1. Define and briefly elaborate Central Tendency using measures with examples?

Central Tendency Measures Measures of central tendency are the measures that are used for describing the data using a single value. Mean, median and mode are the three measures of central tendency and are frequently used to describe data and make comparisons between different datasets.

Measures of central tendency help users to summarize and comprehend the data.

Mean: Mean is the arithmetic average value of the data and is one of the most frequently used measures of central tendency.

Assume that the data has n records in a sample and let Xi be the value of the i th record. Then the mean value of the data is given by: –

Mean or average= X = – X1 + X 2 + ...+ Xn

n = Xi i=1 n The symbol X is frequently used to represent the estimated value of the mean from a sample.

Mean = $\mu = \frac{1}{n}\sum_{i}^{n} x_i$

= sum of all the terms / total number of sum

the entire population is available and if we calculate mean based on records from the entire population, then we have the population mean which is usually denoted by μ. Amongst all the measures of central tendency,

mean is the most frequently used measure, since it uses values of all records (all Xi values) in the data set (either sample or population) to calculate the mean value. The salary of graduating students from a business school; then the average salary (or mean salary) is given by –

Mean = (270 + 220 + 240 + 250 +180 + 300 + 240 + 235 + 425 + 240)×1000 / 10 = 26000

2.Median : mid point of sorted data

N is odd = n+1 / 2

N is even = (n/2) th + (n/2 +1) th

Ex: 10,20,30,40,50,60 , n =6

Median = 30+40 / 2 = 35

3.Mode : most frequent term :

10,20,30,40,20,40,20,60

Here mode is 20

To apply central tendency measures (mean, median, mode) using SciPy, you can use functions available in the `scipy.stats` module. Here's how you can use SciPy to calculate these measures:

1. Mean :

You can calculate the mean using the `scipy.stats` module's `mean()` function.

```python
from scipy.stats import mean

data = [1, 2, 3, 4, 5]
mean_value = mean(data)
print("Mean:", mean_value)
```

2. Median:

You can calculate the median using the `scipy.stats` module's `median()` function.

```python
from scipy.stats import median

data = [1, 2, 3, 4, 5]
```

```
median_value = median(data)
print("Median:", median_value)
```

3. Mode:

SciPy does not have a built-in function to directly calculate the mode. However, you can use other libraries like NumPy to compute the mode.

```python
import numpy as np

data = [1, 2, 3, 4, 5, 2, 3, 4, 5]
mode_value = np.mode(data)
print("Mode:", mode_value)
```

Make sure you have installed SciPy and NumPy libraries in your Python environment using `pip install scipy numpy`. These examples assume you have imported the necessary functions from the `scipy.stats` and `numpy` modules.

2. What do you understand by the empirical rule of normal distribution?

The empirical rule, also known as the 68–95–99.7 rule, is a statistical rule that states that in a normal distribution, 68% of values are within one standard deviation of the mean, 95% within two standard deviations, and 99.7% within three standard deviations. It's also known as the three-sigma rule.

The empirical rule can be used to quickly get an overview of data and check for outliers or extreme values. It's often used in statistics for forecasting final outcomes.

The empirical rule is an approximation, and there are always chances of outliers that don't fall in the distribution. For example, in the social sciences, a result may be considered "significant" if its confidence level is of the order of a two-sigma effect.

To calculate the probability density function (PDF), cumulative distribution function (CDF), and other properties of the normal distribution using SciPy, you can utilize the `scipy.stats.norm` module, which provides various functions for working with the normal distribution. Here's how you can use SciPy to work with the normal distribution:

1. Probability Density Function (PDF):

To compute the probability density function (PDF) of the normal distribution at a given point, you can use the `pdf()` function.

```python
from scipy.stats import norm

# Parameters for the normal distribution
mean = 0  # Mean of the distribution
std_dev = 1  # Standard deviation of the distribution

# Calculate the PDF at a specific point (e.g., x = 0)
x = 0
pdf_value = norm.pdf(x, mean, std_dev)
print("PDF at x =", x, ":", pdf_value)
```

2. Cumulative Distribution Function (CDF) :

To compute the cumulative distribution function (CDF) of the normal distribution up to a given point, you can use the `cdf()` function.

```python
# Calculate the CDF up to a specific point (e.g., x = 0)
x = 0
```

```
cdf_value = norm.cdf(x, mean, std_dev)
print("CDF up to x =", x, ":", cdf_value)
```

3. Quantiles:

To compute quantiles (percentiles) of the normal distribution, you can use the `ppf()` function.

```python
# Calculate the quantile for a given probability (e.g., p = 0.95)
p = 0.95
quantile_value = norm.ppf(p, mean, std_dev)
print("Quantile for p =", p, ":", quantile_value)
```

4. Random Sampling:

To generate random samples from the normal distribution, you can use the `rvs()` function.

```python
# Generate 10 random samples from the normal distribution
random_samples = norm.rvs(mean, std_dev, size=10)
print("Random samples:", random_samples)
```

```

These are some of the basic operations you can perform with the normal distribution using SciPy. Adjust the parameters `mean` and `std_dev` as needed for your specific use case.

3. Describe the Hypothesis Testing and why do we conduct it?

Hypothesis testing is a statistical method used to make inferences about a population parameter based on sample data. It involves formulating a hypothesis about the population parameter and then using sample data to assess the likelihood of the hypothesis being true.

The two main types of hypotheses in hypothesis testing are:

1. Null Hypothesis (H0): The null hypothesis represents the default assumption or the status quo. It states that there is no significant difference or effect, or no relationship between variables. It is typically denoted as H0.

2. Alternative Hypothesis (H1 or Ha): The alternative hypothesis contradicts the null hypothesis. It represents what the researcher is trying
```

to find evidence for. It states that there is a significant difference, effect, or relationship between variables. It is denoted as H1 or Ha.

