Sampling Distributions - Difference in Means

December 4, 2017

0.0.1 Confidence Interval - Difference In Means

Here you will look through the example from the last video, but you will also go a couple of steps further into what might actually be going on with this data.

```
In [1]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt

        %matplotlib inline
        np.random.seed(42)

        full_data = pd.read_csv('coffee_dataset.csv')
        sample_data = full_data.sample(200)
```

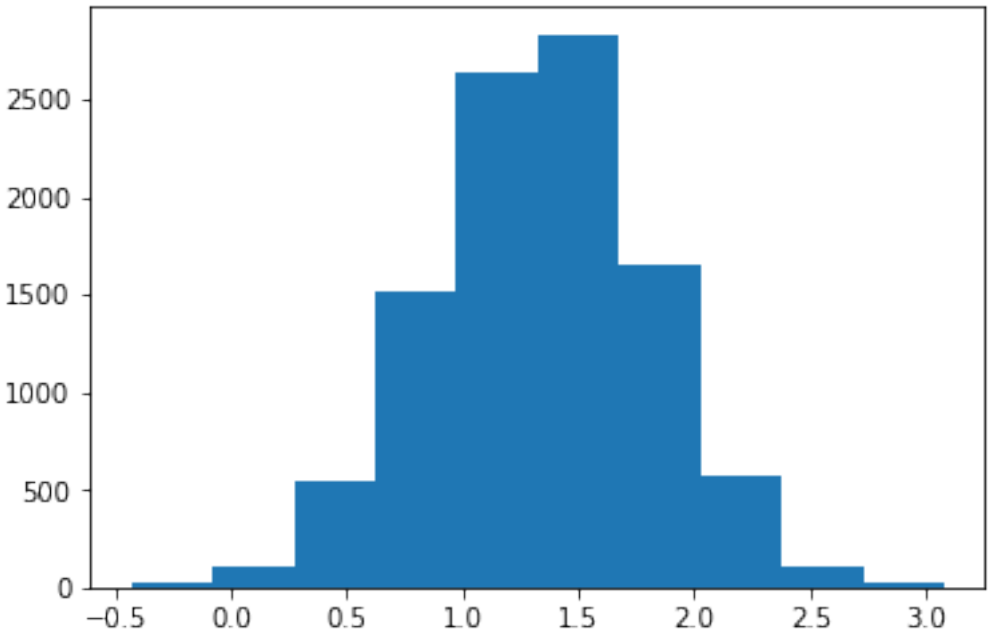
```
In [2]: sample_data.head()
```

```
Out[2]:       user_id   age  drinks_coffee     height
        2402     2874   <21           True  64.357154
        2864     3670  >=21           True  66.859636
        2167     7441   <21          False  66.659561
        507      2781  >=21           True  70.166241
        1817     2875  >=21           True  71.369120
```

1. For 10,000 iterations, bootstrap sample your sample data, compute the difference in the average heights for coffee and non-coffee drinkers. Build a 99% confidence interval using your sampling distribution. Use your interval to start answering the first quiz question below.

```
In [ ]: diffs1 = []
        for _ in range(10000):
            bs = sample_data.sample(200, replace=True)
            cy = bs[bs["drinks_coffee"] == True]
            cn = bs[bs["drinks_coffee"] == False]
            cy_hm = cy["height"].mean()
            cn_hm = cn["height"].mean()
            diff = cy_hm - cn_hm
            diffs1.append(diff)
```

```
In [11]: plt.hist(diffs1);
```



