The Basics of Statistical Inference

This lesson will introduce statistical inference and point estimates, along with the central limit theorem.

WE'LL COVER THE FOLLOWING

In the last chapter, we learned how to explore data using different graphs and extract information from the data. We discovered relationships between different variables in our datasets, but how do we decide whether these relationships are real or just by coincidence? How do we decide whether we can make a generalization about the dataset or not? This is where *inferential statistics* comes in.

Inferential statistics

Before we see what inferential statistics is all about, we need to understand the concept of a *population* and a *sample*.

A **population** includes all the elements from a set of data that we are studying. For example, if we are interested in finding out how grades of students in a school are affected by alcohol consumption, all the students in that school form a population. A measurable characteristic of the population, such as a mean or standard deviation, is known as a **parameter**.

A **sample** is a subset of the population. In our school example, if we randomly choose a single grade and analyze students in that grade, then those chosen students form a sample. A measurable characteristic of a sample is known as a **statistic**.

Inferential statistics is the branch of statistics that helps us make inferences about the general *population* from a *sample*. It aims at gaining insights about the population from which the data was collected, based on the collected samples. There are two main areas of inferential statistics:

- **Estimation**: This involves taking a statistic of a sample and using that to say something about a population parameter.
- **Hypothesis testing**: This involves answering some research questions about a population using a sample of that population.

Point estimates

Point estimates are estimates of population parameters based on sample data. Let's say we want to measure the average salary of all data scientists in the US, but taking the data of all the data scientists will be cumbersome. Therefore, what we do is take the salary data of 100 random data scientists and take the average of this sample. The average of this sample is known as **sample mean**.

The sample mean is usually not the same as the population mean. The difference could be due to randomness in choosing a sample or biased sampling. Let's investigate this further using our Default of Credit Card Clients Dataset. We will be investigating the `AGE` column of the dataset.

```
import pandas as pd
df = pd.read_csv('credit_card_cleaned.csv')

# Population mean
pop_mean = df['AGE'].mean()

# drawing a sample and calulate mean
sample = df['AGE'].sample(n = 200,random_state = 7)
sample_mean = sample.mean()

diff = pop_mean - sample_mean

print('population mean : ',pop_mean)
print('Sample mean : ',sample_mean)
print('Difference : ',diff)
```

We are considering the entire dataset as our population in this example since we are concerned with only credit card customers. In **line 5**, we calculate the population mean using the function `mean`, and store it in `pop_mean`. Then in **line 8**, we sample from the `AGE` column. We use the function `sample` and give it `n=200` as the number of samples that we want. We also give it a `random_state`. We can give any number as `random_state`. This is just to ensure that every time we run the code with the same `random_state`, we get the same random sample. In the next line, we calculate the mean of the sample by using the `mean` function. In **line 11**, we simply calculate the difference between the two means by subtraction. Then we print the three quantities in **lines 13-15**.

We see that the difference is approximately only 0.59 years. This means that we get a fairly accurate estimate of a population parameter.

Sampling distributions

You might have heard or read somewhere about the normal distribution. In a normal distribution, the data is symmetrically distributed and is gathered in a few standard deviations around the mean. However, real-world data that we collect is not normally distributed. Therefore, when we draw a sample from that data, the sample tends to follow the distribution of the population.

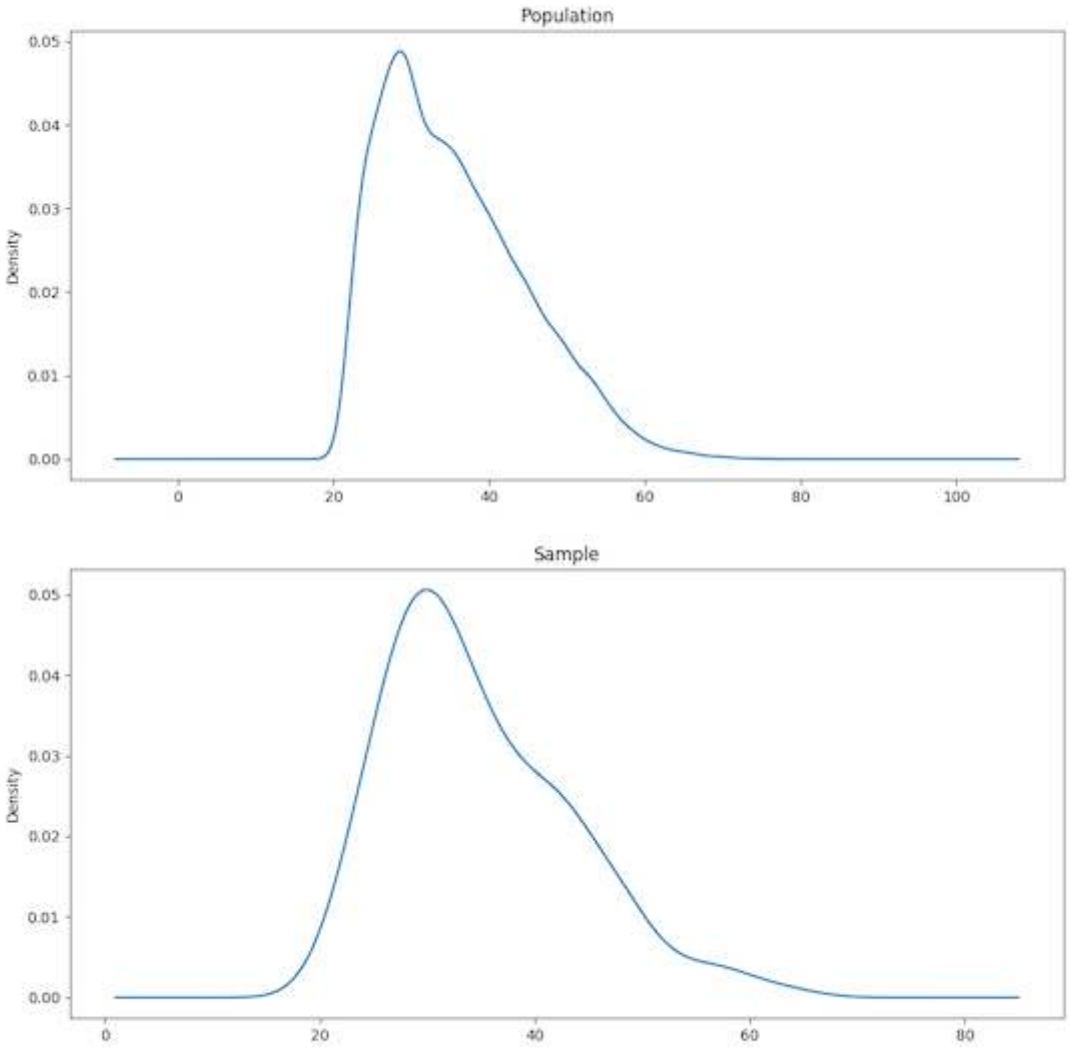
Let's see if this is true using the above sample that we took from the `AGE` column. We will be drawing two density plots, one of the population, and one of the sample, on the same figure to compare them.

```
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_csv('credit_card_cleaned.csv')

fig,subplots = plt.subplots(2,1,figsize = (12,12))

# plot population
df['AGE'].plot(kind = 'density',ax = subplots[0],title = 'Population')
pop_skew = df['AGE'].skew()

# draw sample and plot
sample = df['AGE'].sample(n = 200,random_state = 7)
sample.plot(kind = 'density',ax = subplots[1],title = 'Sample')
sample_skew = sample.skew()

plt.show()
```

```
print('Population Skew : ', pop_skew)
print('Sample Skew : ', sample_skew)
```

