# **Lab 6: Data Distribution and Normality Testing**

**Prelab Questions**

1. What is a data distribution, and why is it important in data analysis?

A **data distribution** refers to how the values of a dataset are spread or arranged. It shows the frequency of different values in the dataset, indicating patterns and trends. Understanding the distribution of data is essential because it helps analysts choose the correct statistical methods, identify potential outliers, and make predictions.

In data analysis, the distribution of the data can impact the results of any analyses or models applied. For example, certain statistical tests assume a particular distribution (e.g., normal distribution). Knowing the distribution can also guide decisions on data transformation or the type of model to use.

1. Define normal distribution and list its key properties.

A **normal distribution** (also known as a Gaussian distribution) is a continuous probability distribution that is symmetric around its mean. It is often called a "bell curve" because of its shape. In a normal distribution:

* The data is symmetrically distributed, with most values clustering around the mean.
* It is described by two parameters: **mean (μ)** and **standard deviation (σ)**.
* The curve has a single peak at the mean, and it decreases as you move away from the mean in either direction.

**Key properties of a normal distribution:**

1. **Symmetry**: The distribution is perfectly symmetrical around the mean.
2. **Mean, Median, Mode are equal**: In a perfectly normal distribution, these three measures of central tendency are the same.
3. **68-95-99.7 Rule**:
   1. Approximately 68% of the data lies within one standard deviation of the mean.
   2. 95% of the data lies within two standard deviations.
   3. 99.7% of the data lies within three standard deviations.
4. **Bell-shaped curve**: The graph of a normal distribution forms a bell-shaped curve, with the highest point at the mean.
5. **Asymptotic tails**: The tails of the normal distribution approach, but never actually touch, the horizontal axis.
6. What is the purpose of normality testing in statistical analysis?

Normality testing helps to determine whether a dataset follows a normal distribution. This is crucial because many statistical methods, including t-tests, ANOVA, and regression, assume that the data are normally distributed. If the data deviate significantly from normality, it may lead to inaccurate conclusions or the need to use different statistical techniques that do not rely on the normality assumption.

1. Name two statistical tests used to check for normality.

* **Shapiro-Wilk Test**: A widely used test for normality, which tests the null hypothesis that the data comes from a normal distribution.
* **Kolmogorov-Smirnov Test**: Compares the sample distribution with a normal distribution to assess if the data fit the normal distribution.

1. Why is normality important in hypothesis testing and machine learning models?

* **Hypothesis Testing**: Many statistical tests, such as t-tests and ANOVA, assume normality. If the data are not normal, these tests may produce incorrect results, leading to false positives or false negatives. Therefore, checking normality helps ensure that the assumptions of the test are met.
* **Machine Learning Models**: Some machine learning algorithms (like linear regression and logistic regression) assume that the data follows a normal distribution for the residuals or errors. If the residuals are not normally distributed, it can affect the model's performance, predictions, and interpretability. Understanding normality helps in choosing the right model and making appropriate adjustments to the data.

**In-Lab Details**

**Objective**:  
Analyze the distribution of a dataset and test for normality using visualization and statistical tests.

**Resources**:

* Python (Jupyter Notebook).
* Libraries: Pandas, Seaborn, Matplotlib, SciPy.
* Dataset: income\_data.csv with columns for age and annual income.

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

from scipy.stats import shapiro, normaltest, kstest, anderson

# Load dataset (Replace with actual file path if running locally)

file\_path = "/content/Salary\_Data.csv"

df = pd.read\_csv(file\_path)

# Rename 'Salary' to 'Annual Income' for consistency

df.rename(columns={'Salary': 'Annual\_Income'}, inplace=True)

# Plot histogram with KDE for annual income distribution

plt.figure(figsize=(10, 6))

sns.histplot(df['Annual\_Income'], kde=True, bins=30, color='skyblue')

plt.title("Distribution of Annual Income with KDE")

plt.xlabel("Annual Income")

plt.ylabel("Frequency")

plt.show()

# Perform normality tests

# 1. Shapiro-Wilk Test (on a subset of 5000 rows for accuracy)

shapiro\_test = shapiro(df['Annual\_Income'].sample(min(5000, len(df)), random\_state=42))

