IDS Lab Internal 1 & 2 Viva

Internal 1

1. Name some popular R packages for data visualization.

Some widely used data visualization packages in R are:

- **ggplot2** – Most famous, grammar of graphics.
- **lattice** – Trellis graphics for multivariate data.
- **plotly** – Interactive plots.
- **highcharter** – Attractive interactive charts.
- **dygraphs** – Time-series interactive graphs.
- **shiny + ggplot2** – For web-based interactive graphics.
- **corrplot** – Correlation matrix visualizations.

2. What are the basic data structures in R? Explain the differences.

R has **six basic data structures**:

1. Vector

- Contains elements of **same data type** (numeric, character, logical).
- Example: `c(1, 2, 3)` or `c("A", "B")`

- **1-dimensional**.

2. List

- Can store **different data types** in one object.
- Can include vectors, matrices, data frames, even other lists.
- Example: `list(1, "A", TRUE)`
- **1-dimensional but heterogeneous**.

3. Matrix

- A 2D structure (rows × columns).
- Stores **same data type**.
- Example: `matrix(1:6, nrow = 2)`
- **2-dimensional and homogeneous**.

4. Data Frame

- Table-like structure (like Excel).
- Columns can have **different data types**.
- Example: `data.frame(Name=c("A"), Age=c(20))`
- **2-dimensional and heterogeneous**.

5. Array

- Can have **2 or more dimensions**.
- All elements must be **same data type**.
- Example: `array(1:12, dim = c(2,3,2))`

6. Factor

- Used for **categorical data**.
- Stores values as **levels**.
- Example: `factor(c("Low","Medium","High"))`
- Internally stored as integers mapped to labels.

3. What is a data frame in R?

A **data frame** is a **table-like** data structure in R where:

- Each column can contain **different data types** (numeric, character, logical, factor).
- All columns must have **equal length**.
- It is the most commonly used structure for datasets (similar to Excel or SQL table).

Example:

```
df ← data.frame(
  Name = c("A", "B"),
  Age = c(20, 25),
  Passed = c(TRUE, FALSE)
)
```

4. What is the purpose of the apply() function in R?

The **apply() function** is used to apply a function over:

- **Rows** of a matrix/data frame (MARGIN = 1)
- **Columns** of a matrix/data frame (MARGIN = 2)

Basic Format

```
apply(X, MARGIN, FUN)
```

Example

```
apply(matrix(1:9, nrow=3), 1, sum)   # Sum of each row
```

5. How do you create a scatter plot in R?

You can create a scatter plot using the **plot()** function.

Example:

```
x ← c(1, 2, 3, 4, 5)
y ← c(2, 4, 3, 6, 5)

plot(x, y, main="Scatter Plot", xlab="X-axis", ylab="Y-axis")
```

6. How do you calculate the mean of a vector in R?

You can calculate the mean using the **mean()** function.

Example:

```
v ← c(10, 20, 30, 40)
mean(v)
```

7. How do you check if a value is NA in R?

Use:

```
is.na(x)
```

- Returns TRUE if value is NA, FALSE otherwise.

8. How do you calculate the p-value in R?

Use built-in statistical test functions.

Example:

```
t.test(x, y)$p.value
```

- Most test functions (t.test, chisq.test, cor.test, etc.) return a p-value.

9. How do you create a histogram in R?

Two ways:

```
hist(x)              # Base R
ggplot(df, aes(x)) + geom_histogram()    # ggplot2
```

10. What is the role of dplyr and ggplot2 in data science?

dplyr

- Used for **data manipulation**: filtering, selecting, grouping, summarizing.
- Makes data cleaning fast and easy.

ggplot2

- Used for **data visualization**.

- Creates high-quality plots based on the Grammar of Graphics.

Together, they help in:

- Cleaning data
- Transforming data
- Visualizing patterns

Essential tools in almost every data science workflow.

11. Role of dplyr and ggplot2 in Data Science

dplyr → Data manipulation

- Used for filtering, selecting, grouping, summarizing.
- Makes data cleaning simple and readable.

ggplot2 → Data visualization

- Creates high-quality plots.
- Follows Grammar of Graphics.

Together, they help in:

- Preparing data
- Analyzing patterns
- Building insights visually

12. Types of Graphs in R

Common graphs:

- Histogram
- Bar Plot
- Line Plot
- Scatter Plot
- Boxplot
- Pie Chart
- Density Plot
- Heatmap

- Violin Plot
- Faceted plots (ggplot2)

13. How to Access Data in a Data Frame?

Common ways:

```
df$column         # by column name
df[ , "column"]    # column access
df[3, ]           # row 3
df[3, 2]          # row 3, column 2
df[1:5, c("A","B")]# subset rows and columns
```

- `$` for columns
- `[]` for rows & columns
- `head(df)` to preview

Internal 2

1. SET–1

1.1 Differentiate between linear regression and logistic regression in terms of objective, assumptions, and output.

Aspect	Linear Regression	Logistic Regression
Objective	Predict a continuous numerical value.	Predict a probability and classify into categories (usually binary).
Assumptions	Linearity between predictors and target, homoscedasticity, normally distributed residuals, no multicollinearity.	No assumption of linearity between predictors and target; assumes linear relationship between predictors and log-odds; no normality assumption.
Output	Direct numeric value (for example, price, temperature).	Probability between 0 and 1 → converted to a class label.

1.2 How does LIME handle categorical vs numerical features when building explanations?

- Numerical features:

  LIME perturbs values by adding small noise and observes how predictions change.

  It uses distance measures like Euclidean distance to weigh samples.

- Categorical features:

  LIME randomly “switches” categories and checks impact on prediction.