Since we want to draw two density plots on the same figure, we initialize our subplots in **line 5** by using `plt.subplots`. We specify that we want the layout to be `2,1` so that there is one column and two rows. We retrieve the subplots axes in `subplots`. In **line 8**, we plot the density plot of `AGE`. We pass `ax=subplots[0]`, to indicate that the top plot in the figure should be this one. In the next line, we calculate the skewness of the `AGE` column.

In **line 12**, we draw a sample from the `AGE` column. We then plot it in the next line and pass `ax=subplots[1]` to indicate that the second plot in the figure should be this one. Then we calculate skewness in the next line. We print the skewness of both the population and the sample in **lines 17-18**.



From the plot, we can see that the distribution of the sample and the population are almost identical. Both are skewed to the right. We can also verify this by looking at the coefficients of skewness of both the sample and the population.

The central limit theorem

The above plots imply that we cannot apply techniques that apply to a normal distribution to a sample drawn from a population. This is where the *central limit theorem* kicks in. The **central limit theorem** is a very important theorem in probability theory and statistics. It states that the distribution of sample means known as **sampling distribution** will be normally distributed. It means the techniques we used on a normal distribution can be used on the sampling distribution too. We can treat the *sample mean* as if it was drawn from a normal distribution. Let's see an example of the theorem.

```
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_csv('credit_card_cleaned.csv')

# draw a sample 500 times and store the mean
sample_means = []
for i in range(500):
    sample = df['AGE'].sample(n = 200)
    s_mean = sample.mean()
    sample_means.append(s_mean)

# plot the distribution of the sample means
sample_means_series = pd.Series(sample_means)
sample_means_series.plot(kind='density')

# compare the population mean and the mean of sample means
mean_of_sample_means = sample_means_series.mean()
pop_mean = df['AGE'].mean()
diff = pop_mean - mean_of_sample_means
print('mean of sample means :', mean_of_sample_means)
print('mean of population :', pop_mean)
print('difference : ',diff)
print('skew of sample means : ',sample_means_series.skew())
```

In **lines 6-10**, our aim is to sample the `AGE` column 500 times and then store the mean of each sample in a list. To accomplish this, we first initialize an

the mean of each sample in a list. To accomplish this, we first initialize an

empty list in **line 6**. We call this `sample_means`. Then in the next line, we use a `for` loop. The `for` loop will run 500 times. In the `for` loop,

- We first draw a sample of 200 items from the `AGE` column. We do that on **line 8**.
- Then we take the mean of the sample using the `mean` function in **line 9** and call it `s_mean`.
- Now we store this `s_mean` in our `sample_means` list by using the `append` method of the list in **line 10**. Because of the loop, **lines 8-10** will run 500 times and we will get 500 sample means in the list `sample_means`.

Now we want to plot the distribution of the sample means that we have collected in `sample_means`. To do this, we first convert our list into a Pandas *Series* in **line 13** by using the function `pd.Series`. We give it our list and it returns a Series. We call it `sample_means_series`. Then we plot it in the next line. By looking at the plot, we can see that it follows roughly a normal distribution as informed by the central limit theorem.

Another interesting point that the central limit theorem implies is that the mean of the sample means is close to the actual population mean. To observe this, we calculate the means of both the population and the sample means and then their difference in **lines 17-19**. We can see from the output of **line 22** that this difference is minor, and this difference will reduce if we increase the number of samples taken from 500. We also observe from the output of **line 23** that the sample means have almost 0 skew.

This is how we can estimate population parameters using samples. In the next lesson, we will continue our discussion and focus on confidence intervals.