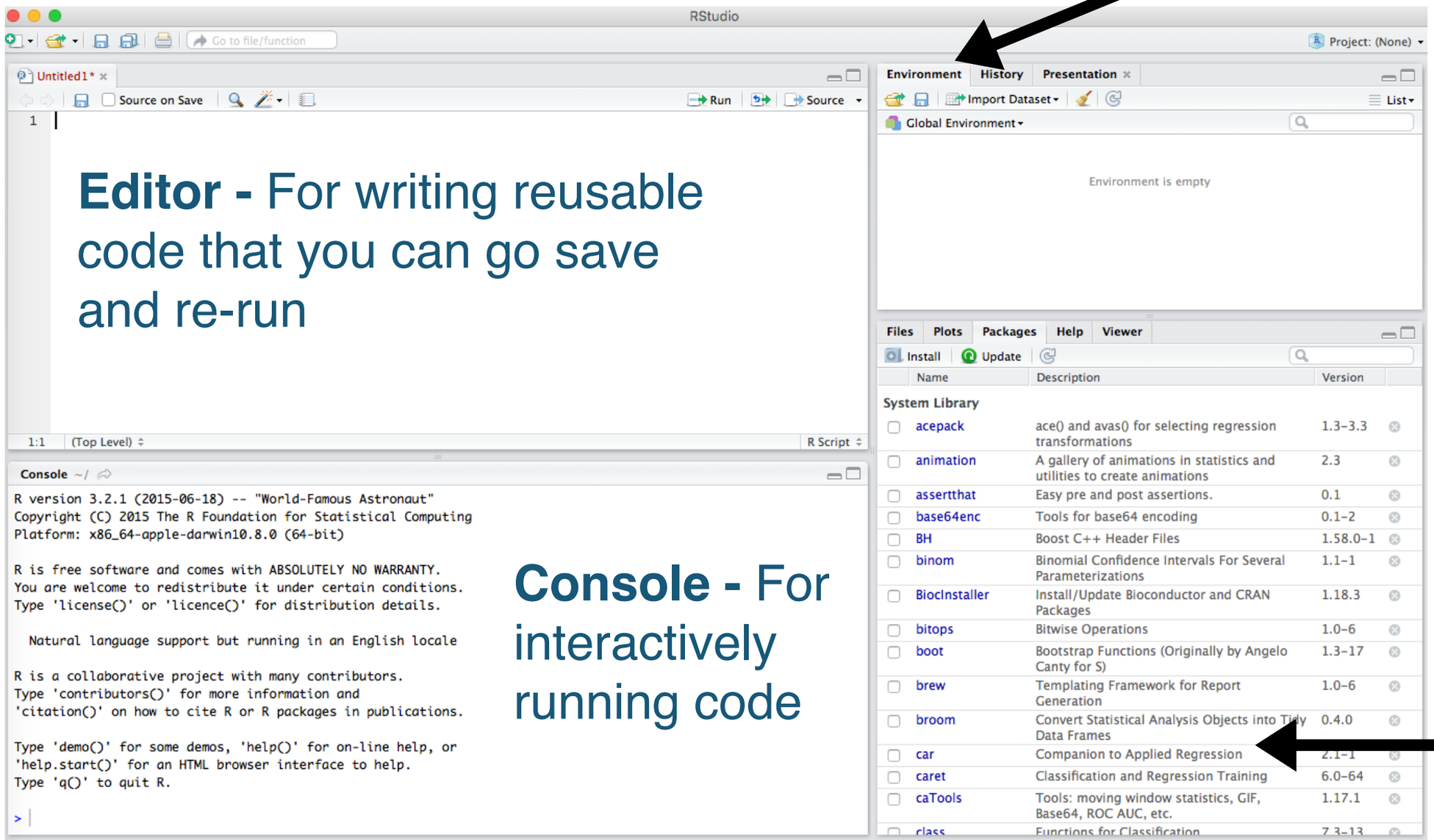
The RStudio Interface

also called an IDE -
Integrated Development Environment

Environment - Lists variables and data sets loaded and saved during your R session



Editor - For writing reusable code that you can go save and re-run

Console - For interactively running code

Help, Package Manager, and plotting -
A useful interface for additional common tasks

Running Code

Performing operations

Type in code and hit enter

```
Console ~/
> 4 + 5
[1] 9
> 5 * 200
[1] 1000
```

R performs the tasks and produces output

Storing variables

```
Console ~/
> x = 4 + 5
>
```

There is no output! That is because R is **storing** this variable.

