Statistics for Data Science - Detailed Notes

1. What is Statistics?

Statistics is the science of collecting, organizing, analyzing, interpreting, and presenting data to support decision-making.
It helps convert raw data into meaningful insights.

In real-world scenarios, statistics is used in business forecasting, healthcare research, government planning, and machine learning model evaluation.

Statistics helps us understand data and make decisions from it.

Example (Real-life)

Suppose a company wants to know:

- How many products they sold
- Which product sells most
- Average customer spending

They collect and analyze data → This process is called Statistics.

2. Types of Statistics

There are two main types:

1. **Descriptive Statistics:**
It **summarizes and describes** the main features of a dataset using measures like mean, median, mode, variance, and graphs.

Example:

Marks of students:
60, 70, 80, 90

Descriptive statistics can tell:

- Average marks = 75
- Highest marks = 90
- Lowest marks = 60

It **describes** the data only.

2. **Inferential Statistics:**
It uses sample data to make **predictions or conclusions** about a population. Examples include hypothesis testing, regression analysis, and confidence intervals.

It answers questions like:

- What might happen?
- What is likely true?

Example:

Survey of **100 customers** shows:

- 70 like Product A

Inferential statistics helps predict:
"About 70% of ALL customers may like Product A."

Descriptive	**Inferential**
Describes data	Predicts using data
Works on given data	Uses sample to conclude
No guessing	Makes predictions
Example: Average marks	Example: Predict future sales

One-Line Summary

Descriptive statistics = What happened
Inferential statistics = What might happen

3. Central Tendency

Central tendency refers to the measure that identifies the center or typical value of a dataset.

- The **center value** of a dataset.
- The value that represents the **whole data.**

- Central tendency tells us **where most of the data is located**.
- It helps to summarize large data into **one single meaningful value**

Types:
Mean = Sum of values / Number of values
Median = Middle value after sorting
Mode = Most frequently occurring value

Python Example:

```
import numpy as np
data = [10,20,30,40,50,50]
print(np.mean(data))
print(np.median(data))

mode_value = stats.mode(data)
```

4. Measures of Dispersion

Dispersion describes how **spread out data** points are.

Example

Two classes have same average marks = 70

Class A: 68, 70, 72

Marks are very close → **Low dispersion**

Class B: 30, 70, 110

Marks are very spread → **High dispersion**

Types of Measures of Dispersion

There are **4 main types**:

1. **Range:** Difference between **highest and lowest** values.

Range = Maximum – Minimum

Code:

```
import numpy as np

data = np.array([10, 20, 30, 40, 50])
```

```
range_value = np.max(data) - np.min(data)

print("Range:", range_value)
```

2. **Variance:**

- How far each value is from the mean.
- How much data **varies.**

Variance = Average squared deviation from the mean

Formula: $\sigma^2 = \Sigma(X{-}\mu)^2$ / n

Code:

```
variance_value = np.var(data)

print("Variance:", variance_value)
```

3. **Standard Deviation:**

It is the square root of variance.

Shows how much data **normally deviates from mean.**

STD Value	**Meaning**
Small	Data close to mean
Large	Data widely spread

Standard Deviation = Square root of variance

The measuring unit of S.D. is same as the Sample values' unit. It indicates the average distance of data points from the mean and is widely used due to its intuitive interpretation.

$$Formula\colon \sigma = \sqrt{(}\sigma^2) = \sqrt{(}\frac{\Sigma(X-\mu)^2}{n}\Big)$$

Code:

```
std_value = np.std(data)

print("Standard Deviation:", std_value)
```

4. **Interquartile Range (IQR):**

IQR measures spread of **middle 50% of data**.

IQR = Q3 – Q1

Where:

- Q1 = 25th percentile
- Q3 = 75th percentile

Why Important?

- Used to detect **outliers.**
- Works well when data has extreme values.

Code:

```
q1 = np.percentile(data, 25)

q3 = np.percentile(data, 75)

iqr = q3 - q1

print("IQR:", iqr)
```

Python Example:

```
import numpy as np
data=[10,12,14,18,100]
print(np.var(data))
print(np.std(data))
```

4. Hypothesis Testing

Hypothesis testing is a statistical method used to make decisions using data.