# 2. D’Agostino’s K² Test (on full dataset)

dagostino\_test = normaltest(df['Annual\_Income'])

# 3. Kolmogorov-Smirnov Test (against normal distribution)

ks\_test = kstest(df['Annual\_Income'], 'norm')

# 4. Anderson-Darling Test

anderson\_test = anderson(df['Annual\_Income'], dist='norm')

# Display results

print("\nShapiro-Wilk Test:")

print("Statistic =", shapiro\_test.statistic, ", p-value =", shapiro\_test.pvalue)

print("\nD’Agostino’s K² Test:")

print("Statistic =", dagostino\_test.statistic, ", p-value =", dagostino\_test.pvalue)

print("\nKolmogorov-Smirnov Test:")

print("Statistic =", ks\_test.statistic, ", p-value =", ks\_test.pvalue)

print("\nAnderson-Darling Test:")

print("Statistic =", anderson\_test.statistic)

print("Critical Values =", anderson\_test.critical\_values)

# Interpretation

alpha = 0.05

print("\n--- Interpretation ---")

if shapiro\_test.pvalue > alpha:

print("\nShapiro-Wilk: Data appears to be normally distributed.")

else:

print("\nShapiro-Wilk: Data does not appear to be normally distributed.")

if dagostino\_test.pvalue > alpha:

print("\nD’Agostino’s K²: Data appears to be normally distributed.")

else:

print("\nD’Agostino’s K²: Data does not appear to be normally distributed.")

if ks\_test.pvalue > alpha:

print("\nKolmogorov-Smirnov: Data appears to be normally distributed.")

else:

print("\nKolmogorov-Smirnov: Data does not appear to be normally distributed.")

# Anderson-Darling interpretation (comparison with critical values)

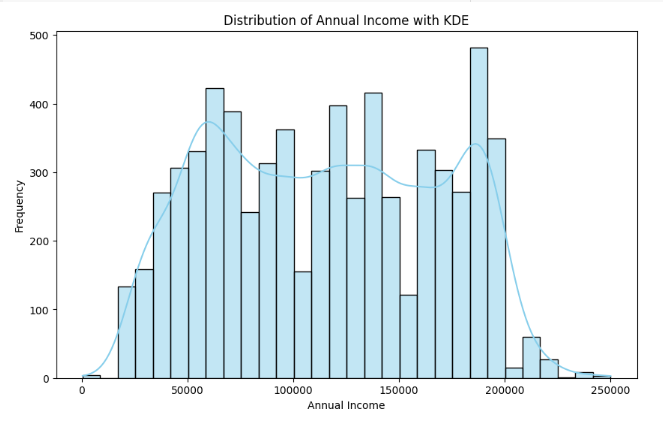
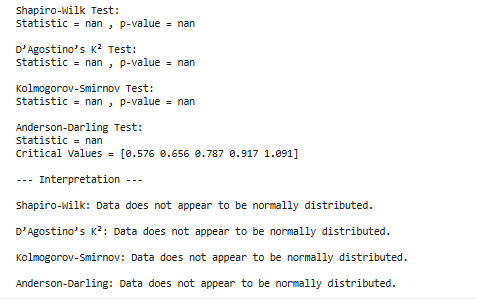
if anderson\_test.statistic < anderson\_test.critical\_values[2]: # 5% significance level

print("\nAnderson-Darling: Data appears to be normally distributed.")

else:

print("\nAnderson-Darling: Data does not appear to be normally distributed.")

**OUTPUT:**

**Expected Output**:

1. **Histogram with KDE**: A plot showing whether the income data follows a normal distribution.
2. **Shapiro-Wilk and D’Agostino’s K² Tests**:
   * **p-value > 0.05**: Data is normally distributed.
   * **p-value ≤ 0.05**: Data is not normally distributed.

**Postlab Questions**

1. How do you interpret the shape of a histogram to determine if data is normally distributed?

To determine if data is normally distributed based on a **histogram**, look for the following characteristics:

* **Symmetry**: The histogram should be roughly symmetrical. In a normal distribution, the left and right sides of the histogram should be mirror images of each other.
* **Bell-shaped curve**: The histogram should resemble a bell curve with the highest frequency of data points clustered around the center (mean), with the frequency of data points decreasing as you move further away from the mean.
* **Single peak (unimodal)**: A normal distribution has only one peak at the mean. If the histogram has multiple peaks, the data might follow a different distribution (e.g., bimodal).

