R-Programming

Q1. What type of programming language is R?

A. Statistical
B. General-purpose
C. Scripting
D. Object-oriented

✅ **Correct Answer:** ***Statistical***

Explanation:

R is primarily a statistical language used for data analysis and graphical representations.

Q2. Which of the following is a core feature of R?

A. GUI development
B. Web programming
C. Statistical computing
D. Mobile development

✅ **Correct Answer:** ***Statistical computing***

Explanation:

R is renowned for its capabilities in statistical computing and graphics.

Q3. R is part of the software environment for which kind of computing?

A. Quantum
B. Statistical
C. Cloud
D. Distributed

✅ **Correct Answer:** ***Statistical***

Explanation:

R is part of a software environment specifically designed for statistical computing.

Q4. Which IDE is most commonly used with R for data analysis?

A. Visual Studio Code
B. Jupyter Notebook
C. RStudio
D. Atom

✅ **Correct Answer:** ***RStudio***

Explanation:

RStudio provides tools specifically designed for R programming and data analysis.

Q5. R is part of which project that aims to provide software for statistical analysis?

A. The R Foundation
B. The R Project for Statistical Computing
C. Open Source Project
D. GNU Project

✅ **Correct Answer:** ***The R Project for Statistical Computing***

Explanation:

R is developed under the R Project, which focuses on statistical computing.

Q6. What was R based on?

A. C++
B. S
C. Java
D. Python

✅ **Correct Answer: *S***

Explanation:
R is based on the S programming language used for statistical analysis.

Q7. What type of data analysis is R particularly well-suited for?
A. Real-time
B. Exploratory
C. Big Data
D. Operational

✅ **Correct Answer: *Exploratory***

Explanation:
R excels in exploratory data analysis (EDA) using statistical summaries and visualizations.

Q8. What function in R can be used to generate a sequence of numbers?
A. seq()
B. gen()
C. createSequence()
D. listNumbers()

✅ **Correct Answer: *seq()***

Explanation:
seq() generates regular sequences of numbers in R.

Q9. What symbol is used in R for commenting out a line of code?
A. #
B. //
C. --
D. %

✅ **Correct Answer: *#***

Explanation:
The hash symbol (#) is used to write comments in R.

Q10. Which data type in R is used to store true or false values?
A. integer
B. logical
C. character
D. double

✅ **Correct Answer: *logical***

Explanation:
Logical data type stores Boolean values: TRUE or FALSE.

Q11. In R, which function is used to check the data type of a variable?
A. type()
B. typeof()

C. datatype()
D. varType()
✅ **Correct Answer:** ***typeof()***
Explanation:
typeof() returns the internal data type of an object.

Q12. Which operator is used to assign values in R?
A. <-
B. =
C. =>
D. :=
✅ **Correct Answer: <-**
Explanation:
The <- operator is the standard assignment operator in R.

Q13. What is the default data type of a number in R, like 42?
A. integer
B. logical
C. numeric
D. character
✅ **Correct Answer:** ***numeric***
Explanation:
Numbers in R default to numeric (double) unless specified otherwise.

Q14. How do you declare a constant in R?
A. Using the const keyword
B. By assigning a value once
C. Using the define keyword
D. By using the constant() function
✅ **Correct Answer:** ***By assigning a value once***
Explanation:
R does not support constants explicitly; values are treated as constants by convention.

Q15. Which function converts a character string to uppercase in R?
A. toUpper()
B. toUpperCase()
C. upperCase()
D. toupper()
✅ **Correct Answer:** ***toupper()***
Explanation:
toupper() converts characters to uppercase.

Q16. Which function creates a vector from 1 to 10?
A. c(1:10)
B. seq(1, 10)
C. vector(1, 10)
D. range(1, 10)
✅ **Correct Answer:** ***c(1:10)***

Explanation:
1:10 generates numbers from 1 to 10, wrapped into a vector.

Q17. How do you start an R script file?
A. With a function declaration
B. With libraries loading
C. With any code
D. With a shebang line
✅ **Correct Answer:** ***With any code***
Explanation:
R scripts can begin with comments, functions, or executable statements.

Q18. Which symbol is used to end a line of code in R?
A. ;
B. .
C. ,
D. |
✅ **Correct Answer:** ***;***
Explanation:
Semicolon is optional but can separate multiple statements on one line.

Q19. What function checks whether a variable is a matrix?
A. is.matrix()
B. is.array()
C. is.vector()
D. is.frame()
✅ **Correct Answer:** ***is.matrix()***
Explanation:
is.matrix() checks if an object is a matrix.

