

AI and Biotechnology/Bioinformatics

R Crash Course (2025)

Session 2: Basics of R Programming language (Variables, Comments & keywords)

Welcome to another session of R Programming Crash Course!

In the previous class we learned about how to install and setup R environment, now that now everything is ready. It's time to start coding in R.




If you missed it  [click here](#) to catch up

In today's session we'll understand some nuts and bolts (syntax) of R programming language and will look at how to write some basic codes in R.

Syntax




Syntax are the predefined structure and rules of any programming language. It shows us how to right codes right way. If we don't follow these rules, codes won't work at all.

So knowing syntax well makes coding easier:

-  Codes become easier to read and understand for you and others
-  Sticking to rules helps to you to avoid mistakes in your code
-  It's easier to debug and fix the issues when you know how the code is supposed to work

Syntax in R

Just like any other programming language, R also have three basic building blocks:

-  **Variables**
-  **Comments**
-  **Keywords**

- ❖ **Variables** stores values in R just like a labeled test holds sample in a lab

Storing a patient's glucose level in a variable

glucose_level <- 98.6

Explanation:

- *glucose_level is the variable name*
- *<- is an assignment operator that stores the value 98.6 into variable*
- *98.6 is now saved in glucose_level*

To check what's inside a variable, type its name and run it

glucose_level

or use print () function

print(glucose_level)

Important Notes:

Overwriting Variables: Assigning a new value replaces the old one

glucose_level <- 45.6

old value (98.6) is replaced

Case Sensitivity: variable names are case sensitive

Glucose_level <- 98.6

different from glucose_level even have same values

Run these lines of code in RStudio to check the output

- ❖ **Comment** help you explain your code

- This is only for your own understanding; R doesn't consider it as a code
- It starts with # and anything after that is ignored by R when the code runs
- If your code begin with # R treats the entire line as a comment and skips it completely

Correct way to write comment

age <- 56 # assigning a value to a variable

Incorrect way to write comment

assigning a value to a variable age <- 56

“Always add comment before a code or at the end for clarity”

- ❖ **Keywords** are the reserved words that perform specific function in R. You cannot use the as variable name. **Common keywords are:**

<i>If</i>	<i>else</i>
<i>for</i>	<i>while</i>
<i>function</i>	<i>return</i>
<i>repeat</i>	<i>break</i>
<i>TRUE</i>	<i>FALSE</i>
<i>NULL</i>	<i>NA</i>

In R, you can view these keywords by using either `help(reserved)` or `?reserved`

➤ If and else in R

If, else are the control flow keywords in R. They help you make decision in your code.

These are easiest decision-making statements in R

If checks a condition/statement

- ✚ If the condition is TRUE, the code inside the if block executes.
- ✚ If the condition is FALSE, the code inside the else blocks runs

Using only if

```
gene_expression <- 30

if (gene_expression > 25) {
  print ("gene is highly expressed")
}
```

The if statement checks if expression_level is greater than 25.

- ✚ If true, it prints: "Gene is highly expressed".
- ✚ If it's FALSE, nothing happens.

Using if, else

```
if (gene_expression > 25) {
  print("gene is highly expressed")
} else {
  print(gene expression is low)
}
```

This checks both condition.

- ✚ The if statement checks if expression_level is greater than 25.
- ✚ If true, it prints: "Gene is highly expressed".
- ✚ Otherwise, the else block executes. It prints "Gene expression is low".

Using if...else if...else

```
bmi <- 17

if (bmi < 18.5) {
  print("underweight")
} else if (bmi >= 18.5 & bmi <= 24.9) {
  print("normal weight")
} else {
  print("overweight")
}
```

This checks multiple conditions

- ✚ If bmi is less than 18.5 prints (underweight)
- ✚ If it's between 18.5 and 24.9 (normal)
- ✚ If it's more than 24.9 (overweight)

Try It Yourself:

1. Check Cholesterol level (using if)

```
cholesterol <- 230
```

Write an If statement to check cholesterol level is greater than 240, if it is print **“High Cholesterol”**

2. Blood Pressure Status (using if...else)

```
Systolic_bp <- 130
```

Write an if...else statement to check if blood pressure is normal.

- ✚ If it's less than 120, print: **“Blood Pressure is normal”**
- ✚ Otherwise print: **“Blood Pressure is high”**

3. White Blood Cell Count (using if...else, if...else condition)

```
wbc_count <- 3000
```

Write if...else, if...else statement to check WBC count.

- ✚ If wbc_count is lower than 4000, print: **“Leukopenia-low WBC count”**
- ✚ If wbc_count in between 4000 & 11000, print: **“WBC count is normal”**
- ✚ If wbc_count is greater than 11000 print: **“Leukocytosis-High WBC count”**

That's all about for today, in this session, we covered:

- Importance of syntax in R programming.
- How to use variables, comments and keywords?
- Writing if, else statement.

Want to try it yourself? Get the R script from this GitHub Link  [click here](#)