1). What is R programming language?

Ans: R is a programming language and software environment for statistical analysis, graphics representation and reporting. R was created by Ross Ihaka and Robert Gentleman at the University of Auckland, New Zealand, and is currently developed by the R Development Core Team.

2). Why this is named as R programming language?

Ans:This programming language was named R based on the first letter of first name of the two R authors (Robert Gentleman and Ross Ihaka), and partly a play on the name of the Bell Labs language S.

3). What is R used for?

Ans:R is used mostly for statistics and data modeling, but it is also used to extract data from grap

s. It contains standard and recommended packages used for storing functions and data sets. R uses features from S, a statistical system that is commonly used by statisticians. S processes statistical analysisin series with only halfway results, but R will provide minimal output and store results for assessment later.

4). Does R allows integration with the procedures written in the C, C++, .Net, Python or FORTRAN languages for efficiency?

Ans:Yes

5). What are the main features of R?

Ans:Following important features of R:

- R is a well-developed, simple and effective programming language which includes conditionals, loops, user defined recursive functions and input and output facilities.
- R has an effective data handling and storage facility.
- A collection of tools for analyzing data.
- Effective for developing methods that require interactive data analysis.
- R provides a suite of operators for calculations on arrays, lists, vectors and matrices.
- R provides a large, coherent and integrated collection of tools for data analysis.
- R provides graphical facilities for data analysis and display either directly at the computer or printing at the papers.
- It includes objects, such as regression models, time series, and geo-spatia coordinates.

6). What do you understand by R-objects?

Ans:While doing programming in any programming language, you need to use various variables to store various information. Variables are nothing but reserved memory locations to store values. This means that when you create a variable you reserve some space in memory.In contrast to other Programming languages like C and java in R the variables are not declared as some data type. The variables are assigned with R-Objects and the data type of the R-object becomes the data type of the variable.

7). What are the frequently used R-objects?

Ans:

- Vectors
- Lists
- Matrices
- Arrays
- Factors
- Data Frames

8). Can you tell me some of the function that R provides?

Ans:The function that R provides are

- Mean
- Median
- Distribution
- Covariance
- Regression
- Non-linear
- Mixed Effects
- GLM
- GAM. etc.

9). What is R Base package?

Ans:This is the package which is loaded by default when R environment is set. It provides the basic functionalities like input/output, arithmetic calculations etc. in the R environment.

10). What do you understand by Data Frame?

Ans:Data frame is a table or a two-dimensional array-like structure in which each column contains values of one variable and each row contains one set of values from each column.

11). Name some file format supported by R?

Ans:R can read and write into various file formats like csv, excel, xml

12). List out some most popular file format used in R programming language?

Ans:

- .RDA files: These are saved R objects that are used to attaching and loading files. They use the .rda or .RData extension. Files with the .RData and the .rda extension are the same.
- .R files: These files are created inside the R editor by the dump function. They include R commands. Some R files may also have the .q extension.
- .TXT files: These are text files that are used to store datasets. R uses theread.table() function and thewrite.table function. R uses theread.table() for data input and reading from text files. It then automatically creates a data frame with it. Thewrite.table() function on the other hand is used to create the text files.
- .CSV files: CSV or comma seprated values files are common data files typesinputted with theread.csv() function.

13). What is csv file, and how you can create it?

Ans:The csv file is a text file in which the values in the columns are separated by a comma. You can create this file using windows notepad by copying and pasting this data. Save the file as input.csv using the save As All files(*.*) option in notepad.

Example

```
id,Firstname,LastName
1,Vikas,Ahlawat
2,Amit,Arya
```

14). Which function is used to read csv file? can you give some example?

Ans: read.csv() function to read a CSV file available in your current working directory.

```
data <- read.csv("input.csv")
print(data)
```

15). What is the use of Scan() function?

Ans: The scan() function is used to read various types of data or data objects, for example data vactors. You can customize the command to read specific data. The command waits for input from the user and then return the value entered at the prompt.

16). What is the difference between an Array and a matrix?

Ans: A matrix is always two dimensional as it has only rows and columns. But an array can be of any number of dimensions and each dimension is a matrix. For example a 3x3x2 array represents 2 matrices each of dimension 3x3.

17). What is the output of runif(5)?

Ans: It generates 5 random numbers between 0 and 1

18). What is the use of apply() in R?

Ans: It is used to apply the same function to each of the elements in an Array. For example finding the mean of the rows in every row.