The process of hypothesis testing involves the following steps:

1. Formulating the hypotheses: Define the null hypothesis (H0) and the alternative hypothesis (H1 or Ha) based on the research question.

2. Choosing a significance level (α): The significance level (α) determines the probability of rejecting the null hypothesis when it is actually true. Commonly used values for α are 0.05 or 0.01, indicating a 5% or 1% chance of a Type I error, respectively.

3. Collecting sample data: Gather data from a sample that is representative of the population under study.

4. Selecting an appropriate statistical test: Choose a statistical test based on the type of data and the research question.

5. Calculating the test statistic: Calculate the test statistic using the sample data and the chosen statistical test.

6. Determining the critical value or p-value: Based on the test statistic, determine whether to reject the null hypothesis by comparing the test statistic to the critical value from the distribution or by calculating the p-value.

7. Making a decision: If the test statistic falls in the rejection region (beyond the critical value) or if the p-value is less than the significance level (α), reject the null hypothesis. Otherwise, fail to reject the null hypothesis.

8. Drawing conclusions: Based on the decision, draw conclusions about the population parameter and the research question.

Example:

Suppose a researcher wants to test whether a new drug is effective in reducing blood pressure. The null hypothesis (H0) would be that the drug has no effect on blood pressure, while the alternative hypothesis (H1) would be that the drug does have an effect on blood pressure.

The researcher collects data from a sample of patients, administers the drug to one group (treatment group) and a placebo to another group

(control group), and measures their blood pressure before and after treatment.

After analyzing the data using a paired t-test, the researcher calculates the test statistic and compares it to the critical value or calculates the p-value. If the test statistic falls in the rejection region or if the p-value is less than the significance level (e.g., 0.05), the researcher rejects the null hypothesis and concludes that the drug is effective in reducing blood pressure. Otherwise, the researcher fails to reject the null hypothesis.

Hypothesis testing using SciPy involves using functions from the `scipy.stats` module to perform various statistical tests. Here's an overview of how to conduct hypothesis testing using SciPy:

1. Choose the appropriate test:

Select the appropriate statistical test based on the research question, data type, and assumptions. Common tests include
t-tests, ANOVA, chi-square tests, and more.

2. Set up hypotheses:

Formulate the null hypothesis (H0) and alternative hypothesis (H1). The null hypothesis represents the default assumption or no effect, while

the alternative hypothesis contradicts the null hypothesis and represents the effect you are testing for.

3. Collect and preprocess data:

Gather the relevant data for analysis. Ensure that the data meet the assumptions of the chosen test, such as normality and independence.

4. Perform the test:

Use the appropriate function from `scipy.stats` to perform the hypothesis test. Pass the necessary arguments, such as sample data and test parameters, to the function.

For example, to conduct a t-test for the difference in means, you can use `ttest_ind()` for independent samples or `ttest_rel()` for paired samples.

5. Interpret the results :

Examine the test statistic, p-value, and any other relevant outputs from the hypothesis test.

Compare the p-value to the chosen significance level (α) to determine whether to reject the null hypothesis.

If the p-value is less than α, reject the null hypothesis and conclude that there is evidence for the alternative hypothesis. Otherwise, fail to reject the null hypothesis.

Here's an example of how to perform a t-test for the difference in means using SciPy:

```python
from scipy.stats import ttest_ind

# Sample data
group1 = [23, 25, 28, 32, 35]
group2 = [18, 20, 24, 28, 30]

# Perform independent t-test
t_statistic, p_value = ttest_ind(group1, group2)

# Interpret the results
alpha = 0.05
print("t-statistic:", t_statistic)
print("p-value:", p_value)
if p_value < alpha:
    print("Reject the null hypothesis")
else:
    print("Fail to reject the null hypothesis")
```

```

This example demonstrates how to conduct an independent samples t-test to compare the means of two groups and interpret the results based on the p-value and chosen significance level (α). Adjust the test and data accordingly for other types of hypothesis tests.

5. What is the difference between Type-I and Type-II error.

Type-I and Type-II errors are two types of errors that can occur in hypothesis testing:

1. Type-I Error (False Positive):
   Type-I error occurs when the null hypothesis (H0) is incorrectly rejected when it is actually true.
   It represents the situation where the researcher concludes that there is a significant effect, difference, or relationship when there is none.
   The probability of making a Type-I error is denoted by the significance level (α), which is the probability of rejecting the null hypothesis when it is true.
   A Type-I error is also known as a false positive or alpha error.
```

2. Type-II Error (False Negative):

Type-II error occurs when the null hypothesis (H0) is incorrectly not rejected when it is actually false.

It represents the situation where the researcher fails to detect a significant effect, difference, or relationship when there is one.

The probability of making a Type-II error is denoted by β (beta), which is the probability of failing to reject the null hypothesis when it is false.

The power of a statistical test (1 - β) is the probability of correctly rejecting the null hypothesis when it is false.

