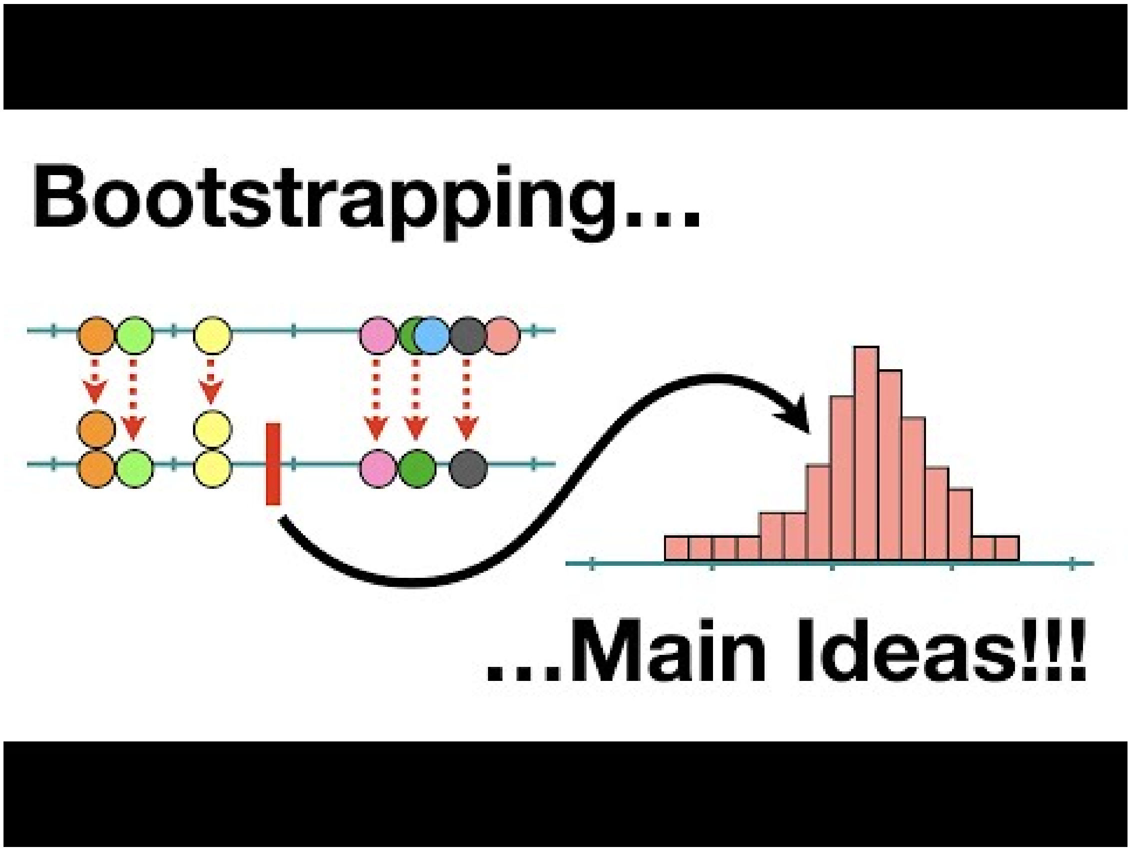
STA130 Week 4 HW

October 4, 2024

[1]:
```
from IPython.display import YouTubeVideo
YouTubeVideo('Xz0x-8-cgaQ', width=800, height=500)
```

[1]:



Question 1

The "Pre-lecture" video (above) mentioned the "standard error of the mean" as being the "standard deviation" of the distribution bootstrapped means. What is the difference between the "standard error of the mean" and the "standard deviation" of the original data? What distinct ideas do each of these capture? Explain this concisely in your own words.

In the pre-lecture video, I referred to the "standard error of the mean" as the "standard deviation" of the bootstrap mean of the distribution. After watching the video provided above, I was able to see the difference between the two and what unique ideas each captures.

Assume for us an existing data collection. Standard deviation measures the general distribution of data points from the mean within the current data collection. It indicates the extent of distribution of the data points around the mean. Data points distant from the mean, for instance, have a significant SD, therefore showing a broad distribution of the data points. Though it differs from SD, SEM is comparable. Standard Error of the Mean gauges the anticipated variation in the mean of the sample data points should we then take another sample from the same population. By means of repeated experiment and sample mean estimation of variability, it helps to raise the population estimate's precision.

While SD assesses the distribution within the acquired real-world data, SEM measures, considering data variability, how representative the mean is of the larger population? Furthermore, the SEM lowers relative to the SD as the sample data size grows. The sample mean more precisely as the sample size grows, therefore approximating the actual population mean.

Question 6

1. What is the process of bootstrapping?

Let's say you roast a bunch of chicken. Not all chickens taste the same. If you want to know if all chickens taste good, bootstrapping is like taking a few chickens out of the roasted chicken at random, tasting them, and putting them back. The process is to randomly pick a chicken and try again. Repeat this process many times, recording how good the chicken tastes each time. This is bootstrapping explained in chicken.

2. What is the main purpose of bootstrapping?

The goal of boosting is to find out if the roasted chicken tastes consistently good, not just one or two. By trying these small, mixed samples over a number of times, you can get a good idea of the overall taste of the batch. This method provides a way to guess what all future batches will taste like if you use the same recipe.

1. 3. If you had a (hypothesized) guess about what the average of a population was, and you had a sample of size n from that population, how could you use bootstrapping to assess whether or not your (hypothesized) guess might be plausible?

Let's use the size of a chicken wing as an example. I measure the length of each wing from 100 chicken wings to get a range of sizes. This is the process of getting the initial data.

