Probability

Probability

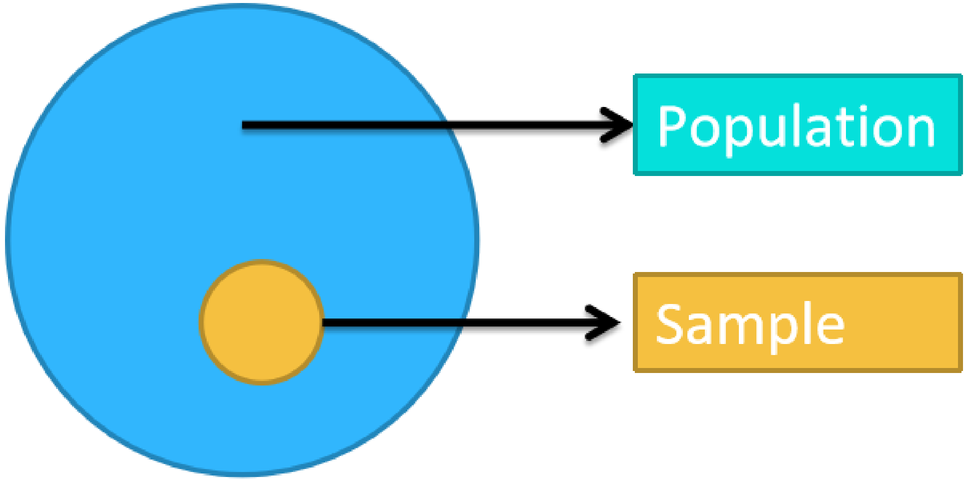
- Probability is the numerical measure of the likelihood that an event will occur
- The probability of an event is equal to the number of outcomes divided by the total number of possibile outcome
  - **p(A) = n(A) / n(s)**
- Probability of an event must be between 0 and 1 inclusively
  - **0 ≤ p(A) ≤ 1 for any event A**

Terminologies

- Experiment
  - A process that produces outcomes
- Event
  - An outcome of an experiment
- Sample space
  - The set of all events for an experiment

Population vs Sample

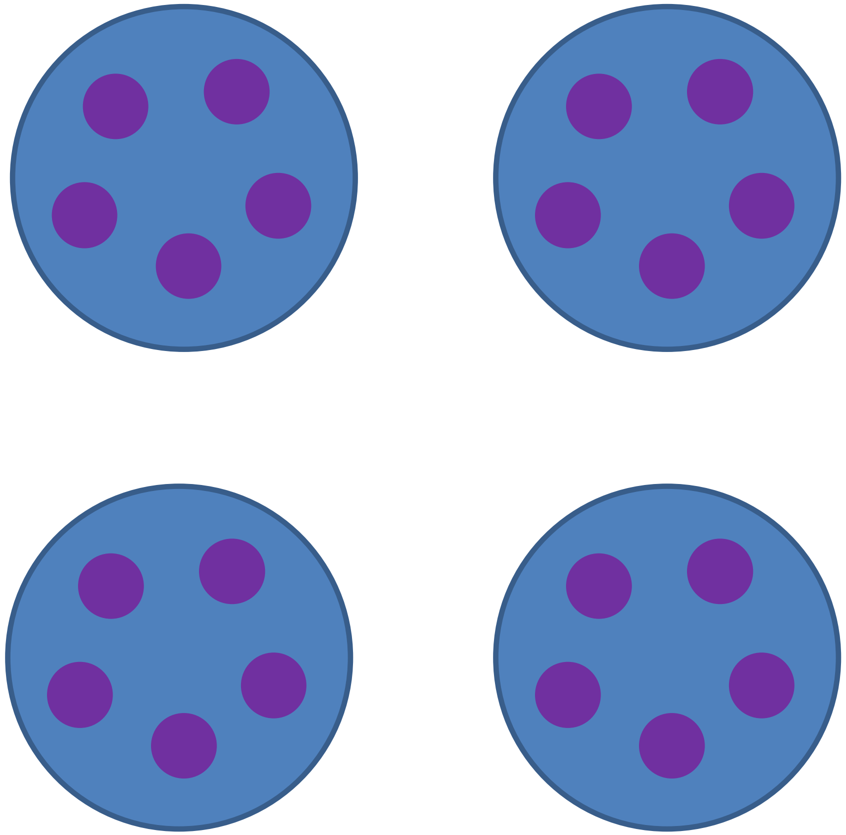
- A dataset may consist of the elements of a population of interest or it may take the form of a sample
- A Sample is a subset of a population



