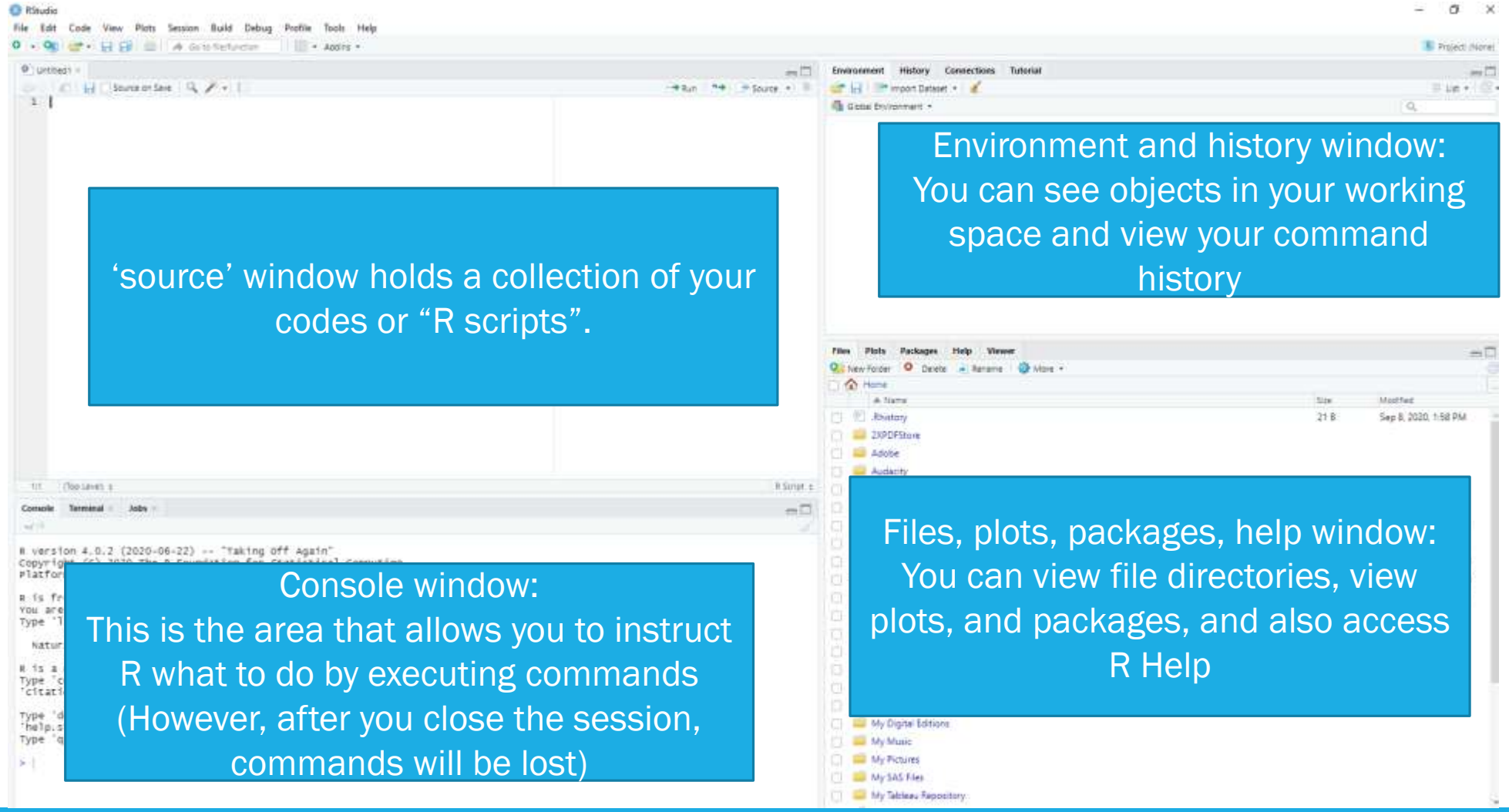
Basic Skills in R: part I

Housekeeping Issues

- Do not wait till the last minute before you do your assignment.
- Make the best use of weekends
- While in class, I would give about 1-2 minutes for you to practice what is being taught so that you could practice faster on your own
- Practice before trying the quizzes

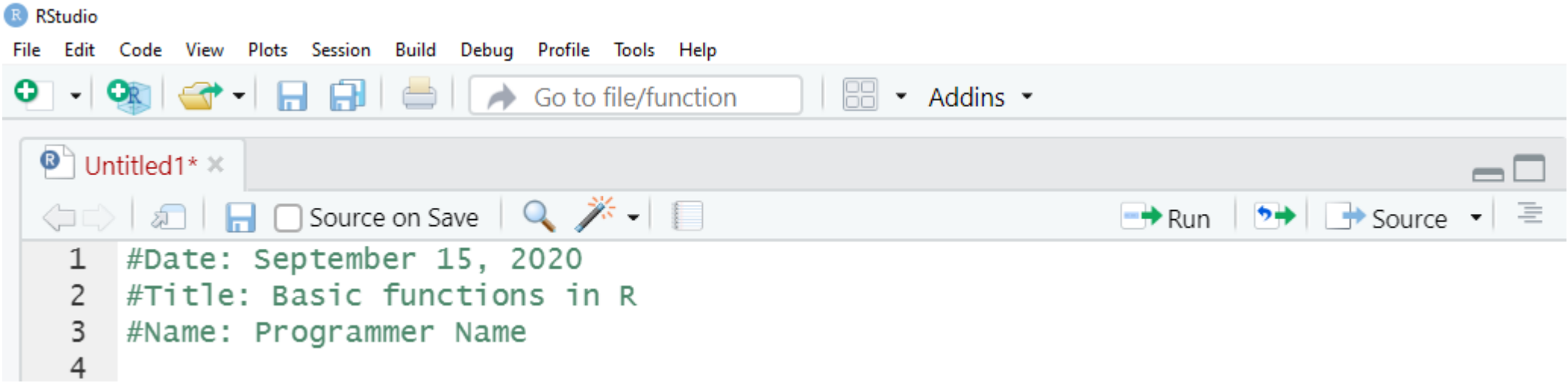
Open RStudio and you should see 4 RStudio Windows



Comments

- Recall when we used to add headings to our assignments in grade school in order to remember? Well, it is good practice to add comments to your R scripts as well.

- You can add comments by including the hashtag symbol before your comments. This will turn the font of the comments green. Example shown below:

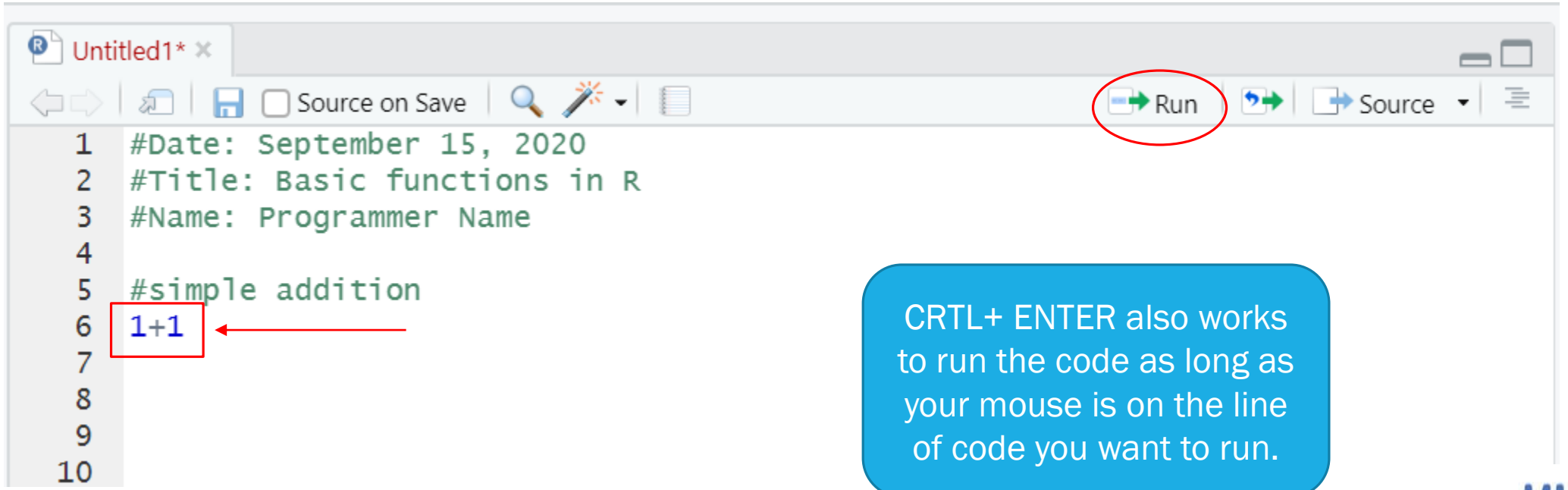


Basic Numerical Operations in R

ADDITION
SUBTRACTION
MULTIPLICATION
DIVISION

Addition

- Let's simply add 1+1 in the R environment. Type in the following code, highlight the code and select 'Run'
- If you don't select a certain section of your code before clicking 'run', all codes present in the RScript file will be executed. You can either highlight the section of code or place your mouse on the same line as the code (i.e., place your mouse on line 6) then click run.



