Introduction to Inferential Statistics

1. What is the difference between the population and the sample? What is the difference between parameter and statistic?

Population and parameter are the universal set.

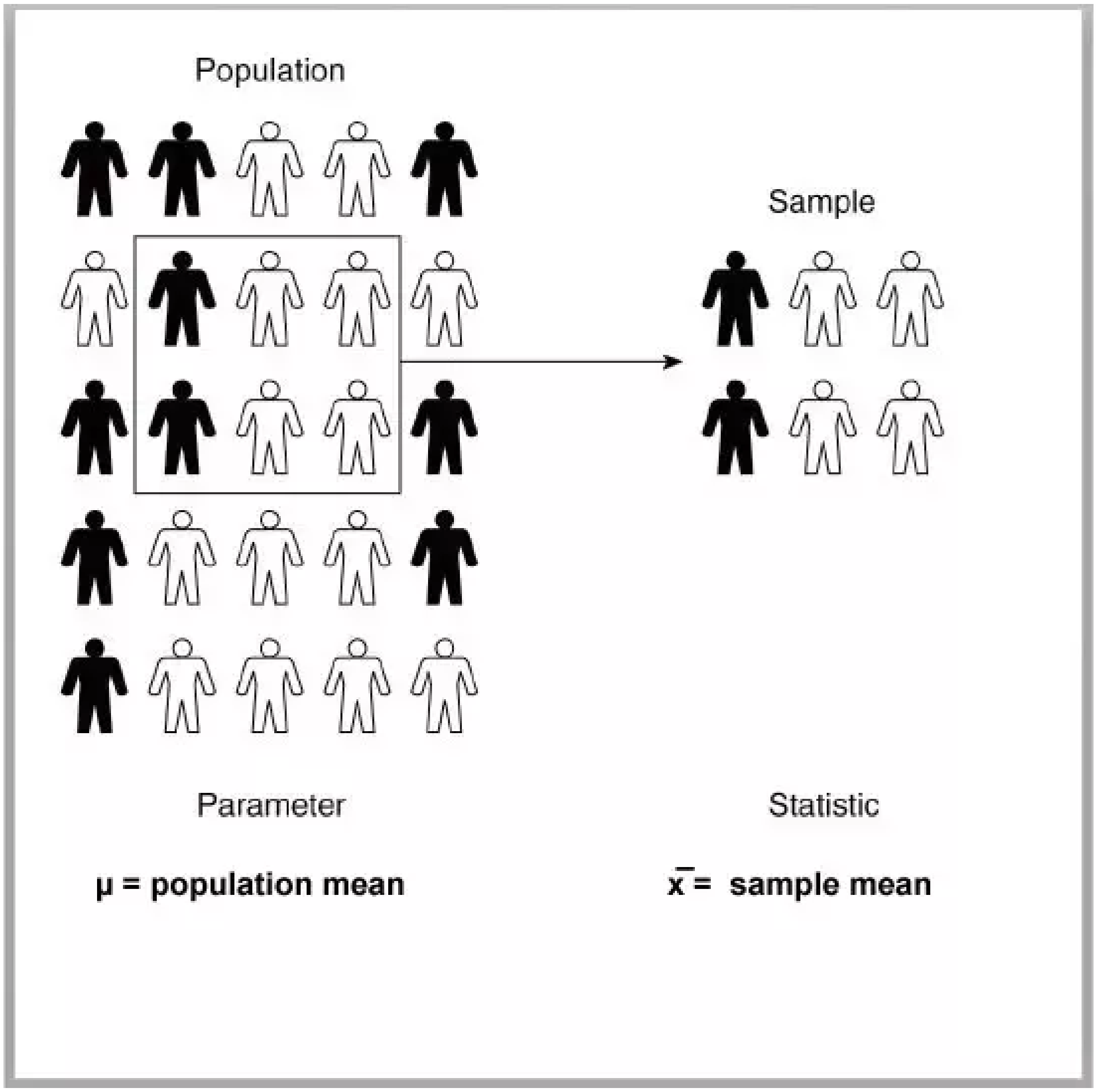
In statistics, a population is an entire pool from which a statistical sample is drawn. Examples of populations can be the number of newborn babies in North America, the total number of tech startups in Asia.