Q20. What causes an "unexpected symbol" error in R?
A. Missing comma
B. Incorrect function name
C. Syntax error
D. Unquoted string
✅ **Correct Answer:** ***Syntax error***
Explanation:
This error usually occurs due to incorrect syntax or misplaced characters.

Q21. If an R script runs but shows no output, what might be the reason?
A. Wrong file format
B. Infinite loop
C. Code is commented out
D. Libraries not loaded
✅ **Correct Answer:** ***Code is commented out***
Explanation:
Commented code does not execute, resulting in no output.

Q22. What type of data structure is a vector in R?
A. Single-element
B. Homogeneous
C. Heterogeneous
D. Multi-dimensional
✅ **Correct Answer:** ***Homogeneous***
Explanation:
Vectors store elements of the same data type.

Q23. Which function combines elements into a list?
A. combine()
B. list()
C. merge()
D. collect()
✅ **Correct Answer:** ***list()***
Explanation:
list() creates lists that can store multiple data types.

Q24. In R, what is a data frame?
A. A special type of list
B. A special type of matrix
C. A low-level data structure
D. A non-relational database
✅ **Correct Answer:** ***A special type of list***
Explanation:
Data frames are lists where each element represents a column.

Q25. How do you access the 3rd element of a vector vec?
A. vec[3]
B. vec(3)
C. vec{3}
D. vec<3>
✅ **Correct Answer:** ***vec[3]***
Explanation:
Square brackets are used to access vector elements.

Q26. Main difference between list and vector in R?
A. Lists contain only numbers
B. Vectors contain mixed types
C. Lists are 1D
D. Lists contain different data types
✅ **Correct Answer:** ***Lists contain different data types***
Explanation:
Vectors are homogeneous; lists are heterogeneous.

Q27. Which function checks if an object is a data frame?
A. is.data.frame()
B. is.list()

C. is.vector()
D. is.frame()
✅ **Correct Answer:** ***is.data.frame()***
Explanation:
This function verifies if an object is a data frame.

Q28. How to create even numbers from 2 to 20?
A. seq(2, 20, by=2)
B. even(2, 20)
C. range(2, 20, step=2)
D. vec_even(2, 20)
✅ **Correct Answer:** ***seq(2, 20, by=2)***
Explanation:
The by argument defines step size in a sequence.

Q29. How do you convert a vector to a list?
A. as.list()
B. to.list()
C. convert.list()
D. make.list()
✅ **Correct Answer:** ***as.list()***
Explanation:
as.list() converts vectors to lists.

Q30. Which function creates a data frame in R?
A. data.frame()
B. createDataFrame()
C. new.data.frame()
D. df()
✅ **Correct Answer:** ***data.frame()***
Explanation:
data.frame() is the standard function to create data frames.

Q31. Which function is most appropriate for adding a new column to a data frame in R?
A. append()
B. insert()
C. **$ operator**
D. add()
Answer: $ operator
Explanation:
The $ operator is used in R to add or access a column in a data frame by name.

Q32. If a vector operation returns unexpected 'NA' values, what is a possible cause?
A. **Missing data in vector**
B. Data type mismatch in operations
C. Incorrect function use
D. All elements are NA
Answer: Missing data in vector

Explanation:
'NA' values in a vector operation often result from operations on missing data within the vector.

Q33. What could be the issue if accessing an element beyond the length of a vector doesn't return an error but returns NA?
A. Vector is defined to have extra length
B. Vector indexing starts at 0
C. **Vector contains explicit NA values**
D. An error in R's internal handling
Answer: Vector contains explicit NA values
Explanation:
In R, attempting to access an index beyond the actual length of a vector typically returns NA, indicating an absence of data at that index.

Q34. When merging two data frames, why might some entries appear as 'NA'?
A. **Missing matches in key columns**
B. Incorrect data types
C. Syntax errors in merge function
D. All columns are incompatible
Answer: Missing matches in key columns
Explanation:
'NA' values appear during a merge operation in R if there are missing matches between the key columns of the data frames being merged.

Q35. Which control structure allows repeating a set of commands a fixed number of times in R?
A. while loop
B. **for loop**
C. if-else statement
D. repeat loop
Answer: for loop
Explanation:
A for loop in R is used to repeat a set of commands a specific number of times, iterating over a sequence or vector.

Q36. In R, what does the if statement do?
A. Executes a loop
B. **Tests a condition and executes an associated block of code**
C. Defines a function
D. Imports a package
Answer: Tests a condition and executes an associated block of code
Explanation:
The if statement in R tests a condition and if true, executes the code block that follows.