In summary, a histogram that shows symmetry, a single peak at the center, and a bell-shaped curve is an indication that the data might be normally distributed.

1. What does a p-value > 0.05 in the Shapiro-Wilk test indicate?

The **Shapiro-Wilk test** is used to assess the normality of a dataset.

* If the **p-value is greater than 0.05**, it suggests that **there is no significant evidence to reject the null hypothesis**, meaning the data **does not significantly deviate from a normal distribution**.
* In other words, a p-value > 0.05 indicates that the data **may** be normally distributed (i.e., the data does not show strong evidence of non-normality).

On the other hand, a p-value **less than 0.05** would indicate that the data significantly deviates from normality, and thus, it is likely not normally distributed.

1. What are some real-world scenarios where normality testing is essential?

Normality testing is important in many fields, especially when using statistical methods that assume normality. Some real-world scenarios include:

* **Medical Research**: When testing the effectiveness of a drug, researchers often assume normality in the distribution of patient outcomes (e.g., blood pressure, cholesterol levels). Normality testing ensures the appropriate statistical tests (like t-tests) can be used.
* **Finance**: In risk management or portfolio optimization, financial returns are often assumed to be normally distributed. Normality testing ensures accurate risk assessments and prediction models.
* **Manufacturing Quality Control**: In process control, manufacturers may assume the distribution of product measurements (e.g., length, weight) follows a normal distribution. Normality testing helps confirm whether their assumptions are valid and if process adjustments are needed.
* **Psychology and Education**: Scores on psychological tests or student assessments may be assumed to follow a normal distribution. Normality testing helps confirm that statistical methods (e.g., ANOVA, regression) are appropriate for data analysis.

1. How can you transform a non-normal dataset to approximate a normal distribution?

There are several methods to transform a non-normal dataset to approximate a normal distribution:

1. **Log Transformation**: For right-skewed data, applying a log transformation can help normalize the distribution (e.g., log(x)).
2. **Square Root Transformation**: Often used for count data, taking the square root of each value can reduce skewness.
3. **Box-Cox Transformation**: A more general family of transformations that includes both log and square root transformations, used to make data more normally distributed.
4. **Inverse Transformation**: For data that is heavily skewed, taking the reciprocal (1/x) of the data can sometimes make it more normal.
5. **Z-score Transformation**: Standardizing the data by converting it into z-scores (subtracting the mean and dividing by the standard deviation) can sometimes improve normality, although it doesn't always fully normalize skewed data.
6. Why is a normal distribution preferred in many machine learning algorithms?

A normal distribution is preferred in many machine learning algorithms for the following reasons:

* **Mathematical Simplicity**: Many machine learning models (e.g., linear regression, logistic regression, and Gaussian Naive Bayes) are based on statistical assumptions that work well with normally distributed data. For example, in linear regression, the residuals are often assumed to be normally distributed to make valid inferences about model parameters.
* **Optimization**: Algorithms that use optimization techniques (e.g., gradient descent) perform better when the data is normally distributed, as normality leads to well-behaved cost functions that are easier to minimize.
* **Robustness**: Models that assume normality tend to have better performance and fewer issues when the data approximately follows a normal distribution. This allows models to make predictions with greater confidence and reduce the potential for overfitting or underfitting.
* **Central Limit Theorem**: For large datasets, the Central Limit Theorem suggests that the sampling distribution of the mean will approximate a normal distribution, even if the data itself isn't normal. This makes many machine learning algorithms more reliable when dealing with large amounts of data.
* **Assumptions of Common Algorithms**: Algorithms like **Gaussian Naive Bayes** assume that the data for each class follows a normal distribution. While these models can handle non-normal data, their performance can be optimized when data is close to normal.

DONE BY

Durga S – 22BCS026

Nandhini Priya K K -22BCS073

Nithiyaa T – 22BCS082

Priya Dharshini P – 22BCS094

Rithika S – 22BCS103

Sandhiya G – 22BCS109

Santhiya P – 22BCS111