Concept of data analysis and maintenance Illustration by WOOBRO DESIGN on Iconscout

Introduction to R for Research

RStudio and R Programming

Penn State, University Libraries, Research Informatics and Publishing

Workshop Housekeeping

Questions?

Use the chat or raise hand feature.

Feedback Survey

After each session, please fill out the Qualtrics survey (link in chat and emailed out after session).

Credits

Many images are sourced from the teaching team at Harvard Chan Bioinformatics Core (HBC).

Content was similarly inspired by HBC.

Slides with `(HBC Source)` in the bottom corner indicate the image/table source.

Original source: https://hbctraining.github.io/Training-modules/IntroR/

About Me

- Master's in Applied Statistics
- Bachelor's in Computational Statistics

R Experience:

- Self-taught for research in bioinformatics (2017)
- Statistics Courses (STAT 184/380)
- Research data processing (2017-Present)

Research Consultant, University Libraries

David Chen

dzc89@psu.edu

Research Consultant

github.com/TheDavidChen

Introduce Yourself!

Name, Major, Year, Favorite Letter

Why do we care?

Motivation - Max Number of Rows

Excel: 1,048,576

R: 2^48 = 281,474,976,710,656

https://support.microsoft.com/en-us/office/excel-specifications-and-limits-1672b34d-7043-467e-8e27-269d656771c3
https://cran.r-project.org/doc/manuals/R-ints.html#Long-vectors

Workshop Goals

- General data analysis skills
- Research Skills
- For grad school/career/research applications/opportunities
- Apply to current class/thesis
- Data Visualization
- Manage Big Data
- General Knowledge
- R

Agenda

- Why R/RStudio?
- The RStudio Interface
- Running Code
- Variable Assignment
- Data Types
- Data Structures
- Importing Datasets
- Basic Data Exploration

A Comment on Time

Pace vs Content tradeoff

Why R?

- Free/Open Source
- Platform Agnostic
- Reproducibility
- Popular in Research
- Designed for Data Analysis
- Customizable Data Visualizations
- Big Data!

Logo: The R Foundation

RStudio

- Free/Open Source
- Platform agnostic
- Interface for R

Reasons to use Base R instead:

General Comments

Expect a learning curve. Expect to struggle.