Check the Environment window

You can **print the output of stored variables** by typing their name into the console and hitting Enter

```
> x
[1] 9
>
```

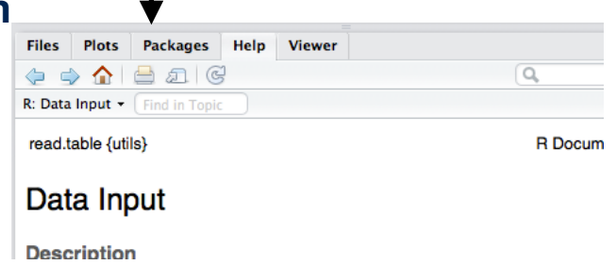
Not all commands produce output in the console: e.g., making plots, getting help, loading libraries

Getting help

? or help('')

Opens help in the Help tab

```
> ?mean
> ?read.delim
> help('read.delim')
>
```



The Console

The place for trial and error

Tips and tricks

The console is stuck on +

This happens if R doesn't know if the line has ended (e.g., missing parenthesis)

```
> x = ((5 + 5/4^2)
+
```

Hit the 'esc' key

```
> x = ((5 + 5/4^2)
+

>
```

Use the up and down arrow keys on your keyboard instead of re-entering lines of code.

You can use the up and down arrows to scroll through all of the commands you've entered into the console!

Clear console: control - L

Variables are not removed (check the Environment window). Up and down arrows will still scroll through the commands you've entered.

You will get Errors!

It's okay!! Errors help us discover bugs in code.

Read the messages. Check for typos in your code. When in doubt, copy and paste into your favorite search engine. Some error messages are very useful (some aren't).

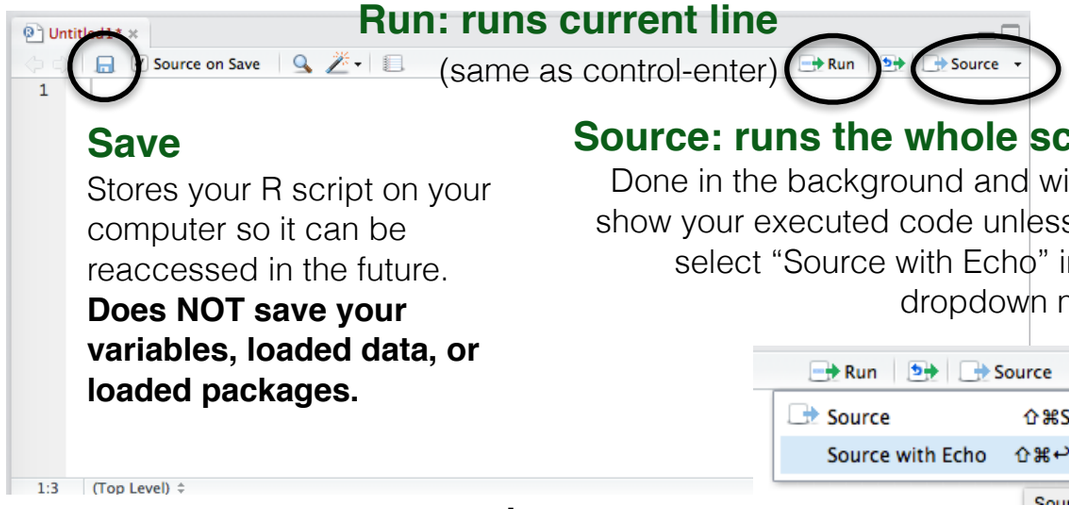
```
> my_variable = 5
> my_varible * 5
Error: object 'my_varible' not found
>
```

Oops! forgot an 'a'

The Editor

Writing, running, and saving re-useable R scripts

Useful icons on Rstudio editor interface



Run: runs current line
(same as control-enter)

Save
Stores your R script on your computer so it can be reaccessed in the future.
Does NOT save your variables, loaded data, or loaded packages.

Source: runs the whole script
Done in the background and will not show your executed code unless you select "Source with Echo" in the dropdown menu

Save your code.

Use the icon above, the file->save menu, or control-s (cmd-s on a mac). If your code is part of a project, save it with your project.

Name of saved script

my_r_script.R

```
1  my_variable = 5
2  my_variable * 5
3
4  my_vector = c(1, 2, 3, 4, 5)
5
6  my_data_frame = data.frame(
7    variable_x = c(1, 2, 3, 4, 5),
8    variable_y = c(5, 4, 3, 2, 1)
9  )
10 #This is comment, R doesn't interpret this
11
12 # this code loads the dplyr package and uses selects
13 # to select variable_x column from my_data_frame
14 library(dplyr)
15 select(my_data_frame, variable_x)
```

Running code from the editor

Single lines of code

Code can be run by placing the cursor at the end of the line and hitting control-enter (cmd-enter on Mac). Code lines can also be highlighted with the mouse and run.

```
1  my_variable = 5
```

OR

```
1  my_variable = 5
```

Control-Enter
(Mac: cmd-enter)

Check the Console window

```
> my_variable = 5
>
```

Note: there is no output, this is storing a variable. Check the Environment window

```
2  my_variable * 5
```

This will produce a result.

```
> my_variable * 5
[1] 25
>
```

Multiple lines of code

Often you will run multiple lines of code, highlight and run it all at once. You can also run each line separately: R displays "+" when it expects additional code.

```
7  my_data_frame = data.frame(
8    variable_x = c(1, 2, 3, 4, 5),
9    variable_y = c(5, 4, 3, 2, 1)
10 )
```

```
> my_data_frame = data.frame(
+   variable_x = c(1, 2, 3, 4, 5),
+   variable_y = c(5, 4, 3, 2, 1)
+ )
>
```