Steps:

1. Define Null Hypothesis
2. Define Alternative Hypothesis
3. Choose significance level
4. Calculate test statistic
5. Interpret p-value

1. Null Hypothesis (H_0): There is no significant difference or effect.

2. Alternative Hypothesis (H_1): There is a significant effect i.e the given statement can be false.

3. Degrees of freedom: in statistics represent the number of values or quantities in the final calculation of a statistic that are free to vary. It is mainly defined as sample size-one (n-1).

4. Level of Significance(α): This is the threshold used to determine statistical significance. Common values are 0.05, 0.01, or 0.10.

5. p-value: probability of observing results if H_0 is true.

- If $p \leq \alpha$: reject H_0
- If $p > \alpha$: fail to reject H_0

6. Type I Error and Type II Error

- Type I Error that occurs when the null hypothesis is true, but the statistical test incorrectly rejects it. It is often referred to as a "false positive" or "alpha error."
- Type II Error that occurs when the null hypothesis is false, but the statistical test fails to reject it. It is often referred to as a "false negative."

7. Confidence Intervals: is a range of values that is used to estimate the true value of a population parameter with a certain level of confidence. It provides a measure of the uncertainty or margin of error associated with a sample statistic, such as the sample mean or proportion.

5. t-Test

A t-test compares means when sample size is small and population variance is unknown.

Types

- One sample t-test
- Independent t-test
- Paired t-test

Python Example

```
from scipy import stats

sample = [10, 12, 14, 16, 18]
stats.ttest_1samp(sample, 15)
```

6. Z-Test

Used when sample size is large and population variance is known.
Formula:
$Z = (X - \mu) / (\sigma / \sqrt{n})$

7. ANOVA

Analysis of Variance compares means of three or more groups.

Assumptions:
Normal distribution, equal variance, independent samples.

Python Example:

```
from scipy.stats import f_oneway

group1 = [10,12,14]

group2 = [20,22,24]

group3 = [30,32,34]

f_oneway(group1, group2, group3)
```

8. p-Value

p-value measures probability that results occurred by chance.

Interpretation: $p < 0.05$ → Significant

- $p > 0.05$ → Not significant

9. Z-score:

Shows how many standard deviations a value is from mean.

Formula

$$Z = \frac{X - Mean}{Std}$$

Code:

```
from scipy.stats import zscore

zscore(data)
```

10. Significance level:

The **significance level**, denoted by **α (alpha)**, is the threshold used in hypothesis testing to decide whether to reject the null hypothesis.

Common values:

- **0.05 (most used)** → 5% risk of wrong rejection
- **0.01** → very strict testing
- **0.10** → more flexible testing

Why Significance Level Is Important

1. Controls False Positives (Type I Error)

It sets how much risk you are willing to accept for making a wrong decision.

Example:
If $\alpha = 0.05$, it means you accept a **5% chance** of falsely concluding something is significant.

2. Provides a Decision Rule

It gives a clear rule:

- If **p-value $\le \alpha$** → Reject null hypothesis
- If **p-value $> \alpha$** → Fail to reject null hypothesis

Without α, we cannot decide whether results are statistically meaningful.

3. Ensures Scientific Reliability

A fixed significance level prevents random or biased decisions and ensures consistent statistical conclusions.

4. Balances Risk and Strictness

Choosing α affects how strict the test is:

Significance Level	**Meaning**
High α (0.10)	Easier to reject H_0, higher false risk
Low α (0.01)	Harder to reject H_0, more reliable results

5. Critical in Research and Machine Learning

In data science and ML, significance level helps in:

- Feature selection
- Model validation
- A/B testing
- Experiment analysis
-

Python Example:

```
from scipy import stats

sample = [45, 47, 49, 50, 46, 48, 52]

t_stat, p_value = stats.ttest_1samp(sample, 50)

alpha = 0.05

if p_value <= alpha:

    print("Reject Null Hypothesis")

else:

    print("Fail to Reject Null Hypothesis")
```

11. Chi-Square Test

Used for categorical data to check relationship.

Used to check whether observed data fits an expected distribution.

Example:
Checking whether a dice is fair.

Formula

$$\chi^2 = \sum \frac{(Observed - Expected)^2}{Expected}$$

Explanation of Formula

The formula measures how far observed values differ from expected values.