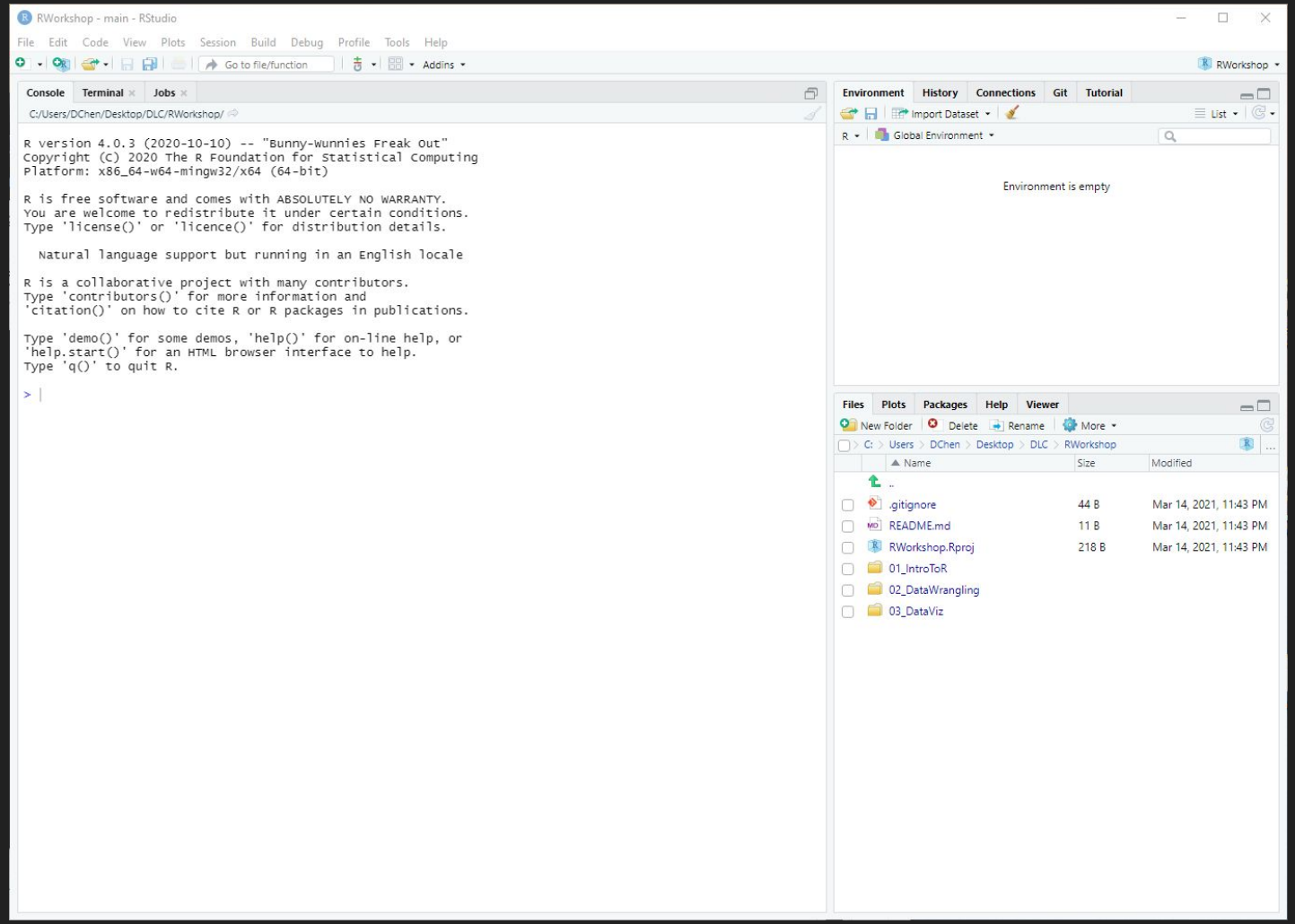
Errors are the best for learning. If they happen:

- Read the error and try to solve it
- Share the code that gave you an error
- Share the error message

