

SECTION A

UNIT-1 & 2

a) A Computer is a concept used to convey a machine (particular electronic) which can perform required tasks based on the instructions provided to it and then return the output in meaningful form. Examples: Laptop, Smart Phones, Calculator, Printer, etc.

d) Primary memory is the main memory of a computer system which handles the processing of complex calculations and temporary storage of data and programs.

g) The shortcut keys are:

- Bold: Ctrl+B
- Italic: Ctrl+I
- Underline: Ctrl+U
- Copy: Ctrl+C
- Paste: Ctrl+V
- Find & Replace: Ctrl+F

UNIT-5 & 6

b) The R software was primarily developed by Ross Ihaka and Robert Gentleman. They began the project in the early 1990s at the University of Auckland, New Zealand. R is now maintained by the R Development Core Team, which is an international group of developers.

c) R is a versatile language widely used in various application areas, including:

1. Statistical Analysis: R is extensively used for statistical modeling and analysis.
2. Data Visualization: With packages like ggplot2 and lattice, R excels at creating detailed and sophisticated graphs.
3. Bioinformatics: R is used for analyzing biological data and is particularly popular in genetics.
4. Machine Learning: R provides numerous tools and packages for machine learning algorithms.
5. Econometrics: Economists use R for analyzing economic data and modeling economic phenomena.
6. Social Sciences: Researchers use R for data analysis and graphical representation in sociology, psychology, and political science.

e) To calculate the correlation coefficient between two variables x and y in R, you use the `cor()` function. Here is the command syntax: `cor(x, y)`

This command computes the Pearson correlation coefficient by default. You can specify different methods such as "pearson", "kendall", or "spearman" using the `method` parameter if needed, like this:

```
cor(x, y, method = "kendall")
```

SECTION B

UNIT-5 & 6

b) In the R console:

- [1] indicates the beginning of the output line, showing the index of the first element in the resulting output vector.
- "> +" is a prompt indicating that the R console expects additional input to complete the current command. It usually appears when a command is not properly closed with a bracket or a quotation mark.

c) The round() function in R is used to round numbers to a specified number of decimal places. The syntax is round(x, digits), where x is the number to be rounded and digits specifies the number of decimal places.

The abs() function returns the absolute value of a number, effectively removing any negative sign. The syntax is abs(x), where x is the number.

e) Packages in R are collections of functions and data sets developed by the R community to extend R's capabilities. Examples include:

- ggplot2: A package for creating complex and customizable visualizations.
- dplyr: A package for data manipulation and transformation.
- shiny: A package for building interactive web applications.
- caret: A package for machine learning.
- tidyr: A package for tidying data.

UNIT-7 & 8

i) sqrt(x): Computes the square root of x.

Example: sqrt(16) returns 4.

```
# Example of sqrt function
```

```
sqrt_value <- sqrt(16)
```

```
print(sqrt_value)
```

```
# Output: 4
```

log(x, base): Computes the logarithm of x to the specified base. If base is not provided, it defaults to the natural logarithm (base e).

Example: log(8, base=2) returns 3.

```
# Example of log function
```

```
log_value <- log(8, base=2)
```

```
print(log_value)
```

```
# Output: 3
```

`exp(x)`: Computes the exponential of x , i.e., e^x where e is the base of the natural logarithm.
Example: `exp(2)` returns approximately 7.389.

```
# Example of exp function
exp_value <- exp(2)
print(exp_value)
```

Output: 7.389056

j) The `rep()` function in R is used to repeat elements of vectors.

Syntax: `rep(x, times)`
`x`: The vector or value to be repeated.
`times`: The number of times to repeat each element of `x`.

Example:
Repeating the number 1 five times and repeating the vector `c(1, 2, 3)` three times.

```
# Repeat a single number
repeated_number <- rep(1, times=5)
print(repeated_number)
```

Output: 1 1 1 1 1

```
# Repeat a vector
repeated_vector <- rep(c(1, 2, 3), times=3)
print(repeated_vector)
```

Output: 1 2 3 1 2 3 1 2 3

In this example, `rep(1, times=5)` generates a vector with the number 1 repeated five times, and `rep(c(1, 2, 3), times=3)` generates a vector where the sequence 1, 2, 3 is repeated three times.

SECTION C

UNIT-5 & 6

d) `sqrt()` Function in R: The `sqrt()` function computes the square root of a given number or each element of a numeric vector.

Usage:

Single value: `sqrt(16)` returns 4.

Vector: `sqrt(c(4, 16, 25))` returns `c(2, 4, 5)`.

Example:

```
x <- 16
result <- sqrt(x)
print(result)
```

Output: 4

`order()` Function in R: The `order()` function returns the indices of the vector in ascending or descending order. It is used to sort or reorder data.

Usage:

Ascending order: `order(c(3, 1, 2))` returns `c(2, 3, 1)`.

Descending order: `order(c(3, 1, 2), decreasing = TRUE)` returns `c(1, 3, 2)`.

Example:

```
x <- c(3, 1, 2)
result <- order(x)
print(result)
```

Output: 2 3 1

In summary, `sqrt()` is used for mathematical operations to find square roots, while `order()` is used for sorting and reordering data.