- If difference is small → Variables are independent
- If difference is large → Variables are related

Example (Conceptual)

Suppose we want to test whether **gender affects product choice**.

Observed data:

	Product A	Product B
Male	30	20
Female	20	30

Chi-Square test checks whether this difference happened by chance

Python Example:

```
from scipy.stats import chi2_contingency

table = [[10,20],[20,40]]

chi2_contingency(table)
```

12. Major Types of Data Distribution

1. Normal Distribution (Gaussian Distribution)

A **normal distribution** is a symmetric, bell-shaped distribution where most values cluster around the mean.

Characteristics

- Mean = Median = Mode
- Symmetrical shape
- Follows bell curve
- 68-95-99.7 rule

Real-World Examples

- Human height
- Exam scores
- IQ levels

- Measurement errors

Normal Distribution Graph (Python)

```
import numpy as np
import matplotlib.pyplot as plt
data = np.random.normal(loc=50, scale=10, size=1000)
plt.hist(data, bins=30)
plt.title("Normal Distribution")
plt.xlabel("Values")
plt.ylabel("Frequency")
plt.show()
```

2. Uniform Distribution

In uniform distribution, **all values have equal probability** of occurring.

Characteristics

- No peaks
- Equal frequency
- Flat shape

Real-World Examples

- Dice roll
- Lottery numbers
- Random number generator

Uniform Distribution Graph

```
data = np.random.uniform(0, 100, 1000)
plt.hist(data, bins=30)
plt.title("Uniform Distribution")
plt.show()
```

3. Skewed Distribution

A skewed distribution is **not symmetrical**. One tail is longer than the other.

Types of Skewness

(A) Positive Skew (Right Skew)

Characteristics

- Tail extends to right
- Mean > Median > Mode

Examples

- Income distribution
- House prices
- Waiting time

Graph Example

```
data = np.random.exponential(scale=2, size=1000)

plt.hist(data, bins=30)

plt.title("Positive Skewed Distribution")

plt.show()
```

(B) Negative Skew (Left Skew)

Characteristics

- Tail extends to left
- Mean < Median < Mode

Examples

- Easy exam scores
- Retirement age data

4. Binomial Distribution

Distribution of **two possible outcomes** (success/failure).

Characteristics

- Discrete values
- Fixed number of trials

Examples

- Tossing a coin
- Pass/fail results
- Defective vs non-defective products

Python Example

```
data = np.random.binomial(n=10, p=0.5, size=1000)

plt.hist(data, bins=20)

plt.title("Binomial Distribution")

plt.show()
```

5. Poisson Distribution

Shows number of events occurring in a fixed interval.

Examples

- Number of customer arrivals
- Calls per minute
- Website traffic

Python Example

```
data = np.random.poisson(lam=5, size=1000)

plt.hist(data, bins=20)

plt.title("Poisson Distribution")

plt.show()
```

6. Exponential Distribution

Shows time between events.

Examples

- Time between customer arrivals
- Failure time of machines
- Waiting time in queue

Python Example

```
data = np.random.exponential(scale=2, size=1000)

plt.hist(data, bins=30)

plt.title("Exponential Distribution")

plt.show()
```

7. Multimodal Distribution

Has more than one peak.

Examples

- Mixed population data
- Sales of different product categories

Python Example

```
data1 = np.random.normal(20, 5, 500)

data2 = np.random.normal(60, 5, 500)

data = np.concatenate([data1, data2])

plt.hist(data, bins=30)

plt.title("Multimodal Distribution")

plt.show()
```

13. Inferential Statistics:

Inferential statistics is the branch of statistics that uses sample data to make predictions, generalizations, or conclusions about a larger population.

Using a small sample to understand or predict the behavior of a large population.

Population = Entire group
Sample = Small part of the population

Inferential statistics helps to move from:

Sample → Population conclusion

14. Correlation

Correlation measures the strength and direction of relationship between variables.
Range: −1 to +1.

Correlation answers this question:

If one variable changes, does the other variable also change?

Examples

1. Study hours and exam marks → positive relationship
2. Price of product and demand → negative relationship
3. Shoe size and intelligence → no relationship

1. Positive Correlation

When both variables move in the **same direction**.

If one increases, the other also increases.

Examples

- Study hours vs marks
- Experience vs salary
- Height vs weight

2. Negative Correlation

When variables move in **opposite directions**.

