# Bootstrapping and Confidence Intervals

CS1090A Introduction to Data Science

Pavlos Protopapas, Kevin Rader, and Chris Gumb

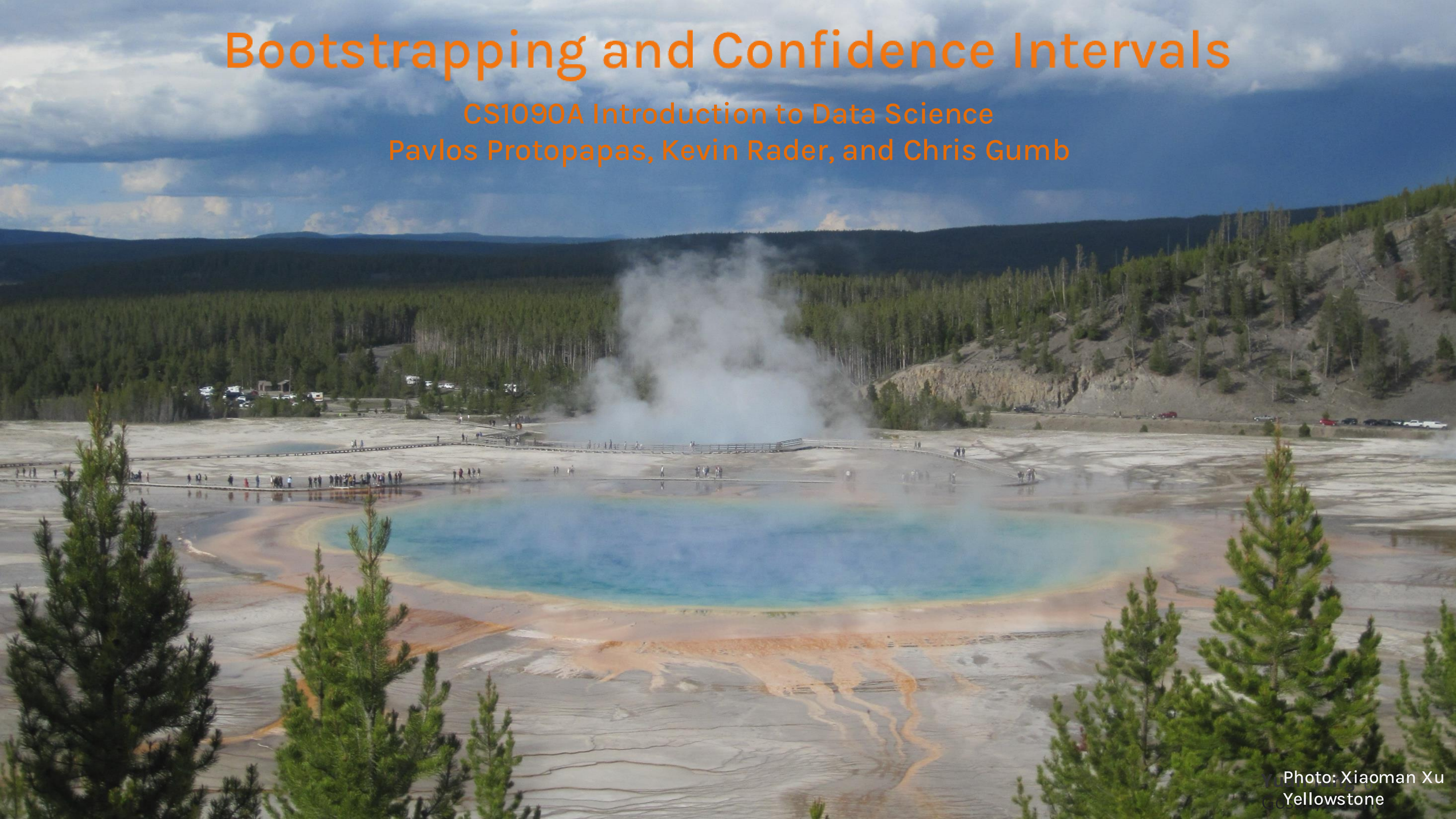


Photo: Xiaoman Xu
Yellowstone

# Outline

**Part A and B: Assessing the Accuracy of the Coefficient Estimates**

Bootstrapping and confidence intervals

**Part C: Evaluating Significance of Predictors**

Does the outcome depend on the predictors?

Hypothesis testing [not]

**Part D: How well do we know $\hat{f}$**

The confidence intervals of $\hat{f}$

# Lack of Active Imagination

In the lack of active imagination, parallel universes and the like, we need an alternative way of producing fake dataset that resemble the parallel universes.

**Bootstrapping** is the practice of sampling from the observed data $(X, Y)$ in estimating statistical properties.

**NOTE:** This is not to create synthetic data to add to the actual observed data. It is to mimic the alternative universes mentioned previously.

# Bootstrap

Imagine we have 5 billiard balls in a bucket.

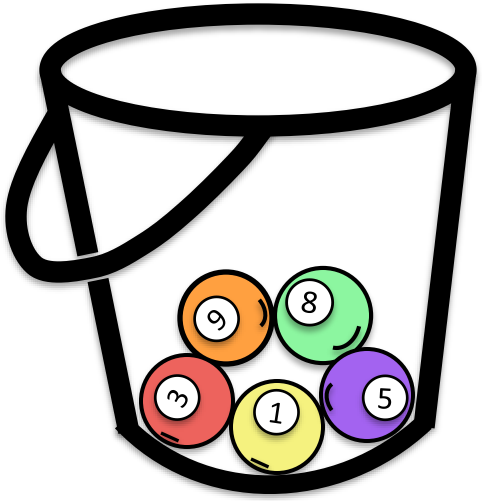
# Bootstrap

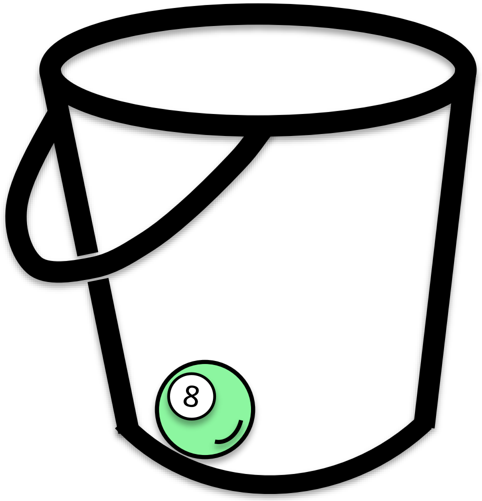
We first pick randomly a ball and replicate it.


