What is the impact of sample size?

December 6, 2017

0.0.1 The Impact of Large Sample Sizes

When we increase our sample size, even the smallest of differences may seem significant.

To illustrate this point, work through this notebook and the quiz questions that follow below.

Start by reading in the libraries and 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')
```

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

Out[2]:

	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

`1.` In this case, imagine we are interested in testing if the mean height of all individuals in `full_data` is equal to 67.60 inches. First, use **quiz 1** below to identify the null and alternative hypotheses for these cases.

$H_0 : \mu_{height} = 67.6\ in$ $H_1 : \mu_{height} \neq 67.6\ in$

`2.` What is the population mean height? What is the standard deviation of the population heights? Create a sample set of data using the code below. What is the sample mean height? Simulate the sampling distribution for the mean of five values to see the shape and plot a histogram. What is the standard deviation of the sampling distribution of the mean of five draws? Use **quiz 2** below to assure your answers are correct.

```
In [3]: sample1 = full_data.sample(5)
```

```
In [4]: sheights = []
        for _ in range(10000):
```

```
        s = sample1.sample(5, replace=True)
        h = s['height'].mean()
        sheights.append(h)

    print("done!")
```

```
done!
```

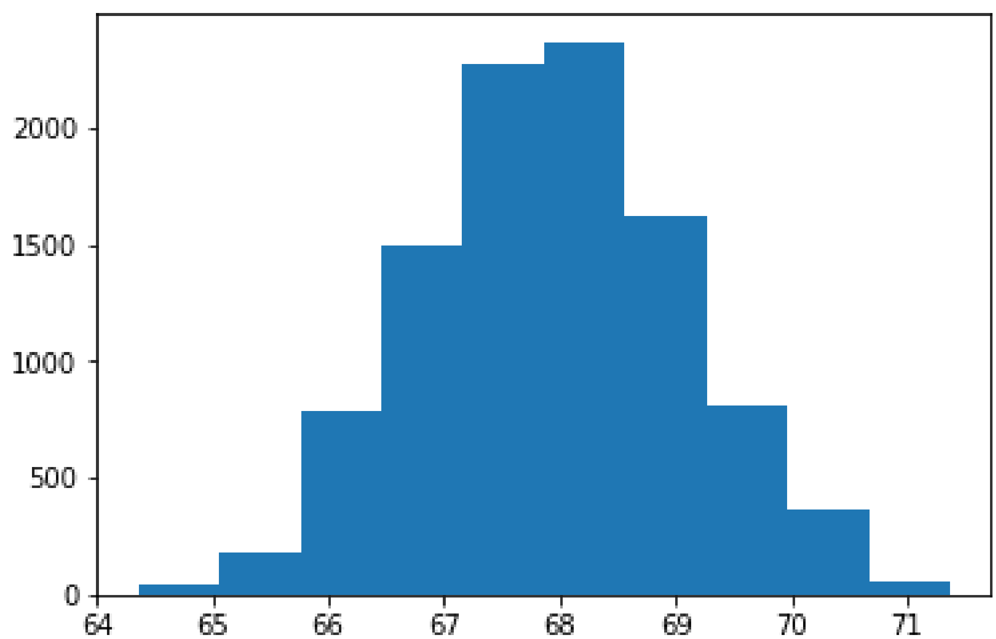
In [5]:
```
sm = np.mean(sheights)
sm
```

Out[5]: 67.902914964404943

In [6]:
```
ssd = np.std(sheights)
ssd
```

Out[6]: 1.141357351999374

In [7]:
```
plt.hist(sheights);
```



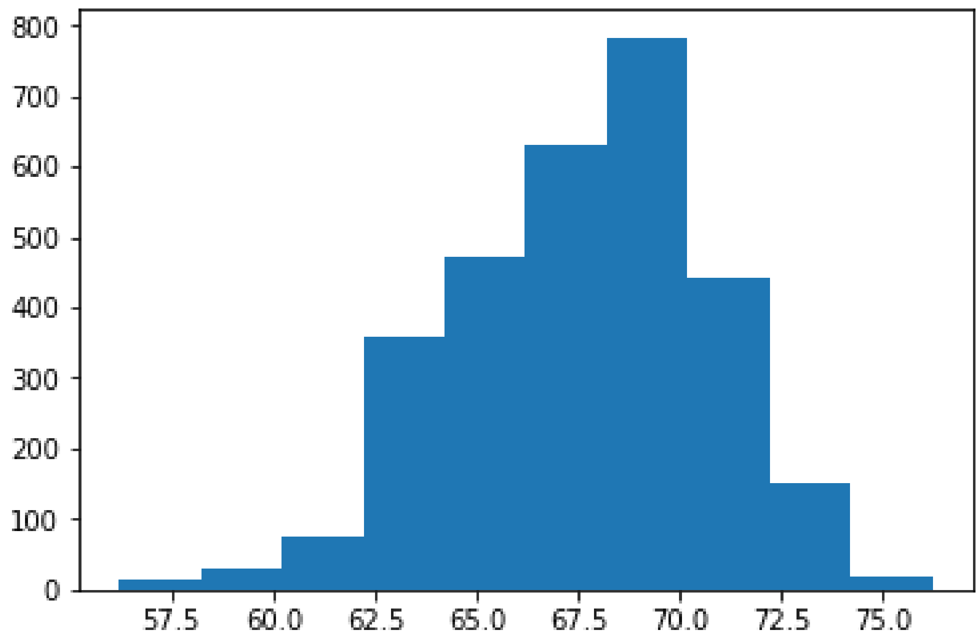
In [8]:
```
umean = full_data['height'].mean()
umean
```

Out[8]: 67.597486973079342

In [9]:
```
ustd = full_data['height'].std()
ustd
```

Out[9]: 3.1194332065503421

