R Programming Assignment-9

1.Can you write an R programme that calculates the first 30 Fibonacci numbers?

ANS:

R Programming Code :

```
Fibonacci <- numeric(30)
Fibonacci[1] <- Fibonacci[2] <- 1
for (i in 3:30) Fibonacci[i] <- Fibonacci[i - 2] + Fibonacci[i - 1]
print("First 30 Fibonacci numbers:")
print(Fibonacci)
```

Output:

```
[1] "First 30 Fibonacci numbers:"
 [1]     1     1     2     3     5     8    13    21    34    55
[11]    89   144   233   377   610   987  1597  2584  4181  6765
[21] 10946 17711 28657 46368 75025 121393 196418 317811
514229 832040
```

2.What are the drawbacks of using R programming?

ANS:

Draw Backs of R Programming:

- Steep Learning Curve. As many have said, R makes easy things hard, and hard things easy. ...
- Some Packages may be of Poor Quality. CRAN houses more than 10,000 libraries and packages. ...
- Poor Memory Management. ...

- Slow Speed. ...
- Poor Security. ...
- No Dedicated Support Team. ...
- Flexible Syntax.

3.How do R and Python vary from one another?

ANS: R and Python are both open-source programming languages with a large community. New libraries or tools are added continuously to their respective catalog. R is mainly used for statistical analysis while Python provides a more general approach to data science.

R

Academics and statisticians have developed R over two decades. R has now one of the richest ecosystems to perform data analysis. There are around 12000 packages available in CRAN (open-source repository). It is possible to find a library for whatever the analysis you want to perform. The rich variety of library makes R the first choice for statistical analysis, especially for specialized analytical work.

The cutting-edge difference between R and the other statistical products is the output. R has fantastic tools to communicate the results. Rstudio comes with the library knitr. Xie Yihui wrote this package. He made reporting trivial and elegant. Communicating the findings with a presentation or a document is easy.

Python

Python can pretty much do the same tasks as R: data wrangling, engineering, feature selection web scrapping, app and so on. Python is

a tool to deploy and implement machine learning at a large-scale. Python codes are easier to maintain and more robust than R. Years ago; Python didn't have many data analysis and machine learning libraries. Recently, Python is catching up and provides cutting-edge API for machine learning or Artificial Intelligence. Most of the data science job can be done with five Python libraries: Numpy, Pandas, Scipy, Scikit-learn and Seaborn.

Python, on the other hand, makes replicability and accessibility easier than R. In fact, if you need to use the results of your analysis in an application or website, Python is the best choice.

KEY DIFFERENCES:

- R is mainly used for statistical analysis while Python provides a more general approach to data science
- The primary objective of R is Data analysis and Statistics whereas the primary objective of Python is Deployment and Production
- R users mainly consists of Scholars and R&D professionals while Python users are mostly Programmers and Developers
- R provides flexibility to use available libraries whereas Python provides flexibility to construct new models from scratch
- R is difficult to learn at the beginning while Python is Linear and smooth to learn
- R is integrated to Run locally while Python is well-integrated with apps
- Both R and Python can handle huge size of database
- R can be used on the R Studio IDE while Python can be used on Spyder and Ipython Notebook IDEs

4.What is the purpose of using a programming language?

ANS:

A programming language is a language used to write computer programs, which instruct a computer to perform some kind of computation, and/or organize the flow of control between external devices (such as a printer, a robot, or any peripheral).

Computer programming languages allow us to give instructions to a computer in a language the computer understands. Just as many human-based languages exist, there are an array of computer programming languages that programmers can use to communicate with a computer. The portion of the language that a computer can understand is called a “binary.” Translating programming language into binary is known as “compiling.” Each language, from C Language to Python, has its own distinct features, though many times there are commonalities between programming languages.

These languages allow computers to quickly and efficiently process large and complex swaths of information. For example, if a person is given a list of randomized numbers ranging from one to ten thousand and is asked to place them in ascending order, chances are that it will take a sizable amount of time and include some errors.

5.Is it possible to write pseudocode for a function?

ANS: Pseudocode has no defined standard, as it is simply a method of writing a human-readable representation of program code. Functions can be defined however you wish....

EXAMPLE:

def FunctionName() , as is the Python syntax for defining functions;

C-style <type> function name() ;

etc.

6.In R, how do we import data?

ANS: Importing Data in R:

Importing data in R programming means that we can read data from external files, write data to external files, and can access those files from outside the R environment. File formats like CSV, XML, xlsx, JSON, and web data can be imported into the R environment to read the data and perform data analysis, and also the data present in the R environment can be stored in external files in the same file formats.

7.Can you write a programme to find the maximum and minimum values of a vector?

ANS: R Programming Code :


```
x = c(10, 20, 30, 25, 9, 26)
print("Original Vectors:")
print(x)
print("Maximum value of the above Vector:")
print(max(x))
print("Minimum value of the above Vector:")
print(min(x))
```

Output:

```
[1] "Original Vectors:"
[1] 10 20 30 25  9 26
[1] "Maximum value of the above Vector:"
```

[1] 30

[1] "Minimum value of the above Vector:"

[1] 9

8.Can you write a program that prints the unique elements of a string?

ANS: R Programming Code :

```
str1 = "The quick brown fox jumps over the lazy dog."
print("Original vector(string)")
print(str1)
print("Unique elements of the said vector:")
print(unique(tolower(str1)))
nums = c(1, 2, 2, 3, 4, 4, 5, 6)
print("Original vector(number)")
print(nums)
print("Unique elements of the said vector:")
print(unique(nums))
```

OUTPUT:

[1] "Original vector(string)"

[1] "The quick brown fox jumps over the lazy dog."

[1] "Unique elements of the said vector:"

[1] "the quick brown fox jumps over the lazy dog."

[1] "Original vector(number)"

[1] 1 2 2 3 4 4 5 6

[1] "Unique elements of the said vector:"

[1] 1 2 3 4 5 6