# EBU5601
# Data Design

## Confidence Intervals

Dr Chao Shu
School of Electronic Engineering and Computer Science
Queen Mary University of London
Nov. 2024

# Learning Outcomes

- The main outcomes are:
  - [**LO6.1**] Understand the relationship of sampling distributions and confidence intervals
  - [**LO6.2**] Apply parametric methods to constructing confidence intervals
  - [**LO6.3**] Apply the bootstrapping method to constructing confidence intervals
  - [**LO6.4**] Analyse population means and mean differences with confidence intervals
  - [**LO6.5**] Correctly interpret confidence intervals from a dataset

# Introduction

- In the previous lesson, we learned how to use sampling distributions and bootstrapping to understand the values of a statistic that are possible. We can also use the sampling distribution to understand the most likely values for a **parameter** as well.
- In the real world, we don't usually know the values of a parameter. How do we use sampling distributions to **infer** where a parameter is located?
- **Statistical inference draws conclusions about a population based on sample data**. It also provides a statement, expressed in terms of probability, of how much confidence we can place in our conclusions.
- Two most common types of statistical inference are **confidence intervals** (*this lesson*) and **tests of significance** (*next lesson*).

# Introduction

## Case Study

- Suppose you want to estimate the TOEFL iBT scores of all BUPT students who took the test this year. You randomly collected a sample of **100** students' scores. The sample mean is **88**.
- The value $\bar{x} = 88$ is your **estimate** of the mean score **μ** that all BUPT students who took TOEFL this year would achieve. But how reliable is this estimate? An estimate without an indication of its variability is of little value. (A second sample would surely not give 88 again)

# Introduction

## Case Study

- We know that if the entire population of TOEFL scores has mean $\mu$ and standard deviation $\sigma$, then in repeated samples of size 100 the sample mean $\bar{x}$ follows the $N(\mu, \sigma^2/100)$ distribution.
- Suppose we know that the standard deviation σ of TOEFL scores of all students at BUPT (population) is $\sigma = 15$. (*not realistic to know population $\sigma$, just for introductory purposes for now*)
- In repeated sampling the sample mean $\bar{x}$ (sampling distribution of the mean) follows a Normal distribution centred at the unknown population mean $\mu$ with a standard deviation:

$$\sigma_{\bar{x}} = \frac{15}{\sqrt{100}} = 1.5$$

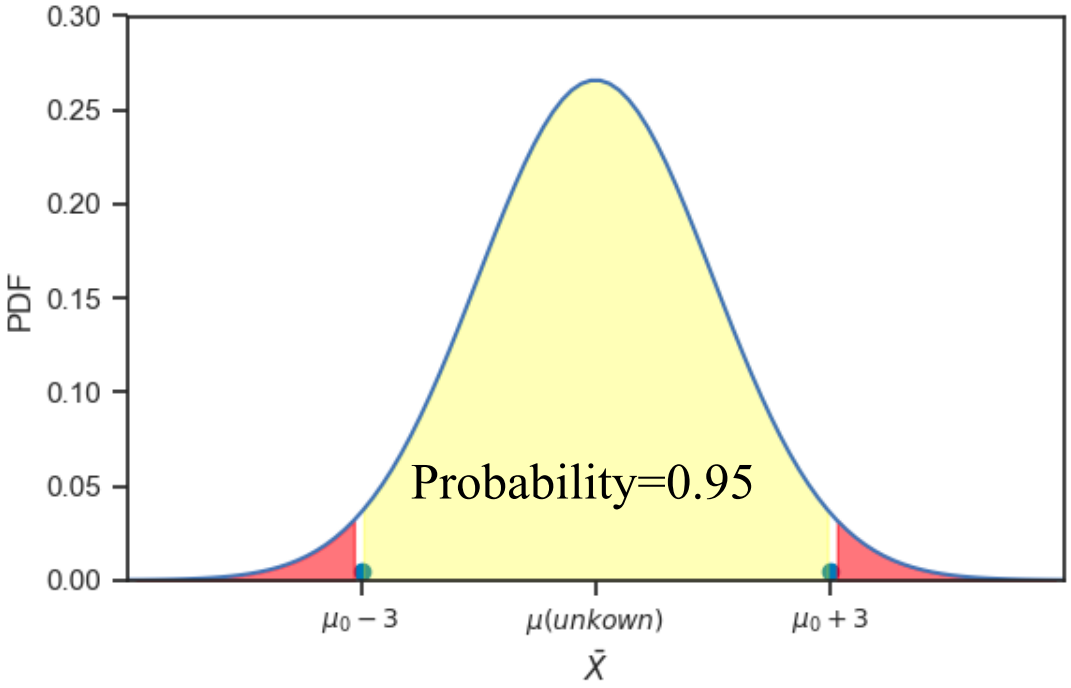
# Introduction

## Case Study

- The 68–95–99.7 rule says that the probability is about 0.95 that $\bar{x}$ will be within 3 points (two standard deviations of $\bar{x}$) of the population mean score $\mu$.
- To say that $\bar{x}$ lies within 3 points of $\mu$ is the same as saying that $\mu$ is within 3 points of $\bar{x}$.
- So 95% of all samples will capture the true $\mu$ in the interval from $\bar{x} - 3$ to $\bar{x} + 3$

We state that we are *95% confident* that the unknown mean score for all BUPT students who took TOEFL this year lies between

- $\bar{x} - 3 = 85$
- $\bar{x} + 3 = 91$



# Confidence Intervals

## Confidence Interval

- A level *C* **confidence interval** for a parameter is an interval computed from sample data by a method that has probability *C* of producing an interval containing the true value of the parameter.
  - It is an interval of the form (a, b), where a and b are numbers computed from the data.
  - It has a property called a confidence level that gives the probability of producing an interval that contains the unknown parameter.
  - Users can choose the confidence level, but 95% is the standard for most situations. Occasionally, 90% or 99% is used. We will use *C* to stand for the confidence level in decimal form. For example, a 95% confidence level corresponds to C = 0.95.
  - In the TOEFL score example, the interval of numbers between the values $\bar{x} \pm 3$ is called ***a 95% confidence interval for $\mu$***.

# Confidence Intervals

## Confidence Interval for a Population Mean

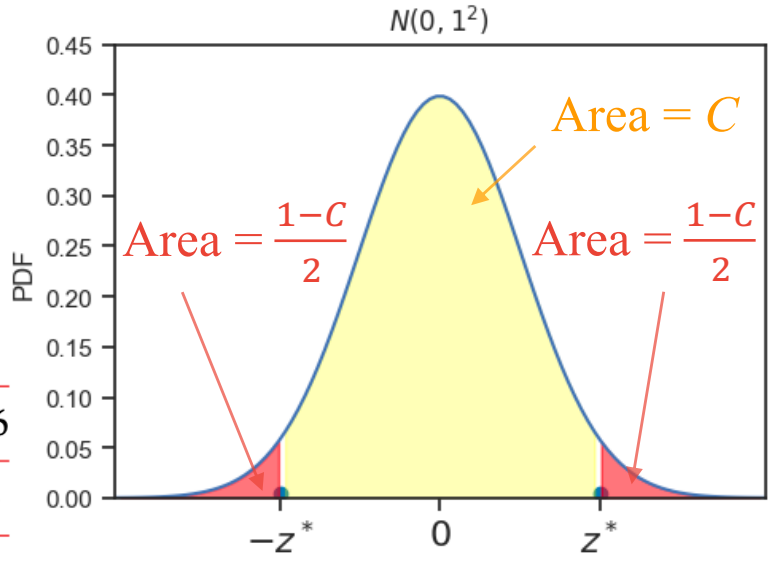
- The sample mean $\bar{x}$ follows the normal distribution with mean $\mu$ and standard deviation $\sigma/\sqrt{n}$. So there is probability $C$ that $\bar{x}$ lies between