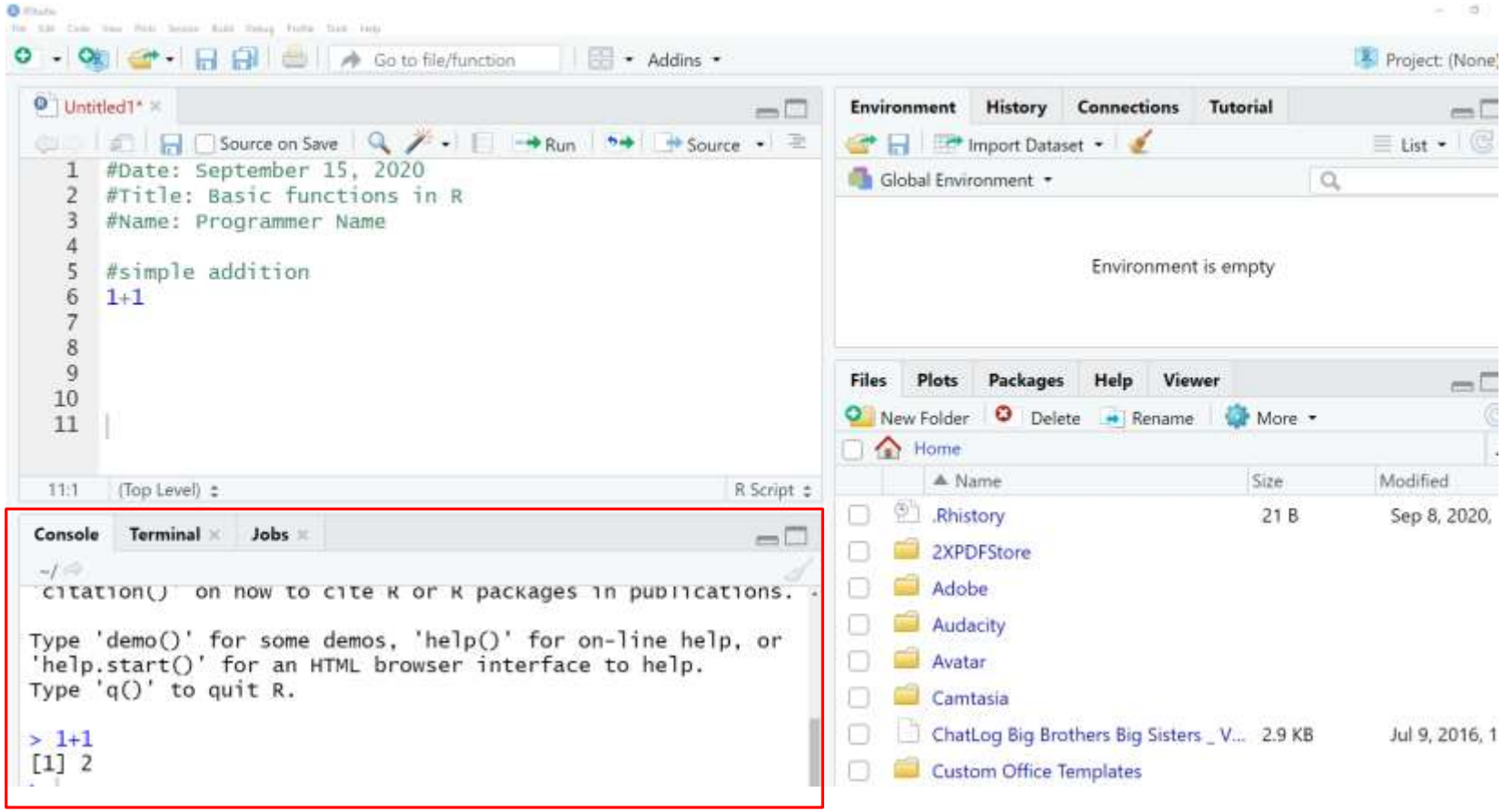
CRTL+ ENTER also works to run the code as long as your mouse is on the line of code you want to run.

Addition Results:

- Your results will be displayed in the Console Window

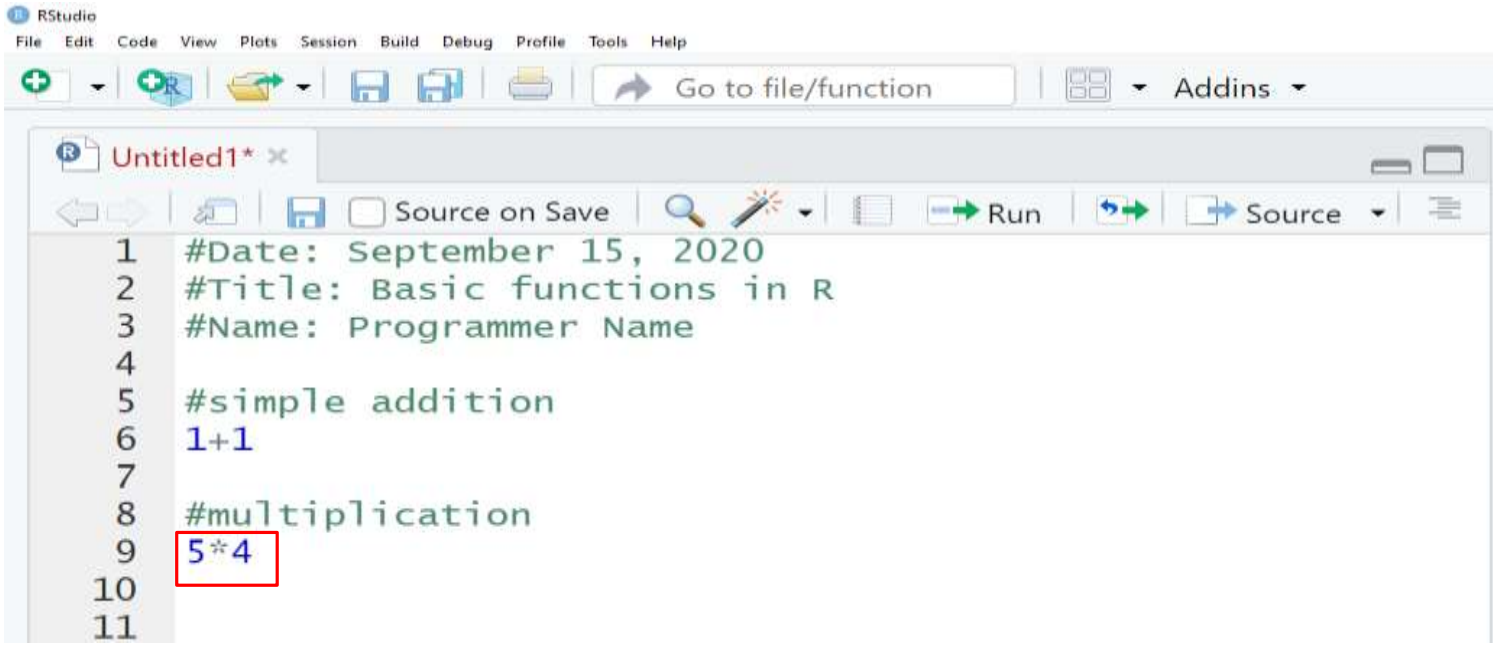
Console window is boxed in red

Result shown here as '2'