This is called **sampling with replacement.**

# Bootstrap

We move the replicated ball to another bucket.



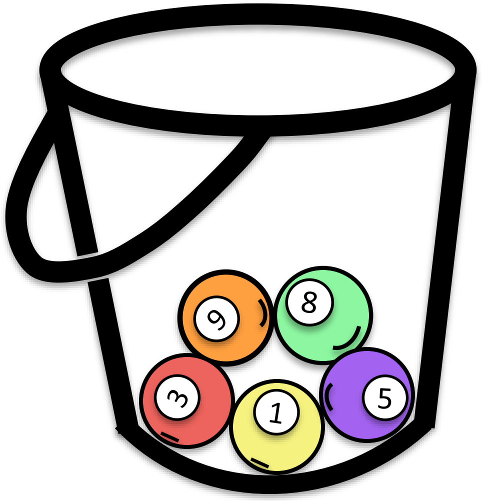



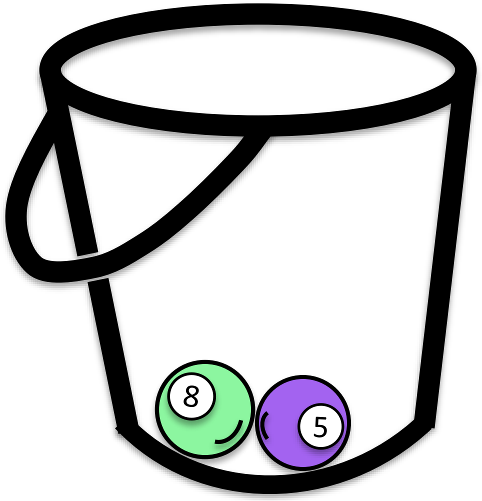
# Bootstrap

We then randomly pick another ball and again we replicate it.

# Bootstrap

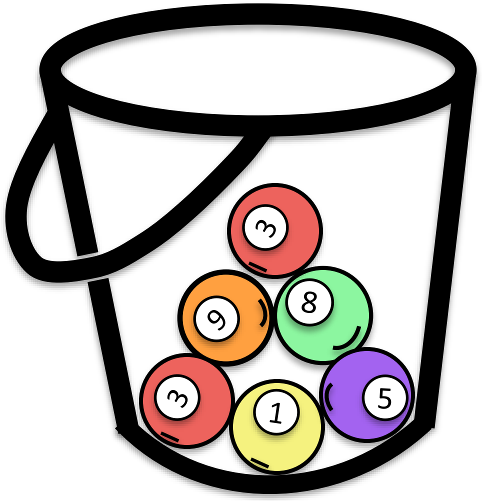
As before, we move the replicated ball to the other bucket.

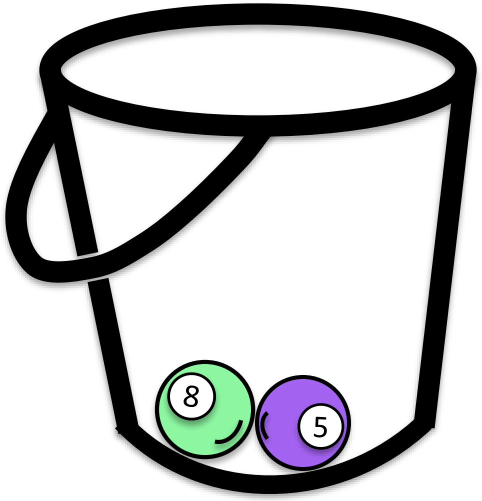


# Bootstrap

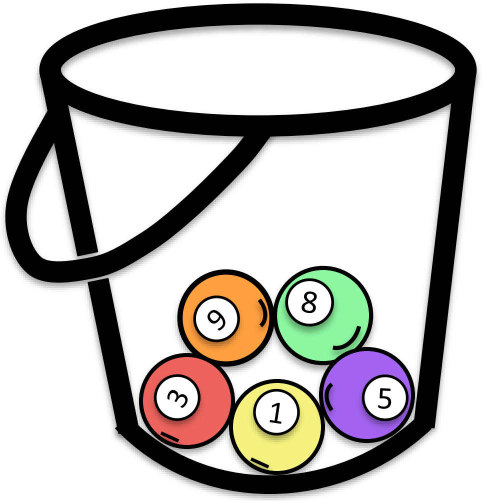
We repeat this process.

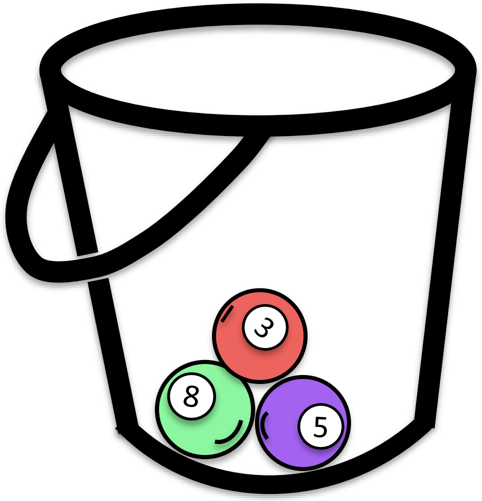


# Bootstrap

We repeat this process.



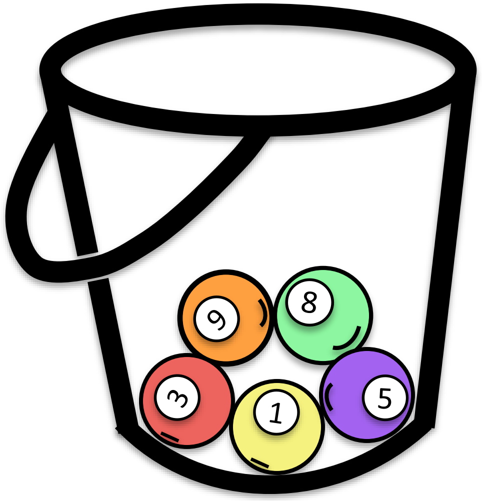



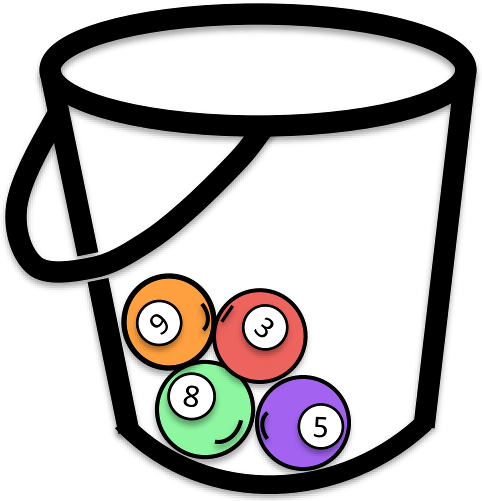
# Bootstrap

We repeat this process.