A Type-II error is also known as a false negative or beta error.

```
from scipy.stats import ttest_ind

# Sample data
group1 = [23, 25, 28, 32, 35]
group2 = [18, 20, 24, 28, 30]

# Perform independent t-test
t_statistic, p_value = ttest_ind(group1, group2)

# Set significance level
alpha = 0.05
```

```
# Determine Type-I error
if p_value <= alpha:
    print("Type-I Error: Null hypothesis rejected when it is actually true")

# Determine Type-II error (indirectly through power)
power = 1 - beta  # beta is the probability of Type-II error
print("Type-II Error (Beta):", beta)
```

In summary:

Type-I error involves incorrectly rejecting a true null hypothesis (false positive).
Type-II error involves incorrectly failing to reject a false null hypothesis (false negative).

These two types of errors are inversely related: decreasing the probability of one type of error usually increases the probability of the other type of error. Therefore, researchers need to consider both types of errors when interpreting the results of hypothesis testing and selecting an appropriate significance level.

6. Confidence Interval, Significance Level, and P-Value are fundamental concepts in statistics, particularly in hypothesis testing and estimation:

1. Confidence Interval:

A confidence interval is a range of values that is likely to contain the population parameter of interest with a certain level of confidence.

It provides a range of plausible values for the parameter, rather than a single point estimate.

The confidence level (often denoted as 1 - α) represents the probability that the interval contains the true population parameter. Common confidence levels include 90%, 95%, and 99%.

For example, a 95% confidence interval for the mean blood pressure might be [120, 130], indicating that we are 95% confident that the true population mean blood pressure falls within this range.

2. Significance Level (α):

The significance level, denoted by α (alpha), is the probability of rejecting the null hypothesis when it is actually true.

It represents the threshold for making a decision about the null hypothesis in hypothesis testing.

Commonly used significance levels include 0.05 (5%) and 0.01 (1%). These values indicate the probability of making a Type-I error (false positive).

A smaller significance level corresponds to a more conservative test, where stronger evidence is required to reject the null hypothesis.

3. P-Value:

The p-value is the probability of obtaining results as extreme as or more extreme than the observed results, assuming that the null hypothesis is true.

It provides a measure of the strength of evidence against the null hypothesis.

If the p-value is less than or equal to the significance level (α), the null hypothesis is rejected in favor of the alternative hypothesis.

A smaller p-value indicates stronger evidence against the null hypothesis.

The p-value allows researchers to make probabilistic statements about the data and the null hypothesis.

For example, if the p-value is 0.03 and the significance level is 0.05, we would reject the null hypothesis at the 0.05 significance level, indicating that the observed results are statistically significant.

In summary:

Confidence interval provides a range of plausible values for the population parameter.

Significance level determines the threshold for making a decision about the null hypothesis in hypothesis testing.

P-value quantifies the strength of evidence against the null hypothesis based on the observed data.

6. List the differences between Parametric and Non Parametric Tests

Parametric and nonparametric tests are two broad categories of statistical tests used in hypothesis testing and data analysis. Here are the key differences between them:

1. Assumptions:

    1. Parametric tests assume that the data follow a specific probability distribution, usually a normal distribution, and/or have specific parameters (e.g., mean, variance).

    2. Nonparametric tests make fewer assumptions about the underlying distribution of the data. They are distribution-free or have fewer distributional assumptions compared to parametric tests.

2. Data Type

    Parametric tests are typically used for interval or ratio data, which are continuous and have a meaningful zero point (e.g., height, weight).

Nonparametric tests are more flexible and can be used for ordinal, nominal, interval, or ratio data. They are suitable for both continuous and categorical data.

3. Test Statistics::

Parametric tests often involve calculating test statistics based on the parameters of the assumed distribution (e.g., t-test, ANOVA, Pearson correlation).

Nonparametric tests use rank-based statistics or other distribution-free methods that do not rely on specific distributional assumptions (e.g., Wilcoxon rank-sum test, Kruskal-Wallis test, Spearman correlation).

4. Power

Parametric tests tend to have higher statistical power (i.e., ability to detect true effects) when the underlying assumptions are met.

Nonparametric tests may have lower power compared to parametric tests, especially when the sample size is small or when the assumptions of parametric tests are satisfied.

5. Robustness

Parametric tests are sensitive to violations of their underlying assumptions. If the data do not meet the assumptions (e.g., normality, homogeneity of variance), the results may be unreliable.

Nonparametric tests are more robust to violations of assumptions and can be used when the data do not meet the assumptions of parametric tests.

6. Sample Size

Parametric tests are generally more efficient (i.e., require smaller sample sizes to achieve the same power) than nonparametric tests when the assumptions are met.

Nonparametric tests may require larger sample sizes to achieve comparable power, especially for detecting small effects.

7. Ease of Interpretation

Parametric tests often provide more straightforward interpretations of results, as they are based on familiar statistical parameters (e.g., means, standard deviations).

Nonparametric tests may have less intuitive interpretations, especially for rank-based statistics or tests involving complex data transformations.