Sample

- Sample data is used due to the time and cost required to analyze an entire population
- It is critical that the sample be really random
- Population are usually denoted in upper case and Samples are usually denoted in lower case

Sample Size vs Number of Samples



Here,

Number of Samples = 4

Sample Size = 5

4 Samples with the sample size of 5

Summary Statistics

Central Tendency

1. Mean
2. Median
3. Mode

Dispersion

1. Range
2. Quartile
3. Interquartile Range
4. Variance
5. Standard Deviation

MEASURE OF CENTRAL TENDENCY

Measure of Central Tendency

- Shows the middle or center of a sample or a population
- Three widely used measures of central tendency
  - *Mean*
  - *Median*
  - *Mode*

Generating Population and Extracting Samples

In [1]:
```python
import numpy as np
```

Generating a population of size 1,00,000 with random integers between 1 and 100

In [6]:
```python
population=np.random.randint(low=1,high=100,
                             size=1_00_000)
print(population)
print("Length of Population",len(population))
```

```
[92 42 37 ... 90 39 94]
Length of Population 100000
```

Extracting a sample of size 20

In [10]:
```python
sample=np.random.choice(population,size=20)
print(sample)
print("Length of Sample",len(sample))
```

```
[59 20  4 86 73 70 45  3  4 14 93 27 77  7  3 86 73 79 36 48]
Length of Sample 20
```

Mean

- Add all the elements in a dataset and then divide it by the number of elements

Population Mean	Sample Mean
$\mu = \frac{\sum_{i=1}^{N} x_i}{N}$	$\overline{X} = \frac{\sum_{i=1}^{n} x_i}{n}$
N = number of items in the population	n = number of items in the sample

```python
pop_mean=np.mean(population)
print("Mean of population",pop_mean)
```

```
Mean of population 49.93998
```

```python
sample_mean=np.mean(sample)
print("Mean of sample",sample_mean)
```

```
Mean of sample 45.35
```

Median

- Midpoint
- Half of the observations are below median and half are above it

```
In [13]:
pop_median=np.median(population)
print("Median of population",pop_median)
```

```
Median of population 50.0
```

```
In [14]:
sample_median=np.median(sample)
print("Median of sample",sample_median)
```

```
Median of sample 46.5
```

Mode

- Most Commonly observed value

In [15]:
```
from statistics import mode
```

In [16]:
```
pop_mode=mode(population)
print("Mode of population",pop_mode)
```

```
Mode of population 59
```

In [28]:
```
sample_mode=mode(sample)
print("Mode of sample",sample_mode)
```

```
Mode of sample 48
```

MEASURE OF DISPERSION

Measure of Dispersion

- Shows how spread out the elements of a sample or populations are.
- Most important measure of dispersion
  - Range
  - Quartile
  - Interquartile range
  - Variance
  - Standard Deviation

Range

- Difference between its largest and smallest elements
- Hypersensitive to outlier

```
print(population)
pop_range=np.max(population)-np.min(population)
print("Range of population",pop_range)
```