# Bootstrap

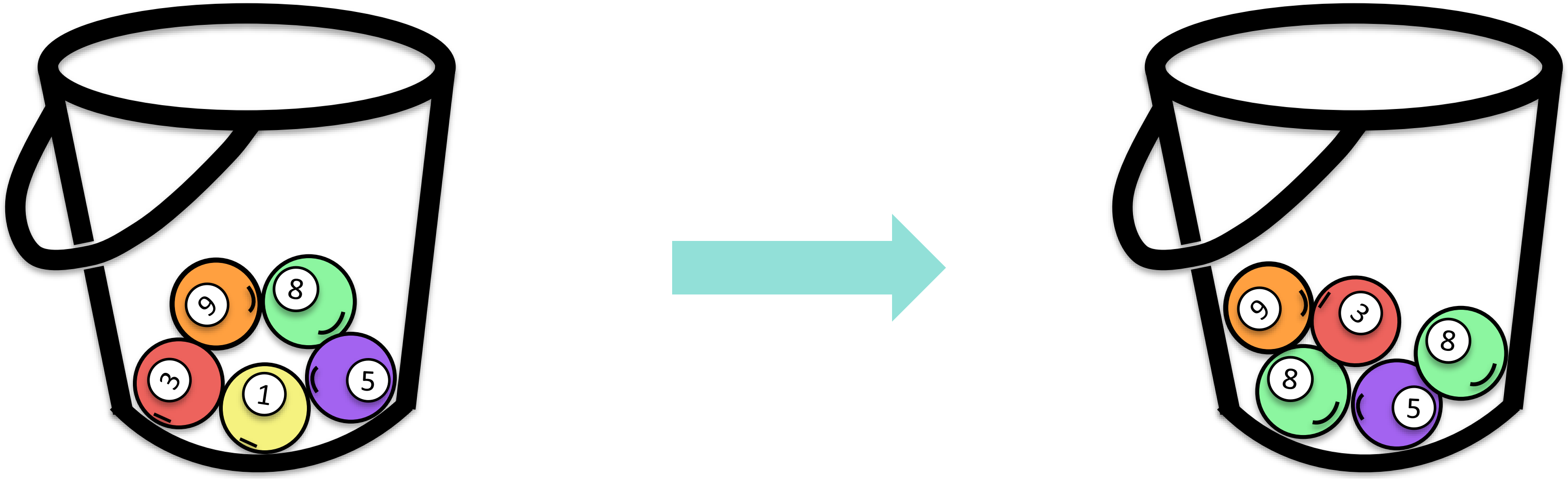
We repeat this process.



# Bootstrap

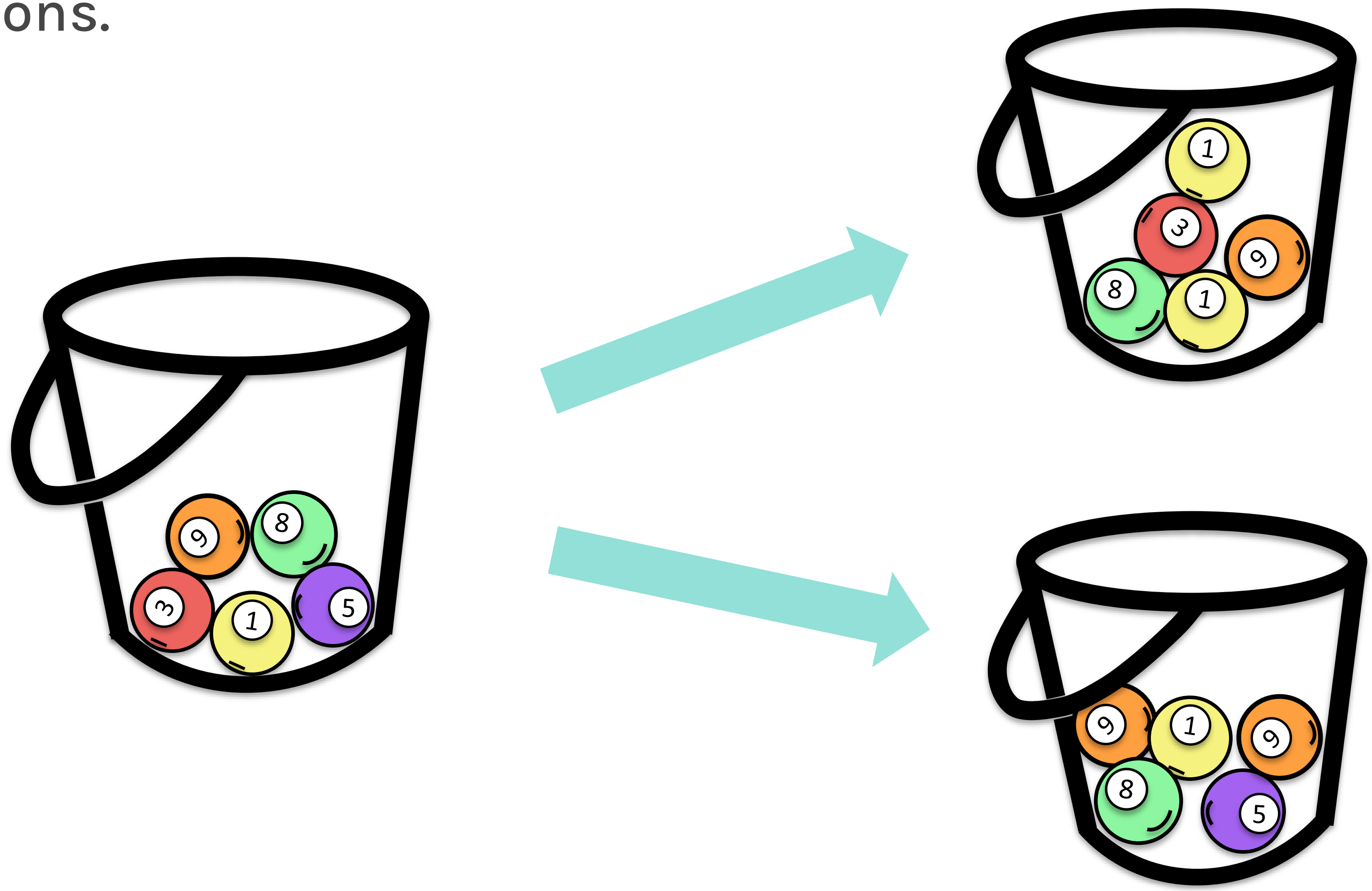
We continue until the “other” bucket has **the same number of balls** as the original one.