Parametric Test using scipy:

```
from scipy.stats import ttest_ind, f_oneway, pearsonr

# Example of independent t-test
group1 = [23, 25, 28, 32, 35]
group2 = [18, 20, 24, 28, 30]
t_statistic, p_value = ttest_ind(group1, group2)
print("Independent t-test: t-statistic =", t_statistic, ", p-value =", p_value)

# Example of one-way ANOVA
group1 = [23, 25, 28, 32, 35]
group2 = [18, 20, 24, 28, 30]
group3 = [15, 17, 20, 22, 25]
f_statistic, p_value = f_oneway(group1, group2, group3)
print("One-way ANOVA: F-statistic =", f_statistic, ", p-value =", p_value)

# Example of Pearson correlation
x = [1, 2, 3, 4, 5]
y = [2, 4, 6, 8, 10]
corr_coef, p_value = pearsonr(x, y)
print("Pearson correlation: correlation coefficient =", corr_coef, ", p-value =", p_value)
```

Non parametric testing using scipy:

```
from scipy.stats import mannwhitneyu, kruskal, spearmanr

# Example of Mann-Whitney U test
group1 = [23, 25, 28, 32, 35]
group2 = [18, 20, 24, 28, 30]
u_statistic, p_value = mannwhitneyu(group1, group2)
print("Mann-Whitney U test: U-statistic =", u_statistic, ", p-value =", p_value)

# Example of Kruskal-Wallis test
group1 = [23, 25, 28, 32, 35]
group2 = [18, 20, 24, 28, 30]
group3 = [15, 17, 20, 22, 25]
h_statistic, p_value = kruskal(group1, group2, group3)
print("Kruskal-Wallis test: H-statistic =", h_statistic, ", p-value =", p_value)

# Example of Spearman correlation
x = [1, 2, 3, 4, 5]
y = [2, 4, 6, 8, 10]
corr_coef, p_value = spearmanr(x, y)
```

```
print("Spearman correlation: correlation coefficient =", corr_coef, ", p-value =", p_value)
```

In summary, parametric tests are more powerful and efficient when the underlying assumptions are met, while nonparametric tests provide more robustness and flexibility, making them suitable for a wider range of data types and situations where parametric assumptions are violated. The choice between parametric and nonparametric tests depends on the nature of the data, the assumptions being made, and the research question being addressed.

8. What is Central Limit Theorem ?

The central limit theorem states that if you have a population with mean μ and standard deviation σ and take sufficiently large random samples from the population with replacement, then the distribution of the sample means will be approximately normally distributed. This will hold true regardless of whether the source population is normal or skewed, provided the sample size is sufficiently large (usually $n \geq 30$). If the population is normal, then the theorem holds true even for samples smaller than 30. In fact, this also holds true even if the population is binomial, provided that $\min(np, n(1-p)) \geq 5$, where n is the sample size

and p is the probability of success in the population. This means that we can use the normal probability model to quantify uncertainty when making inferences about a population mean based on the sample mean.

For the random samples we take from the population, we can compute the mean of the sample means:

$$\mu_{\bar{X}} = \mu$$

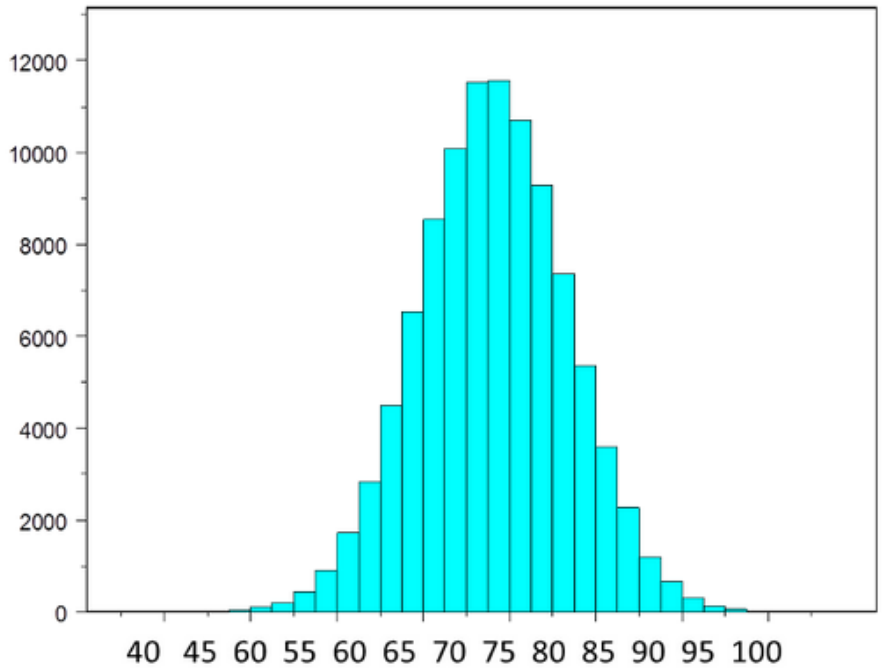
and the standard deviation of the sample means:

$$\sigma_{\bar{X}} = \frac{\sigma}{\sqrt{n}}$$

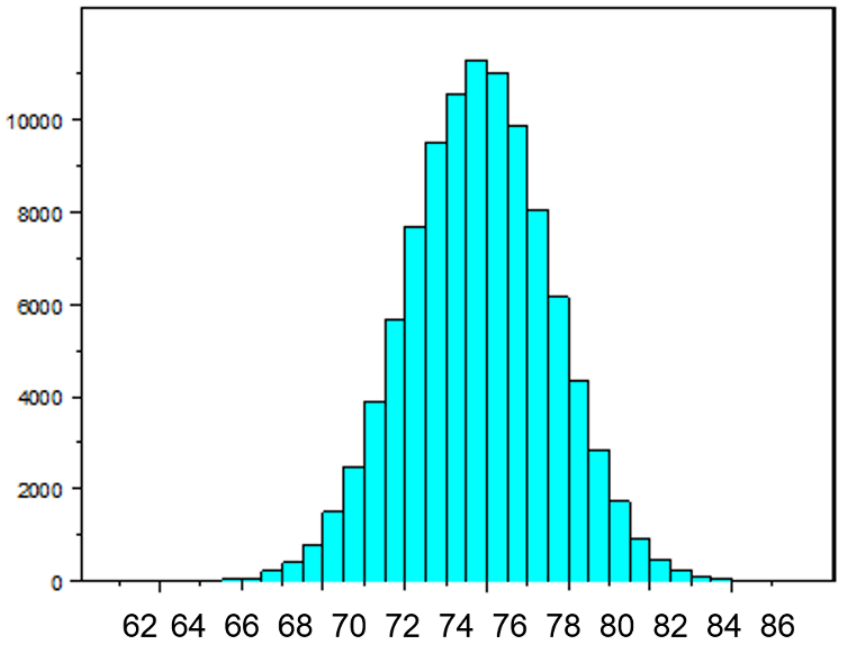
Before illustrating the use of the Central Limit Theorem (CLT) we will first illustrate the result. In order for the result of the CLT to hold, the sample must be sufficiently large ($n \geq 30$). Again, there are two exceptions to this. If the population is normal, then the result holds for samples of any size (i..e, the sampling distribution of the sample means will be approximately normal even for samples of size less than 30).

Central Limit Theorem with a Normal Population

The figure below illustrates a normally distributed characteristic, X, in a population in which the population mean is 75 with a standard deviation of 8.



If we take simple random samples (**with replacement)** of size n=10 from the population and compute the mean for each of the samples, the distribution of sample means should be approximately normal according to the Central Limit Theorem. Note that the sample size (n=10) is less than 30, but the source population is normally distributed, so this is not a problem. The distribution of the sample means is illustrated below. Note that the horizontal axis is different from the previous illustration, and that the range is narrower.



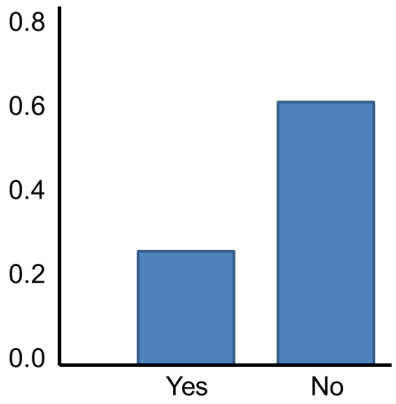
The mean of the sample means is 75 and the standard deviation of the sample means is 2.5, with the standard deviation of the sample means computed as follows:

$$\sigma_{\bar{X}} = \frac{\sigma}{\sqrt{n}} = \frac{8}{\sqrt{10}} = 2.5$$

If we were to take samples of n=5 instead of n=10, we would get a similar distribution, but the variation among the sample means would be larger. In fact, when we did this we got a sample mean = 75 and a sample standard deviation = 3.6.

Central Limit Theorem with a Dichotomous Outcome

Now suppose we measure a characteristic, X, in a population and that this characteristic is dichotomous (e.g., success of a medical procedure: yes or no) with 30% of the population classified as a success (i.e., p=0.30) as shown below.



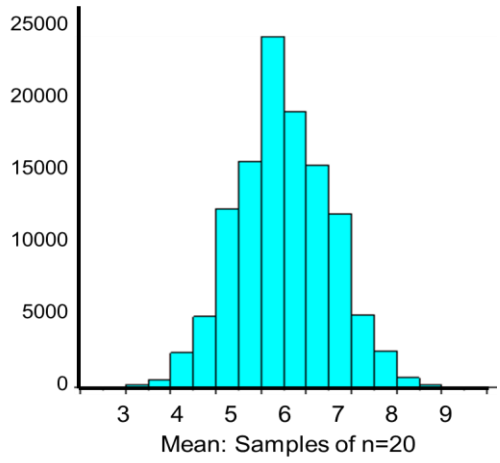
The Central Limit Theorem applies even to binomial populations like this provided that the minimum of np and n(1-p) is at least 5, where "n" refers to the sample size, and "p" is the probability of "success" on any given trial. In this case, we will take samples of n=20 with replacement, so min(np, n(1-p)) = min(20(0.3), 20(0.7)) = min(6, 14) = 6. Therefore, the criterion is met.

We saw previously that the ***population*** mean and standard deviation for a binomial distribution are:

Mean binomial probability: $\mu = np$

Standard deviation: $\sigma = \sqrt{n(p)(1-p)}$

The distribution of sample means based on samples of size n=20 is shown below.



The mean of the ***sample means*** is

$$\bar{X} = np = 20\left(0.3\right) = 6$$

and the standard deviation of the sample means is:

$$\sigma_{\bar{X}} = \frac{\sigma}{\sqrt{n}} = \frac{\sqrt{n(p)(1-p)}}{\sqrt{n}}$$

$$\sigma_{\bar{X}} = \frac{\sqrt{20(0.3)(0.7)}}{\sqrt{20}} = 0.46$$

Now, instead of taking samples of n=20, suppose we take simple random samples (with replacement) of size n=10. Note that in this scenario we do not meet the sample size requirement for the Central

Limit Theorem (i.e., min(np, n(1-p)) = min(10(0.3), 10(0.7)) = min(3, 7) = 3).The distribution of sample means based on samples of size n=10 is shown on the right, and you can see that it is not quite normally distributed. The sample size must be larger in order for the distribution to approach normality.