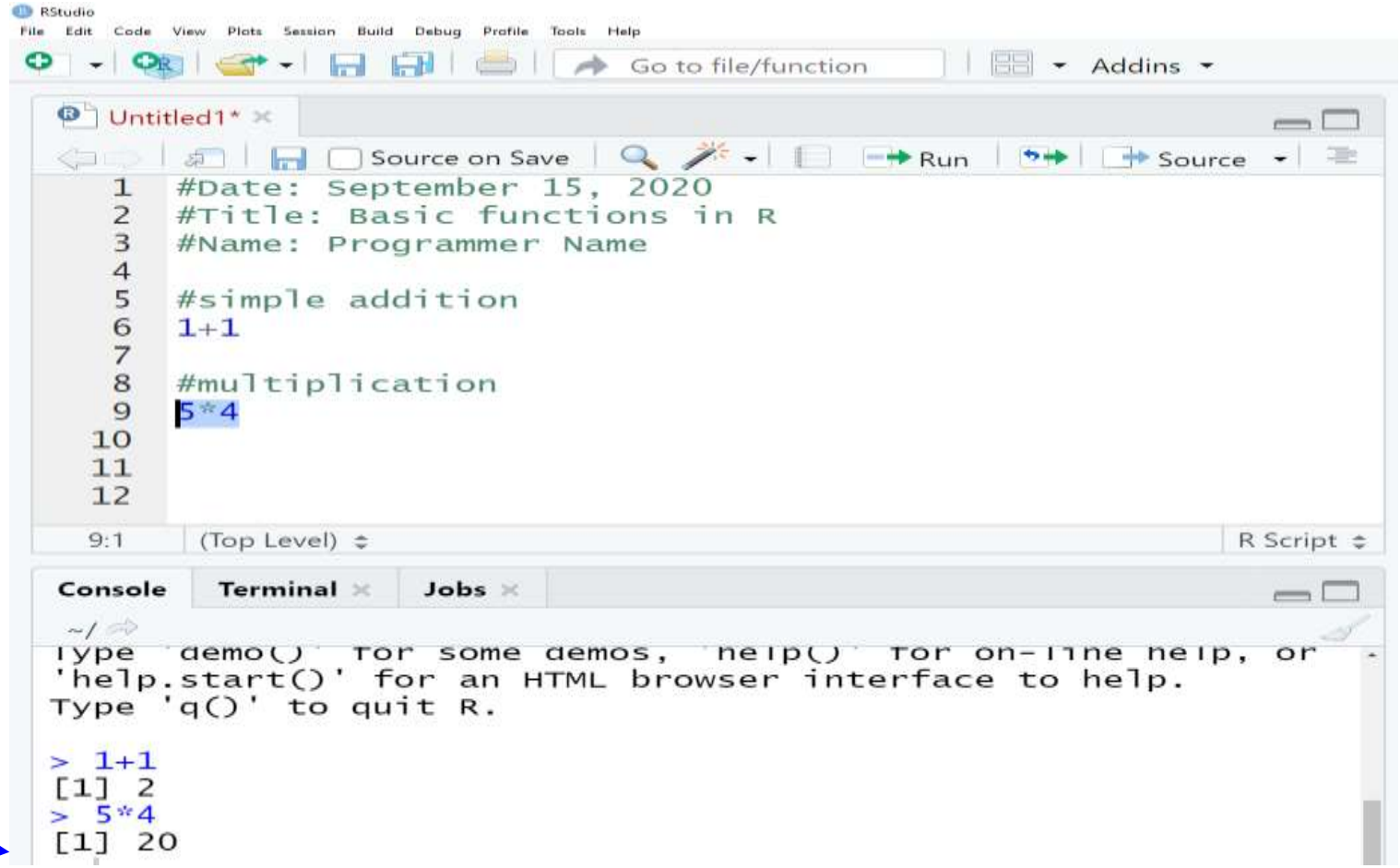
Multiplication

- Simply multiply two numbers by adding the asterisk (*) sign in between them, highlight the code, and click ‘run’



Multiplication results

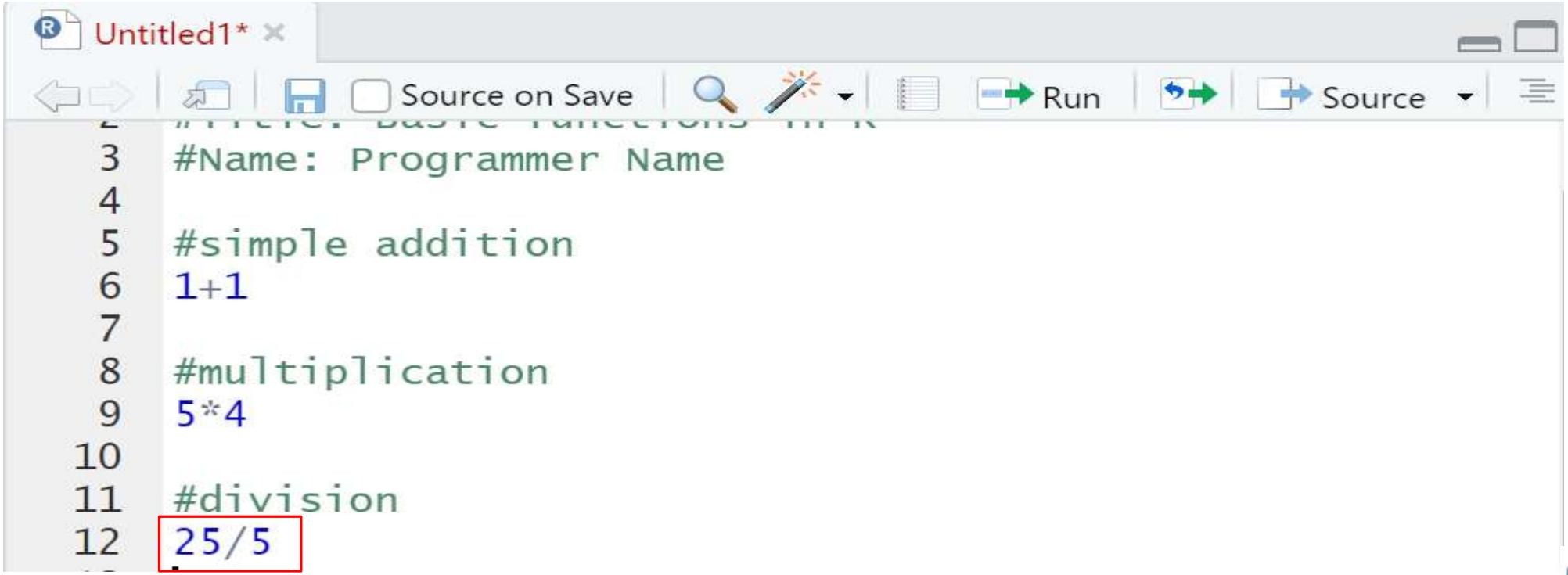
- Your results will be displayed in the Console Window



Result shown here as '20'