If one increases, the other decreases.

Examples

- Price vs demand
- Speed vs travel time
- Stress vs productivity

3. Zero Correlation

Definition

No relationship between variables.

Examples

- Shoe size vs intelligence
- Hair color vs salary

```
import numpy as np

x = [10, 20, 30, 40, 50]

y = [15, 25, 35, 45, 55]

corr = np.corrcoef(x, y)[0,1]

print("Correlation:", corr)
```

15. Regression

Regression predicts dependent variable values based on independent variables.
Used in forecasting and machine learning models.

Regression analysis is a statistical technique used to **study the relationship between a dependent variable and one or more independent variables**.

Its main purpose is to **predict the value of an outcome variable** based on input variables.

Regression answers this question:

How does the output change when input variables change?

It is mainly used for **prediction and forecasting**.

Example to Understand

Suppose you want to predict **house price** based on:

- Size of house
- Location
- Number of rooms

Here:

- House price → Dependent variable
- Size, location, rooms → Independent variables

Regression finds the mathematical relationship between them.

The equation for regression: $y = \alpha + \beta x$

Where,

- y is the dependent variable,
- x is the independent variable
- α is the intercept
- β is the regression coefficient.

Regression coefficient is a measure of the strength and direction of the relationship between a predictor variable (independent variable) and the response variable (dependent variable) $\beta = \frac{\sum(X_i - \bar{X})(Y_i - \bar{Y})}{\sum(X_i - \bar{X})^2}$

```
from sklearn.linear_model import LinearRegression

X = [[1],[2],[3]]

y = [2,4,6]

model = LinearRegression()

model.fit(X,y)

model.predict([[5]])
```

16. Independent vs Dependent Variables

Type	Meaning
Independent	Input variable
Dependent	Output variable

Example:
Study hours → Exam score

17. Statistics impact on ML:

Statistics helps in:

- Feature selection
- Data cleaning
- Model evaluation
- Probability prediction
- Hypothesis testing

18. Data Attributes:

Characteristics of data.

Examples:

- Age
- Salary
- Gender

19. Qualitative vs Quantitative Data:

Qualitative data describes qualities or characteristics and cannot be measured numerically.

It is also called categorical data.

Features:

Non-numeric

Descriptive

Represents categories or labels

Cannot perform mathematical operations

Examples

- Gender (Male/Female)
- Color (Red, Blue, Green)
- Brand names
- Education level
- Customer feedback

Types of Qualitative Data

1. Nominal Data

No order between categories.

Examples:

- Blood group
- Gender
- Religion

2. Ordinal Data

Has a meaningful order.

Examples:

- Ranking (1st, 2nd, 3rd)
- Satisfaction level (Low, Medium, High)
- Grades (A, B, C)

Quantitative Data

Quantitative data represents **numerical values** that can be measured and analyzed mathematically.

Features

- Numeric
- Measurable
- Can perform calculations

Examples

- Age
- Height
- Salary
- Temperature
- Distance

Types of Quantitative Data

1. Discrete Data

Countable values.

Examples:

- Number of students
- Number of cars
- Number of customers

2. Continuous Data

Measured values within a range.

Examples:

- Height
- Weight
- Time

- Temperature

Feature	**Qualitative**	**Quantitative**
Nature	Descriptive	Numeric
Measurement	Cannot be measured	Can be measured
Example	Gender	Salary
Mathematical operations	Not possible	Possible
Data type	Categorical	Numerical

20. Difference Between Continuous and Categorical Data

Continuous Data: Data that can take **any value within a range**.

Characteristics:

- Infinite possibilities
- Measured using instruments

Examples:

- Height = 165.5 cm
- Temperature = 37.2°C
- Time = 2.75 hours

Categorical Data: Data divided into **groups or categories**.

Characteristics:

- Limited values
- Represents labels

Examples:

- Gender
- Color
- Product type

Feature	**Continuous Data**	**Categorical Data**
Nature	Numeric	Non-numeric
Values	Infinite	Limited
Measurement	Measured	Classified
Example	Weight	Gender

21. Data:

Data refers to raw facts, observations, measurements, or information collected for analysis, decision-making, or research.

It can be numbers, text, images, audio, or any form of recorded information.