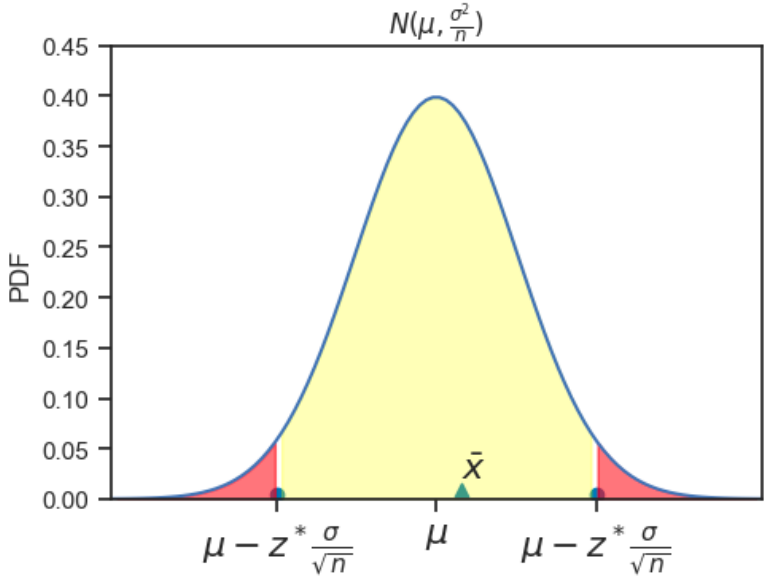
$$\mu - z^* \frac{\sigma}{\sqrt{n}} \quad and \quad \mu + z^* \frac{\sigma}{\sqrt{n}}$$

- This is exactly the same as saying that the unknown population mean $\mu$ lies between $\bar{x} - z^* \frac{\sigma}{\sqrt{n}}$ $and$ $\bar{x} + z^* \frac{\sigma}{\sqrt{n}}$ *The level C confidence interval for $\mu$*

where $z^*$ is the value on the standard Normal curve with area C between -$z^*$ and $z^*$

| $z^*$ | 1.645 | 1.960 | 2.576 |
|---|---|---|---|
| $C$ | 90% | 95% | 99% |



# Confidence Intervals

## Example: Average debt of undergraduate borrowers

- The National Student Loan Survey collects data to examine questions related to the amount of money that borrowers owe. The survey selected a sample of **1280** borrowers who began repayment of their loans between four and six months prior to the study. The **mean** of the debt for undergraduate study was **$18,900** and the **standard deviation** was about **$49,000**. This distribution is clearly skewed but because our sample size is quite large, we can rely on the central limit theorem to assure us that the confidence interval based on the Normal distribution will be a good approximation.
- Let's compute a **95% confidence interval for the true mean debt** for all borrowers. Although the standard deviation is estimated from the data collected, we will treat it as a known quantity for our calculations here.

For 95% confidence, we can find from z-table that $z^* = \mathbf{1.96}$. The 95% confidence interval is

$$\bar{x} \pm z^* \frac{\sigma}{\sqrt{n}} = 18900 \pm 1.96 \frac{49000}{\sqrt{1280}}$$

# Construct Confidence Intervals using Bootstrapping

- We construct a confidence interval based on the sampling distribution of the statistic of interest: if the entire population has a mean $\mu$ and a standard deviation $\sigma$, then the sample mean $\bar{x}$ follows the $N(\mu, \sigma^2/\sqrt{n})$ distribution given a sufficiently large sample size.
- We use:
  - the sample statistic (e.g., sample mean $\bar{x}$ ) as the **estimate** of the value of the unknown population parameter
  - the product of the z-score and sample standard deviation $z^*\frac{\sigma}{\sqrt{n}}$) as the **margin of error** to reflects how accurate we believe our estimate is, based on the variability o the estimate, and how confident we are that the procedure will catch the true population parameter (e.g., mean $\mu$), where the z-score is based on the confidence level.
- It is not realistic to know the population standard deviation $\sigma$, but we can use bootstrapping to obtain the sampling distribution directly. Then we can construct the confidence interval by obtaining the percentile of the sampling distribution based on the confidence level *C*.