Central Limit Theorem with a Skewed Distribution

The Poisson distribution is another probability model that is useful for modeling discrete variables such as the number of events occurring during a given time interval. For example, suppose you typically receive about 4 spam emails per day, but the number varies from day to day. Today you happened to receive 5 spam emails. What is the probability of that happening, given that the typical rate is 4 per day? The Poisson probability is:

$$P(x;\mu) = \frac{(e^{-\mu})(\mu^{x})}{x!}$$

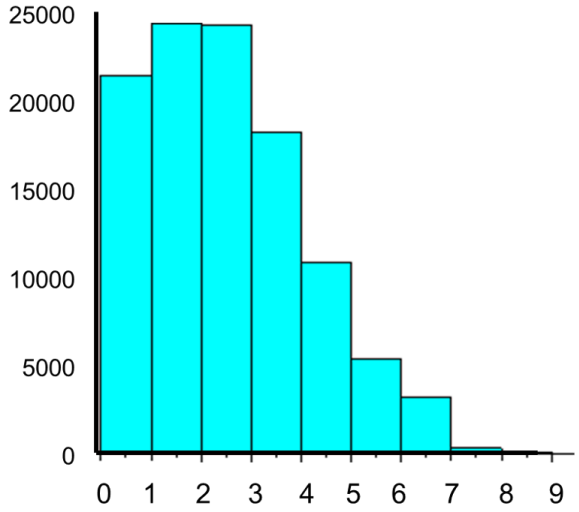
Mean = μ

Standard deviation = $\sigma = \sqrt{\mu}$

The mean for the distribution is μ (the average or typical rate), "X" is the actual number of events that occur ("successes"), and "e" is the constant approximately equal to 2.71828. So, in the example above

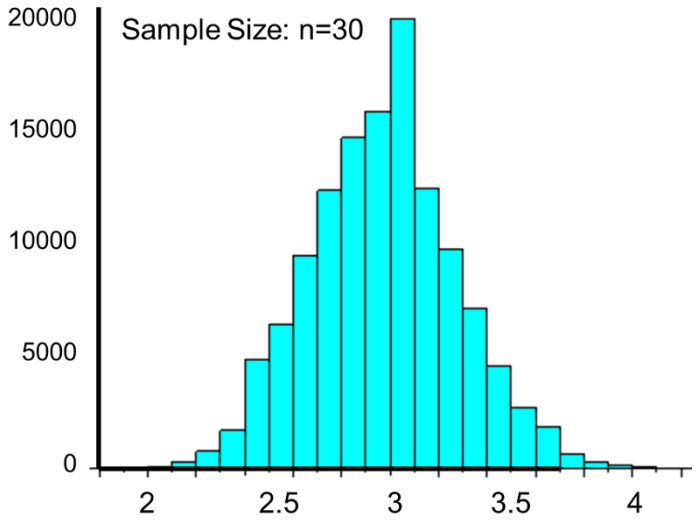
$$P(5;4) = \frac{(2.71828^{-\mu})(\mu^{x}))}{X!} = 0.15829 = 15.8\%$$

Now let's consider another Poisson distribution. with μ=3 and σ=1.73. The distribution is shown in the figure below.

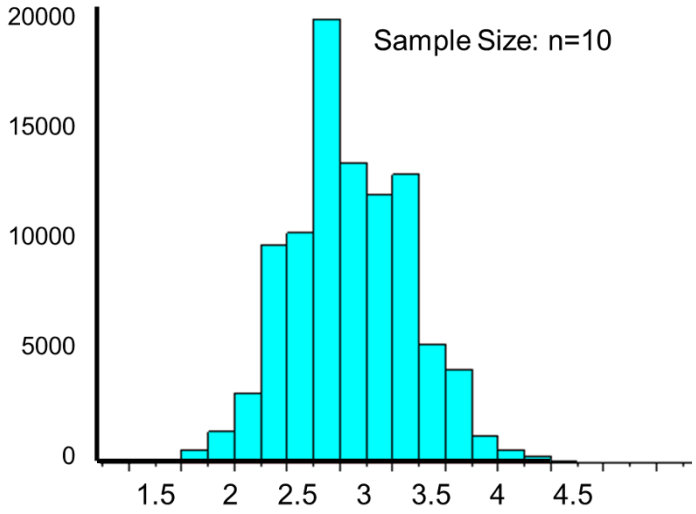


This population is not normally distributed, but the Central Limit Theorem will apply if n ≥ 30. In fact, if we take samples of size n=30, we obtain samples distributed as shown in the first graph below with a mean of 3 and standard deviation = 0.32. In contrast, with small samples of n=10, we obtain samples distributed as shown in the lower graph. Note that n=10 does not meet the criterion for the Central Limit Theorem, and the small samples on the right give a distribution that is not quite normal. Also note that the sample standard deviation (also called the "**standard error**") is larger with smaller samples, because it is obtained by dividing the population standard deviation by the square root of the sample size. Another way of thinking about this is that

extreme values will have less impact on the sample mean when the sample size is large.