RStudio®

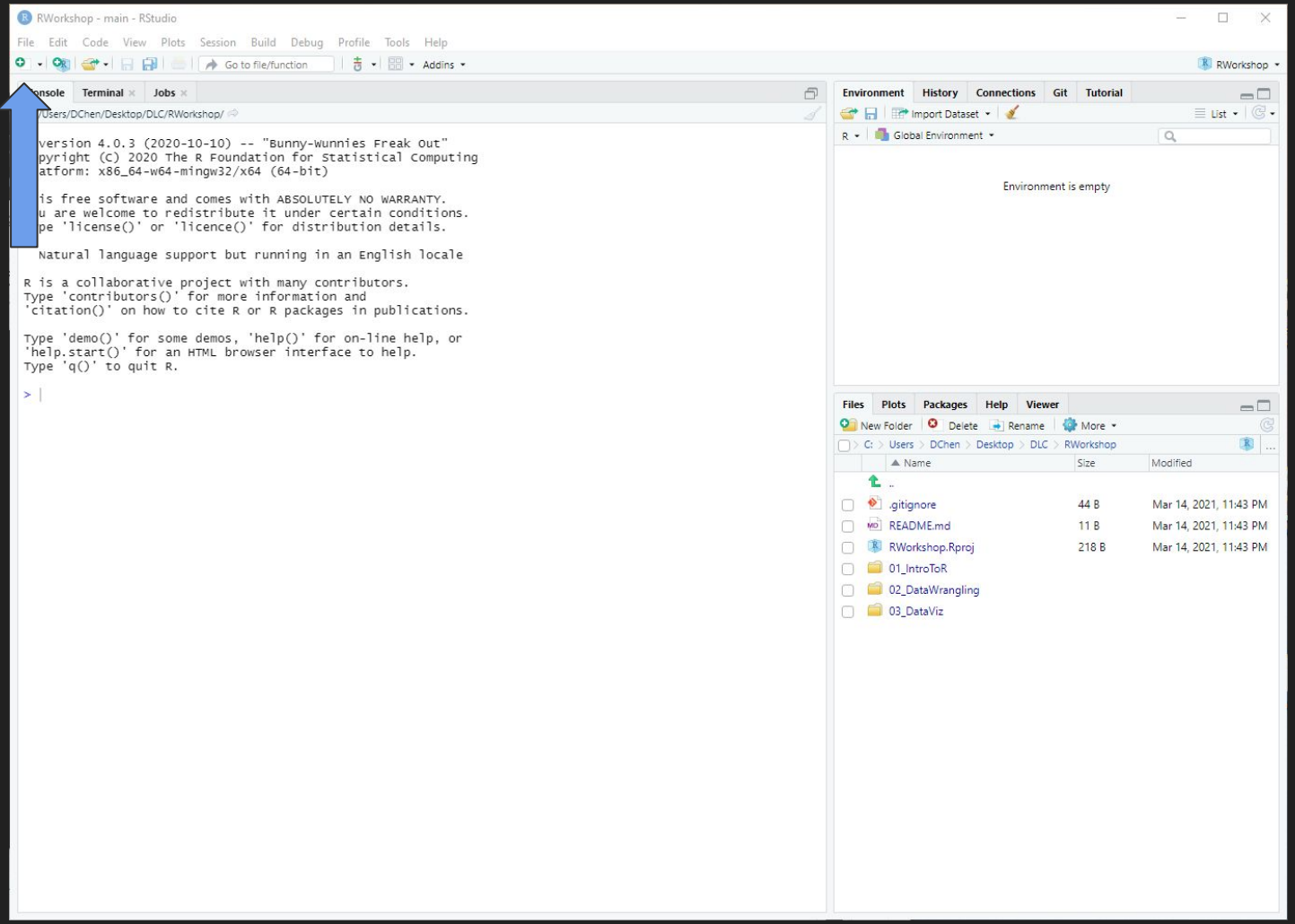
RStudio

Open RStudio!



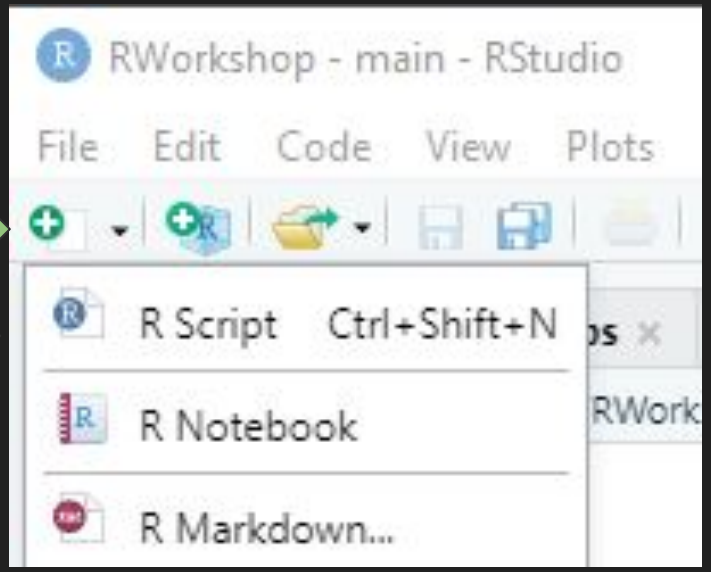
RStudio

Open RStudio!