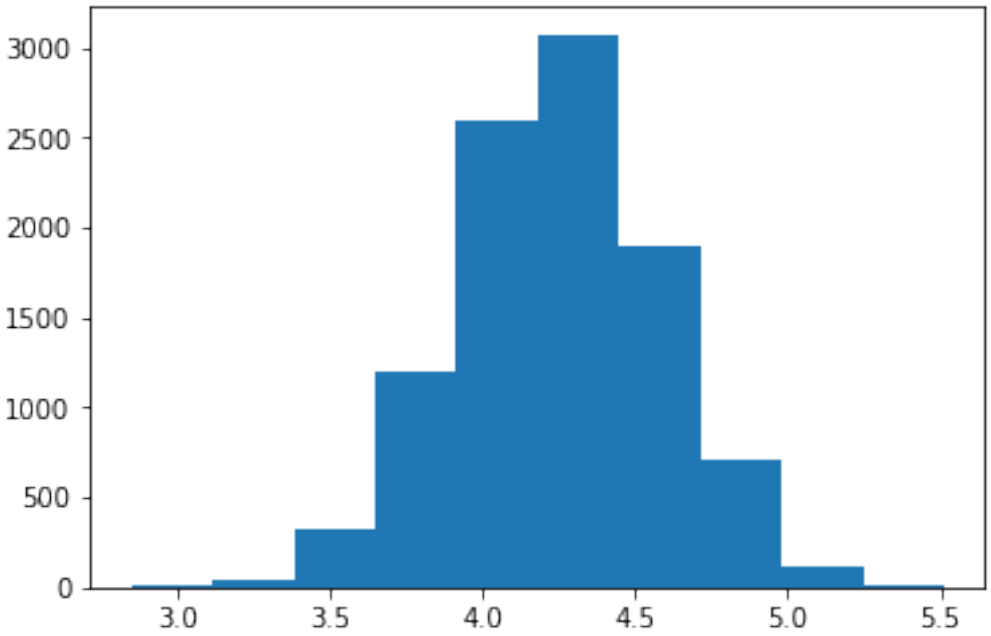

```
In [12]: np.percentile(diffs1, 2.5), np.percentile(diffs1, 97.5)

Out[12]: (0.39656867909086274, 2.2409418186017551)
```

2. For 10,000 iterations, bootstrap sample your sample data, compute the difference in the average heights for those older than 21 and those younger than 21. Build a 99% confidence interval using your sampling distribution. Use your interval to finish answering the first quiz question below.

```
In [17]: diffs2 = []
         for _ in range(10000):
             bs = sample_data.sample(200, replace=True)
             younger = bs[bs["age"] == "<21"]
             older = bs[bs["age"] == ">=21"]
             y_hm = younger["height"].mean()
             o_hm = older["height"].mean()
             diff = o_hm - y_hm
             diffs2.append(diff)

In [19]: plt.hist(diffs2);
```



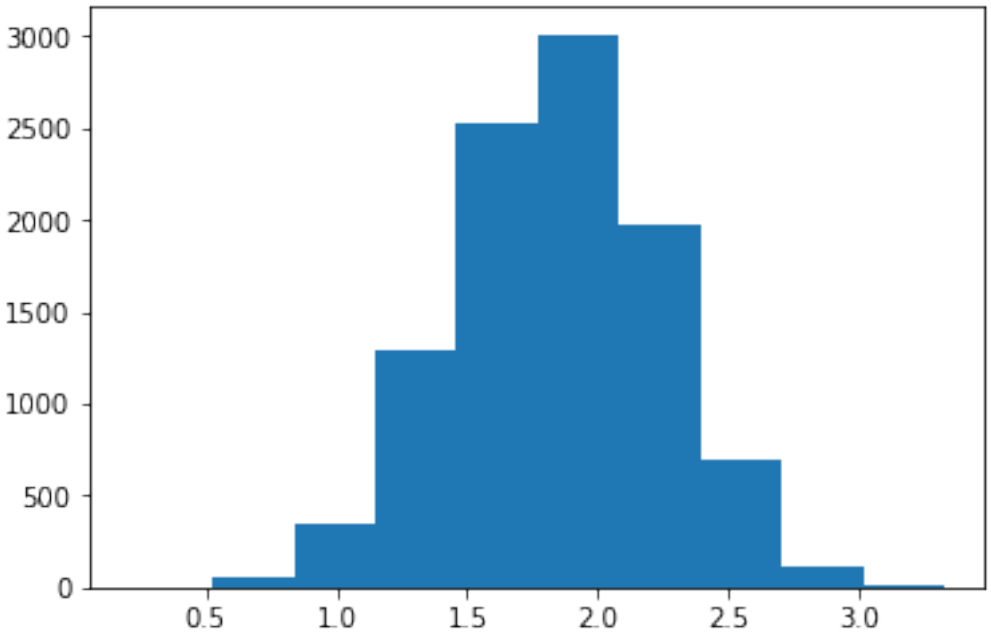

```
In [20]: np.percentile(diffs2, 2.5), np.percentile(diffs2, 97.5)
```

```
Out[20]: (3.576119649609014, 4.8937985328397238)
```