19). What is the latest version of R?

Ans: When this question written, R latest version is 3.2.2

20). How you will create an object in R?

Ans: Objects can be created in R by reading in data from a file or directly from the internet, by creation within a program, or through direct creation within R, as shown below:

Example:

```
x <-10
```

This will create an object called as x which has a value equal to 10

21). How you will create a function for divide two numbers, give example?

```
Ans: divider <- function(x,y){
result <-x/
print(result)
}
```

22). Can you give example of an array?

Ans: f<-c(1,2,3,4,5,6)

23). How you will make comment in R?, Example

Ans: A comment is easy to apply in R, via the hash symbol(#)

Example:

```
x<-10 #This is a simple object example
```

24). What command is used for stop/close R?

Ans: "q()"

25). Name some text editors for R?

Ans:

- Tinn-R : This is an easy to use GUI text editor for R programming in Windows.
- RKward : An easy to R text editor that works with GNU/Linux, Windows, and Mac OS X environments. It is an extensible IDE/GUI for R.
- JGR: This is like Rstudio with similar GUI that integrates with the R command line console.

26). What is One-factor ANOVA?

Ans: The one-factor analysis of variance, or ANOVA, is used to compare the means of several different groups on a single, quantitative outcome variable.

Example

```
# CREATE SIMULATION DATA
# Step 1: Each group in a separate variable.
x1 <- rnorm(30, mean = 40, sd = 8) # Group 1, mean = 40
x2 <- rnorm(30, mean = 41, sd = 8) # Group 1, mean = 41
x3 <- rnorm(30, mean = 44, sd = 8) # Group 1, mean = 44
x4 <- rnorm(30, mean = 45, sd = 8) # Group 1, mean = 45
```

27). What's the difference between "=" and "<-" in R?

Ans: The operators <- and = assign into the environment in which they are evaluated. The operator <- can be used anywhere, whereas the operator = is only allowed at the top level (e.g., in the complete expression typed at the command prompt) or as one of the subexpressions in a braced list of expressions.

28). What's the difference between seq_len() and seq_along() in the R programming language?

Ans:

seq_along(x): takes a vector for x, and it creates a sequence upto the count of elements in the vector.
seq_len(y): takes numeric for y, it creates a sequence upto the number y

29). What are the most popular R packages?

Ans:

- Ggplot2
- Plyr
- Reshape
- Lme4
- RODBC

30). What is R-Forge?

Ans: R-Forge offers a central platform for the development of R packages, R-related software and further projects.

31). How can I sort the rows of a data frame?

Ans: To sort the rows within a data frame, with respect to the values in one or more of the columns, simply use order() (e.g., DF[order(DF$a, DF[["b"]]),] to sort the data frame DF on columns named a and b).

32). What are the competitive benefits is using R for data analysis?

Ans: 6000 packages on CRAN spread across various domains of study.

Strong support on stackoverflow and good documentation reducing the learning curve for beginners

Availability of *almost all* machine learning packages

Incredible plotting system (ggplot2).

Give you exposure to the latest analytics techniques, including forecasting, socail network analytics and text mining.

Able to add-on to your existing analytics knowledge and methodology.

36). What does library() do?

Ans: Loading a package in order to make its functions/processes/dependencies available to the user. Just type

```
?library
```

and you will see the help for library ().

37). How to read selective lines in R programming language?

Ans: How do I read selective lines in R from an external file. For example: if the first column is a date, how do I read only lines between two given dates without reading the whole file into memory.

Ans:

Here's a workable, but non-robust solution -- others might have better ones:

This will store everything in a character variable called input_file:

```
input_file <- scan(file="x", skip = y, n = z, what = "raw", sep = "\n")
```

Replace x with file path

Replace y with the count of rows up to (but not including) the row that you want as the start date.

Replace z with the count of of rows up to (and including) the row you want as the end-date.

The option sep = "\n" specifies that each item is on a new row.

The option what = "raw" specifies that you're inputting a raw text file.

38). What's the difference between Hadoop and R Programming?

Ans: Hadoop is an open source framework user for distributed data storage and very huge data processing whereas R is a programming language used to form data science patterns and predictions.

39). How can I draw N beta distribution curves in one plot?