I randomly select one of the 100 values, record it, and put it back into the data set. I repeat this until have 100 measurements again. This new set of measurements is our first bootstrap sample.

I generate many bootstrap samples in the same way. I do this 1,000 times to get a lot of data. For each sample, I calculate and record the average wing size. This is the process of repeating the process.

After bootstrapping, I have 1,000 average sizes. I can look at the recorded averages to see how often I reach or exceed our target size of inches. I can record them all and visualize them in a way I like. This is the process of analyzing the results.

Question 8

Explanation of the Null Hypothesis of "no effect"

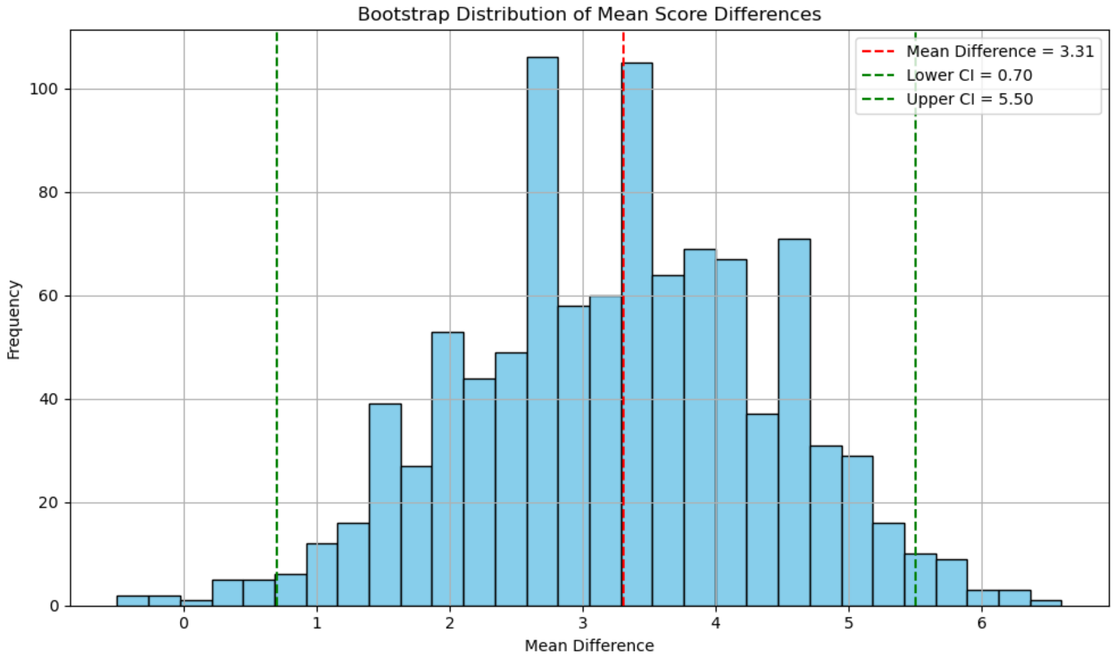
In statistical testing, the null hypothesis is a default hypothesis asserting either no effect or no connection between events. Within the framework of the AliTech vaccination research, the null hypothesis especially asserts that the vaccination has no influence on patient health ratings. This implies that any reported changes in the health ratings from before to after vaccination are ascribed entirely to random fluctuations anticipated in any biological assessment, not to the efficiency of the immunization.

Data Visualization

[9]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

# Step 1: Define the data
data = {
    'PatientID': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
    'InitialHealthScore': [84, 78, 83, 81, 81, 80, 79, 85, 76, 83],
    'FinalHealthScore': [86, 86, 80, 86, 84, 86, 86, 82, 83, 84]
}

df = pd.DataFrame(data)

# Step 2: Calculate score difference
df['ScoreDifference'] = df['FinalHealthScore'] - df['InitialHealthScore']

# Step 3: Bootstrap function to compute mean difference and confidence intervals
def bootstrap_mean_diff(data, n_bootstrap=1000, random_seed=42):
    np.random.seed(random_seed)
    means = []
    for _ in range(n_bootstrap):
        sample = np.random.choice(data, size=len(data), replace=True)
        means.append(np.mean(sample))
    return np.percentile(means, [2.5, 97.5]), np.mean(means)

# Step 4: Execute bootstrapping
confidence_interval, mean_difference =␣
 ↪bootstrap_mean_diff(df['ScoreDifference'])

print(f"Mean Difference: {mean_difference}")
print(f"95% Confidence Interval: {confidence_interval}")

# Step 5: Visualize the bootstrap distribution
bootstrap_samples = [np.mean(np.random.choice(df['ScoreDifference'],␣
 ↪size=len(df['ScoreDifference']), replace=True)) for _ in range(1000)]

plt.figure(figsize=(10, 6))
plt.hist(bootstrap_samples, bins=30, color='skyblue', edgecolor='black')
plt.axvline(mean_difference, color='red', linestyle='--', label=f'Mean␣
 ↪Difference = {mean_difference:.2f}')
plt.axvline(confidence_interval[0], color='green', linestyle='--',␣
 ↪label=f'Lower CI = {confidence_interval[0]:.2f}')
plt.axvline(confidence_interval[1], color='green', linestyle='--',␣
 ↪label=f'Upper CI = {confidence_interval[1]:.2f}')
plt.title('Bootstrap Distribution of Mean Score Differences')
plt.xlabel('Mean Difference')
plt.ylabel('Frequency')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```

```
Mean Difference: 3.3075
95% Confidence Interval: [0.7 5.5]
```



```
[ ]:
```

```
[ ]:
```

```
[ ]:
```