Share on your Social Media

Top 20 R Programming Interview Questions and Answers

Published On: September 14, 2019

R Programming Interview Questions and Answers

R Programming language is the most demanded programming language in the realm of statistical computing and graphics. Due to its vibrant community of users and developers who contribute to its ecosystem by creating and sharing packages, tutorials, and resources, R Programming language is widely used in academia, research, and industry for data analysis, statistical modeling, and data visualization. Therefore, a career in R Programming is surely a long-lasting one. These **R Programming Interview Questions and Answers** will surely make you land a job in the R Programming sector. So, explore our interview questions and answers to get a breakthrough job in R Programming.

Related Courses at SLA

- R Programming Training in OMR
- R Programming Training in Chennai

Related Posts

Top 12 Business Intelligence and Data Analytics Interview

Download R Programming Interview Questions PDF

R Programming Interview Questions and Answers

1. What is R Programming?

R is a programming language and environment widely utilized for statistical computing and graphics. It offers numerous statistical and graphical techniques like linear and nonlinear modeling, time-series analysis, classification, and clustering. R's extensibility through packages, which contain functions, data, and documentation, enhances its capabilities. With thousands of available packages, R is a versatile tool for tasks such as data analysis, visualization, and **machine learning**.

2. What distinguishes apply() from lapply() functions in R?

apply() operates on array margins, such as matrices or arrays, while lapply() operates on lists.

3. How do you manage missing values in an R dataset?

Missing values can be managed using functions like na.omit() to remove rows with missing values, complete.cases() to identify complete cases, or na.fill() to replace missing values with specified values.

4. Define closures in R.

Closures are functions created within another function that preserve the environment in which they were created, allowing access to the variables in their parent function's environment even after the parent function has finished executing.

5. What strategies can you employ to accelerate calculations in R?

Questions and Answers
Published On: June 19, 2024

Business Intelligence and Data Analytics Interview Questions and Answers Our Business Intelligence and Data Analytics...

Top 20 Azure DevOps Interview Questions and Answers
Published On: June 19, 2024

Azure DevOps Interview Questions and Answers One of the most in-demand skills in the IT...

Top 20 AWS DevOps Interview Questions and Answers
Published On: June 19, 2024

AWS DevOps Interview Questions and Answers The integration of AWS and DevOps is doing magic...

Top 14 MEAN Stack Interview Questions and Answers
Published On: June 19, 2024

Techniques for speeding up calculations include vectorization, utilizing specialized packages like data.table or dplyr, employing parallel processing with packages like parallel or foreach, and optimizing code for performance.

Download R Programming Syllabus PDF

6. Explain the concept of method dispatch in S3 object-oriented programming in R.

Method dispatch in S3 involves R determining which method to use for a generic function based on the class of the object passed to it. It searches for methods with matching names and class attributes, utilizing a class hierarchy if necessary.

7. How does R manage memory, and what tactics can you employ to enhance memory usage?

R utilizes garbage collection for memory management, but users can also manage memory manually using functions like gc() to trigger garbage collection. Techniques for optimizing memory usage include avoiding unnecessary object copying, removing objects from memory when they're no longer needed, and utilizing memory-efficient data structures like matrices over data frames when feasible.

8. Define lazy evaluation in R and discuss its impact on performance.

Lazy evaluation in R refers to delaying the evaluation of expressions until their results are needed. While this can enhance performance by avoiding unnecessary computations, it may lead to unexpected behavior if side effects are involved.

9. How would you profile R code to pinpoint performance bottlenecks?

R provides profiling tools such as Rprof() and the profvis package to identify performance bottlenecks. These tools generate reports highlighting which functions consume the most time and memory, enabling developers to optimize those sections of the code.

10. What role does the %>% operator serve in R, and how do you use it?

The %>% operator, known as the pipe operator, facilitates chaining multiple operations in a sequence. It takes the output from the left-hand expression and feeds it as the first argument to the function on the right-hand side.

11. Define factor variables in R and discuss their significance.

Factor variables in R are employed to represent categorical data, storing both the values of the categorical variable and their associated levels. They are particularly valuable for statistical modeling and analysis tasks.

R Programming Salary

12. What do closures signify in R, and why are they beneficial?

Closures in R pertain to functions crafted within another function, retaining access to the environment in which they were created. They prove advantageous for developing functions with specialized behavior dependent on variables from their enclosing environment.

13. How do you import data from a CSV file into R?

Data from a CSV file can be brought into R via the read.csv() function. For instance:

```
data <- read.csv(“data.csv”)
```

14. Enumerate some common data

manipulation tasks achievable using the dplyr package.

The dplyr package offers functions catering to various data manipulation tasks, including filtering rows (filter()), selecting columns (select()), arranging rows (arrange()), mutating columns (mutate()), summarizing data (summarize()), and grouping data (group_by()).

