Randomness_and_Reproducibility

October 18, 2019

1 Randomness and Reproducibility

The concept of randomness is a cornerstone for statistical inference when drawing samples from larger populations.

In this tutorial, we are going to cover the following:

- Randomness and its uses in python.

- Utilizing python seeds to reproduce analysis.

- Generating random variables from a probability distribution.

- Random sampling from a population.

1.1 What is Randomness?

If we have complete randomness, our estimates of means, proportions, and totals are ubiased. This means our estimates are equal to the population values on average.

In Python, we refer to randomness as the ability to generate data, strings, or, more generally, numbers at random.

However, when conducting analysis it is important to consider reproducibility. If we are creating random data, how can we enable reproducible analysis?

We do this by utilizing pseudo-random number generators (PRNGs). PRNGs start with a random number, known as the seed, and then use an algorithm to generate a psuedo-random sequence based on it.

This means that we can replicate the output of a random number generator in python simply by knowing which seed was used.

We can showcase this by using the functions in the python library *random*.

1.1.1 Setting a Seed and Generating Random Numbers

```
In [ ]: import random

In [ ]: random.seed(1234)

        random.random()

In [ ]: random.seed(1234)

        random.random()
```

1.1.2   Random Numbers from Real-Valued Distributions

Uniform

```
In [ ]: random.uniform(25,50)

In [ ]: unifNumbers = [random.uniform(0,1) for _ in range(1000)]

In [ ]: unifNumbers
```

Normal

```
In [ ]: mu = 0

        sigma = 1

        random.normalvariate(mu, sigma)

In [ ]: mu = 5

        sigma = 2

        random.normalvariate(mu, sigma)

In [ ]: mu = 0

        sigma = 1

        [random.normalvariate(mu, sigma) for _ in range(10000)]
```

1.1.3   Random Sampling from a Population

From lecture, we know that **Simple Random Sampling (SRS)** has the following properties:

- Start with known list of N population units, and randomly select n units from the list
- Every unit has **equal probability of selection = n/N**
- All possible samples of size n are equaly likely
- Estimates of means, proportions, and totals based on SRS are **UNBIASED** (meaning they are equal to the population values on average)

```
In [ ]: import random
        import numpy as np

In [ ]: mu = 0

        sigma = 1

        Population = [random.normalvariate(mu, sigma) for _ in range(10000)]

In [ ]: SampleA = random.sample(Population, 500)
        SampleB = random.sample(Population, 500)
```

```
In [ ]: np.mean(SampleA)

In [ ]: np.std(SampleA)

In [ ]: np.mean(SampleB)

In [ ]: np.std(SampleB)

In [ ]: means = [np.mean(random.sample(Population, 1000)) for _ in range(100)]

        np.mean(means)

In [ ]: standarddevs = [np.]
```