# Construct Confidence Intervals using Bootstrapping

## Demo: Confidence Interval for a Population Mean using Bootstrapping

- Suppose we've run a survey about whether people drink coffee or not and their heights.
- In this demo, we would like to estimate the height of all coffee drinkers.
- A quick look of the dataset

`coffee_full_df.head()`

| | user_id | age | drinks_coffee | height |
|---|---|---|---|---|
| 0 | 4509 | <21 | False | 64.538179 |
| 1 | 1864 | >=21 | True | 65.824249 |
| 2 | 2060 | <21 | False | 71.319854 |
| 3 | 7875 | >=21 | True | 68.569404 |
| 4 | 6254 | <21 | True | 64.020226 |

`coffee_full_df.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2974 entries, 0 to 2973
Data columns (total 4 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   user_id        2974 non-null   int64
 1   age            2974 non-null   object
 2   drinks_coffee  2974 non-null   bool
 3   height         2974 non-null   float64
dtypes: bool(1), float64(1), int64(1), object(1)
memory usage: 72.7+ KB
```

# Construct Confidence Intervals using Bootstrapping

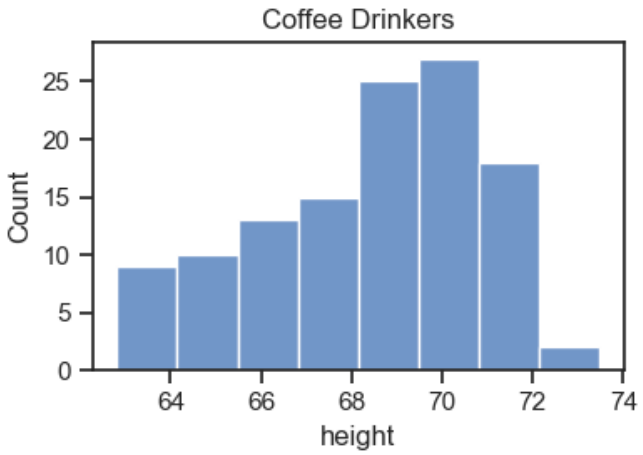
## Demo: Confidence Interval for a Population Mean using Bootstrapping

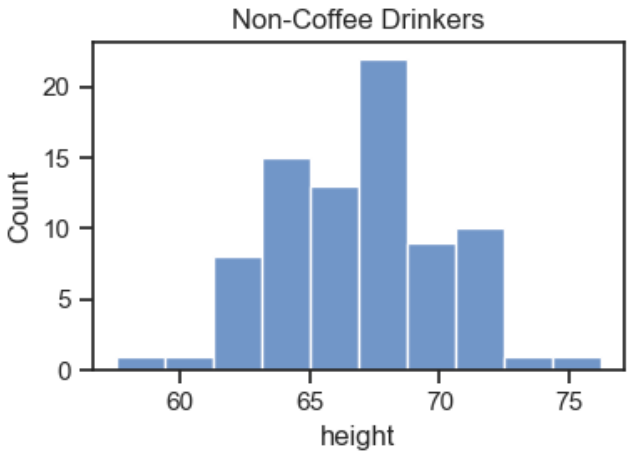
- Assume the 2974 people form a population and get a random sample of size **200** from the full dataset assuming we can only access the sample.

```
1  coffee_sample_df = coffee_full_df.sample(200)
✓  0.0s
```

- We find 119 coffee drinkers and 81 non-coffee drinkers in the sample, the means and standard deviations of the heights for the two groups and their distributions:

| | mean | std |
|---|---|---|
| **drinks_coffee** | | |
| False | 66.977648 | 3.198533 |
| True | 68.351804 | 2.419505 |



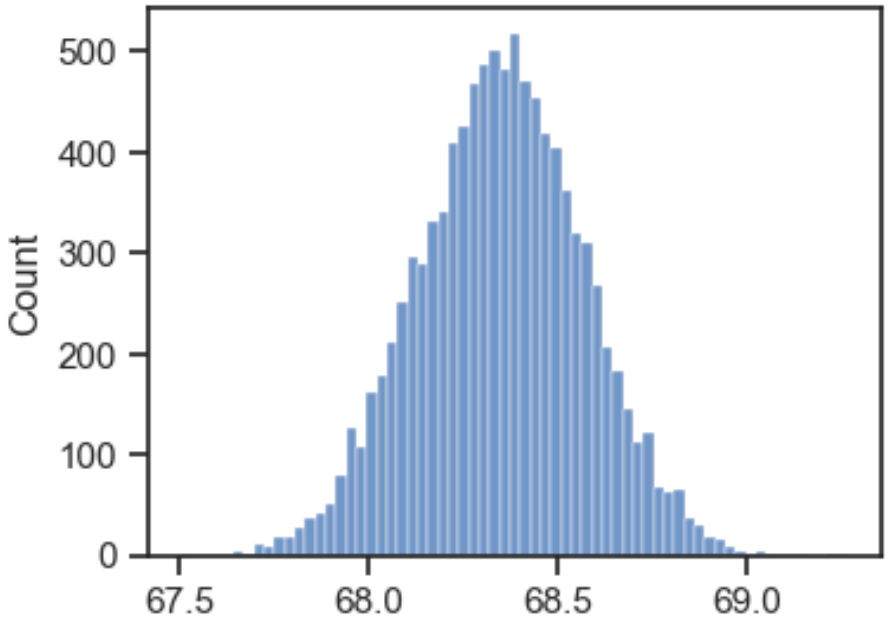
# Construct Confidence Intervals using Bootstrapping

## Demo: Confidence Interval for a Population Mean using Bootstrapping

- Generate a sampling distribution of the average height of the **coffee drinkers** by bootstrapping our sample.