Q37. What is the use of the break statement in R?
A. **Terminates a loop or switch statement**
B. Starts a new iteration of a loop
C. Skips the current iteration of a loop
D. Exits the program
Answer: Terminates a loop or switch statement

Explanation:
The break statement is used in R to immediately terminate the execution of a loop or switch statement.

Q38. Which statement is used in R to skip the current iteration of a loop and begin the next one?
A. continue
B. **next**
C. skip
D. break
Answer: next
Explanation:
The next statement in R is used to skip the current iteration of a loop and start the next iteration.

Q39. How can you execute a block of code multiple times conditionally in R?
A. Using the if loop
B. Using the for loop
C. **Using the while loop**
D. Using the do loop
Answer: Using the while loop
Explanation:
The while loop in R executes a block of code repeatedly as long as the specified condition is true.

Q40. What is the purpose of the else if statement in R?
A. **To specify a new condition to test, if the first condition is false**
B. To exit a loop
C. To define a function
D. To repeat a condition
Answer: To specify a new condition to test, if the first condition is false
Explanation:
The else if statement allows for multiple conditional tests within an if-else structure, providing alternative conditions if the first is false.

Q41. Which syntax correctly starts a for loop in R that iterates from 1 to 10?
A. for (i = 1 to 10) { }
B. for i in 1:10 { }
C. **for (i in 1:10) { }**
D. for i from 1 to 10 { }
Answer: for (i in 1:10) { }
Explanation:
In R, the correct syntax for a for loop that iterates from 1 to 10 is for (i in 1:10) { }.

Q42. How do you write an infinite loop in R?
A. while (true) { }
B. while (1) { }
C. **Both A and B are correct**
D. Neither A nor B are correct
Answer: Both A and B are correct
Explanation:
Both while (true) { } and while (1) { } create infinite loops in R, as the condition remains always true.

Q43. What does the following loop do?
for (i in 1:5) print(i^2)
A. **Prints the squares of numbers 1 to 5**
B. Prints numbers from 1 to 5
C. Counts from 1 to 5
D. None of the above
Answer: Prints the squares of numbers 1 to 5
Explanation:
This loop iterates over numbers 1 to 5, and for each number, it prints its square (i^2).

Q44. In a while loop, what happens if the condition never becomes false?
A. The loop terminates after a set number of iterations
B. **The loop continues indefinitely**
C. The loop skips iterations
D. The loop throws an error
Answer: The loop continues indefinitely
Explanation:
If the condition in a while loop never becomes false, the loop will continue to execute indefinitely, potentially creating an infinite loop.

Q45. What is a common mistake when using nested loops in R?
A. **Mismanaging loop counters**
B. Using the wrong loop type
C. Not nesting loops correctly
D. Forgetting to close brackets
Answer: Mismanaging loop counters
Explanation:
A common issue with nested loops is mismanaging the loop counters, which can lead to incorrect results or infinite loops.

Q46. If a loop is supposed to run but doesn't start, what could be a potential issue?
A. **The loop condition is initially false**
B. The loop is improperly nested
C. There are syntax errors in the loop code
D. All of the above
Answer: The loop condition is initially false
Explanation:
If a loop's initial condition is false, the loop will not execute even once, which might be overlooked when setting up the loop's conditions.

Q47. Which function in R is used to install packages from CRAN?
A. **install.packages()**
B. get.packages()
C. load.packages()
D. add.packages()
Answer: install.packages()
Explanation:
The install.packages() function is used to install new packages from CRAN, making them available for use in R.

Q48. What is a primary use of the library() function in R?
A. Create new libraries
B. Delete existing libraries
C. **Load installed packages into the R session**
D. Update libraries
Answer: Load installed packages into the R session
Explanation:
The library() function is used to load installed packages into the current R session, enabling their functions and datasets.

Q49. In R, what is the purpose of the source() function?
A. To load data from external sources
B. **To execute R scripts from files**
C. To generate reproducible random numbers
D. To integrate R with other programming languages
Answer: To execute R scripts from files
Explanation:
The source() function in R is used to execute R code contained in files, making it useful for running entire scripts.

Q50. What does the apply() function do in R?
A. Modifies data frames
B. **Applies a function over the margins of an array or matrix**
C. Creates graphical plots
D. None of the above
Answer: Applies a function over the margins of an array or matrix
Explanation:
The apply() function is used to apply a function to the rows or columns of an array or matrix in R.