15. How can you create a scatter plot in R?

To create a scatter plot in R, utilize the plot() function with the type = “p” argument. For instance:

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

y <- c(2, 3, 5, 7, 11)

plot(x, y, type = “p”)
```

16. Elaborate on the concept of vectorization in R and its advantages.

Vectorization in R denotes the capacity to execute operations on entire vectors or arrays simultaneously, bypassing the need for individual element looping. It results in cleaner, more concise code and often leads to faster execution times.

17. How would you generate random numbers conforming to a normal distribution in R?

To generate random numbers adhering to a normal distribution in R, utilize the rnorm() function. For example:

```
random_numbers <- rnorm(100, mean = 0, sd = 1)
```

18. What are the advantages of using R Programming?

The following are the advantages of using R Programming:

- **Rich Statistical and Graphical Capabilities:** R offers an extensive array of statistical and graphical techniques, empowering users with robust tools for data analysis, visualization, and modeling.

- **Extensive Package Ecosystem:** With a dynamic community behind it, R boasts a plethora of packages spanning diverse domains such as machine learning, time series analysis, and bioinformatics. These packages greatly enhance R's functionality, enabling users to tap into advanced analytical methods effortlessly.
- **Open Source and Free:** R stands as an open-source language, granting users the liberty to utilize, adapt, and distribute it without cost. This accessibility democratizes access to R, catering to a broad audience including academics, researchers, and professionals.

R Programming Tutorials

19. What are some of the famous data visualization packages in R Programming?

The following are some of the famous data visualization packages in R Programming:

- **ggplot2:** Crafted by Hadley Wickham, ggplot2 stands as one of R's premier data visualization packages. Employing a layered grammar of graphics approach, it empowers users to fashion intricate and highly adaptable visualizations effortlessly.
- **plotly:** A dynamic graphing library, plotly facilitates the creation of interactive web-based visualizations directly from R. It encompasses a diverse array of chart types and customization possibilities, alongside features like zooming, panning, and tooltips for enhanced interactivity.
- **ggvis:** Developed by Hadley Wickham, ggvis extends the grammar of graphics framework

to interactive web-based visualizations. Seamlessly integrating with Shiny, it enables users to craft interactive dashboards and web applications with ease.

- **leaflet:** Renowned for its prowess in interactive mapping, leaflet is a favored choice for creating interactive maps in R. It offers extensive support for various mapping features such as markers, polygons, heatmaps, and overlays, while harmonizing seamlessly with other spatial data analysis packages.
- **dygraphs:** Tailored for interactive time series plotting, dygraphs facilitates the creation of dynamic visualizations replete with features like zooming, panning, and mouseover tooltips. Its versatility renders it an ideal tool for visualizing time series data comprehensively.
- **lattice:** Recognized for its potency in generating trellis plots, lattice stands as a powerful package for crafting multi-panel displays of data. Equipped with a high-level interface, it expedites the creation of conditioned plots, facilitating swift visualization of relationships between variables across multiple dimensions.
- **rgl:** Catering to the realm of three-dimensional visualization, rgl emerges as a formidable package for crafting interactive 3D plots in R. Particularly adept at visualizing intricate three-dimensional datasets and exploring spatial relationships, it offers a powerful toolset for immersive data exploration.

20. What are the requirements for naming variables in R Programming?

The following are the requirements for naming variables in R Programming:

- **Validity:** In R programming, variable names must initiate with a letter (uppercase or lowercase) or a dot (.), followed by letters, digits, underscores (_), or dots. However, they cannot commence with a digit or dot followed by a digit.

- **Length:** Variable names in R can extend up to 10,000 characters.
- **Case Sensitivity:** R distinguishes between uppercase and lowercase letters, hence “myVar” and “myvar” would be recognized as distinct variables.
- **Reserved Keywords:** Variable names cannot coincide with reserved keywords in R, such as “if,” “else,” “for,” “function,” etc. Trying to use a reserved keyword as a variable name will prompt an error.
- **Descriptive and Readable:** It’s advisable to employ descriptive and meaningful names for variables to enhance code readability and maintainability.
- **Avoid Special Characters:** While underscores (_) and dots (.) are permissible in variable names, other special characters like spaces, hyphens (-), arithmetic operators (+, -, *, /), and logical operators (&, |) are not allowed.
- **Consistency:** Maintaining consistency in variable naming across your codebase is a recommended practice to prevent confusion and enhance code clarity.

Conclusion

We have curated these interview questions and answers purely for the welfare of our candidates. By learning from our **R Programming Interview Questions and Answers**, students can easily secure a job in the R Programming sector without any hassle. Therefore, we hope that this interview questions and answers serves you in the best possible manner.

R Programming Training

Share on your Social Media

Navigation

- About Us
- Blog Posts
- Careers
- Contact
- Placement Training
- Corporate Training
- Hire With Us
- Job Seekers
- SLA’s Recently Placed Students
- Reviews
- Sitemap

Important Links

- Disclaimer
- Privacy Policy
- Terms and Conditions

Softlogic Academy

Softlogic Systems

KK Nagar [Corporate Office]

No.10, PT Rajan Salai, K.K. Nagar, Chennai – 600 078.
Landmark: Karnataka Bank Building
Phone: +91 86818 84318
Email: enquiry@softlogicsys.in
Map: Google Maps Link

OMR

No. E1-A10, RTS Food Street
92, Rajiv Gandhi Salai (OMR),
Navalur, Chennai - 600 130.
Landmark: Adj. to AGS Cinemas
Phone: +91 89256 88858
Email: info@softlogicsys.in
Map: Google Maps Link

Courses

- Python
- Software Testing
- Full Stack Developer
- Java

Social Media Links

Review Sources

- Google

Power BI

Clinical SAS

Data Science

Embedded

Cloud Computing

Hardware and Networking

VBA Macros

Mobile App Development

DevOps

Trustpilot

Glassdoor

Mouthshut

Sulekha

Justdial

Ambitionbox

Indeed

Software Suggest

Sitejabber