RStudio

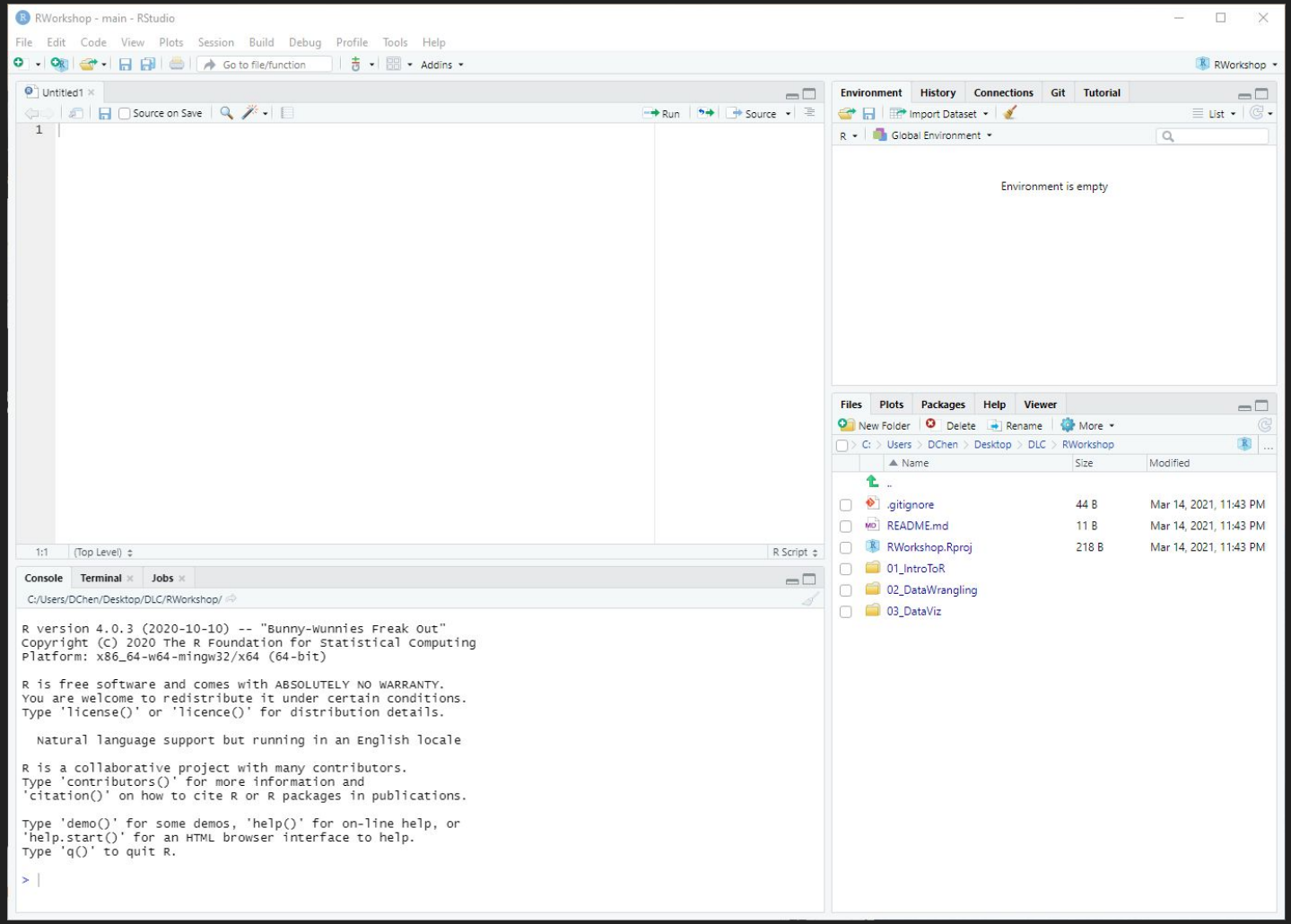
Create an R Script



Three options:

- Click the top left button
- Click `File -> New File -> R Script`
- Press `Ctrl`+`Shift`+`N`

RStudio


RWorkshop - main - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
RWorkshop
Untitled1
Source on Save
Run
Source
1:1
(Top Level)
R Script
Console Terminal Jobs
C:/Users/DChen/Desktop/DLC/RWorkshop/
R version 4.0.3 (2020-10-10) -- "Bunny-Wunnies Freak Out"
Copyright (C) 2020 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
Natural language support but running in an English locale
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
>
Environment History Connections Git Tutorial
Import Dataset
List
R
Global Environment
Environment is empty
Files Plots Packages Help Viewer
New Folder
Delete
Rename
More
C: Users DChen Desktop DLC RWorkshop
Name
Size
Modified
..
.gitignore 44 B Mar 14, 2021, 11:43 PM
README.md 11 B Mar 14, 2021, 11:43 PM
RWorkshop.Rproj 218 B Mar 14, 2021, 11:43 PM
01_IntroToR
02_DataWrangling
03_DataViz