Q51. Which statement best describes the role of user-defined functions in R?
A. They replace all basic functions of R
B. They are only for advanced users
C. **They allow customization of tasks**
D. They are not recommended
Answer: They allow customization of tasks
Explanation:
User-defined functions in R allow for task customization, enabling specific operations to be packaged into reusable blocks of code.

Q52. What is the function set.seed() used for in R?
A. To establish the starting point for loops
B. **To control the reproducibility of random number generation**
C. To seed the R environment with data
D. To initialize package settings
Answer: To control the reproducibility of random number generation
Explanation:
The set.seed() function is crucial for ensuring that random number generation is reproducible, allowing for consistent results across sessions and environments.

Q53. How do you create a simple function in R that takes one argument x and returns x squared?
A. def square(x) { return xx }
B. function(x) { return xx }
C. function(x) { xx }
D. **square <- function(x) { x*x }**
Answer: square <- function(x) { x*x }
Explanation:
In R, functions are defined using the function keyword, and square <- function(x) { x*x } correctly defines a function to return the square of its argument.

Q54. How do you ensure that a package is loaded only if it is already installed in R?
A. if (require(package)) library(package)
B. if (library(package))
C. if (installed.packages(package)) library(package)
D. **if (require(package))**
Answer: if (require(package))
Explanation:
The require() function in R checks if a package is installed and loads it if it is, returning FALSE otherwise, which is useful for conditional loading.

Q55. When defining a function in R, what is the effect of setting a default value for an argument?
A. **It makes the argument optional**
B. It makes the function faster
C. It limits the function to specific data types
D. It creates a constant
Answer: It makes the argument optional
Explanation:
Setting a default value for an argument in an R function makes that argument optional, allowing the function to be called without explicitly passing that parameter.

Q56. If a function fails because an argument is not correctly passed, what could be the probable issue?
A. **Incorrect data type of argument**
B. Missing function definition
C. Syntax error in function call
D. Incompatible library version
Answer: Incorrect data type of argument
Explanation:
The most common cause of function failure related to arguments is the incorrect data type being passed, which doesn't match expected parameter requirements.

Q57. What should you check if a user-defined function in R isn't recognized in your script?
A. **Whether the function is defined after it's called**
B. Whether the package containing the function is loaded
C. Whether the script file is corrupted
D. Whether the R version is outdated
Answer: Whether the function is defined after it's called
Explanation:
In R, functions must be defined before they are called in the script. If a function is used before its definition, it will not be recognized.

Q58. Which dplyr function is used to select columns from a data frame?
A. **select()**
B. filter()
C. arrange()
D. mutate()
Answer: select()
Explanation:
The select() function in dplyr is used to select columns from a data frame, based on specific criteria.

Q59. What does the mutate() function in dplyr do?
A. **Adds new variables to a data frame and preserves existing ones**
B. Filters rows based on conditions
C. Sorts a data frame
D. Summarizes complex calculations
Answer: Adds new variables to a data frame and preserves existing ones
Explanation:
The mutate() function in dplyr is used to add new columns to a data frame that are functions of existing columns, effectively "mutating" the data frame by adding new data.

Q60. How does the filter() function in dplyr determine which rows to keep in a data frame?
A. **By evaluating conditions set on columns**
B. By matching patterns in row names
C. By comparing row indices
D. By data type of rows
Answer: By evaluating conditions set on columns
Explanation:
The filter() function in dplyr selects rows in a data frame based on conditions that evaluate to true for the specified columns.

Q61. Which dplyr function is used to summarize multiple values into a single value?
A. summarize()
B. collect()
C. condense()
D. reduce()
Answer: summarize()
Explanation:
The summarize() function in dplyr is used to reduce multiple values down to a single summary value per group, often used with group_by() to apply summaries within groups.

Q62. What is the purpose of the group_by() function in dplyr?
A. To split a data frame into groups based on one or more variables
B. To arrange groups in a specific order
C. To merge two groups
D. To compare groups
Answer: To split a data frame into groups based on one or more variables

Explanation:
The group_by() function in dplyr groups data by one or more variables, allowing subsequent summarization and mutation operations to be performed "by group."

Q63. What does the arrange() function in dplyr do when applied to a data frame?
A. It arranges the rows in descending order
B. It arranges the rows based on the values in one or more columns
C. It randomly arranges the rows
D. It converts the rows into columns
Answer: It arranges the rows based on the values in one or more columns
Explanation:
The arrange() function in dplyr sorts the rows of a data frame ascending by default, based on the values in specified columns, which can be set to descending with additional parameters.

Q64. Which function in dplyr is best for changing the names of columns in a data frame?
A. rename()
B. alter()
C. change()
D. modify()
Answer: rename()
Explanation:
The rename() function in dplyr is used to change column names in a data frame, providing a simple way to update or correct column identifiers.