**This new bucket represents a new parallel universe**

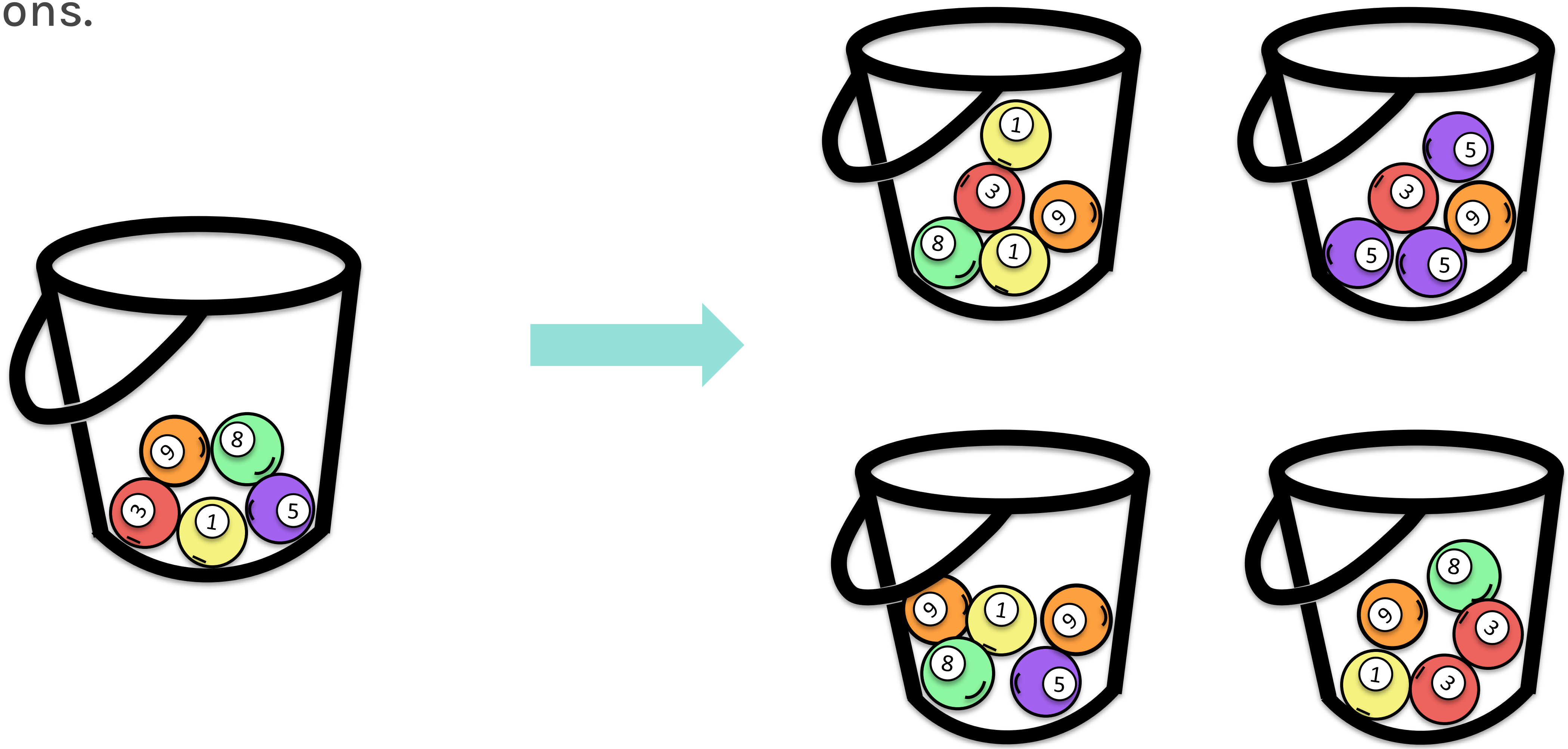
# Bootstrap

We repeat the same process and acquire another set of bootstrapped observations.



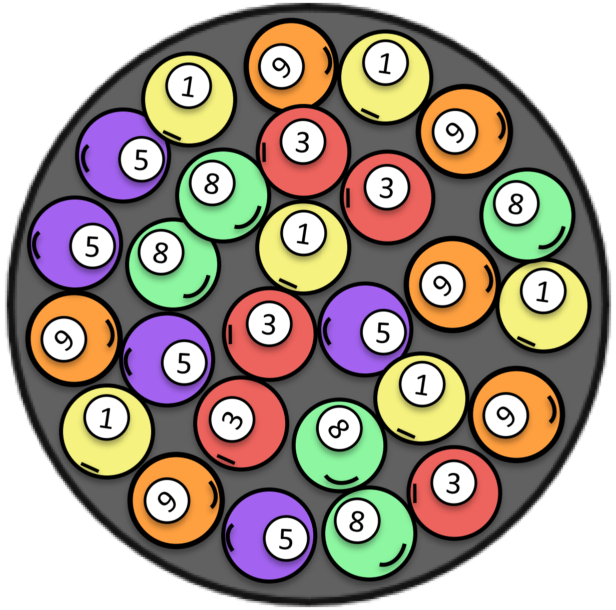
# Bootstrap

We repeat the same process and acquire another set of bootstrapped observations.