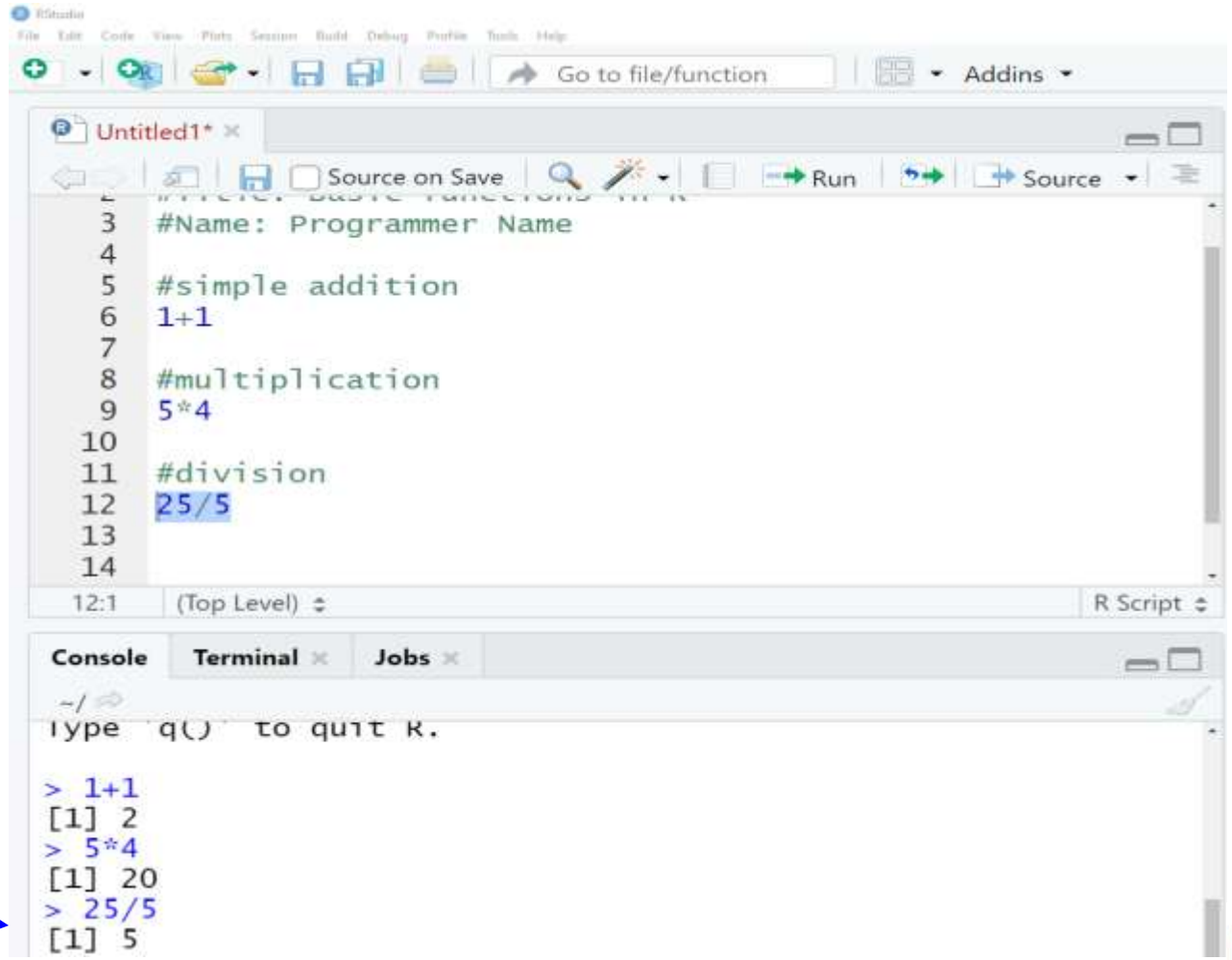
Division

- You can divide two numbers by including a forward slash (/) in between them



Division Results

- Your results will be displayed in the Console Window



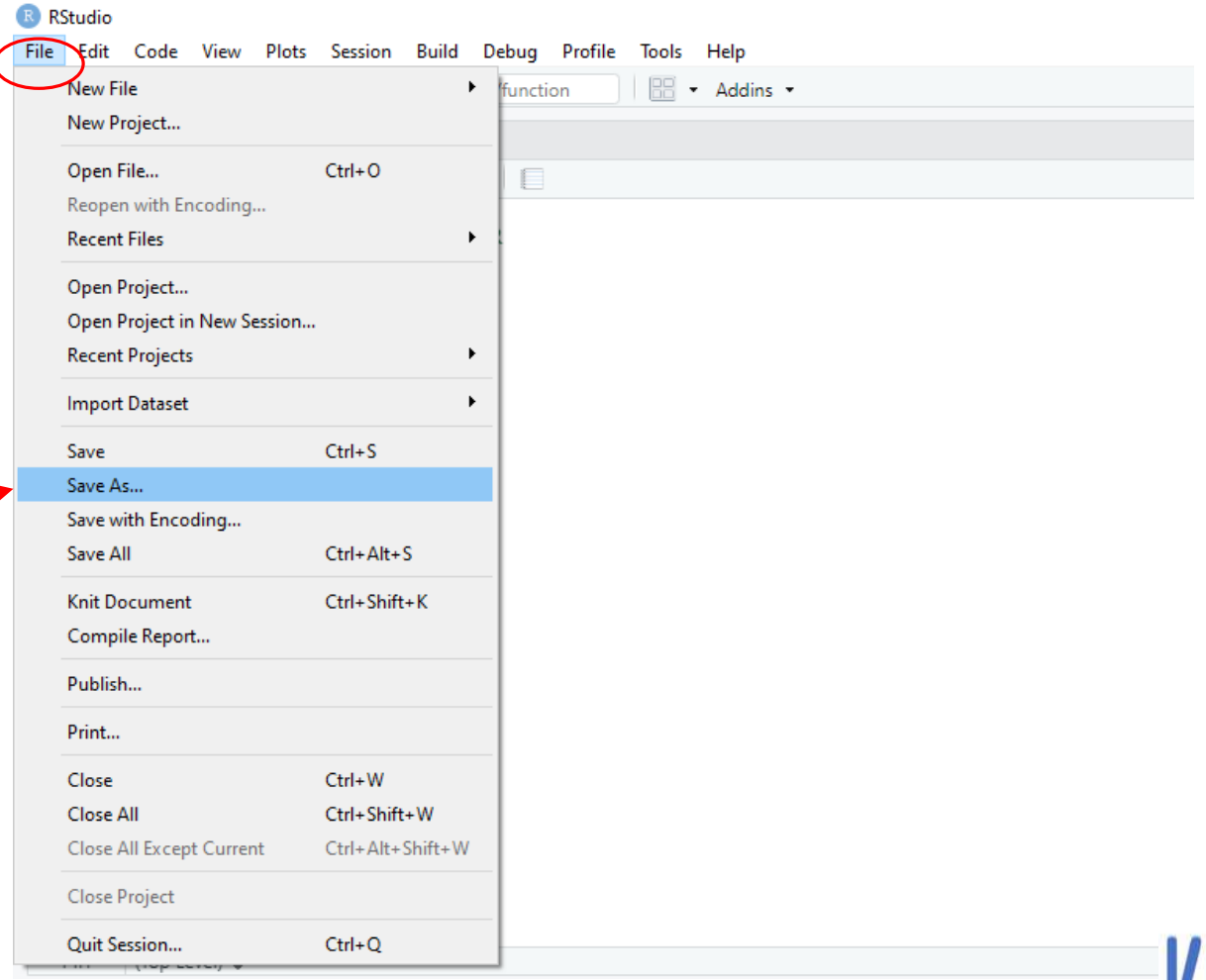
Result shown here as '5'

Saving a RScript

Please save your RScript

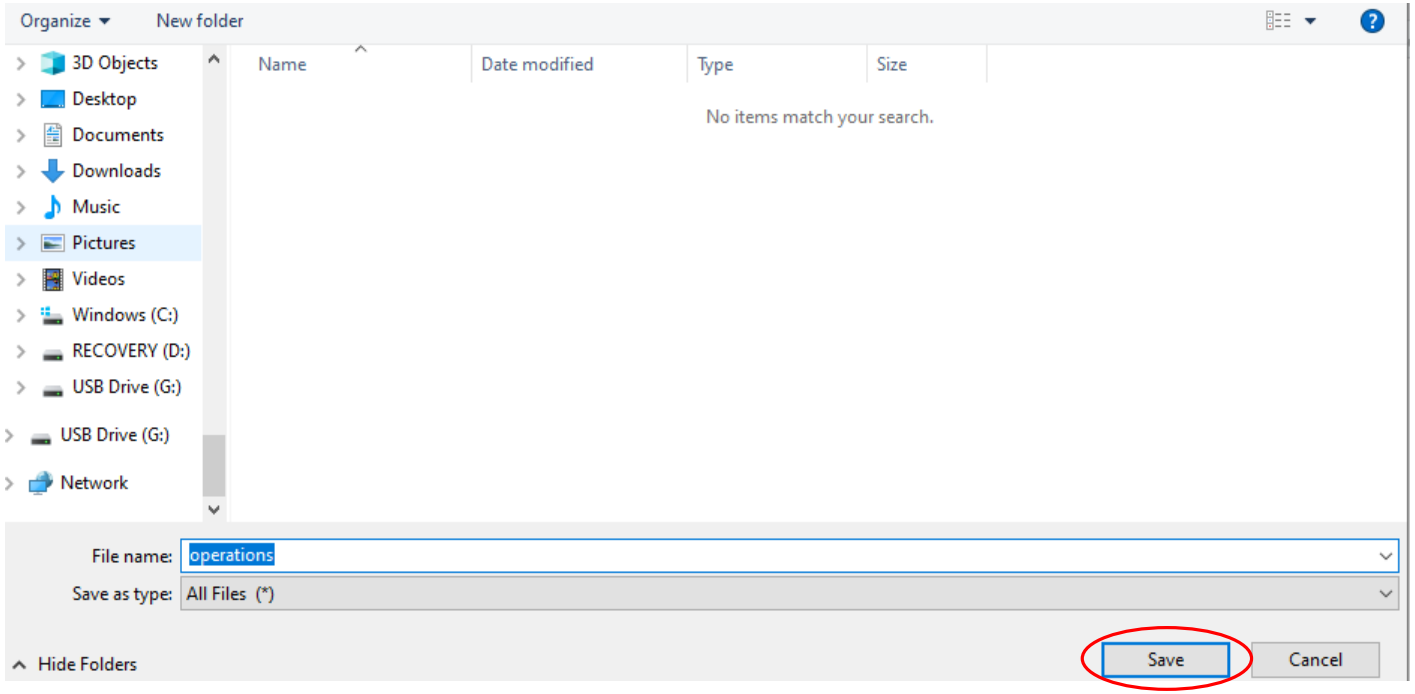
- It is good practice to save your R Script so that you can continue where you left off just in case you need to leave the session or if you want to come back to the code at a later point.

- Click ‘file’ → ‘save as’



Please save your R Script

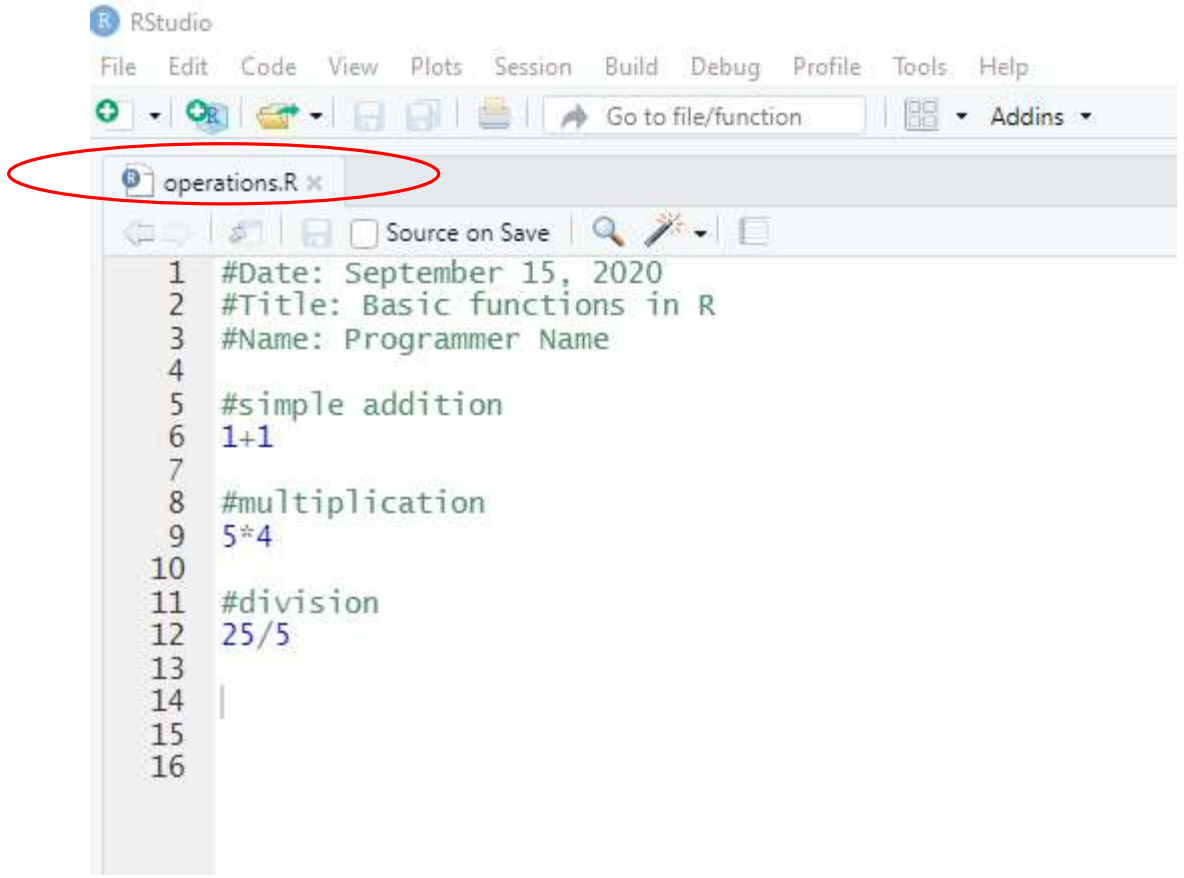
- After clicking ‘file’ → ‘save as’
- You will be taken to a window that will allow you select where to save the RScript and also how to name the RScript
- We will call this RScript ‘operations’.
- Then, select ‘save’.



