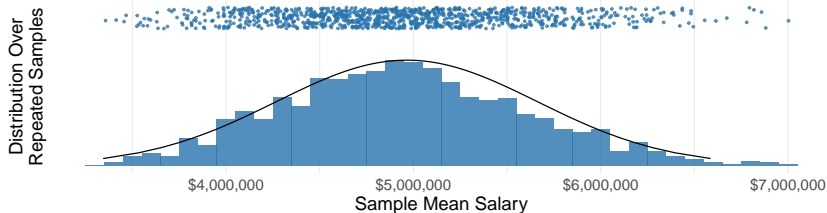
As the sample size grows



Asymptotic Normality

- ▶ As the sample size gets large (asymptotic)
- ▶ This becomes a Normal distribution

Each dot is the mean from one sample of 100 players.

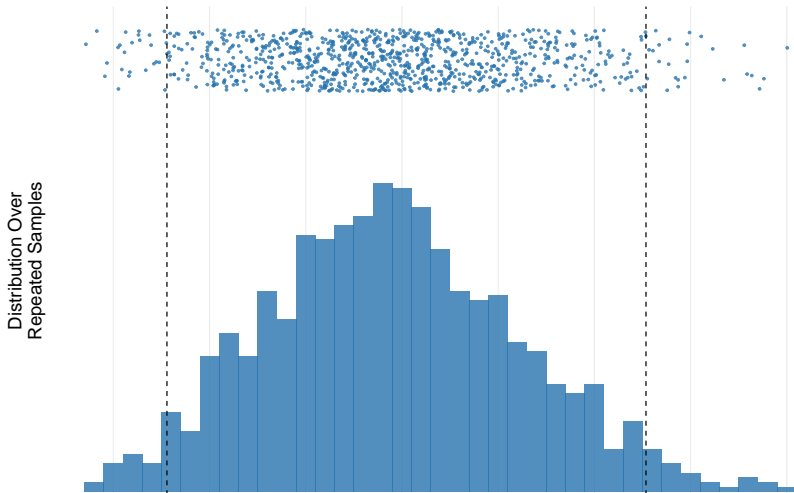


Middle 95% sampling interval

We might want to summarize:

- ▶ The mean of the estimator
- ▶ A range containing the middle 95% of sample estimates

Each dot is the mean from one sample of 100 players.



Confidence interval via the bootstrap

What we want:

1. We would want many samples: `sample_1`, `sample_2`, `sample_3`,...
2. We estimate with each
3. We summarize the middle 95%

Confidence interval via the bootstrap

What we can do:

1. We get only one sample
 - ▶ So we simulate hypothetical `sample_sim_1`, `sample_sim_2`,...
2. We estimate with each
3. We summarize the middle 95%

How to generate bootstrap samples

Start with your one sample.

```
sampled_players <- baseball |>  
  slice_sample(n = 100)
```

How to generate bootstrap samples

Start with your one sample.

```
sampled_players <- baseball |>  
  slice_sample(n = 100)
```

Resample n players with replacement.

```
sampled_players_bootstrap <- sampled_players |>  
  slice_sample(prop = 1, replace = TRUE)
```


How to generate bootstrap samples: Example

Here is a sample of 3 players:

```
a_small_sample <- baseball |>  
  slice_sample(n = 3) |>  
  print()
```

A tibble: 3 x 3

	player <chr>	team <chr>	salary <dbl>
1	Arenado, Nolan	St. Louis	32822071
2	Rodriguez, Christopher*	L.A. Angels	740000
3	Ward, Taylor	L.A. Angels	2750000

How to generate bootstrap samples: Example

Here is a bootstrap sample of those 3 players.

```
a_small_sample |>  
  slice_sample(prop = 1, replace = TRUE) |>  
  print()
```

```
# A tibble: 3 x 3
```

	player <chr>	team <chr>	salary <dbl>
1	Rodriguez, Christopher*	L.A. Angels	740000
2	Ward, Taylor	L.A. Angels	2750000
3	Rodriguez, Christopher*	L.A. Angels	740000

How to generate bootstrap samples: Example

Here is a bootstrap sample of those 3 players.

```
a_small_sample |>  
  slice_sample(prop = 1, replace = TRUE) |>  
  print()
```

```
# A tibble: 3 x 3
```

	player <chr>	team <chr>	salary <dbl>
1	Rodriguez, Christopher*	L.A. Angels	740000
2	Arenado, Nolan	St. Louis	32822071
3	Ward, Taylor	L.A. Angels	2750000