Ans: I was able to do this In R using the add property :

```
curve(dbeta(x,642,101),col="green");
curve(dbeta(x,1286,130),add=TRUE,col="blue");
curve(dbeta(x,2058,634),add=TRUE,col="orange");
curve(dbeta(x,2131,651),add=TRUE,col="brown");
```

33). Mostly all thing which are done by R, can be done by other language like Java then why R?

Ans: R programming for data science can by use to acive following thing

1. Loading data from file or from database
2. Data exploration like summary,scatter plots , box plots etc
3. Processing data like fixing missing data.
4. Segregating data into Training and testing set
5. Creating a model based and predicting.
6. Validating your results.
7. Data visualization.

All the above can be done by mainstream programming languages like Java and C++ , but will be cumbersome but with R all the above can be achieved in a fraction of seconds as all these functions are built in within R.

34). Why is R programming used instead of MS excel?

Ans: There are a number of reasons:

- Excel can't deal with data sets that are too large to fit in the memory of a single machine, and there's no way to extend it to do so.
- Excel's computation model is somewhat difficult to extend. It's possible to use VBA to write your own functions, but the language is extremely simple, so it's going to take a lot longer to write anything complicated.
- On a somewhat related note, we don't have good debugging tools for spreadsheets. If you're getting the wrong answer, it's very hard to find out why. R has much better tools for this.
- Excel can't handle data that lives in more than two dimensions. R has no problems with this.
- Excel only provides the absolute bare minimum of statistical and text processing functions. It doesn't even support regular expressions out of the box.
- Furthermore, the implementations of Excel's statistical functions are somewhat suspect.
- Excel's charts are limited to the small number that it provides. I guess you can extend that too with VBA, but base R graphics and ggplot2 are much easier to use.

35). What are the most common methods used in R programming?

Ans: Personally, I use a lot of data frame manipulation routines like apply, subset and others. plyr is also my most commonly used module and it involves data frame manipulation too.

40). What is the difference between package and library in R?

Ans: A package is a standardized collection of material extending R, e.g. providing code, data, or documentation. A library is a place (directory) where R knows to find packages it can use (i.e., which were installed). R is told to use a package (to "load" it and add it to the search path) via calls to the function library. I.e., library() is employed to load a package from libraries containing packages.

41). What machines/OS does R run on?

Ans: R is being developed for the Unix-like, Windows and Mac families of operating systems. Support for Mac OS Classic ended with R 1.7.1.

42). What do you understand by CRAN?

Ans: The "Comprehensive R Archive Network" (CRAN) is a collection of sites which carry identical material, consisting of the R distribution(s), the contributed extensions, documentation for R, and binaries.

43). What mailing lists exist for R?

Ans: Thanks to Martin Maechler(Martin Maechler is a member of R-Core), there are four mailing lists devoted to R.

R-announce :

- A moderated list for major announcements about the development of R and the availability of new code.

R-packages :

- A moderated list for announcements on the availability of new or enhanced contributed packages.

R-help :

- The 'main' R mailing list, for discussion about problems and solutions using R, announcements (not covered by 'R-announce' and 'R-packages') about the development of R and the availability of new code.

R-devel :

- This list is for questions and discussion about code development in R.

44). What are the differences between R and S?

Ans: Some known differences are the following.

- In R, if x is a list, then x[i] <- NULL and x[[i]] <- NULL remove the specified elements from x. The first of these is incompatible with S, where it is a no-op. (Note that you can set elements to NULL using x[i] <- list(NULL).)
- In S, the functions named .First and .Last in the .Data directory can be used for customizing, as they are executed at the very beginning and end of a session, respectively.
- In R, T and F are just variables being set to TRUE and FALSE, respectively, but are not reserved words as in S and hence can be overwritten by the user. (This helps e.g. when you have factors with levels "T" or "F".) Hence, when writing code you should always use TRUE and FALSE.
- In R, dyn.load() can only load shared objects, as created for example by R CMD SHLIB.
- In R, attach() currently only works for lists and data frames, but not for directories. (In fact, attach() also works for R data files created with save(), which is analogous to attaching directories in S.) Also, you cannot attach at position 1.
- Categories do not exist in R, and never will as they are deprecated now in S. Use factors instead.
- In R, For() loops are not necessary and hence not supported.
- In R, assign() uses the argument envir= rather than where= as in S.
- The random number generators are different, and the seeds have different length.
- R passes integer objects to C as int * rather than long * as in S.
- R has no single precision storage mode. However, as of version 0.65.1, there is a single precision interface to C/FORTRAN subroutines.