Please save your R Script

- Notice how the heading is now called ‘operations.R’

- RScripts have a ‘.R’ extension
  - Similar to how word documents end in ‘.doc’ or ‘.docx’



Data objects in R

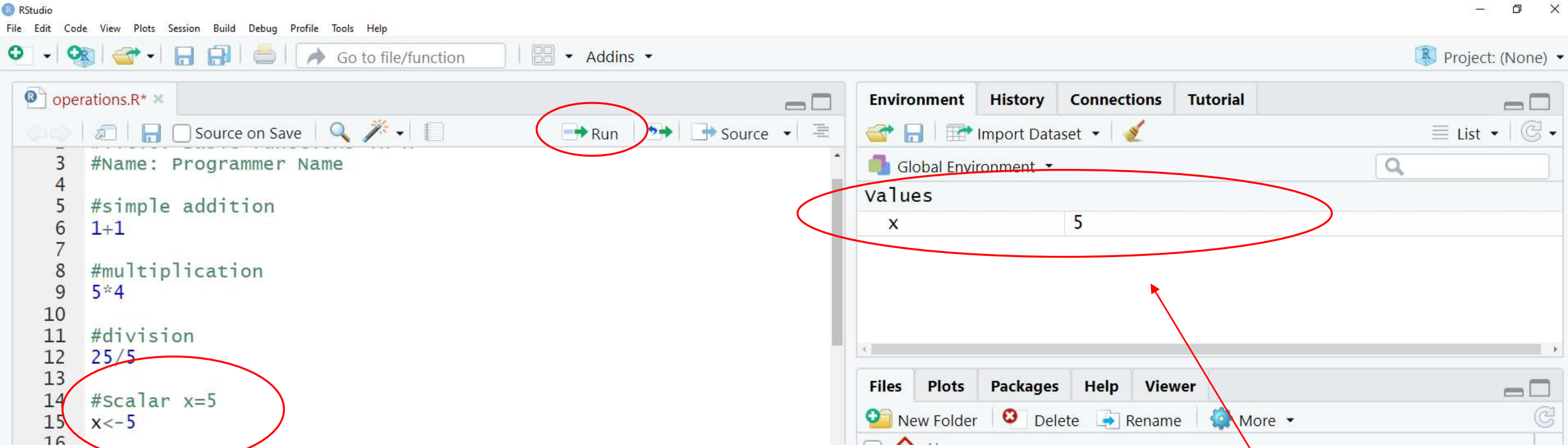
Scalar

- Scalars are the most simple data type. They are an object with one character or numeric value.

- For example: x=5, w='Sunday'
  - x and w are the scalars

- You use "<-" symbols to create data objects that we will see in subsequent slides and lectures. This symbol ("<-") is known as the "assignment operator".

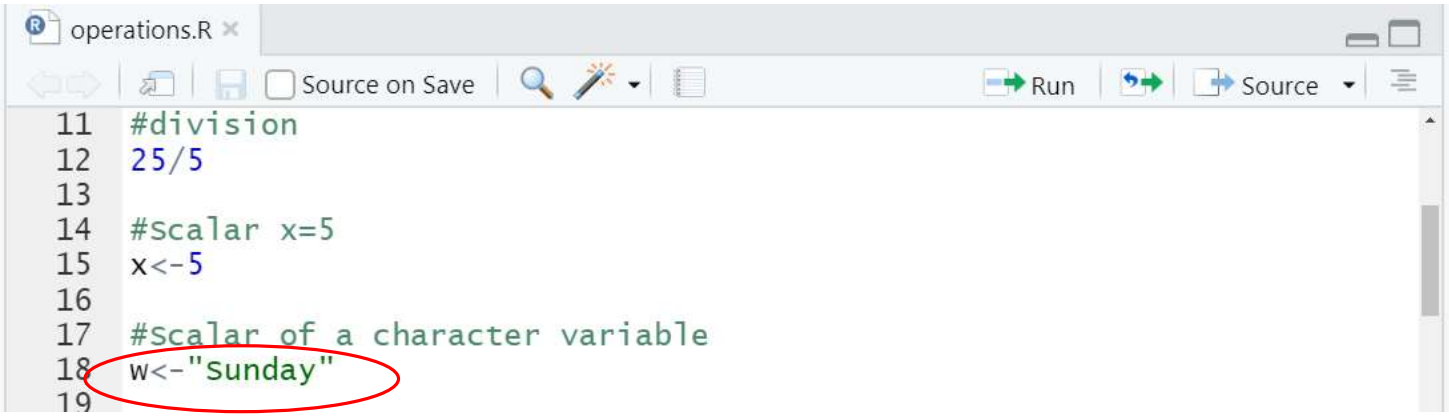
- The operationalization of scalars in R is shown on the next slide

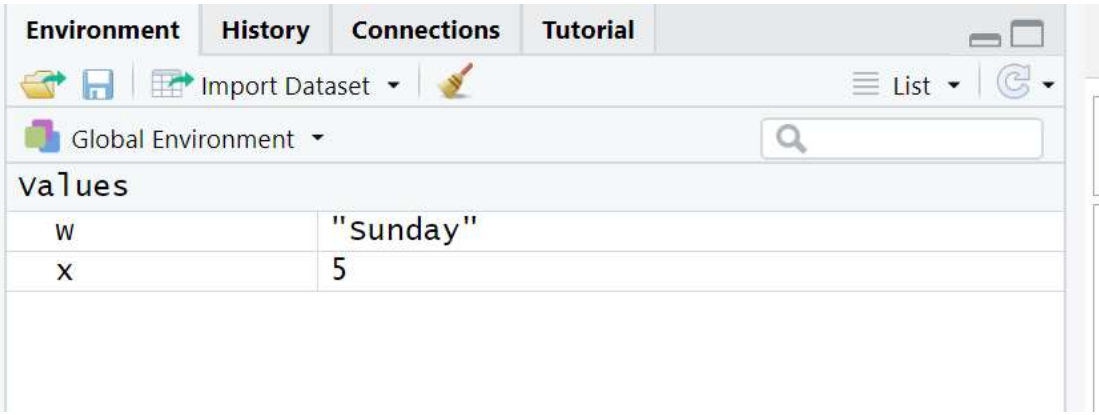


Type the code x<-5, highlight the code, then select run

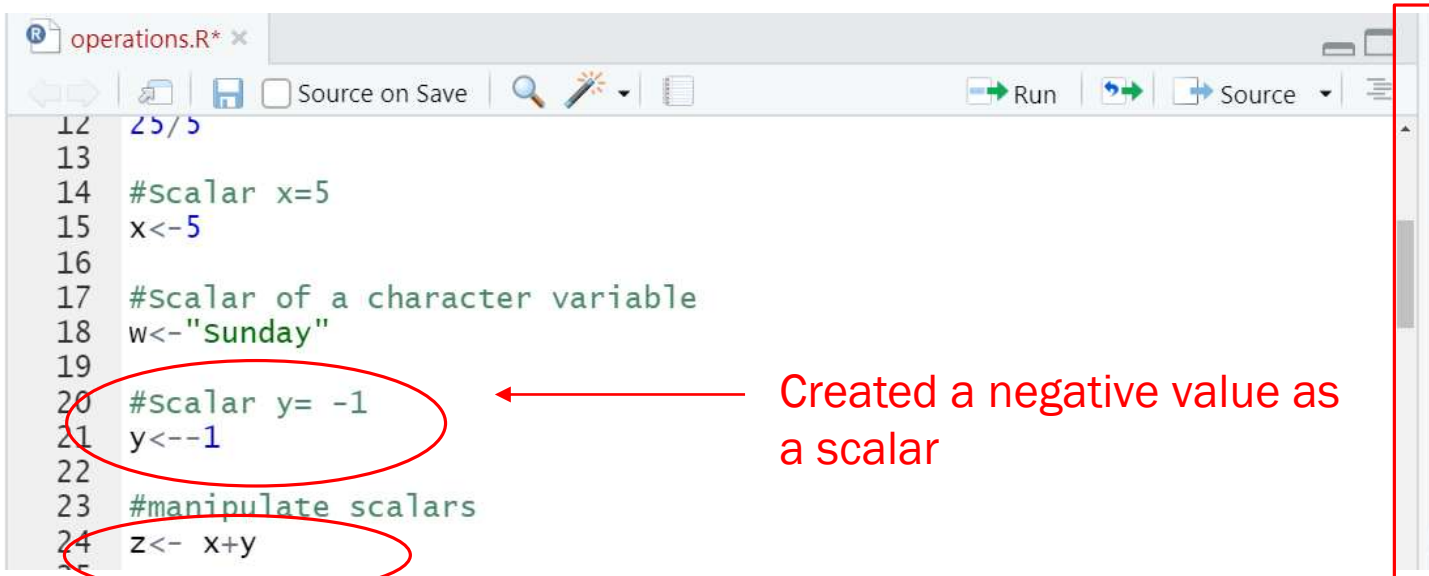
*Note: if it is a single line of code, you could also place your cursor on the line and then select run