```
In [10]: plt.hist(full_data['height']);
```



3. Using the null and alternative set up in question 1 and the results of your sampling distribution in question 2, simulate the mean values you would expect from the null hypothesis. Use these simulated values to determine a p-value to make a decision about your null and alternative hypotheses. Check your solution using **quiz 3** and **quiz 4** below.

Hint: Use the numpy documentation here to assist with your solution.

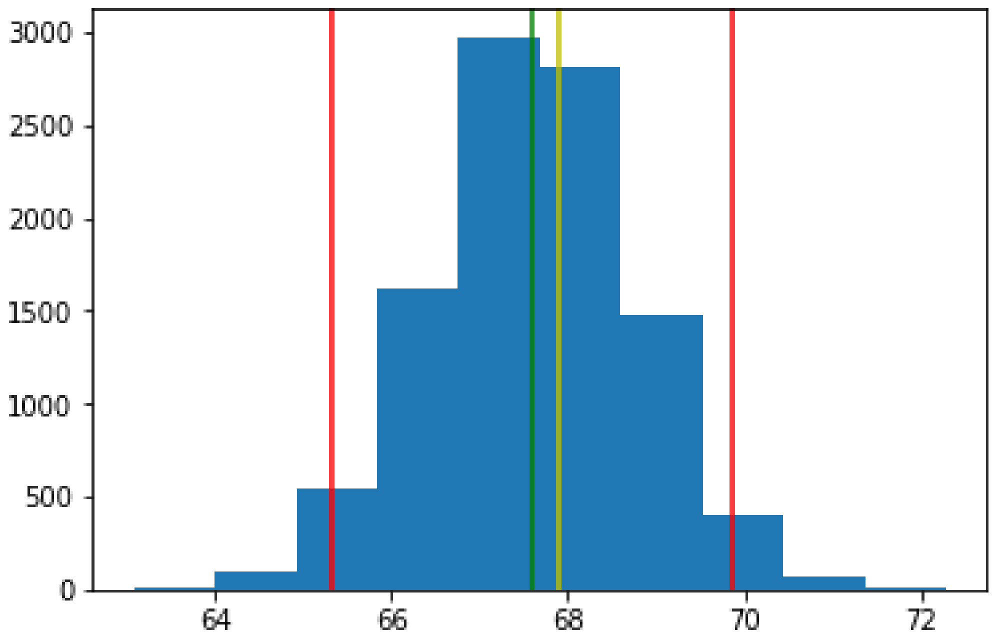
```
In [11]: test_val = 67.6
         null_vals = np.random.normal(test_val, ssd, 10000)
         print("done")
```

done

```
In [12]: p1 = np.percentile(null_vals, 2.5)
         p2 = np.percentile(null_vals, 97.5)
         (p1, p2)
```

Out[12]: (65.316817696525959, 69.857925665577156)

```
In [13]: plt.hist(null_vals);
         plt.axvline(x=test_val, color="g");
         plt.axvline(x=sm, color="y");
         plt.axvline(x=p1, color="r");
         plt.axvline(x=p2, color="r");
```



this helpful bit was given in the quiz, once answered. if how to do this was covered in the material, i surely missed it.

null_mean = 67.60
# this is another way to compute the standard deviation of the sampling distribution theoretically
std_sampling_dist = full_data.height.std()/np.sqrt(5)
num_sims = 10000

null_sims = np.random.normal(null_mean, std_sampling_dist, num_sims)
low_ext = (null_mean - (sample1.height.mean() - null_mean))
high_ext = sample1.height.mean()

(null_sims > high_ext).mean() + (null_sims < low_ext).mean()

4. Now, imagine you received the same sample mean you calculated from the sample in question 2 above, but with a sample of 1000. What would the new standard deviation be for your sampling distribution for the mean of 1000 values? Additionally, what would your new p-value be for choosing between the null and alternative hypotheses you set up? Simulate the sampling distribution for the mean of five values to see the shape and plot a histogram. Use your solutions here to answer the second to last quiz question below.

Hint: If you get stuck, notice you can use the solution from the quiz regarding finding the p-value earlier to assist with obtaining this answer with just a few small changes.

```
In [15]: # this is the sample mean from question 2
         sm

Out[15]: 67.902914964404943

In [16]: # this is the new sample size
         nss = 1000
```

```
In [19]: # this is the population std dev for height
         ustd

Out[19]: 3.1194332065503421

In [23]: # calculate the new std dev of sampling distrubution
         ssd_1000 = ustd / np.sqrt(nss)
         ssd_1000

Out[23]: 0.098645139414615612

In [24]: # get new p values
         null_vals_1000 = np.random.normal(test_val, ssd_1000, nss)
         print("done")

done

In [28]: lowp = (test_val - (sm - test_val))
         highp = sm
         prop_1000 = (null_vals_1000 > highp).mean() + (null_vals_1000 < lowp).mean()
         (lowp, highp, prop_1000)

Out[28]: (67.297085035595046, 67.902914964404943, 0.0030000000000000001)
```

5. Reflect on what happened by answering the final quiz in this concept.

the quiz question/answer indicated we had a sufficiently high p value to not reject the null. i'm unclear how my prop_1000 value reflects that.