Q65. How can you combine two data frames vertically using dplyr?
A. bind_rows()
B. merge()
C. concat()
D. union()
Answer: bind_rows()
Explanation:
The bind_rows() function in dplyr is used to combine data frames by rows, effectively appending one data frame beneath another.

Q66. What dplyr function would you use to join two data frames by columns when the columns have different names in each frame?
A. join()
B. merge()
C. full_join()
D. left_join(), with the by argument specifying the matching columns
Answer: left_join(), with the by argument specifying the matching columns
Explanation:
Using left_join() with the by argument allows specifying how columns from different data frames should be matched when they have different names, facilitating a flexible merging process.

Q67. How do you create a new column in a data frame that calculates the logarithm of an existing column in dplyr?
A. mutate(log_column = log(existing_column))
B. create(log_column = log(existing_column))

C. add(log_column = log(existing_column))
D. set(log_column = log(existing_column))
Answer: mutate(log_column = log(existing_column))
Explanation:
The mutate() function in dplyr is used to add new columns to a data frame, here creating a new column by applying the logarithm function to values in an existing column.

Q68. Why might changes made by mutate() in dplyr not appear in the original data frame?
A. mutate() does not modify in place
B. mutate() syntax errors
C. mutate() used on an empty data frame
D. mutate() is not supported for data frames
Answer: mutate() does not modify in place
Explanation:
In dplyr, the mutate() function does not modify the original data frame in place. Changes need to be explicitly saved to a variable to update the data frame with the new columns or values.

Q69. What could cause summarize() in dplyr to return fewer rows than expected when using a group_by() statement?
A. Incorrect grouping
B. Exclusion of NA values in groups
C. Overly restrictive aggregation functions
D. Summarize is case-sensitive
Answer: Incorrect grouping
Explanation:
If summarize() returns fewer rows than expected, it could be due to incorrect grouping in group_by(), which affects how data is aggregated. This often happens if the groups are not defined as intended.

Q70. What is ggplot2 primarily used for in R?
A. Database management
B. Statistical analysis
C. Data visualization
D. Machine learning
Answer: Data visualization
Explanation:
ggplot2 is a data visualization package in R, widely used for creating complex and aesthetically pleasing graphical representations of data.

Q71. Which ggplot2 function is used to create a scatter plot?
A. ggsave()
B. ggplot() + geom_point()
C. ggplot() + geom_line()
D. ggplot() + geom_histogram()
Answer: ggplot() + geom_point()
Explanation:
The ggplot() + geom_point() combination in ggplot2 is used to create scatter plots by mapping variables to the x and y axes and displaying points.

Q72. In ggplot2, what does the aes() function specify?

A. Data sources
B. Aesthetics mappings
C. Plot dimensions
D. Saving options
Answer: Aesthetics mappings
Explanation:
The aes() function in ggplot2 is used to specify aesthetic mappings like x, y, color, fill, etc., which determine how data is displayed in plots.

Q73. What is the purpose of the facet_wrap() function in ggplot2?
A. To wrap text labels
B. To create multi-panel plots
C. To apply themes to plots
D. To adjust plot margins
Answer: To create multi-panel plots
Explanation:
The facet_wrap() function in ggplot2 is used to create multi-panel plots, allowing the display of different subsets of the data in separate panels based on a factor.

Q74. How does theme() function affect a ggplot2 plot?
A. It changes the data source for the plot
B. It modifies aesthetic attributes
C. It alters plot appearance by modifying non-data components
D. It adjusts data alignment
Answer: It alters plot appearance by modifying non-data components
Explanation:
The theme() function in ggplot2 is used to fine-tune the appearance of non-data elements of a plot, such as text, labels, and grid lines.

Q75. Which function saves a ggplot2 plot to a file?
A. ggsave()
B. ggplot_save()
C. save_ggplot()
D. export()
Answer: ggsave()
Explanation:
The ggsave() function in ggplot2 is designed to save plots to a file, supporting various file formats like PNG, PDF, and JPEG.

Q76. To add a linear regression line to a scatter plot in ggplot2, which function would you use?
A. geom_smooth(method = "lm")
B. geom_line()
C. geom_path()
D. geom_segment()
Answer: geom_smooth(method = "lm")
Explanation:
The geom_smooth(method = "lm") function adds a linear regression line to a scatter plot, indicating trends in the data.