$$\sigma_{\bar{X}} = \frac{\sigma}{\sqrt{n}} = \frac{1.73}{\sqrt{30}} = \mathbf{0.32}$$



$$\sigma_{\bar{X}} = \frac{\sigma}{\sqrt{n}} = \frac{1.73}{\sqrt{10}} = \mathbf{0.55}$$

```
import numpy as np

import matplotlib.pyplot as plt

from scipy.stats import norm


# Parameters

population_mean = 50

population_std = 10

sample_size = 1000

num_samples = 1000


# Generate samples from a uniform distribution

samples = np.random.uniform(low=0, high=100, size=(num_samples, sample_size))


# Calculate sample means

sample_means = np.mean(samples, axis=1)
```

```
# Plot the histogram of sample means
plt.figure(figsize=(8, 6))
plt.hist(sample_means, bins=30, density=True, alpha=0.6, color='blue')

# Plot the theoretical normal distribution
x = np.linspace(0, 100, 1000)
pdf = norm.pdf(x, loc=population_mean, scale=population_std/np.sqrt(sample_size))
plt.plot(x, pdf, color='red', linestyle='--', linewidth=2)
plt.title('Central Limit Theorem')
plt.xlabel('Sample Mean')
plt.ylabel('Probability Density')
plt.legend(['Theoretical Normal Distribution', 'Sample Means'])
plt.grid(True)
plt.show()
```

8. Find the probability of P(x<400) given that mean is = 1000 variance is =100

To find the probability (P(x < 400) \) given that the mean (μ) is 1000 and the variance (σ^2) is 100, we need to first standardize the variable x using the z-score formula:

$$z = \{x - \mu\} / \{\sigma\}$$

Then, we find the corresponding z-value for x = 400, and calculate the probability using the standard normal distribution (Z-distribution) table or a statistical software.

Given:
- Mean (μ) = 1000
- Variance (σ^2) = 100
- Standard deviation (σ) = √(100) = 10

We can calculate the z-score for x = 400:

z = (400 – 1000) / (10) = -60

Then, we find the probability P(z < -60)

However, it's important to note that a z-score of -60 is extremely rare and practically impossible in most contexts. Such an extreme value suggests that there may be an error in the calculations or an incorrect understanding of the problem. Please double-check the values and the problem statement to ensure accuracy. If the variance is indeed 100, then it's highly unlikely to observe a value of 400 within a distribution with mean 1000 and variance 100.

```
from scipy.stats import norm

# Mean and variance
mean = 1000
variance = 100

# Standard deviation
std_dev = variance ** 0.5

# Value of x
x = 400
```

```
# Calculate the z-score
z_score = (x - mean) / std_dev

# Calculate the cumulative probability
probability = norm.cdf(z_score)

print("Probability of P(x < 400):", probability)
```

9. Provide a proper description as to when to apply z-test, t-test, Chi-Square and Annona, with examples.

POPULATION VS. SAMPLE :

In statistics, population refers to the total set of observations we can make. For example, if we want to calculate the average human height, the population will be the total number of people actually present on Earth.

A sample, on the other hand, is a set of data collected or selected from a predefined procedure. For our example above, a sample is a small group of people selected randomly from different regions of the globe.

To draw inferences from a sample and validate a hypothesis, the sample must be random.

For instance, if we select people randomly from all regions on Earth, we can assume our sample mean is close to the population mean, whereas if we make a selection just from the United States, then our average height estimate/sample mean cannot be considered close to the population mean. Instead, it will only represent the data of a particular region (the United States). That means our sample is biased and is not representative of the population.

DISTRIBUTION

Another important statistical concept to understand is distribution. When the population is infinitely large, it's not feasible to validate any hypothesis by calculating the mean value or test parameters on the entire population. In such cases, we assume a population is some type of a distribution.

While there are many forms of distribution, the most common are binomial, Poisson and discrete.

You must determine the distribution type to calculate the critical value and decide on the best test to validate any hypothesis.

Now that we're clear on population, sample and distribution, let's learn about different kinds of tests and the distribution types for which they are used.

Types of Statistical Tests:

1. T-TEST

We use a t-test to compare the mean of two given samples. Like a z-test, a t-test also assumes a normal distribution of the sample. When we don't know the population parameters (mean and standard deviation), we use t-test.

There are multiple variations of the t-test.

THE THREE VERSIONS OF A T-TEST

1. **Independent sample t-test**: compares mean for two groups

2. **Paired sample t-test**: compares means from the same group at different times
3. **One sample t-test**: tests the mean of a single group against a known mean

The statistic for this hypothesis testing is called t-statistic, the score for which we calculate as:

$t=(x1—x2) / (\sigma / \sqrt{n1} + \sigma / \sqrt{n2})$, where

$x1$ = mean of sample 1

$x2$ = mean of sample 2

$n1$ = sample size 1

$n2$ = sample size 2

There are multiple variations of the t-test.

Note: This article focuses on normally distributed data. You can use z-tests and t-tests for data which is non-normally distributed as well if the sample size is greater than 20, however there are other preferable methods to use in such a situation.

```
from scipy.stats import ttest_ind


# Sample data

group1 = [23, 25, 28, 32, 35]

group2 = [18, 20, 24, 28, 30]


# Perform independent t-test

t_statistic, p_value = ttest_ind(group1, group2)


print("T-statistic:", t_statistic)

print("p-value:", p_value)
```

2. CHI-SQUARE TEST

We use the chi-square test to compare categorical variables.

THE TWO TYPES OF CHI-SQUARE TEST

1. **Goodness of fit test**: determines if a sample matches the population
2. **A chi-square fit test for two independent variables**: used to compare two variables in a contingency table to check if the data fits

A small chi-square value means that data fits.

A large chi-square value means that data doesn't fit.

The hypothesis we're testing is:

- **Null**: Variable A and Variable B are independent.
- **Alternate**: Variable A and Variable B are not independent.