Note: there will be no output, this is storing a data frame. Check the Environment window

If your console is stuck on +, **hit Esc**

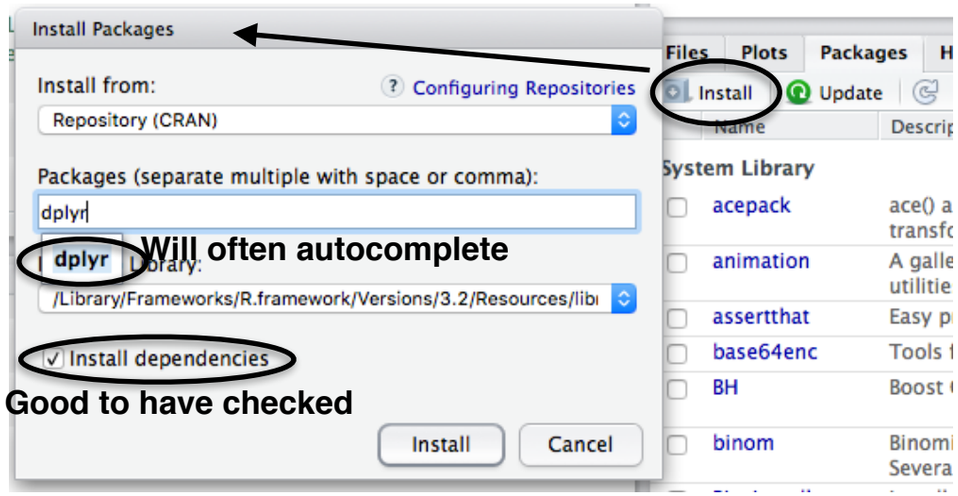
Comments

Comments are used to document your code. Comments starts with # and do not carry across lines (the text becomes green in the editor). You can 'run' comments and they will display in the console but R will not try to interpret (execute) the code. Commenting and documentation is critical in coding and data analysis.

Installing and loading packages

Installing packages using RStudio package manager

Code can be run by placing the cursor at the end of the line and hitting control-enter (cmd-enter on Mac). Code lines can also be highlighted with the mouse and run.



Installing packages using code

The primary function to use is install.packages(). You will need to know the name of the package and it is a good idea to set **dependencies = TRUE**. The name of the packages is entered in quotes. RStudio's package manager is just running this code for you.

```
install.packages("dplyr", dependencies = TRUE)
```

Loading packages

This has to be done each time you open a new R session. Run the library() function with the package name.

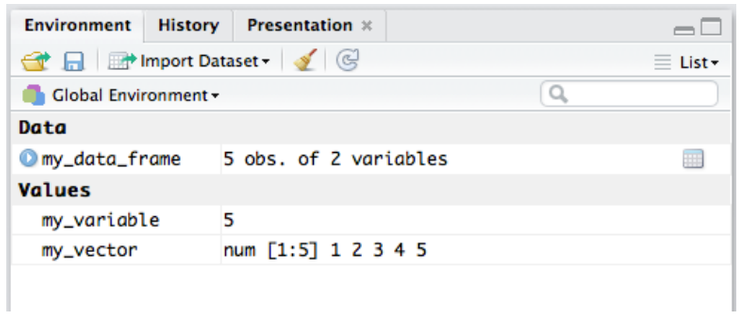
```
> library(dplyr)
>
```

Note: there is no output.

Installing and loading packages may give Warnings. Warnings are not the same as Errors. Warnings should be read but you can generally continue without addressing them..

Environment window

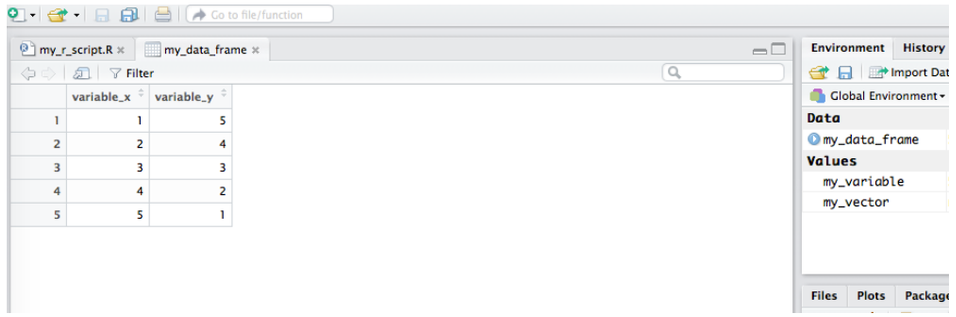
Tells you what data and variables you've loaded or created



Clicking on an item in Data, opens a spreadsheet for viewing in the Editor window.

Runs this in the console

```
> View(my_data_frame)
```



Tab completion

(Intermediate tip)

Hitting the 'tab' key while typing (in the Editor or Console) will often fill in variable names, function parameters, and directories. Use of tab completion is very useful once you become more comfortable with R and RStudio