Q77. How do you add labels to the axes in a ggplot2 plot?
A. labs(x = "X-axis label", y = "Y-axis label")
B. annotate()
C. labels()
D. axis_lab()
Answer: labs(x = "X-axis label", y = "Y-axis label")
Explanation:
The labs() function is used in ggplot2 to add or modify labels for the x and y axes, enhancing the clarity and information of the plot.

Q78. What does setting the alpha parameter in ggplot2 accomplish?
A. It adjusts the plot's aspect ratio
B. It changes the data points' color
C. It sets the transparency of elements
D. It scales the axes
Answer: It sets the transparency of elements
Explanation:
The alpha parameter in ggplot2 controls the transparency of elements in the plot, useful for overlapping points or when working with dense data.

Q79. If a ggplot2 plot appears empty, what might be a common issue?
A. Missing data
B. Incorrect data mappings in aes()
C. Data not loaded into R
D. All listed issues
Answer: Incorrect data mappings in aes()
Explanation:
An empty ggplot2 plot often results from incorrect mappings specified in the aes() function, such as incorrect variable names or data types.

Q80. What could cause the colors in a ggplot2 plot to not display as expected?
A. Incorrect color names in the aes() function
B. Incompatible color values
C. Misconfigured display settings
D. Both incorrect color names and incompatible color values

Q81. What function is used to calculate the mean of a numeric vector in R?
A. mean()
B. average()
C. median()
D. sum()
Answer: mean()
Explanation:
The mean() function in R is used to calculate the average value of a numeric vector.

Q82. Which function in R provides a summary of the central tendency, dispersion, and shape of a dataset's distribution?
A. summary()
B. describe()

C. info()
D. data_summary()
Answer: summary()
Explanation:
The summary() function in R generates a statistical summary of the dataset, including measures like the mean, median, min, and max.

Q83. In R, what does the sd() function calculate?
A. Standard deviation
B. Sum of deviations
C. Sample distribution
D. Standard data
Answer: Standard deviation
Explanation:
The sd() function calculates the standard deviation of a given numeric vector, indicating the dispersion around the mean.

Q84. What is the purpose of the cor() function in R?
A. To perform correlation analysis between two or more variables
B. To compare datasets
C. To merge data frames
D. To create plots
Answer: To perform correlation analysis between two or more variables
Explanation:
The cor() function calculates the correlation coefficient between two or more variables, helping to understand the degree of linear relationship between them.

Q85. How do you test for normality of a dataset in R?
A. Using the norm.test() function
B. Using the shapiro.test() function
C. Using the normality() function
D. Using the gauss.test() function
Answer: Using the shapiro.test() function
Explanation:
The shapiro.test() function is used for testing the normality of data distribution, checking how well the data conforms to a normal distribution.

Q86. What R function is used to generate a linear model for regression analysis?
A. lm()
B. linear()
C. regression()
D. model()
Answer: lm()
Explanation:
The lm() function in R is used to fit linear models, commonly used for regression analysis to describe relationships between variables.

Q87. How can you generate a frequency table of a categorical variable in R?

A. table()
B. freq()
C. categorical()
D. group()
Answer: table()
Explanation:
The table() function in R generates a frequency table, which is useful for summarizing categorical variables.

Q88. What is the R command to calculate the median of a numeric vector?
A. median()
B. middle()
C. center()
D. mean()
Answer: median()
Explanation:
The median() function in R is used to find the median value of a numeric vector, representing the middle point of the data.

Q89. How do you perform a t-test to compare the means of two groups in R?
A. t.test(data$group1, data$group2)
B. compare.means(data$group1, data$group2)
C. groups.ttest(data$group1, data$group2)
D. test.t(data$group1, data$group2)
Answer: t.test(data$group1, data$group2)
Explanation:
The t.test() function in R is used for conducting a t-test, which helps in comparing the means of two independent groups to see if there is a statistically significant difference.

Q90. What might be the reason if lm() fails to fit a model in R?
A. Incorrect data types
B. Missing data
C. Both incorrect data types and missing data
D. None of the above
Answer: Incorrect data types
Explanation:
Incorrect data types in the variables used for fitting a model using lm() can prevent the function from executing properly.

Q91. Why might the sd() function return an error when applied to a dataset in R?
A. The dataset contains non-numeric values
B. The dataset is empty
C. Both reasons mentioned
D. None of the above
Answer: The dataset contains non-numeric values
Explanation:
The sd() function requires numeric input, and applying it to non-numeric data or an empty dataset will result in an error.

Q92. What function is used to read CSV files into R?

A. **read.csv()**
B. load.csv()
C. import.csv()
D. fetch.csv()
Answer: read.csv()
Explanation:
The read.csv() function is the standard R function used to load data from CSV files into R as a data frame.