You will see the newly assigned value in the environment and history window

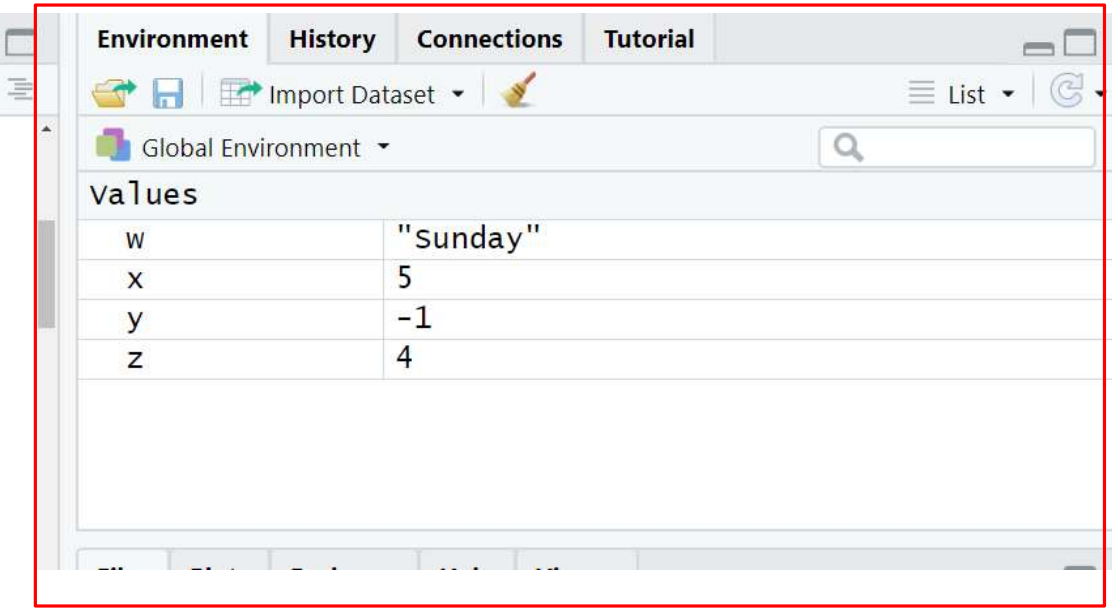




Although you could use double quotations or single quotations to create a character vector, be sure to be consistent and preferably stick to the use of double quotations.



Created a negative value as a scalar



Summing two scalars to create a new scalar called "z"

Vector: Synonym of Variable

- **Most common data objects used in R**
- **Variables are generally stored as vectors**
- **Sequence of data elements of the same type:**
  - Numerical vectors
    - 1, 3, 5, 6
  - Character vectors
    - “normal” “prehypertensive” “hypertensive”
  - And more…
- **Create a vector using “c()” function (shown on the next slide)**

operations.R*

Source on Save | Run | Source

```
13
14  #Scalar x=5
15  x<-5
16
17  #Scalar y= -1
18  y<--1
19
20  #manipulate scalars
21  z<- x+y
22
23  #Vector
24  bpstatus <- c("normal", "prehypertensive", "hypertensive")
```

Environment | History | Connections | Tutorial

Import Dataset | List

Global Environment

Values

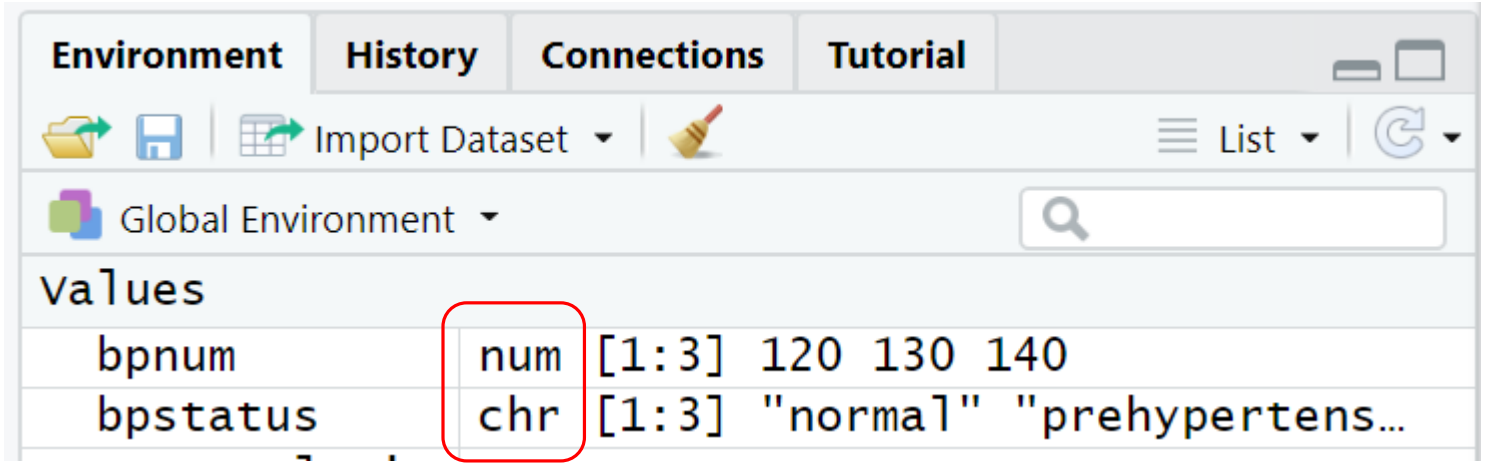
bpstatus	chr [1:3] "normal" "prehypertensive" "hypertensive"

Notice that in your environment and history window, you will see your newly created character vector designated as ‘chr’

operations.R*

Source on Save | Run | Source

```
13
14  #Scalar x=5
15  x<-5
16
17  #Scalar y= -1
18  y<--1
19
20  #manipulate scalars
21  z<- x+y
22
23  #Vector
24  bpstatus <- c("normal", "prehypertensive", "hypertensive")
25  bpnum <- c(120,130,140)
```



'num' stands for numerical vector and 'chr' stands for character vector

Factor

- Factors are vectors that are categorized.
- This has a special utility for example, in modeling or when constructing frequency tables
- The assigned labels (values) could be character, numeric, or Boolean (“AND”, “OR” and “NOT”)
- Creating a factor in R options (as shown on the next slide)

operations.R* ×

Source on Save | Run | Source

```
20  #manipulate scalars
21  z<- x+y
22
23  #Vector
24  bpstatus <- c("normal", "prehypertensive", "hypertensive")
25  bpnum <- c(120,130,140)
26
27  #Converting vector to a factor
28  bpstatus_f<-factor(bpstatus)
29  bpstatus_f
```

After you run the section you will see this in the console window

Previously, we created this vector. We will now convert it into a factor called "bpstatus_f" by instructing using 'factor' option.

```
> bpstatus_f
[1] normal          prehypertensive hypertensive
Levels: hypertensive normal prehypertensive
```

After you print the newly created factor variable, it appears with the assigned 'levels' or categories

Lists

- Simply put, lists are a collection of items in a particular order
  - Can accommodate heterogenous elements
- Lists can be useful for organizing information.
- Unlike vectors, they can contain different modes/types of data.
  - List of names: [Jill, Bill, Sally]
  - List of numbers: [1, 5, 96]
  - List made up of names and numbers: [“Jill”, 5, 6.8, “B”]
- How to think of a list: ‘a general container’
  - Movie: each movie has a cast, crew, budget, script, etc.
  - List: can also contain multiple data frames (ie., datasets)

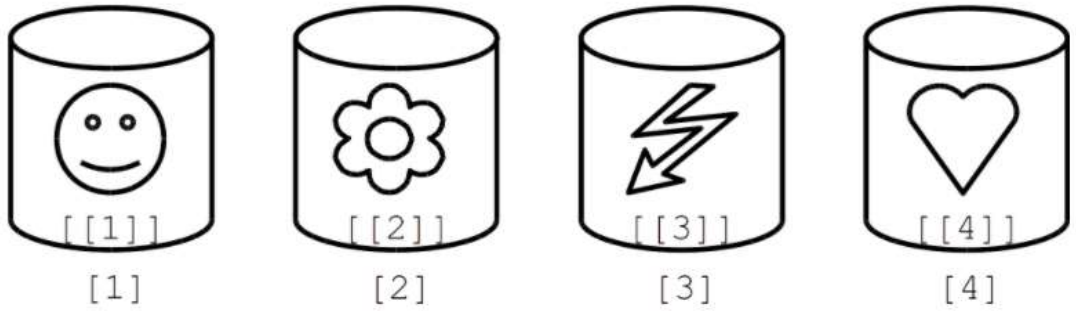


Figure 3.1: Schematic representation of a list of length four.

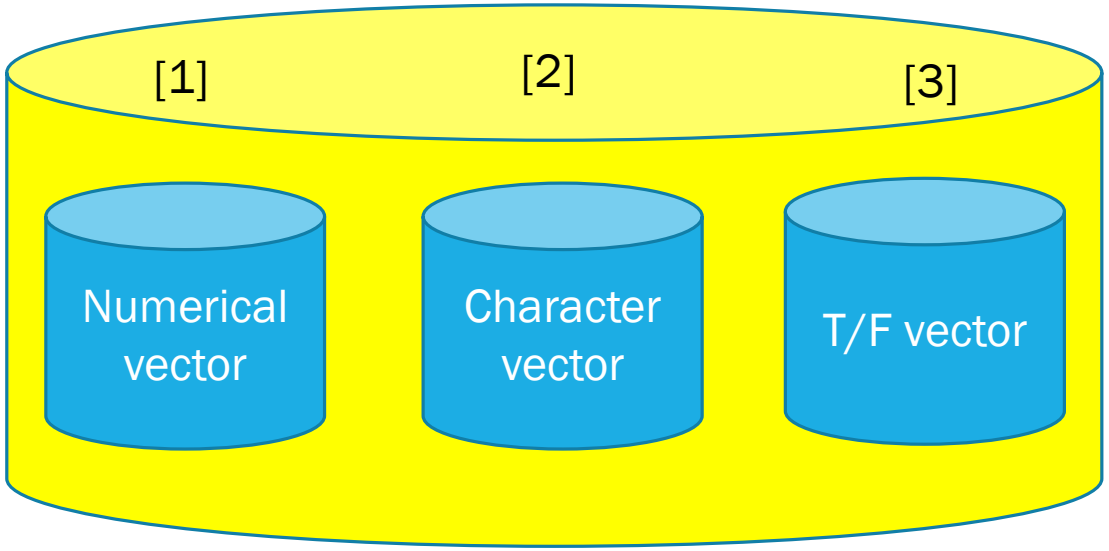
https://bookdown.org/medepi/phds/working-with-lists-and-data-frames.html

Lists in R

Here, we are creating a list to store 3 vectors with information we are interested in.