A parameter is any summary number, like an average or percentage, that describes the entire population.

Since we can't measure the entire population, therefore, measuring its parameter is also difficult.

For example, the average height of adult women in the United States is a parameter that has an exact value—we just don't know what it is! or the average height of all CFA exam candidates in the world, the mean weight of U.S. taxpayers, and so on.

We know what we have to collect and that collection is termed as Sample (a specific set from the population). Parameters associated with it would be sample parameters like sample mean or sample standard deviation and are referred to as statistic.



Uniform distribution

2. How to choose loc and scale for a continuous random variable following uniform distribution?

In scipy.stats.uniform(), the parameters **loc** and **scale** refer to the starting point and range of the uniform distribution. Using the parameters **loc** and **scale**, one obtains the uniform distribution on **[loc, loc + scale]**. For a continuous random variable following uniform distribution on [1, 4], **loc** and **scale** will be 1 and 3 respectively.

3. To calculate continuous uniform distribution using uniform.cdf(), we included the x value to calculate the probability to the right i.e. 96 (96, loc=90, scale=11). But for the discrete uniform distribution, we took a step further to calculate the probability to the right i.e. 96 (93, loc=90, scale=11). What is the reason for doing so?

The cdf() function behaves differently for a continuous distribution and a discrete distribution. In the case of a continuous distribution, the cdf is a strictly monotonic increasing continuous function;

whereas the cdf of a discrete distribution is a step function. Here, we are using the continuous uniform distribution to approximate the discrete uniform distribution.

The question asks to find the probability that less than or equal to 92 books (refer to the problem statement) will be sold on a given day. Given that we are using the continuous uniform distribution, we will try to find the area under the uniform curve that is below 93. The correct code to find the same is

```
uniform.cdf(93, loc=90, scale=11)
```

The intuitive approach is, to sum up, the probability of buying 90, 91, and 92 books. As the probability of each is 1/11, the cdf will be

```
3 * (1/11) = 3/11 = 0.2727
```

Normal distribution and z-score

4. What is z-score and how is it used in real-life scenarios?

A z-score (also called the standard score) measures how many standard deviations below or above the mean a data point lies. The z-score is very useful as it enables us to compare two scores coming from two different normal populations. The two scores might be on two different scales however we can compare them using the z-score.

Real-life application – Suppose you have appeared for two different competitive exams having different scoring systems. How will you compare your scores' in two exams? Let's assume that the competitive exams are popular and the distributions of their scores follow two different normal distributions. To compare your scores (coming from two different normal populations), you need to standardize each of your scores. Then, you can easily compare them.

5. What does norm.ppf() do?

The norm.ppf() function is the inverse of the norm.cdf() function. It takes a percentage p and returns a point such that the probability of the normal random variable being less than or equal to that number is p%. Thus, it just does the opposite work of norm.cdf().

For example, if the percentage p is equal to 0.92, you will get the point below which 92% of data falls. This also means that 8% of data falls above that point.

Hypothesis testing

6. Why do you get the below error and how to resolve it?

```
ttest_1samp() got an unexpected keyword argument ‘alternative’
```

The parameter alternative = 'greater' and alternative = 'less' has been introduced in the SciPy version 1.6.0. onwards. So, you will get an error if you run the code in an old SciPy version. When you remove the parameter 'alternative', you get the p-value for the two-tailed test as implemented by the previous version. But, we would recommend that you install the latest SciPy version in your system to perform the hypothesis test.

You can run the below command in the jupyter notebook to install the SciPy version 1.6.1.

```
!pip install scipy==1.6.1
```

After installing the new SciPy version, restart the jupyter notebook and run the below code to check the current Scipy version in your system. If you get ‘1.6.1’ as the output, then you can use the ‘alternative’ parameter with values 'less' and 'greater'.

```
import scipy
scipy.__version__
```