R Introduction

Basic R Introductions and Operations

Jingyu Sun

March 29, 2025

Ocean University of China, High Performance Computing Club, Qingdao 266100

What is R?

Initially R language was originied from S language, which was developed at Bell Laboratories by John Chambers and colleagues.

R language is designed for data analysis and statistical computing, which is a free software environment for statistical computing and graphics. Now it is a part of GNU project (so that's why I like it).

R language is widely used among statisticians and data miners for developing statistical software and data analysis.

Other hot statistical languages: Python (we will discuss later), Julia (especially for high performance computing), Stata (designed for applcations), etc.

R Installation

Steps for install R and RStudio:

- Download R from CRAN (TUNA mirror): `https://mirrors.tuna.tsinghua.edu.cn/CRAN/bin/windows/base/R-4.4.3-win.exe`
- Install R
- Download RStudio (A Integrated Developer Environment for R) `https://posit.co/download/rstudio-desktop/` (Remember to select the version that suit your systemm)
- Install RStudio

Begin with R: read and write data

Basically everytime we use R is for data analysing, so we need to know how to read and write data.

Suppose we have a data file named `iris.csv`, we can use the following code to read the data into R:

```
iris <- read.csv("iris.csv")
```

Begin with R: read and write data

Now wondering if we have the iris data and we want to write it into a file, we can use the following code:

```
write.csv(iris, "iris2.csv")
```

Begin with R: read and write data

Now as you can see, read and write are probably easier than any other languages, which means that R is a good language for data analysis.

To learn this further, let's move on how to do basic operations in R.

Basic Operations: Structure of data

The most common scene we would face is we do not know how the data is strcutured. So we need to know how to see the structure of data.

- `head(iris)`: Show the first 6 rows of the data
- `tail(iris)`: Show the last 6 rows of the data
- `str(iris)`: Show the structure of the data
- `summary(iris)`: Show the summary of the data

Basic Operations: Access Data

Also, similar to other languages, R has some methods to access data in the data frame.

- `iris$SepalLengthCm`: Access the column named SepalLengthCm
- `iris[1, ]`: Access the first row of the data
- `iris[1:5, ]`: Access the first 5 rows of the data
- `iris[iris$SepalLengthCm > 6, ]`: Access the rows that SepalLengthCm is greater than 6

Basic Operations: Statistics of Data

While we already knows how to access the data, we can do some statistics on the data.

- `mean(iris$SepalLengthCm)`: Calculate the mean of SepalLengthCm
- `sd(iris$SepalLengthCm)`: Calculate the standard deviation of SepalLengthCm
- `cor(iris$SepalLengthCm, iris$SepalWidthCm)`: Calculate the correlation between SepalLengthCm and SepalWidthCm
- `table(iris$Species)`: Calculate the frequency of each species

Basic Operations: Matrices and Arraies

R also supports matrices and arraies, which are very useful in data analysis.

- `matrix(1:9, nrow=3, ncol=3)`: Create a 3x3 matrix
- `array(1:9, dim=c(3, 3, 3))`: Create a 3x3x3 array
- `t(matrix(1:9, nrow=3, ncol=3))`: Transpose the matrix
- `solve(matrix(1:4, nrow=2, ncol=2))`: Inverse the matrix

Basic Operations: Assign Values

The previous code are all about how to get the data, but how to assign values to the data?

- `iris$SepalLengthCm[1] <- 5.1`: Assign 5.1 to the first row of SepalLengthCm
- `iris[1, 1] <- 5.1`: Assign 5.1 to the first row of the first column
- `iris[iris$SepalLengthCm > 6, 1] <- 6`: Assign 6 to the rows that SepalLengthCm is greater than 6

Basic Operations: Assign Values

Another operator related to this is %<%, which is used to assign values to a variable.

This operator means that the value on the right side is assigned to the variable on the left side.

- `a %<% 5`: Assign 5 to a
- `a %<% b %<% 5`: Assign 5 to b and a

Work Directory

Every R session has a working directory, which is the default directory for file input and output.

We will simply use the configuration in R Studio to set the working directory.

Progamming Logic

It's not often use the programming logic in R, but it's still useful to know how to use it.

- `if (condition) { ... } else { ... }`: If the condition is true, execute the first block, otherwise execute the second block
- `for (i in 1:10) { ... }`: Execute the block 10 times
- `while (condition) { ... }`: Execute the block until the condition is false
- `break`: Break the loop

Applications of R: Machine Learning (Linear Regression)

The basic machine learning method is regression, which here we will discuss linear regression. Still, we will use iris data as an example.

- `lm <- lm(SepalLengthCm ~SepalWidthCm, data=iris)`: Fit a linear model to the data
- `predict(lm, newdata=iris)`: Predict
- `summary(lm)`: Show the summary
- `plot(lm)`: Plot
- `abline(lm)`
- `anova(lm)`: Show the ANOVA
- `confint(lm)`: Show the CI

Applications of R: Machine Learning (Nonlinear Regression)

Also we could conduct some nonlinear regression.

- `lm <- lm(SepalLengthCm ~ SepalWidthCm^2, data=iris)`: Fit a linear model to the data
- `predict(lm, newdata=iris)`: Predict
- `summary(lm)`: Show the summary
- `plot(lm)`: Plot
- `abline(lm)`
- `anova(lm)`: Show the ANOVA
- `confint(lm)`: Show the CI

Applications of R: Machine Learning (Random Forest)

Other methods like random forest are also available in R.

- `library(randomForest)`: Load the random forest library
- `rf <- randomForest(SepalLengthCm ~SepalWidthCm + PetalLengthCm + PetalWidthCm, data=iris)`: Fit a random forest model to the data
- `predict(rf, newdata=iris)`: Predict
- `importance(rf)`: Variable importance
- `plot(rf)`: Plot

Summary of R Language

Now we have discussed some basic operations in R, and also some applications of R. Later we would like to introduce some further applications by R.

The reason that why R does not take much time is that R is a very simple language (for learners), and it is initially designed for mathematicians, not for computing.