The statistic used to measure significance, in this case, is called chi-square statistic. The formula we use to calculate the statistic is:

$X2 = \Sigma [(Or,c — Er,c)2 / Er,c]$ where

Or,c = observed frequency count at level r of Variable A and level c of Variable B

Er,c = expected frequency count at level r of Variable A and level c of Variable B

T-TEST VS. CHI-SQUARE

We use a t-test to compare the mean of two given samples but we use the chi-square test to compare categorical variables.

```
from scipy.stats import chi2_contingency


# Contingency table (observed frequencies)
observed = [[15, 20, 25],
            [10, 15, 20]]


# Perform Chi-Square test
chi2_stat, p_value, dof, expected = chi2_contingency(observed)


print("Chi-Square Statistic:", chi2_stat)
print("p-value:", p_value)
```

3. Z-TEST

In a z-test, we assume the sample is normally distributed. A z-score is calculated with population parameters such as population mean and population standard deviation. We use this test to validate a hypothesis that states the sample belongs to the same population.

- **Null**: Sample mean is same as the population mean.
- **Alternate**: Sample mean is not same as the population mean.

The statistic used for this hypothesis testing is called z-statistic, the score for which we calculate as:

$z = (x — \mu) / (\sigma / \sqrt{n})$, where

x = sample mean

μ = population mean

$\sigma / \sqrt{n}$ = population standard deviation

If the test statistic is lower than the critical value, accept the hypothesis.

```
from scipy.stats import norm



# Sample mean, population mean, and population standard deviation

sample_mean = 30

pop_mean = 25

pop_std_dev = 5

sample_size = 100
```

```
# Calculate z-score

z_score = (sample_mean - pop_mean) / (pop_std_dev / (sample_size ** 0.5))



# Calculate p-value

p_value = norm.cdf(z_score)  # One-tailed test



print("Z-score:", z_score)

print("p-value:", p_value)
```

4. ANOVA

We use analysis of variance (ANOVA) to compare three or more samples with a single test.

THE TWO MAJOR TYPES OF ANOVA

1. **One-way ANOVA**: Used to compare the difference between three or more samples/groups of a single independent variable.
2. **MANOVA**: Allows us to test the effect of one or more independent variables on two or more dependent variables. In addition, MANOVA can also detect the difference in correlation between dependent variables given the groups of independent variables.

The hypothesis we're testing with ANOVA is:

- **Null**: All pairs of samples are the same (i.e. all sample means are equal).
- **Alternate**: At least one pair of samples is significantly different.

The statistics used to measure the significance in this case are F-statistics. We calculate the F-value using the formula:

F= ((SSE1—SSE2)/m)/ SSE2/n-k, where

SSE = residual sum of squares

m = number of restrictions

k = number of independent variables

There are multiple tools available such as SPSS, R packages, Excel etc. to carry out ANOVA on a given sample.

```
from scipy.stats import f_oneway


# Sample data for each group

group1 = [80, 85, 90, 95, 100]

group2 = [75, 80, 85, 90, 95]

group3 = [70, 75, 80, 85, 90]


# Perform one-way ANOVA

f_statistic, p_value = f_oneway(group1, group2, group3)


print("F-statistic:", f_statistic)

print("p-value:", p_value)
```

10. Explain the Bayes Theorem using an example.

Bayes Theorem :

Bayes' Theorem is a fundamental principle in probability theory .

It describes the probability of an event based on prior knowledge of conditions that might be related to the event.

The formula for Bayes' Theorem is as follows:

$P(A|B) = P(B|A) * P(A) / P(B)$

P(A|B) : The probability of event A occurring given that event B has occurred.

(P(B|A) : The probability of event B occurring given that event A has occurred.

P(A) : The prior probability of event A.

P(B) : The prior probability of event B.

Bayes Theorem Example:

If Aircraft Black box manufactured by company A,B,C

A manufactures 75 % of Black box

B manufactures 15 % of Black box

C manufacture's 10 % of Black box

The defective rate of black boxes manufactured by A is 4%, B is 6%,C is 8%

If a Black box tested randomly what is the probability is found to be defective.

What is the probability that it was manufactured by company A

P(D) be the probability of defective black box

If the black box found defective what is the probability of defective manufactured by company A.

The defect rates of A,B,C

P(D|A) = 0.04 ,P(D|B) = 0.06, P(D|C) =0.08

Let P(A) , P(B), P(C) be events corresponding to the black box being manufactured by companies A,B, and C

P(A) = 0.75 ,P(B) = 0.15, P(C)= 0.10

P(D) = 0.75 * 0.04 + 0.15 * 0.06 +0.10 * 0.08

P(D) = 0.047

P(A|D) = P(D|A) * P(A) / P(D)

P(A|D) = 0.04 * 0.75 / 0.047 = 0.6382

P(B|D) = 0.06 * 0.15 / 0.047 = 0.1914

P(C|D) = 0.08 * 0.10 / 0.047 = 0.1702

Bayes theorem using scipy:

```
# Prior probability: P(A), probability of the event occurring without considering any evidence

prior_probability = 0.01  # Prior belief about the event's probability
```

```
# Likelihood: P(B|A), probability of observing the evidence given that the event has occurred

likelihood = 0.9  # Probability of observing the evidence if the event has occurred


# Evidence: P(B), probability of observing the evidence (including both cases where the event occurs and where it doesn't)

evidence = (likelihood * prior_probability) + ((1 - likelihood) * (1 - prior_probability))


# Posterior probability: P(A|B), updated belief about the probability of the event given the evidence

posterior_probability = (likelihood * prior_probability) /     evidence

print("Posterior Probability (P(A|B)):", posterior_probability)
```