  It encodes categories using one-hot or dummy encoding before building the local surrogate model.

1.3 Discuss the trade-off between support and confidence in generating strong association rules.

- High support → rule applies to many transactions but may miss rare but valuable patterns.
- High confidence → rule is reliable when the antecedent occurs, but might have low support.

Trade-off:

You must balance both—too high support removes interesting rare rules; too low confidence gives weak or unreliable rules.

1.4 Explain the use of the %>% (pipe) operator in dplyr. Why is it useful?

- %>% passes the output of one function directly as input to the next.
- It improves readability and removes nested function calls.
- Allows writing step-by-step data manipulation.

```
data %>% filter(age > 20) %>% select(name, salary)
```

1.5 How can TF-IDF improve sentiment analysis accuracy compared to raw word frequencies? Illustrate with an example.

- Raw frequency treats all words equally—even stopwords like the, is, very.
- TF-IDF reduces weight of common words and increases weight of meaningful, rare words.

Example:

Sentence: “The movie was absolutely fantastic.”

- Word “fantastic” appears rarely → high TF-IDF → strong positive signal.
- Word “the” appears in almost every document → low TF-IDF → not misleading.

---

2. SET–2

2.1 What does a lift value greater than 1 indicate?

- Lift > 1 → Antecedent and consequent occur together more frequently than expected, meaning the rule shows positive association.

Example:

Buying bread increases likelihood of buying butter.

2.2 How do you interpret the coefficients in a logistic regression model?

- Coefficients represent change in log-odds of the target per unit change in the predictor.
- Exponentiating them gives odds ratio.
  - Value > 1 → increases probability of event.
  - Value < 1 → decreases probability of event.

2.3 Why is LIME called “model-agnostic”?

Because it does not depend on internal details of the model.

It works with any model—SVM, neural network, random forest—by treating it as a black box and using only input–output behavior.

2.4 Why do we split data into training and testing sets before building a regression model?

- To evaluate how the model performs on unseen data.
- Prevents overfitting.

- Ensures the model generalizes well in real-world predictions.

2.5 What are some common visualization functions in R (for example, from ggplot2) that help infer patterns in data?

- ggplot()
- geom_point() – scatter plot
- geom_bar() – bar plot
- geom_histogram() – distribution
- geom_boxplot() – detect outliers
- geom_line() – trends over time

---

3. SET–3

3.1 Example of a business application where association rule mining is effective.

- Market basket analysis:

  Identifying that customers who buy laptops also buy laptop bags.

Other examples:

- Cross-selling in e-commerce
- Product placement in retail stores
- Recommendation systems

3.2 What does LIME stand for and what is its purpose?

- LIME = Local Interpretable Model-agnostic Explanations
- Purpose:

  To explain predictions of any black-box ML model by approximating it locally with a simple, interpretable model.

3.3 How does sentiment analysis determine whether text is positive, negative, or neutral?

Two ways:

1. Lexicon-based:

   Counts positive and negative words from a predefined dictionary.
2. Machine learning-based:

   Uses labeled data to train a classifier that predicts sentiment.

3.4 What does a lift value tell you about the strength of a rule?

- Lift > 1 → strong positive relationship
- Lift = 1 → no relationship
- Lift < 1 → negative relationship

Higher lift means stronger association.

3.5 What is the role of the bag of words model in text mining?

- Converts text into numerical vectors.
- Treats each document as a bag of words → ignores grammar and order.
- Basis for TF-IDF, Naïve Bayes, and many ML models.

---

4. SET–4

4.1 How do gather() and spread() differ in tidyr?

Function	Purpose
gather()	Converts wide → long format.
spread()	Converts long → wide format.

4.2 Difference between lexicon-based and ML-based sentiment analysis.

Lexicon-based	Machine learning-based
Uses predefined list of sentiment words	Learns patterns from labeled data
Easy, no training	Needs training data
Less accurate	More accurate
Doesn't handle context well	Handles context and complex patterns

---

4.3 Difference between dependent and independent variables in regression analysis.

- Dependent variable (Y):

  The output we want to predict.

- Independent variables (X):

  Features used to predict Y.

Example:

Predicting house price →

Price = dependent, size or rooms or location = independent.

4.4 What is meant by support and confidence in association rule mining?

- Support:

  Frequency of the itemset in all transactions.

- Confidence:

  Probability that Y occurs when X occurs (strength of the rule X → Y).

4.5 How does LIME explain the prediction of a single instance?

- LIME perturbs the instance to create many similar samples.
- Gets predictions from the black-box model.
- Fits a simple local model (like linear regression).
- Shows which features contributed most to that specific prediction.

---

5. SET–5

5.1 What output does logistic regression produce and when is it preferred over linear regression?

- Produces probability between 0 and 1.
- Preferred when the target is categorical, especially binary (yes or no, spam or not spam).

5.2 One real-life example where LIME helps explain model predictions.

- Explaining why a loan application is rejected by a bank's ML model.

  LIME shows features like low income, high debt, low credit score.

5.3 Advantage of using data.table over regular R data frame for large datasets.

- Extremely fast for filtering, grouping, merging.
- Uses less memory.
- Syntax is concise and optimized in C.

5.4 Differentiate between melt() and dcast() in reshape2.

Function	Converts	Purpose
melt()	wide → long	Makes data tidy by stacking columns
dcast()	long → wide	Spreads rows into multiple columns

5.5 How can you summarize data using group_by() and summarise() in dplyr?

```
data %>%
  group_by(gender) %>%
  summarise(avg_salary = mean(salary), count = n())
```

- group_by() → groups rows
- summarise() → calculates summary statistics per group