# DATASET

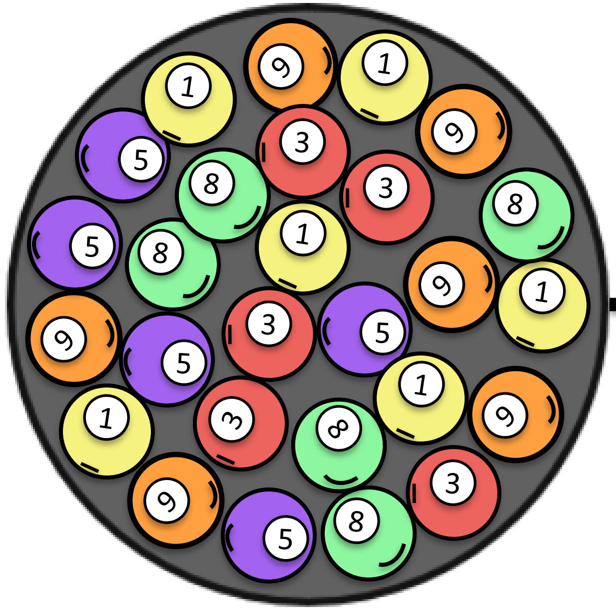
Size N



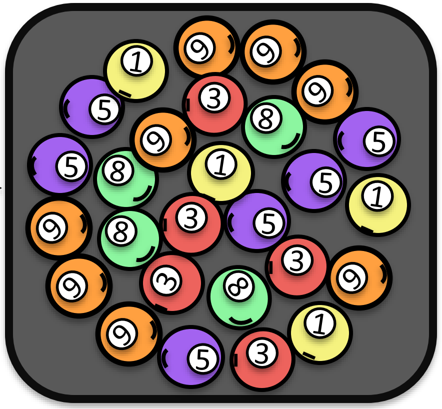
# Bootstrap

## DATASET

Size N



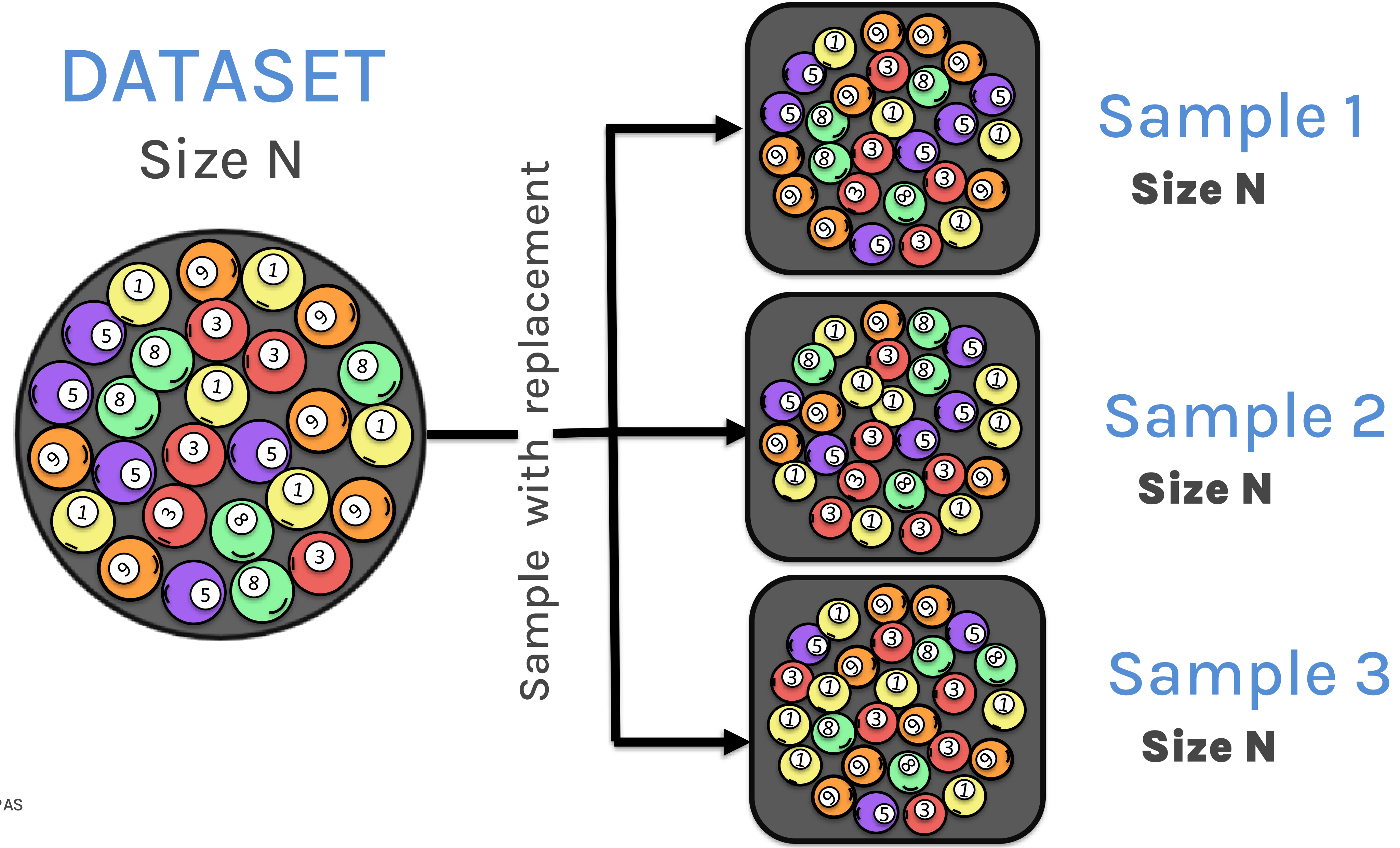

Sample with replacement



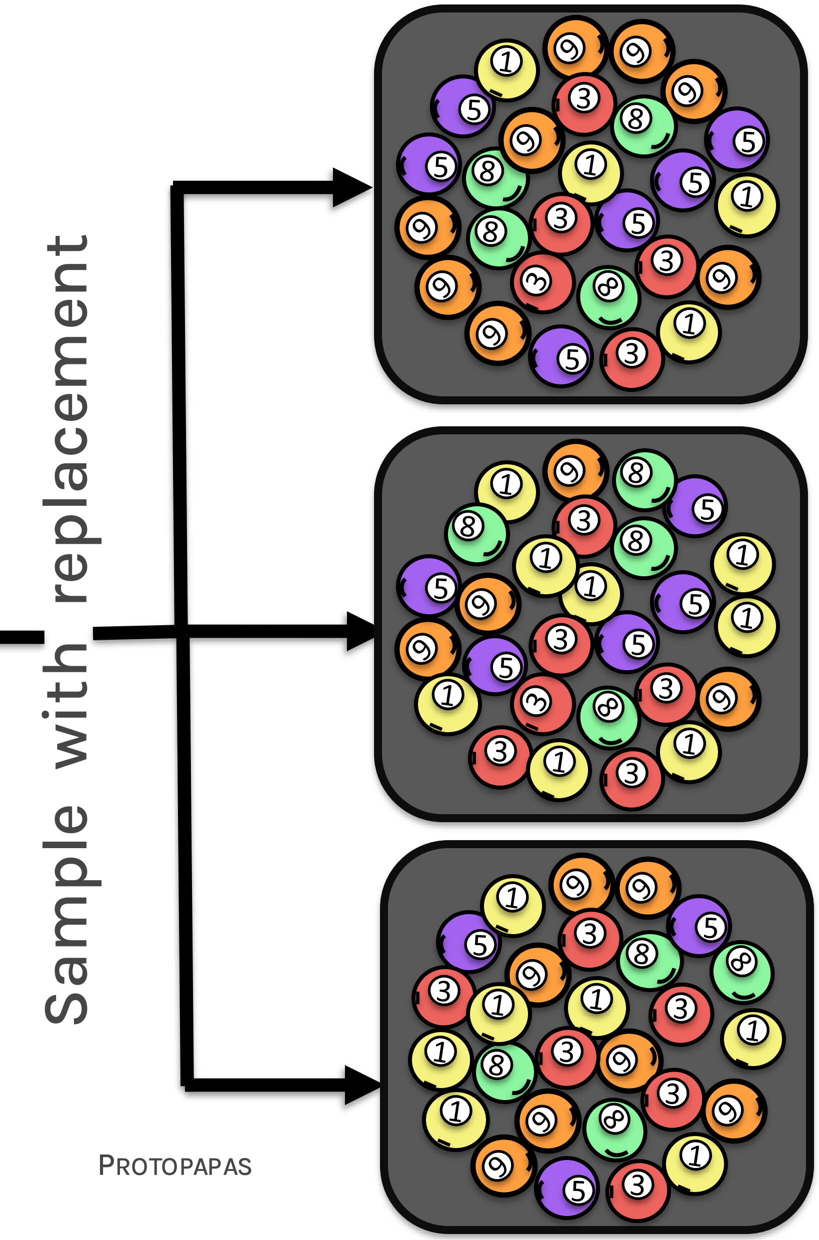

## Sample 1

**Size N**

# Bootstrap


DATASET
Size N
Sample with replacement
Sample 1
Size N
Sample 2
Size N
Sample 3
Size N

# Bootstrap




**Sample 1**

**Size N**

$\xrightarrow{\text{Train}}$ Model 1: $\hat{y} = \hat{\beta}_0^{(1)} + \hat{\beta}_1^{(1)} x$

**Sample 2**

**Size N**

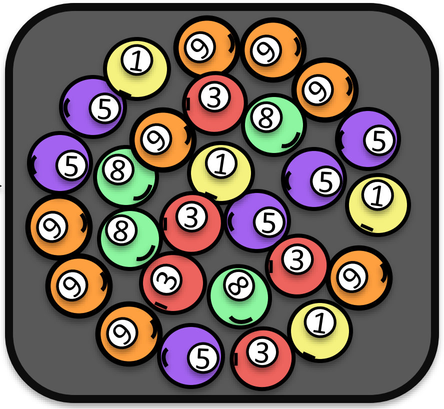
$\xrightarrow{\text{Train}}$ Model 2: $\hat{y} = \hat{\beta}_0^{(2)} + \hat{\beta}_1^{(2)} x$

**Sample 3**

**Size N**

$\xrightarrow{\text{Train}}$ Model s: $\hat{y} = \hat{\beta}_0^{(s)} + \hat{\beta}_1^{(s)} x$

