If you have the chance to work with an experienced programmer, you may be amazed by how fast she can write code. In this article, I share some tips and shortcuts you can use in RStudio and R Markdown to speed up the writing of your code.

Run code

You most probably already know this shortcut but I still mention it for new R users. From your script you can run a chunk of code with:

```
command + Enter on Mac
Ctrl + Enter on Windows
```

Insert a comment in R and R Markdown

To insert a comment:

```
command + Shift + C on Mac
Ctrl + Shift + C on Windows
```

This shortcut can be used both for:

- R code when you want to comment your code. It will add a `#` at the beginning of the line
- for text in R Markdown. It will add around the text

Note that if you want to comment more than one line, select all the lines you want to comment then use the shortcut. If you want to uncomment a comment, apply the same shortcut.

Knit a R Markdown document

You can knit R Markdown documents by using this shortcut:

```
command + Shift + K on Mac
Ctrl + Shift + K on Windows
```

Code snippets

Code snippets is usually a few characters long and is used as a shortcut to insert a common piece of code. You simply type a few characters then press `Tab` and it will complete your code with a larger code. `Tab` is then used again to navigate through the code where customization is required. For instance, if you type `fun` then press `Tab`, it will auto-complete the code with the required code to create a function:

```
name <- function(variables) {

}
```

Pressing `Tab` again will jump through the placeholders for you to edit it. So you can first edit the name of the function, then the variables and finally the code inside the function (try by yourself!).

There are many code snippets by default in RStudio. Here are the code snippets I use most often:

- `lib` to call `library()`

```
library(package)
```

- `mat` to create a matrix

```
matrix(data, nrow = rows, ncol = cols)
```

- `if`, `el`, and `ei` to create conditional expressions such as `if() {}`, `else {}` and `else if () {}`

```
if (condition) {

}

else {

}

else if (condition) {

}
```

- `fun` to create a function

```
name <- function(variables) {

}
```

- `for` to create for loops

```
for (variable in vector) {

}
```

- `ts` to insert a comment with the current date and time (useful if you have very long code and share it with others so they see when it has been edited)

```
# Tue Jan 21 20:20:14 2020 ------------------------------
```

- `shinyapp` everytime I create a new shiny apps

```
library(shiny)

ui <- fluidPage(

)

server <- function(input, output, session) {

}

shinyApp(ui, server)
```

You can see all default code snippets and add yours by clicking on Tools > Global Options… > Code (left sidebar) > Edit Snippets…

Ordered list in R Markdown

In R Markdown, when creating an ordered list such as this one:

1. Item 1
2. Item 2
3. Item 3

Instead of bothering with the numbers and typing

```
1. Item 1
2. Item 2
3. Item 3
```

you can simply type

```
1. Item 1
1. Item 2
1. Item 3
```

for the exact same result (try it yourself or check the code of this article!). This way you do not need to bother which number is next when creating a new item.

To go feven further, any numeric will actually render the same result as long as the first item is the number you want to start from. For example, you could type:

```
1. Item 1
7. Item 2
3. Item 3
```

which renders

1. Item 1
2. Item 2
3. Item 3

However, I suggest always using the number you want to start from for all items because if you move one item at the top, the list will start with this new number. For instance, if we move `7. Item 2` from the previous list at the top, the list becomes:

```
7. Item 2
1. Item 1
3. Item 3
```

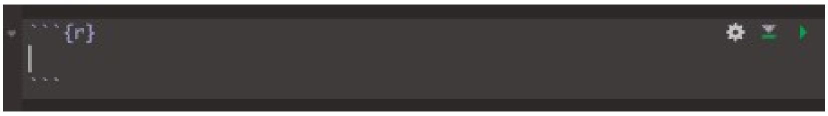
which incorrectly renders

7. Item 2
8. Item 1
9. Item 3

New code chunk in R Markdown

When editing R Markdown documents, you will need to insert a new R code chunk many times. The following shortcuts will make your life easier:

```
command + option + I on Mac (or command + alt + I depending on your keyboard)
Ctrl + ALT + I on Windows
```



~~~~
```{r}

```
~~~~

New R code chunk in R Markdown

Reformat code

A clear and readable code is always easier and faster to read (and look more professional when sharing it to collaborators). To automatically apply the most common coding guidelines such as whitespaces, indents, etc., use:

```
cmd + Shift + A on Mac
Ctrl + Shift + A on Windows
```

So for example the following code which does not respect the guidelines (and which is not easy to read):

```
1+1
  for(i in 1:10){if(!i%%2){next}
print(i)
```

```
 }
```

becomes much more neat and readable:

```
1 + 1
for (i in 1:10) {
  if (!i %% 2) {
    next
  }
  print(i)
}
```

Others

Similar to many other programs, you can also use:

- `command + Shift + N` on Mac and `Ctrl + Shift + N` on Windows to open a new R Script
- `command + S` on Mac and `Ctrl + S` on Windows to save your current script or R Markdown document

Thanks for reading. I hope you find these tips and tricks useful. If you are using others, feel free to share them in the comment section.…