```
# Bootstrap the sample of coffee drinkers' heights
boot_means = np.array([coffee_drinker_sample_df.sample(frac=1, replace=True)['height'].mean() for i in range(10000)])
std_err_boot = boot_means.std()
print("standard error of sampling distribution from bootstrapping:", std_err_boot)
```

- `standard error of the mean from bootstrapping: 0.22092122939286196`



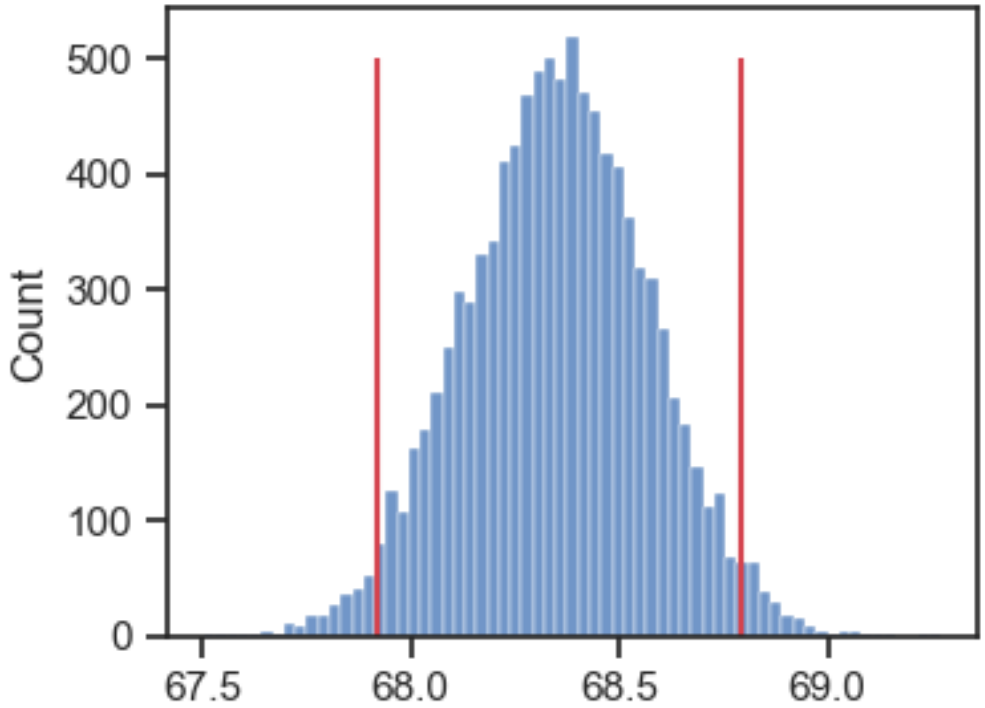
# Construct Confidence Intervals using Bootstrapping

## Demo: Confidence Interval for a Population Mean using Bootstrapping

- Construct the 95% confidence interval for the mean height of the coffee drinkers.