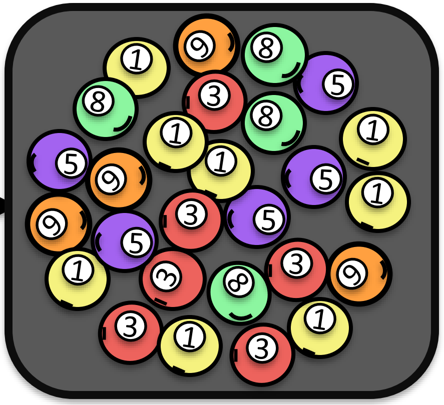
# Bootstrap



**Sample 1**

**Size N**

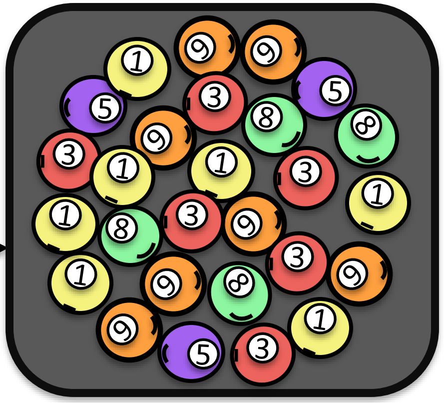
$\xrightarrow{\text{Train}}$ Model 1: $\hat{y} = \hat{\beta}_0^{(1)} + \hat{\beta}_1^{(1)} x$



**Sample 2**

**Size N**

$\xrightarrow{\text{Train}}$ Model 2: $\hat{y} = \hat{\beta}_0^{(2)} + \hat{\beta}_1^{(2)} x$



**Sample 3**

**Size N**

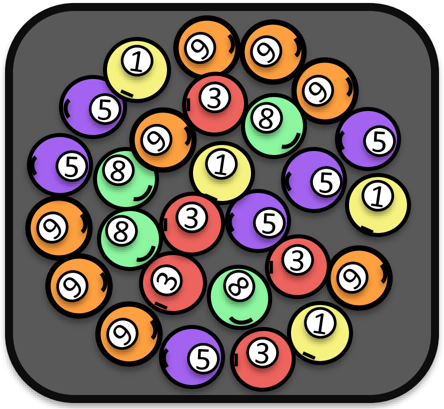
$\xrightarrow{\text{Train}}$ Model s: $\hat{y} = \hat{\beta}_0^{(s)} + \hat{\beta}_1^{(s)} x$

Combine models

$$\boxed{\mu_{\hat{\beta}} = \frac{1}{s}\sum_{i=1}^{s} \hat{\beta}^{(i)}}$$

$$\boxed{\sigma_{\hat{\beta}} = \sqrt{\frac{1}{s-1}\sum_{i=1}^{s}\left(\hat{\beta}^{(i)} - \mu_{\hat{\beta}}\right)^2}}$$