How to generate bootstrap samples: Example

Here is a bootstrap sample of those 3 players.

```
a_small_sample |>
  slice_sample(prop = 1, replace = TRUE) |>
  print()
```

```
# A tibble: 3 x 3
```

	player	team	salary
	<chr>	<chr>	<dbl>
1	Arenado, Nolan	St. Louis	32822071
2	Arenado, Nolan	St. Louis	32822071
3	Ward, Taylor	L.A. Angels	2750000

How to generate bootstrap samples: Example

Here is a bootstrap sample of those 3 players.

```
a_small_sample |>  
  slice_sample(prop = 1, replace = TRUE) |>  
  print()
```

```
# A tibble: 3 x 3
```

	player <chr>	team <chr>	salary <dbl>
1	Ward, Taylor	L.A. Angels	2750000
2	Rodriguez, Christopher*	L.A. Angels	740000
3	Ward, Taylor	L.A. Angels	2750000

Coding concepts

We will analyze hundreds of bootstrap samples.

We need two coding concepts.

1. How to write an estimator function
2. How to write a for loop

How to write an estimator function

A function (like `mean`) takes an input and returns an output. You can write your own.

```
estimator <- function(data) {  
  data |>  
    summarize(estimate = mean(salary)) |>  
    pull(estimate)  
}
```

The function takes data and returns an estimate.

```
estimator(data = sampled_players)
```

```
[1] 4039386
```

How to write a for loop

Useful for tasks you will repeat.

How to write a for loop

Useful for tasks you will repeat.

First, initialize a vector to hold results.

```
vector_for_results <- rep(NA, 3)
```

The rep function repeats the value NA 3 times.

How to write a for loop

Useful for tasks you will repeat.

First, initialize a vector to hold results.

```
vector_for_results <- rep(NA, 3)
```

The rep function repeats the value NA 3 times.

Second, loop through and fill your vector.

```
for (index in 1:3) {  
  vector_for_results[index] <- index  
}
```

Square brackets [] extract an element of a vector.

Analyze 500 bootstrap samples

Initialize a vector to hold the result.

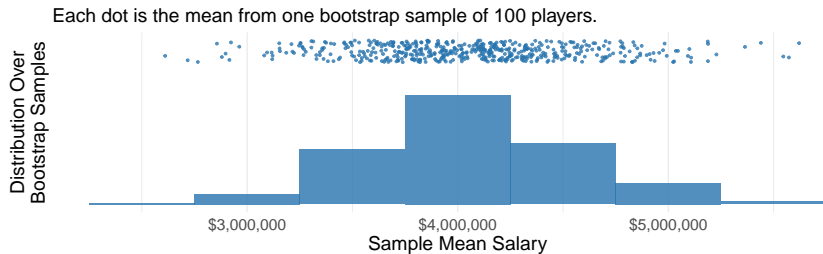
```
bootstrap_estimates <- rep(NA, times = 500)
```

Analyze 500 bootstrap samples

Write a for loop that will repeat 500 times.

```
for (index in 1:500) {  
  
  # Draw a bootstrap sample  
  bootstrap_sample <- sampled_players |>  
    slice_sample(prop = 1, replace = TRUE)  
  
  # Construct an estimate  
  estimate_this_index <- estimator(bootstrap_sample)  
  
  # Store that estimate  
  bootstrap_estimates[index] <- estimate_this_index  
}
```

Bootstrap results



Bootstrap results: Summary statistics

Bootstrap estimate of the standard error.

```
sd(bootstrap_estimates)
```

```
[1] 495720.8
```

Middle 95% of bootstrap estimates

```
quantile(x = bootstrap_estimates, prob = c(.025, .975))
```

2.5%	97.5%
3120228	5070844

Confidence interval

An interval from $\text{lower}(\text{sample})$ to $\text{upper}(\text{sample})$ with the property: across repeated samples, 95% of intervals constructed this way would contain the population parameter.

Confidence interval: Example

Middle 95% of bootstrap estimates is a confidence interval.

- ▶ The true population mean salary is \$4,965,481
- ▶ Our sample mean is \$4,039,386
- ▶ Our confidence interval is:

```
quantile(x = bootstrap_estimates, prob = c(.025, .975))
```

2.5%	97.5%
3120228	5070844

Across repeated samples, 95% of intervals constructed this way will contain the population mean salary.

Recap

- ▶ Statistical concepts
- ▶ Coding concepts

Recap: Statistical concepts

Statistical concepts

- ▶ Sampling distribution
 - ▶ Cannot be directly observed. We have one sample.
- ▶ Standard error
 - ▶ Spread of the sampling distribution
- ▶ Confidence interval
 - ▶ Covers truth in 95% of samples
- ▶ Bootstrap
 - ▶ Method of constructing the CI with one sample

Recap: Coding concepts

- ▶ Writing a custom function (R4DS Ch 25)
- ▶ Writing a for loop (R4DS Ch 27.5)