```
1  ci_lower, ci_upper = np.percentile(boot_means, 2.5), np.percentile(boot_means, 97.5)
2  print("Confidence Interval: ({}, {})".format(ci_lower, ci_upper))
```

```
Confidence Interval: (67.9213961166786, 68.79058635723293)
```



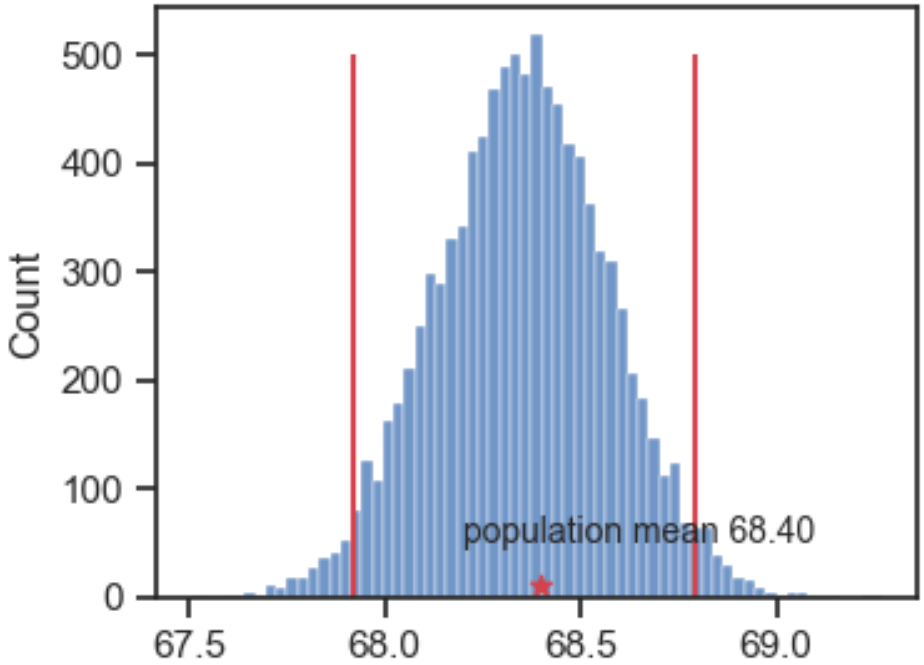
# Construct Confidence Intervals using Bootstrapping

## Demo: Confidence Interval for a Population Mean using Bootstrapping

- Check if our confidence interval captures the actual average height of coffee drinkers in the population. We can see the actual average height of coffee drinkers falls in the 95% confidence interval we constructed.

```
1  pop_mean = coffee_full_df[coffee_full_df['drinks_coffee'] == True]['height'].mean()
2  print("Population mean: ", pop_mean)
```