1) Numerical vector
2) Character vector
3) Vector with true false information

We first create the vectors and then combine them into a list.



Yellow container represents list

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins

3-data mgmt.R | operations.R*

Source on Save

```
52 #Lists
53 numvector <- c(5,6,7)
54 charvector <- c( "Jill", "John", "Meg")
55 tfvector <- c("True", "False", "False", "True", "True")
56 newlist <- list(numvector, charvector, tfvector)
57 newlist
58
59
60
61
62
63
64
65
66
67
```

68:1 (Top Level)

Console Terminal Jobs

~/

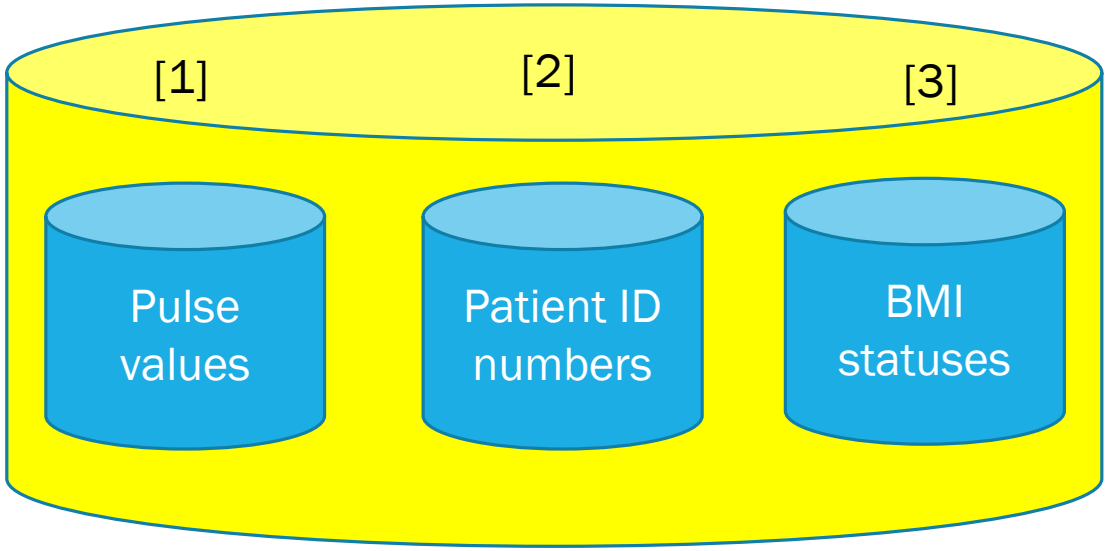
```
> numvector <- c(5,6,7)
> charvector <- c( "Jill", "John", "Meg")
> tfvector <- c("True", "False", "False", "True", "True")
> newlist <- list(numvector, charvector, tfvector)
> newlist
[[1]]
[1] 5 6 7

[[2]]
[1] "Jill" "John" "Meg"

[[3]]
[1] "True"  "False" "False" "True"  "True"
```

Lists in R

Here, we are creating a list to store 3 vectors with information we are interested in:

1) Pulse values
2) Patient ID numbers
3) BMI status (underweight, normal weight, overweight, obese)

We first create the vectors and then combine them into a list.



Yellow container represents list

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins

operations.R

Source on Save

```
62
63  #Another list
64  pulse <- c(64,50,100,70,97,59)
65  patid <- c(57964, 14597,30146)
66  BMIstat <- c("underweight", "normal","overweight","obese")
67
68  cliniclist <- list(pulse,patid,BMIstat)
69  cliniclist
70
71
72
73
74
75
```

75:1 (Top Level)

Console Terminal Jobs

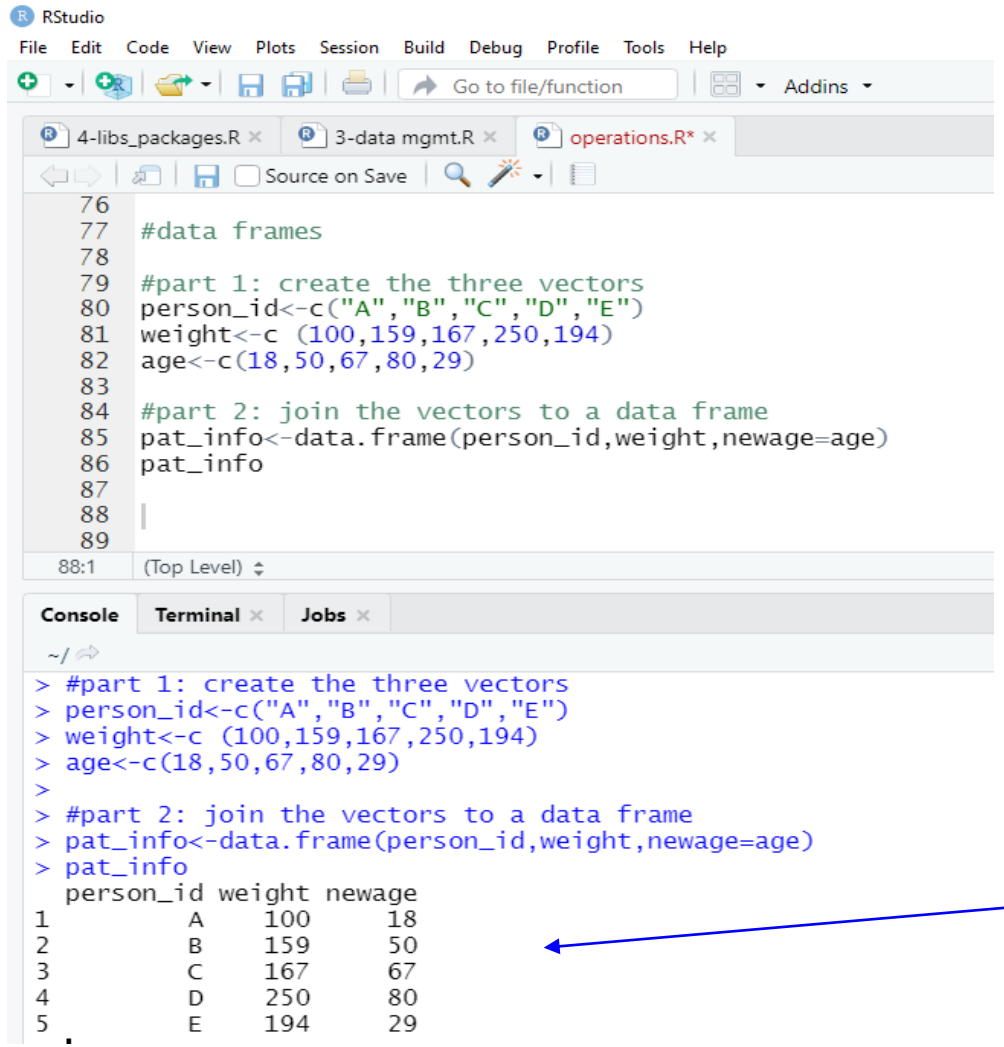
~/

```
> pulse <- c(64,50,100,70,97,59)
> patid <- c(57964, 14597,30146)
> BMIstat <- c("underweight", "normal","overweight","obese")
> cliniclist <- list(pulse,patid,BMIstat)
> cliniclist
[[1]]
[1]  64  50 100  70  97  59

[[2]]
[1] 57964 14597 30146

[[3]]
[1] "underweight" "normal"      "overweight"  "obese"
```

Data frames

- Data frames are versatile data objects in R and can be thought of as spreadsheets.
- Each column of a data frame is a vector with its own data elements
- Example on next slide

Data frames



Part 1: Creating 3 vectors called: person_id, weight, and age

Part 2: creating a dataframe named pat_info containing all previously created vectors using the 'data.frame' function

You can choose to rename your columns in the new data frame. For instance, I'm creating a column called newage that will contain all data elements from the age vector created in part 1.

Type pat_info to view your dataframe in the console window.