# In summary, for each “Parallel Universe”…



Train

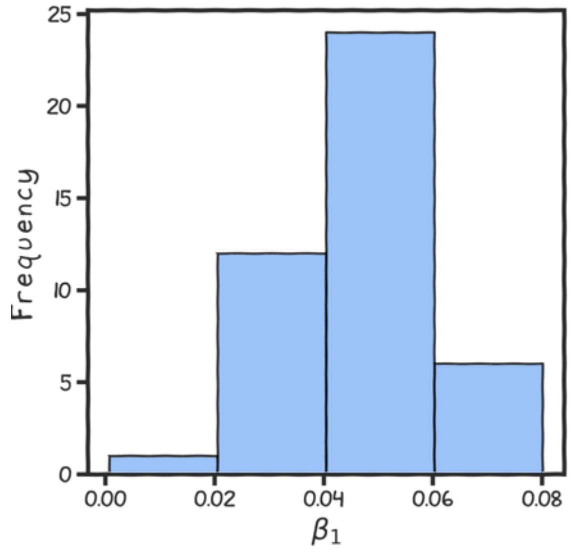
**Model i:** $\hat{y} = \hat{\beta}_0^{(i)} + \hat{\beta}_1^{(i)} x$

s models

Combine all models

$$\mu_{\hat{\beta}} = \frac{1}{s}\sum_{i=1}^{s} \hat{\beta}^{(i)}$$

$$\sigma_{\hat{\beta}} = \sqrt{\frac{1}{s-1}\sum_{i=1}^{s}\left(\hat{\beta}^{(i)} - \mu_{\hat{\beta}}\right)^2}$$



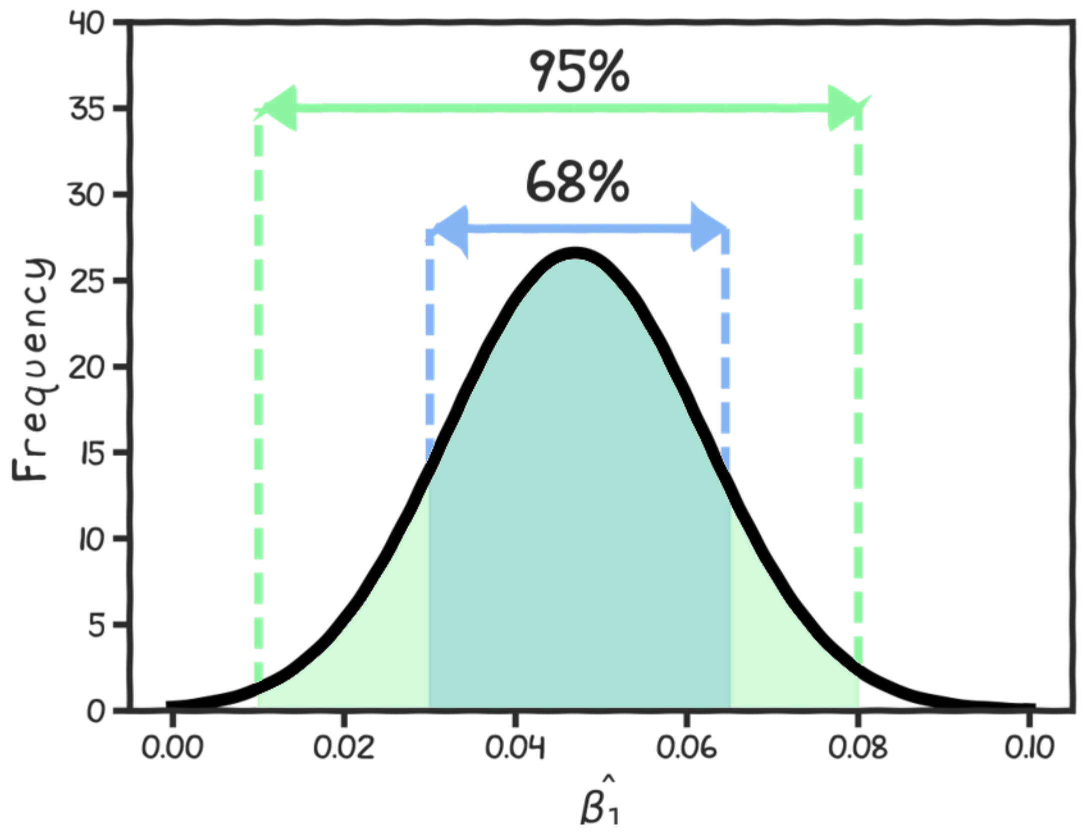
# Bootstrapping for Estimating Sampling Error

## Definition

**Bootstrapping** is the practice of estimating properties of an estimator by measuring those properties by sampling from the observed data.

For example, we can compute $\hat{\beta}_0$ and $\hat{\beta}_1$ multiple times by randomly sampling from our data set. We then use the variance of our multiple estimates to approximate the true variance of $\hat{\beta}_0$ and $\hat{\beta}_1$.

# Confidence intervals for the predictor estimates (cont.)




The samples of $\beta's$ allow us to estimate the 95% confidence interval, which is the range of values that would contain the **true value** of $\hat{\beta}_1$ with $95\%$ probability.

How do we calculate confidence intervals?

# Confidence intervals for the predictor estimates (cont.)

Let's look at an example of how to construct a confidence interval for $\hat{\beta}_1$ using our bootstrap samples.

These numbers represent bootstrap estimates of $\hat{\beta}_1$ obtained by repeatedly resampling our dataset.

[13.75, 15.21, 13.65, 13.58, 12.93, 14.23, 12.81, 11.50, 13.09, 12.26, …]

Step #1: Sort the bootstrap estimates of $\hat{\beta}_1$ from lowest to highest.

# Confidence intervals for the predictor estimates (cont.)

[13.75, 15.21, 13.65, 13.58, 12.93, 14.23, 12.81, 11.50, 13.09, 12.26, …]


Step #1: Sort the bootstrap estimates of $\hat{\beta}_1$ from lowest to highest.