```
Population mean:  68.40021025548381
```

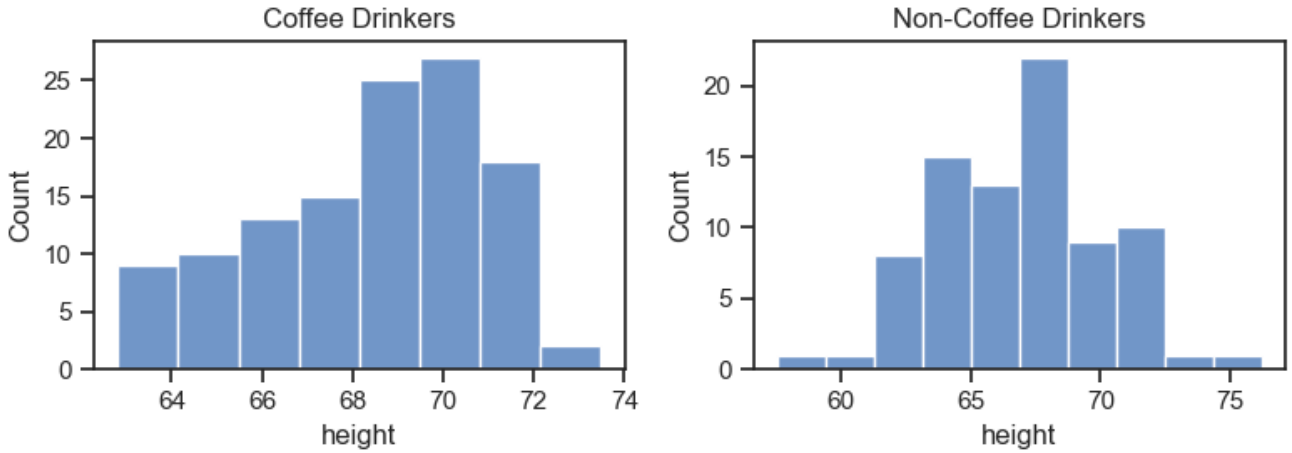


# Construct Confidence Intervals using Bootstrapping

## Demo: Confidence Interval for Difference in Population Means using bootstrapping

- In this demo, we would like to estimate the difference in the average heights of coffee drinkers and non-coffee drinkers, trying to answer the question: *Do we have evidence of a difference in average heights*?
- We'll use the same sample data `coffee_sample_df` from the population data `coffee_full_df`.

| drinks_coffee | mean | std |
|---|---|---|
| False | 66.977648 | 3.198533 |
| True | 68.351804 | 2.419505 |



# Construct Confidence Intervals using Bootstrapping

## Demo: Confidence Interval for Difference in Population Means using bootstrapping

- Generate a sampling distribution of the **difference in average heights** between the **coffee drinkers** and **non-coffee drinkers** by bootstrapping our sample.

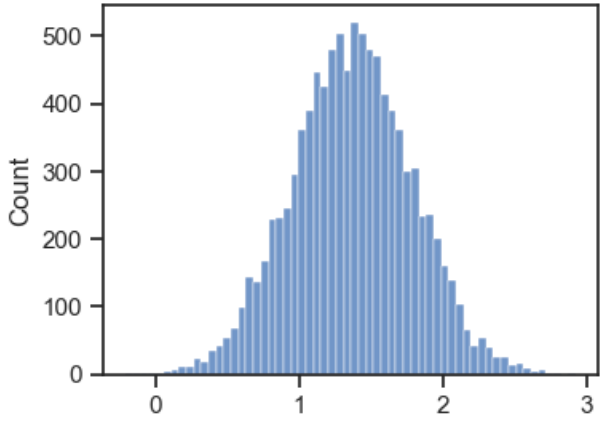
```
boot_coffee_drinker_means, boot_noncoffee_drinker_means, boot_diff_means = [], [], []

for i in range(10000):
    # Boostrapping the original sample to generate a new sample
    boot_sample = coffee_sample_df.sample(frac=1, replace=True)

    # Calculate the mean heights for coffee drinkers and non-coffee drinkers respectively.
    coffee_drinker_mean = boot_sample[boot_sample['drinks_coffee'] == True]['height'].mean()
    noncoffee_drinker_mean = boot_sample[boot_sample['drinks_coffee'] == False]['height'].mean()

    # Insert current mean to the list of means to generate the sampling distributions of the mean sleep hours of female and male studen
    # as well as the sampling distribution of the mean difference between female and male students.
    boot_coffee_drinker_means.append(coffee_drinker_mean)
    boot_noncoffee_drinker_means.append(noncoffee_drinker_mean)
    boot_diff_means.append(coffee_drinker_mean - noncoffee_drinker_mean)
```

- `standard error of the mean difference from bootstrapping: 0.42095385358706866`



# Construct Confidence Intervals using Bootstrapping

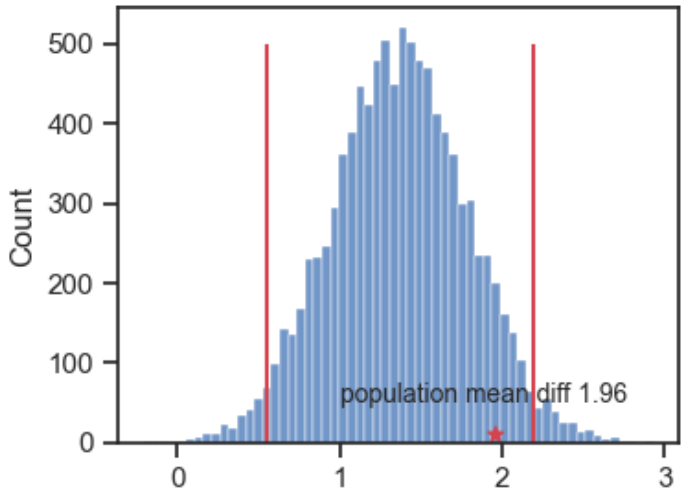
## Demo: Confidence Interval for Difference in Population Means using bootstrapping

- Construct the 95% confidence interval for the mean difference between the average heights of the coffee drinkers and the non-coffee drinkers. Check if our confidence interval captures the actual difference in average heights.