Assessing online help manuals

HELP.START()

You will notice the help window pop up in the bottom right corner

operations.R*

```
#part 1: create the three vectors
person_id<-c("A","B","C","D","E")
weight<-c (100,159,167,250,194)
age<-c(18,50,67,80,29)

#part 2: join the vectors to a data frame
pat_info<-data.frame(person_id=person_id,weight=weight,newage=age)
pat_info

#assessing online help manuals
help.start()


```

Type and run help.start()

```
1         A    100     18
2         B    159     50
3         C    167     67
4         D    250     80
5         E    194     29
>
>
> #assessing online help manuals
> help.start()
If nothing happens, you should open
'http://127.0.0.1:15295/doc/html/index.html' yourself
>
```

Environment | History | Connections | Tutorial

Import Dataset | List

Global Environment

Data

newlist	List of 5
pat_info	5 obs. of 3 variables

Values

Files | Plots | Packages | Help | Viewer

The R Language | Find in Topic

Statistical Data Analysis

Manuals

An Introduction to R
Writing R Extensions
R Data Import/Export
The R Language Definition
R Installation and Administration
R Internals

Reference

Packages
Search Engine & Keywords

Lecture summary

- **In this lecture, you learned how to perform basic operations in R including:**
  - Basic mathematical operations: addition, subtraction, multiplication, and division
- **You also learned about the different data elements in R**
  - Scalars
  - Vectors
  - Factors
  - Lists
  - Data frames
- **Finally, you learned about assessing online help manuals**

References

- https://bookdown.org/medepi/phds/working-with-vectores-matrices-and-arrays.html#understanding-arrays

QUIZ

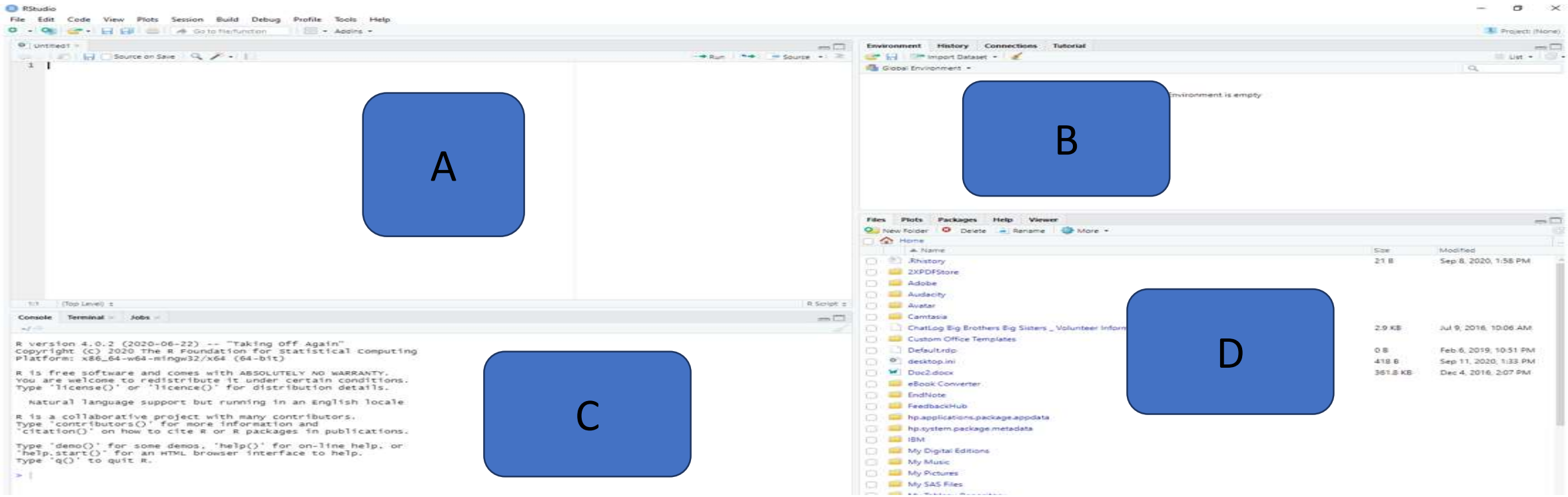
Question 1: Select the proper way to add a comment in your program

a) *this is how you add a comment
b) */this is how you add a comment*/
c) #this is how you add a comment
d) ~!this is how you add a comment

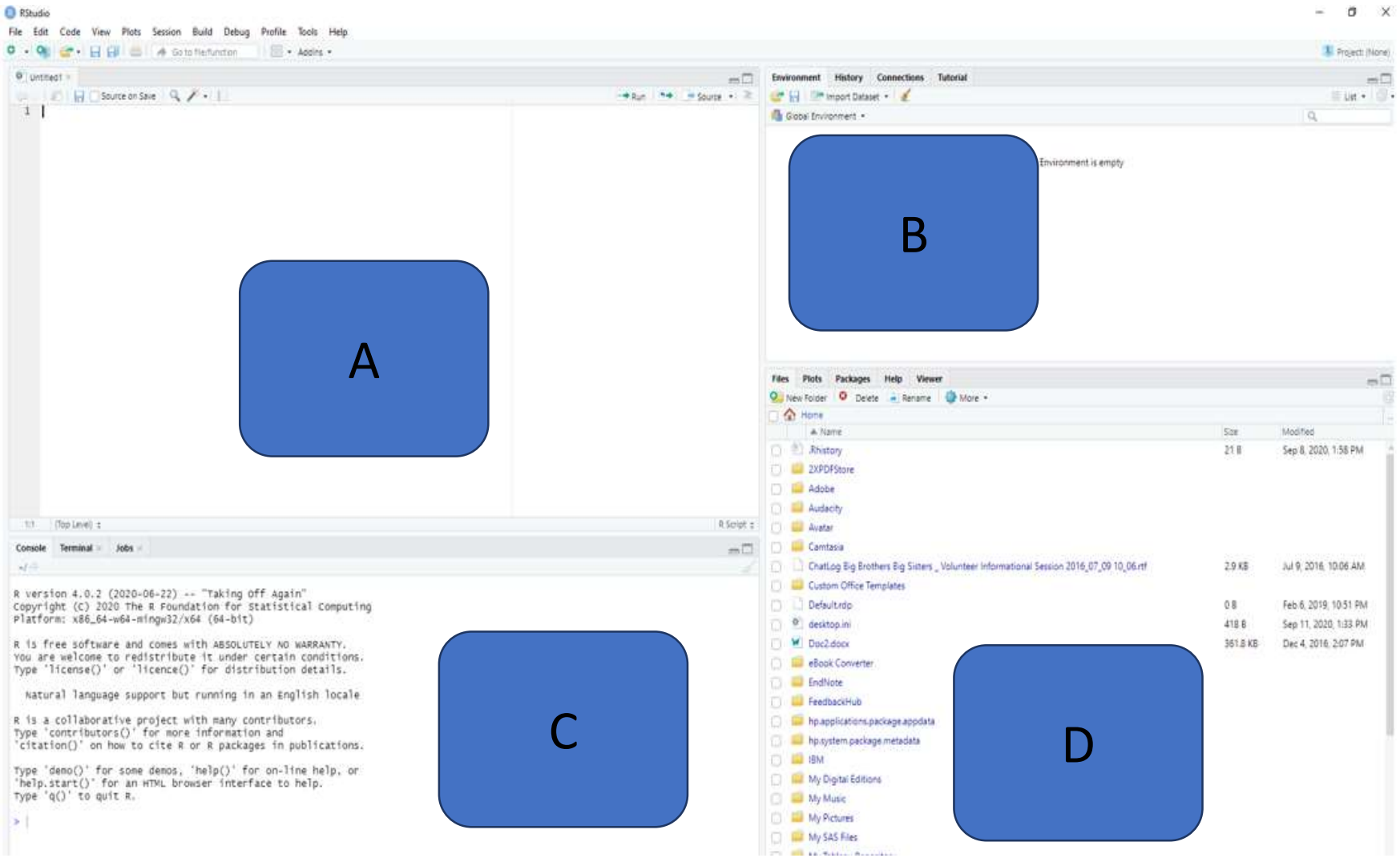
Question 2: True or False: R can be used as a scientific calculator

a) True
b) False

Question #3: Select the window in which you can write your “RScript”



a) Window A
b) Window B
c) Window C
d) Window D



Question #4: Select the window in which your output or results will be displayed

Window A
Window B
Window C
Window D

Question #5: True or False, RScripts are saved with a ".R" extension

a) True
b) False

Question #6: If I type, z <- 3 in R it will result in:

a) Data frame
b) Vector
c) Matrix
d) Scalar

Question #7: Which one of the following options is the most common data object in R?

a) Scalar
b) Vector
c) Matrix
d) Array

Question #8: Suppose you want to create a numerical vector called "num_vec" containing the following values: 2, 6, 7, 9. Which one of the options below will give you the desired result?

a) #num_vec <- b (2, 6, 7, 9)
b) c <- num_vec (2, 6, 7, 9)
c) num_vec <- (2, 6, 7, 9)
d) num_vec <- c (2, 6, 7, 9)

Question #9: True or false: A special utility of the factor variable is for statistical modeling and/or constructing frequency tables

a) True
b) False

Question #10: True or false: Lists can only contain homogenous data objects

a) True
b) False

Question #11: Suppose we have already created three vectors called "id", "sex", "BMI". Which of the following options would be the correct way to create a dataframe named "mydata" out of the three vectors?

a) Mydata <- c (id, sex, BMI)
b) Mydata <- dataframe (id, sex, BMI)
c) Mydata <- data.frame (id, sex, BMI)
d) Mydata <- maxtrix (id, sex, BMI

Question #12: How do you create a variable named x with the numeric value 5?

a) int X = 5
b) X: 5
c) X <- 5

Question #13: How can you assign the same value to multiple variables in one line?

a) var1,var2,var3 =“Orange”
b) var1 <- var2<-var3 <- “Orange”
c) Var1,var2,var3 <- “Orange”

14. Can you solve for x using R?

a <- 9 + 3 * 6

x <- a ÷ 2