[11.50, 12.26, 12.81, 12.93, 13.09, 13.58, 13.65, 13.75, 14.23, 15.21, …]


Step #2: Find the lower confidence range using np.percentile()

# Confidence intervals for the predictor estimates (cont.)

[ 11.50, 12.26, 12.81, 12.93, 13.09, 13.58, 13.65, 13.75, 14.23, 15.21,, … ]

Step #2: Find the lower confidence range using np.percentile()

`np.percentile(` [11.50, 12.26,12.81,12.93,13.09,13.58,13.65,13.75,14.23, 15.21,, … ]`,2.5)` **= 12.80**

[ 11.50, 12.26, 12.81, 12.93, 13.09, 13.58, 13.65, 13.75, 14.23, 15.21,, … ]

2.5% of data are on the left of this value

# Confidence intervals for the predictor estimates (cont.)

[ 11.50, 12.26, 12.81, 12.93, 13.09, 13.58,13.65,13.75, 14.23, 15.21,, … ]

Step #3: Find the upper confidence range again using np.percentile()

`np.percentile(` [11.50, 12.26,12.81,12.93,13.09,13.58, 13.65, 13.75,14.23, 15.21, , … ] `,97.5)` = **13.71**

[ 11.50, 12.26, 12.81, 12.93, 13.09, 13.58,13.65,13.75, 14.23, 15.21, … ]

2.5% of data are on the right of this value

# Confidence intervals for the predictor estimates (cont.)

Lower bound
(2.5th percentile)

Upper bound
(97.5th percentile)

[ 11.50, 12.26, 12.81, 12.93, 13.09, 13.58, 13.65, 13.75, 14.23, 15.21, ... ]

95% confidence intervals

# Confidence intervals for the predictor estimates (cont.)

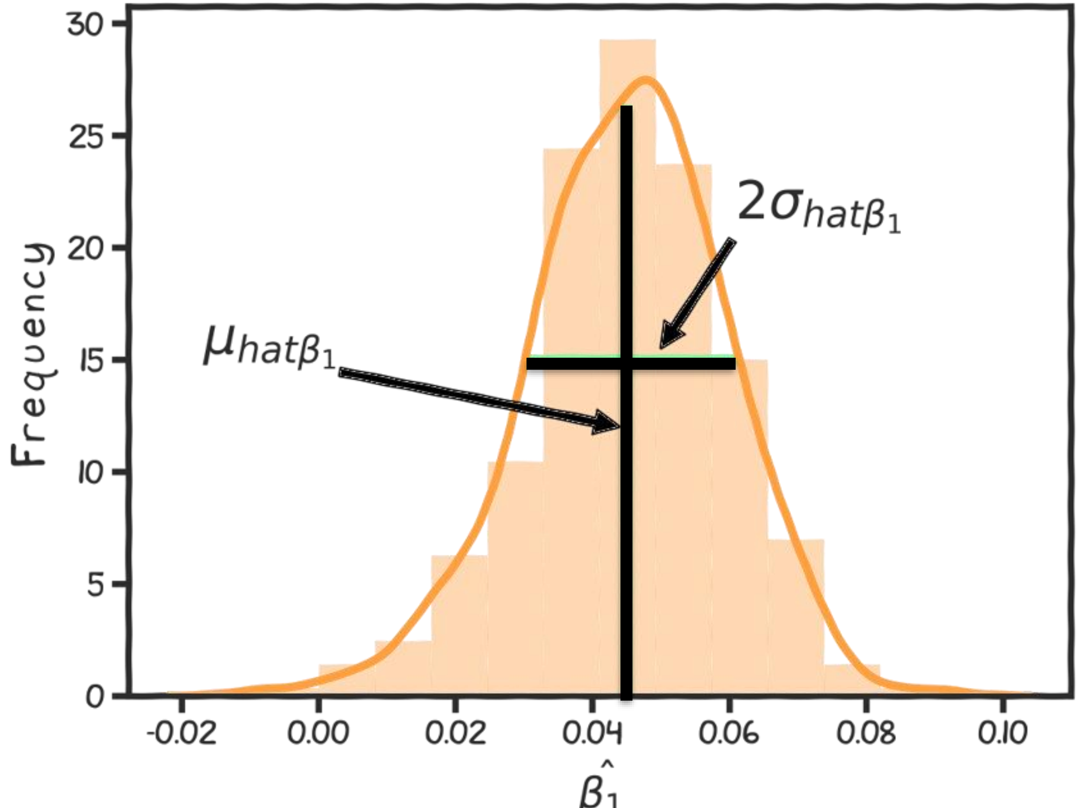
Confidence intervals provide bounds,
but the precision of $\hat{\beta}$ can also be **summarized** by its standard deviation, $\sigma$,
which we call the **standard error**.

95% confidence intervals

# Confidence intervals for the predictors estimates (cont)

In other words we empirically estimate the standard deviations $\hat{\sigma}_{\hat{\beta}}$ which are called the **standard errors,** $SE_{\hat{\beta}_0}, SE_{\hat{\beta}_1}$ through bootstrapping.

**Using these**, one can calculate the 95% confidence intervals as approximately $\hat{\beta} \pm 2SE_{\beta}$.




We are making an assumption here. What is it?

# Standard Errors based on probability theory

**Alternatively:** If we assume normality, then:

And if we know the variance $\sigma_\epsilon^2$ of the noise $\epsilon$, we can compute $SE(\hat{\beta}_0), SE(\hat{\beta}_1)$ analytically using the formulae below (no need to bootstrap):

$$SE_{\widehat{\beta}_0} = \sigma_\epsilon \sqrt{\frac{1}{n} + \frac{\mu_x^2}{\sum_i (x_i - \mu_x)^2}}$$

$$SE_{\widehat{\beta}_1} = \frac{\sigma_\epsilon}{\sqrt{\sum_i (x_i - \mu_x)^2}}$$

$$\boldsymbol{CI_{\widehat{\beta}}(95\%) = [\widehat{\beta} - 2SE_{\widehat{\beta}}, \widehat{\beta} + 2SE_{\widehat{\beta}}]}$$

Where $n$ is the number of observations.

$\mu_x$ is the mean value of the predictor.

# Standard Errors

In practice, we do not know the value of $\sigma_\epsilon$ since we do not know the exact distribution of the noise $\epsilon$.

However, if we make the following assumptions:

- the errors $\epsilon_i = y_i - \hat{y}_i$ and $\epsilon_j = y_j - \hat{y}_j$ are uncorrelated, for $i \neq j$ ,
- each $\epsilon_i$ has a mean 0 and variance $\sigma_\epsilon^2$,

then, we can empirically estimate $\sigma_\epsilon$, from the data and our regression line:

$$\sigma_\epsilon = \sqrt{\frac{n \cdot MSE}{n-2}} = \sqrt{\sum \frac{\left(\hat{f}(x) - y_i\right)^2}{n-2}}$$

Remember: $y_i = f(x_i) + \epsilon_i \Longrightarrow \epsilon_i = y_i - f(x_i)$