Q93. Which function in R is used to write data frames to a CSV file?
A. **write.csv()**
B. export.csv()
C. save.csv()
D. output.csv()
Answer: write.csv()
Explanation:
The write.csv() function is used to save R data frames to CSV files, allowing for data exportation.

Q94. What is the purpose of the file() function in R?
A. To delete files
B. **To create a connection to a file for reading or writing**
C. To list files
D. To copy files
Answer: To create a connection to a file for reading or writing
Explanation:
The file() function in R creates a connection object to a file, which can be used for reading from or writing to the file.

Q95. In R, what does the read.table() function do?
A. **Reads data stored in a table format**
B. Creates a table from vector data
C. Visualizes data in table format
D. None of the above
Answer: Reads data stored in a table format
Explanation:
The read.table() function is used for reading data from file in table format, typically used for more complex data arrangements than read.csv().

Q96. What is the main difference between read.csv() and read.csv2() in R?
A. The former is used for reading large files, the latter for small files
B. **The former uses comma as delimiter, the latter uses semicolon**
C. The former supports UTF-8 encoding, the latter does not
D. There is no difference
Answer: The former uses comma as delimiter, the latter uses semicolon

---

Q97. How do you save an R data frame as a tab-separated values file?
A. **write.table(df, "data.tsv", sep="\t")**
B. save.table(df, "data.tsv", sep="\t")
C. write.csv(df, "data.tsv", sep="\t")
D. output.tsv(df, "data.tsv", sep="\t")

Answer: write.table(df, "data.tsv", sep="\t")
Explanation:
The write.table() function with sep="\t" is used to write data frames to a file with tab-separated values, suitable for .tsv file formats.

Q98. Which function would you use to read a JSON file into R?
A. read.json()
B. **fromJSON()**
C. json()
D. load.json()
Answer: fromJSON()
Explanation:
The fromJSON() function from the jsonlite package in R is commonly used to parse JSON files into R readable formats like lists or data frames.

Q99. What is the correct way to read an Excel file into R using the readxl package?
A. **read_excel("data.xlsx")**
B. load_excel("data.xlsx")
C. excel_read("data.xlsx")
D. get_excel("data.xlsx")
Answer: read_excel("data.xlsx")
Explanation:
The read_excel() function from the readxl package is specifically designed for importing Excel files into R.

Q100. How can you connect to and read from a text file that is continuously being updated, in R?
A. **file("log.txt", "r") and then use readLines()**
B. open("log.txt", "r") and use readText()
C. connect("log.txt") and use getText()
D. link("log.txt", "read")
Answer: file("log.txt", "r") and then use readLines()
Explanation:
Using file() with "r" opens a read connection to a file, and readLines() can be used to read data from it, useful for logs or streaming data.

Q101. If write.csv() fails to save a data frame with error in R, what might be a common cause?
A. Incorrect file path
B. File is open in another program
C. **Both reasons**
D. None of the above
Answer: Both reasons
Explanation:
A common cause for write.csv() failures includes using an incorrect file path or attempting to write to a file that is currently open in another program.

Q102. Why might read.table() return an error when reading a file that includes different data types?
A. **The function expects uniform data types**
B. Incorrect delimiter used
C. File contains improper formatting
D. Both the first and second options

Answer: The function expects uniform data types
Explanation:
read.table() may return errors when encountering columns with mixed data types that it cannot correctly interpret as one type, especially without specifying colClasses.

Q103. What is an environment in R?
A. A collection of data types
B. A collection of packages
C. **A collection of symbol-value pairs**
D. A collection of functions
Answer: A collection of symbol-value pairs
Explanation:
An environment in R is a collection of symbol-value pairs, where symbols are the variable names and values are the data associated with them.

Q104. How does R determine the value of a variable when it is used in a function?
A. By checking global variables only
B. **By searching the function's environment and then parent environments**
C. By checking the global environment first
D. By looking only in the base environment
Answer: By searching the function's environment and then parent environments
Explanation:
R uses lexical scoping to find the value of a variable, searching the function's environment and then parent environments up to the global environment.

Q105. Which environment is at the top of the environment hierarchy in R?
A. Base environment
B. Global environment
C. **Empty environment**
D. Parent environment
Answer: Empty environment
Explanation:
The empty environment is at the top of R’s environment hierarchy and has no parent, thus ending the chain of environment searches.