Data is the **basic input** used to generate meaningful information.

Example:

- Student marks
- Customer names
- Temperature readings
- Sales records

Characteristics of Data

- It is unprocessed information
- It can be structured or unstructured
- It can be numeric or non-numeric
- It is used for analysis and prediction

Types of Data (Major Classification)

Data is broadly classified into:

1. Qualitative Data
2. Quantitative Data

22. Structured and Unstructured Data:

Structured data is data that is **organized in a fixed format**, typically in rows and columns, making it easy to store, search, and analyze.

It follows a predefined schema (structure).

Key Characteristics

- Organized in tabular form
- Stored in databases or spreadsheets
- Easy to query using SQL
- Highly structured and consistent

Examples

- Excel sheets
- SQL databases
- CSV files
- Employee records
- Sales tables

Example of Structured Data

ID	Name	Age	Salary
1	Ravi	25	30000
2	Anu	28	40000

Python Example:

```
import pandas as pd

data = {

   "Name": ["Ravi", "Anu"],

   "Age": [25, 28],

   "Salary": [30000, 40000]

}

df = pd.DataFrame(data)

print(df)
```

Unstructured data is data that **does not have a predefined format** or organization.

It cannot be easily stored in tables.

Key Characteristics

- No fixed structure
- Complex to analyze
- Requires special tools like AI or NLP
- Large in volume

Examples

- Images
- Videos
- Audio files
- Emails
- Social media posts
- Text documents

Example

A WhatsApp message or photo is unstructured because it cannot be stored directly in rows and columns.

Feature	Structured Data	Unstructured Data
Format	Fixed	No fixed format
Storage	Tables/Databases	Files/Cloud storage
Analysis	Easy	Complex
Example	Excel	Images, videos

23. Outliers

Outliers are extreme values far from other observations.

Outliers are data points that are **significantly different from other observations** in a dataset.

They are unusually high or low values.

Example

Dataset:
10, 12, 14, 15, 16, 200

Here, **200** is an outlier.

Why Outliers Are Important

Outliers matter because they can:

1. Affect Statistical Measures

They can distort:

- Mean
- Variance
- Standard deviation

2. Impact Machine Learning Models

Outliers can:

- Reduce model accuracy
- Cause wrong predictions
- Mislead algorithms

3. Indicate Real Events

Sometimes outliers represent:

- Fraud detection
- Equipment failure
- Rare but important events

4. Help in Data Cleaning

They help identify:

- Data entry errors
- Measurement mistakes

24.How to detect Outliers

Method 1: Using IQR (Most Common Method)

Steps

1. Find Q1 (25th percentile)
2. Find Q3 (75th percentile)
3. Compute IQR = Q3 − Q1
4. Define limits:

$$Lower = Q1 - 1.5 \times IQR$$
$$Upper = Q3 + 1.5 \times IQR$$

Values outside these limits are outliers.

Python Example (IQR Method):

```
import numpy as np

data = np.array([10,12,14,15,16,200])

q1 = np.percentile(data, 25)

q3 = np.percentile(data, 75)

iqr = q3 - q1

lower = q1 - 1.5 * iqr

upper = q3 + 1.5 * iqr

outliers = data[(data < lower) | (data > upper)]

print("Outliers:", outliers)
```

Method 2: Using Z-Score Method

Formula

$$Z = \frac{X - Mean}{StandardDeviation}$$

If |Z| > 3 → Outlier.

Python Example:

```
from scipy.stats import zscore

import numpy as np

data = np.array([10,12,14,15,16,200])

z = np.abs(zscore(data))

outliers = data[z > 3]

print("Outliers:", outliers)
```

Method 3: Using Boxplot Visualization

Boxplots visually show outliers.

Python Example:

```
import matplotlib.pyplot as plt

plt.boxplot(data)

plt.title("Outlier Detection using Boxplot")

plt.show()
```

Method 4: Using Interquartile Range in Pandas

```
import pandas as pd

df = pd.DataFrame({"values":[10,12,14,15,16,200]})


Q1 = df["values"].quantile(0.25)

Q3 = df["values"].quantile(0.75)

IQR = Q3 - Q1

outliers = df[(df["values"] < Q1 - 1.5*IQR) | (df["values"] > Q3 + 1.5*IQR)]

print(outliers)
```