3. For 10,000 iterations bootstrap your sample data, compute the difference in the average height for coffee drinkers and the average height for non-coffee drinkers for individuals under 21 years old. Using your sampling distribution, build a 95% confidence interval. Use your interval to start answering question 2 below.

```
In [29]: diffs3 = []
         for _ in range(10000):
             bs = sample_data.sample(200, replace=True)
             younger = bs[bs["age"] == "<21"]
             cy = younger[younger["drinks_coffee"] == True]
             cn = younger[younger["drinks_coffee"] == False]
             cy_hm = cy["height"].mean()
             cn_hm = cn["height"].mean()
             diff = cn_hm - cy_hm
             diffs3.append(diff)
```

```
In [30]: plt.hist(diffs3);
```




```
In [31]: np.percentile(diffs3, 2.5), np.percentile(diffs3, 97.5)
```

```
Out[31]: (1.0604833025073632, 2.5932349683121609)
```

4. For 10,000 iterations bootstrap your sample data, compute the difference in the average height for coffee drinkers and the average height for non-coffee drinkers for individuals under 21 years old. Using your sampling distribution, build a 95% confidence interval. Use your interval to finish answering the second quiz question below. As well as the following questions.

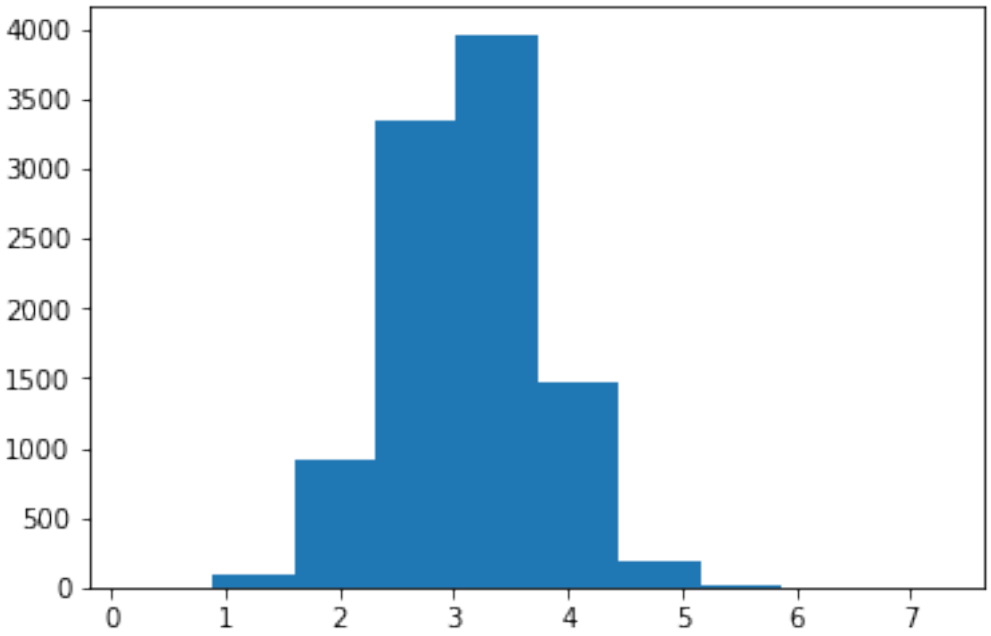
(i'm assuming they mean 21 and over for this one...)

```
In [36]: diffs4 = []
         for _ in range(10000):
             bs = sample_data.sample(200, replace=True)
             older = bs[bs["age"] == ">=21"]
             cy = older[older["drinks_coffee"] == True]
             cn = older[older["drinks_coffee"] == False]
             cy_hm = cy["height"].mean()
             cn_hm = cn["height"].mean()
             diff = cn_hm - cy_hm
             diffs4.append(diff)

         print("done")
```

```
done
```

```
In [37]: plt.hist(diffs4);
```




```
In [38]: np.percentile(diffs4, 2.5), np.percentile(diffs4, 97.5)
```

```
Out[38]: (1.8280535113036145, 4.3961104980494934)
```