Q106. What does the <<- operator do in R?
A. Assigns a value in the current environment
B. Assigns a value in the global environment
C. **Assigns a value in the parent environment**
D. None of the above
Answer: Assigns a value in the parent environment
Explanation:
The <<- operator assigns a value to a variable in the parent environment, used to modify variables outside the current function scope.

Q107. How do you create a new environment in R and assign a variable to it?
A. **new.env(); e$a <- 1**
B. env(); e$a <- 1

C. create.env(); e$a <- 1
D. new.environment(); e$a <- 1
Answer: new.env(); e$a <- 1
Explanation:
The new.env() function creates a new environment, and variables can be assigned to it using the $ operator.

Q108. How do you list all objects in the current environment?
A. **ls()**
B. objects()
C. list()
D. all_objects()
Answer: ls()
Explanation:
The ls() function lists all objects in the current environment, including variables and functions.

Q109. How can you access the parent environment of a given environment env in R?
A. **parent.env(env)**
B. env$parent
C. parent(environment)
D. get.parent(env)
Answer: parent.env(env)
Explanation:
The parent.env() function is used to access the parent environment of a given environment, allowing navigation through the environment hierarchy.

Q110. If a variable in a function is not found in the current environment, what is R's next step in finding it?
A. R throws an error
B. R searches in the global environment
C. **R searches in the parent environment**
D. R returns NULL
Answer: R searches in the parent environment
Explanation:
If a variable is not found in the current environment, R follows lexical scoping rules and searches in the parent environment.

Q111. How can you avoid potential scoping issues in R functions when using global variables?
A. Always use <<- operator
B. Prefix global variables with global::
C. **Avoid using global variables**
D. Use local variables with the same name
Answer: Avoid using global variables
Explanation:
Avoiding global variables helps prevent unintended side effects and scoping issues in R functions.

Q112. What is the primary purpose of the try() function in R?
A. **To attempt execution of code that might fail without stopping the entire script**
B. To handle errors silently
C. To debug code
D. To improve code performance

Answer: To attempt execution of code that might fail without stopping the entire script
Explanation:
The try() function allows code execution to continue even if an error occurs, preventing the script from stopping completely.

Q113. What does the traceback() function do in R?
A. **Shows the call stack leading to the error**
B. Provides detailed error descriptions
C. Fixes the error automatically
D. Logs the error to a file
Answer: Shows the call stack leading to the error
Explanation:
The traceback() function displays the sequence of function calls that led to an error, helping to identify where the error occurred in the code.

Q114. Which function in R can be used to provide custom error messages?
A. **stop()**
B. error()
C. alert()
D. throw()
Answer: stop()
Explanation:
The stop() function in R can be used to generate custom error messages, terminating the execution and displaying the specified message.

Q115. How do you use the tryCatch() function to handle errors in R?
A. **tryCatch(expr, error = function(e) { ... })**
B. tryCatch(expr, silent = TRUE)
C. try(expr)
D. catch(expr)
Answer: tryCatch(expr, error = function(e) { ... })
Explanation:
The tryCatch() function allows you to handle errors by specifying an expression to try and providing a function to execute if an error occurs.

Q116. How can you use debug() to step through a function named my_func in R?
A. **debug(my_func); my_func()**
B. step(my_func)
C. trace(my_func)
D. watch(my_func)
Answer: debug(my_func); my_func()
Explanation:
The debug() function enables step-by-step execution of the specified function, allowing you to inspect its behavior during execution.

Q117. If you encounter an "object not found" error in R, what is a likely cause?
A. **Misspelled variable name**
B. Incorrect function call

C. Package not loaded
D. All of the above
Answer: Misspelled variable name
Explanation:
An "object not found" error typically occurs due to a misspelled variable name, causing R to search for a variable that does not exist in the current environment.

Q118. How can you debug an R script that runs without error but produces incorrect results?
A. Use print statements to inspect variables
B. Use a step-through debugger
C. Check for logical errors
D. **All of the above**
Answer: All of the above
Explanation:
All these techniques help identify logical or calculation issues in scripts that execute without runtime errors.

Q119. When using tryCatch(), how can you capture and analyze the error object?
A. **error = function(e) { print(e) }**
B. catch = function(e) { analyze(e) }
C. errorHandler(e) { inspect(e) }
D. tryError(e) { log(e) }
Answer: error = function(e) { print(e) }
Explanation:
The error handler function within tryCatch() can capture the error object, allowing you to print or analyze it for further debugging.

Q120. Which function is used to get the current date and time in R?
A. now()
B. Sys.Date()
C. **Sys.time()**
D. current()
Answer: Sys.time()
Explanation:
The Sys.time() function in R returns the current date and time.