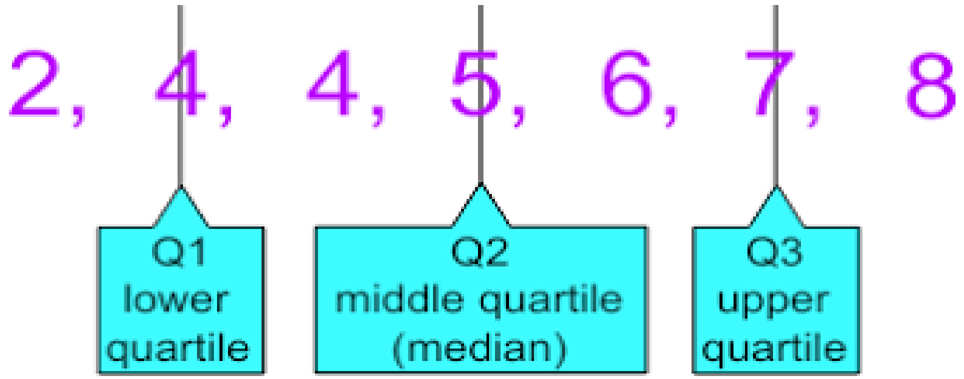
```
[92 42 37 ... 90 39 94]
Range of population 98
```

```
print(sample)
sample_range=np.max(sample)-np.min(sample)
print("Range of sample",sample_range)
```

```
[59 81 89 26 84 85 66 24 28 74 21  4  3 77 11 93 48 48 83 68]
Range of sample 90
```

Quartiles

- Measures of central tendency that divide a group of data into four subgroups
- Q1 -> 25% of dataset is below first quartile
- Q2 -> 50% of dataset is below second quartile **[MEDIAN]**
- Q3-> 75% of dataset is below third quartile



Quartiles of Population

```python
# Population Quartile
pop_q1=np.percentile(population,25)
pop_q2=np.percentile(population,50)
pop_q3=np.percentile(population,75)
print("First Quartile of population",pop_q1)
print("Second Quartile of population",pop_q2)
print("Third Quartile of population",pop_q3)
```

```
First Quartile of population 25.0
Second Quartile of population 50.0
Third Quartile of population 75.0
```

Quartiles of Sample

```python
# Sample Quartile
sample_q1=np.percentile(sample,25)
sample_q2=np.percentile(sample,50)
sample_q3=np.percentile(sample,75)
print("First Quartile of sample",sample_q1)
print("Second Quartile of sample",sample_q2)
print("Third Quartile of sample",sample_q3)
```

```
First Quartile of sample 25.5
Second Quartile of sample 62.5
Third Quartile of sample 81.5
```

Interquartile Range (IQR)

- Range of Values between Q1 and Q2
- Range of the middle half
- Less influenced by extremes
  - **`IQR = Q3 - Q1`**

```
pop_IQR=pop_q3-pop_q1
print("IQR of population",pop_IQR)
```

```
IQR of population 50.0
```

```
sample_IQR=sample_q3-sample_q1
print("IQR of sample",sample_IQR)
```

```
IQR of sample 56.0
```

Variance

- Average squared difference between the elements of a dataset and the mean value of the dataset
- The more spread out the elements, larger the variance

Population Variance	Sample Variance
$$\sigma^2 = \frac{\sum_{i=1}^{N}(x_i - \mu)^2}{N}$$	$$s^2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}$$
$\sigma^2 =$ population variance $x_i =$ value of i^{th} element $\mu =$ population mean $N =$ population size	$s^2 =$ sample variance $x_i =$ value of i^{th} element $\bar{x} =$ sample mean $n =$ sample size

Variance of Population and Sample

```python
pop_var=np.var(population)
print("Variance of Population",pop_var)
```

```
Variance of Population 814.1846375995999
```

```python
sample_var=np.var(sample,ddof=1)
print("Variance of Sample",sample_var)
```

```
Variance of Sample 944.1473684210528
```

Standard Deviation

- Standard deviation is the square root of the variance
- This ensures that the deviation of a dataset is measured in the same units as the dataset

Standard Deviation of Population and Sample

```
print("Population SD",np.sqrt(pop_var))
print("Sample SD",np.sqrt(sample_var))
```

```
Population SD 28.533920824162948
Sample SD 30.72698111466619
```

```
pop_sd=np.std(population)
print("Standard Deviation of Population",pop_sd)
```

```
Standard Deviation of Population 28.533920824162948
```

```
sample_sd=np.std(sample,ddof=1)
print("Standard Deviation of Sample",sample_sd)
```

```
Standard Deviation of Sample 30.72698111466619
```