UNIT-7 & 8

a) Define correlation in R with an example.

Correlation in R: Correlation is a statistical measure that describes the extent to which two variables change together. In R, correlation can be computed using the `cor()` function, which calculates the Pearson correlation coefficient by default. The Pearson correlation coefficient ranges from -1 to 1, where:

- > 1 indicates a perfect positive linear relationship,
- > -1 indicates a perfect negative linear relationship,
- > 0 indicates no linear relationship.

Example:

Let's consider two sets of data, `x` and `y`. To compute their correlation in R, you can use the following code:

```
# Define the data
x <- c(1, 2, 3, 4, 5)
y <- c(2, 4, 6, 8, 10)

# Compute the correlation
correlation <- cor(x, y)

# Print the correlation
print(correlation)
```

```
# Output
[1] 1
```

In this example, the correlation coefficient is 1, indicating a perfect positive linear relationship between x and y.

b) Explain regression analysis in R with a suitable example.

Regression Analysis in R: Regression analysis is a statistical method for modeling the relationship between a dependent variable and one or more independent variables. In R, the `lm()` function is commonly used for linear regression.

Example:

Suppose we have data on the number of hours studied (hours) and the corresponding scores (scores) obtained by students. We want to model the relationship between these two variables.

```
# Define the data
```

```
hours <- c(1, 2, 3, 4, 5)
```

```
scores <- c(2, 4, 6, 8, 10)
```

```
# Perform linear regression
```

```
model <- lm(scores ~ hours)
```

```
# Summarize the model
```

```
summary(model)
```

```
#Output
```

```
Call:
```

```
lm(formula = scores ~ hours)
```

```
Residuals:
```

```
    1    2    3    4    5
8.88e-16 4.44e-16 0.00e+00 -4.44e-16 -8.88e-16
```

```
Coefficients:
```

```
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.0000    0.0000   0.000    1
hours       2.0000    0.0000   Inf <2e-16 ***
```

```
Residual standard error: 0 on 3 degrees of freedom
```

```
Multiple R-squared:  1, Adjusted R-squared:  1
```

```
F-statistic: Inf on 1 and 3 DF, p-value: < 2.2e-16
```

The output indicates that the relationship between hours and scores is perfectly linear, with the equation $\text{scores} = 2 * \text{hours}$.

f) What is the difference between Histogram and Bar graph? Define in R Software.

Difference between Histogram and Bar Graph:

Histogram:

Used to represent the distribution of a continuous variable by dividing the data into bins (intervals) and counting the number of observations in each bin.

The x-axis represents the bins (intervals), and the y-axis represents the frequency of observations within each bin.

There are no gaps between the bars.

Bar Graph:

Used to represent categorical data with rectangular bars.

The length or height of each bar is proportional to the value it represents.

The bars can be separated by gaps to emphasize that the categories are distinct.

Example in R:

Histogram:

```
# Define data
```

```
data <- c(1, 2, 2, 3, 3, 3, 4, 4, 4, 4, 5, 5, 5, 5, 5)
```

```
# Create histogram
```

```
hist(data, main="Histogram Example", xlab="Value", ylab="Frequency", col="blue", border="black")
```

Bar graph:

```
# Define data
```

```
categories <- c("A", "B", "C", "D", "E")
```

```
values <- c(3, 7, 2, 8, 5)
```

```
# Create bar graph
```

```
barplot(values, names.arg=categories, main="Bar Graph Example", xlab="Category", ylab="Value",  
col="red", border="black")
```

i) Define and list the built-in functions of a binomial distribution in R.

Binomial Distribution in R can be handled using functions like `dbinom()`, `pbinom()`, `qbinom()`, and `rbinom()`.

`dbinom(x, size, prob)`: Probability mass function.

`pbinom(q, size, prob)`: Cumulative distribution function.

`qbinom(p, size, prob)`: Quantile function.

`rbinom(n, size, prob)`: Generates random numbers.

Example:

```
# Parameters
```

```
size <- 10
```

```
prob <- 0.5
```

```
# Probability of exactly 5 successes
```

```
dbinom(5, size, prob)
```

```
# Cumulative probability of 5 or fewer successes
```

```
pbinom(5, size, prob)
```

```
# Quantile for cumulative probability 0.5  
qbinom(0.5, size, prob)
```

```
# Generate 10 random binomial variables  
rbinom(10, size, prob)
```

j) The Poisson distribution is used to model the number of events occurring within a fixed interval of time or space. The parameter used is λ ($\hat{\lambda}$), which is the average number of events.

R functions for Poisson distribution:

```
dpois(x, lambda): Probability mass function.  
ppois(q, lambda): Cumulative distribution function.  
qpois(p, lambda): Quantile function.  
rpois(n, lambda): Generates random numbers.
```

Example:

```
# Parameter  
lambda <- 5
```

```
# Probability of exactly 3 events  
dpois(3, lambda)
```

```
# Cumulative probability of 3 or fewer events  
ppois(3, lambda)
```

```
# Quantile for cumulative probability 0.5  
qpois(0.5, lambda)
```

```
# Generate 10 random Poisson variables  
rpois(10, lambda)
```

These explanations and examples cover the specified questions in a detailed manner, suitable for a descriptive answer type section.