```
ci_lower, ci_upper = np.percentile(boot_diff_means, 2.5), np.percentile(boot_diff_means, 97.5)
print("Confidence Interval: ({}, {})".format(ci_lower, ci_upper))

pop_mean_diff = coffee_full_df[coffee_full_df['drinks_coffee'] == True]['height'].mean() - \
    coffee_full_df[coffee_full_df['drinks_coffee'] == False]['height'].mean()
print("Population mean difference: ", pop_mean_diff)
```

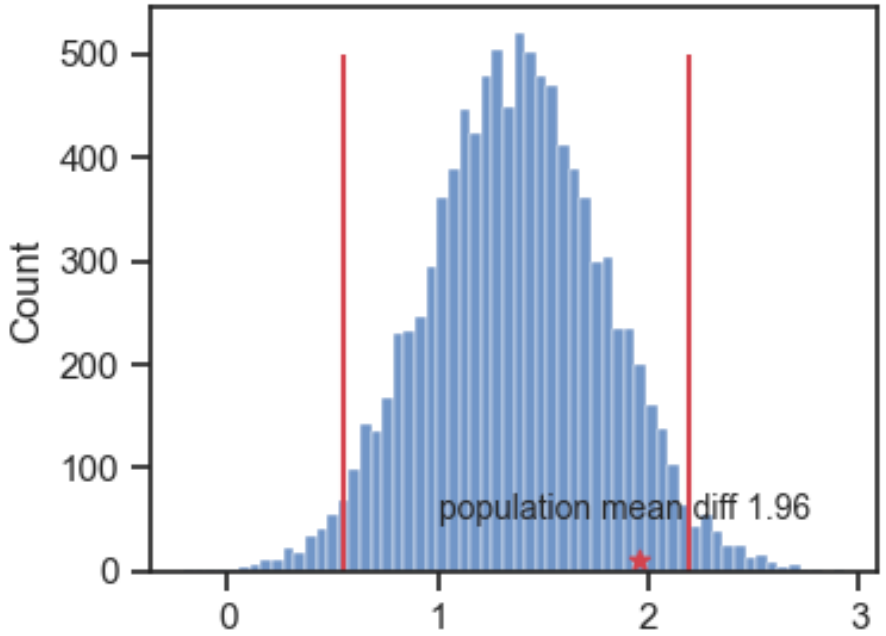
```
Confidence Interval: (0.5507390525103695, 2.1884315671703294)
Population mean difference: 1.9568024933368093
```



# Construct Confidence Intervals using Bootstrapping

## Demo: Confidence Interval for Difference in Population Means using bootstrapping

- We can be 95% confident that the difference between the average heights of coffee drinkers and non-coffee drinkers is in the interval of 0.55 to 2.19 inches.
- Since the difference did not contain zero, this suggested there is truly a difference in the average heights in the population of coffee drinkers as compared to non-coffee drinkers.



# RECAP

Confidence Intervals

# Recap

- **Statistical inference:**
  - Confidence Intervals
    - use sampling distributions to infer where a parameter is located
  - Test of Significance
- **Confidence Interval**
  - An interval of the form (a, b)
  - a confidence level *C*
- **Construct a Confidence Interval**
  - Parametric method
    - estimate ± margin of error (e.g., $\bar{x} \pm z^* \frac{\sigma}{\sqrt{n}}$)
  - Bootstrapping
    - sampling distribution + percentile

# Questions

Use student forum on QM+

chao.shu@qmul.ac.uk