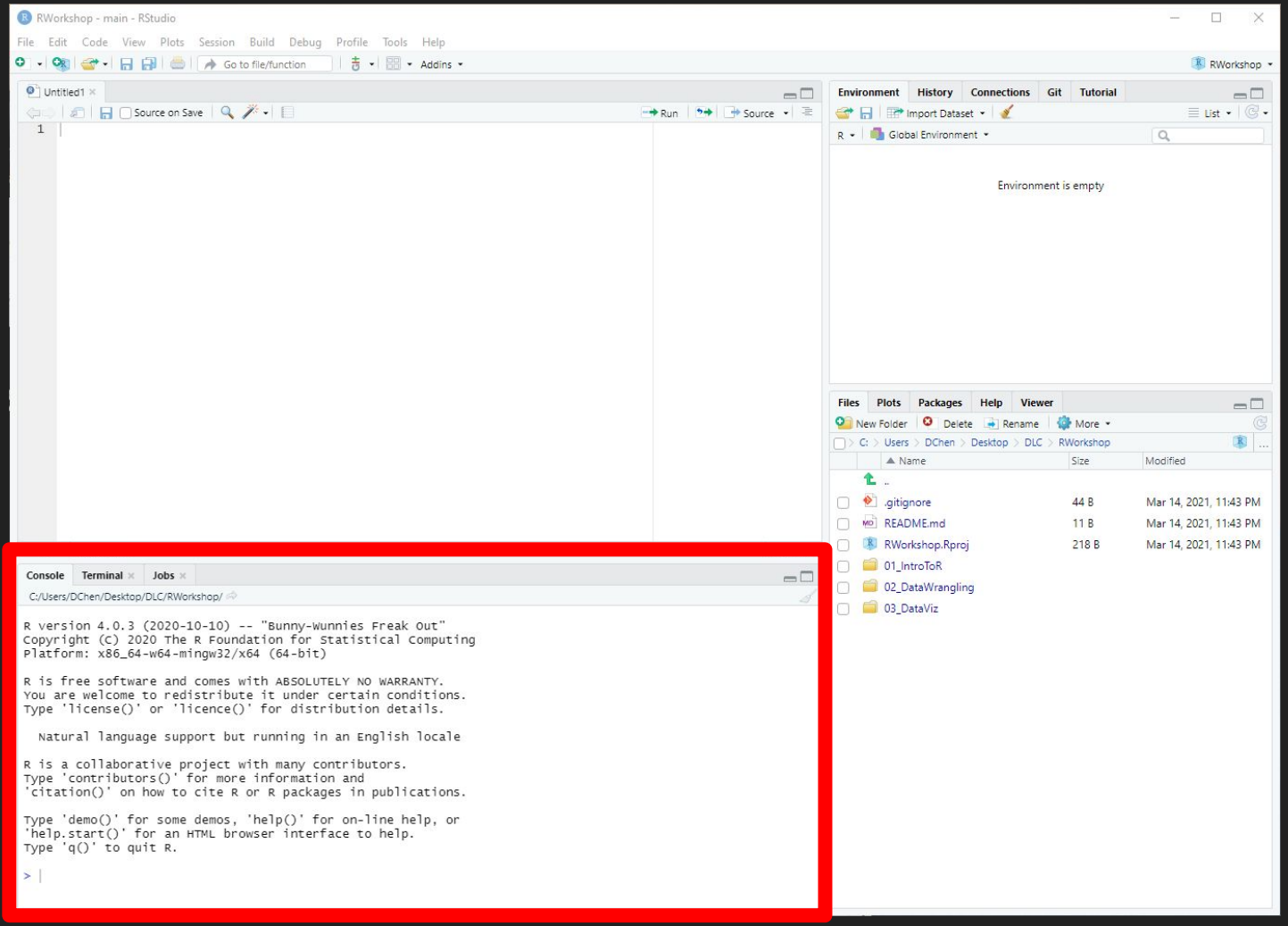
The Console

Run Code

Show Output

Show Errors

Code <u>Unsaved</u>

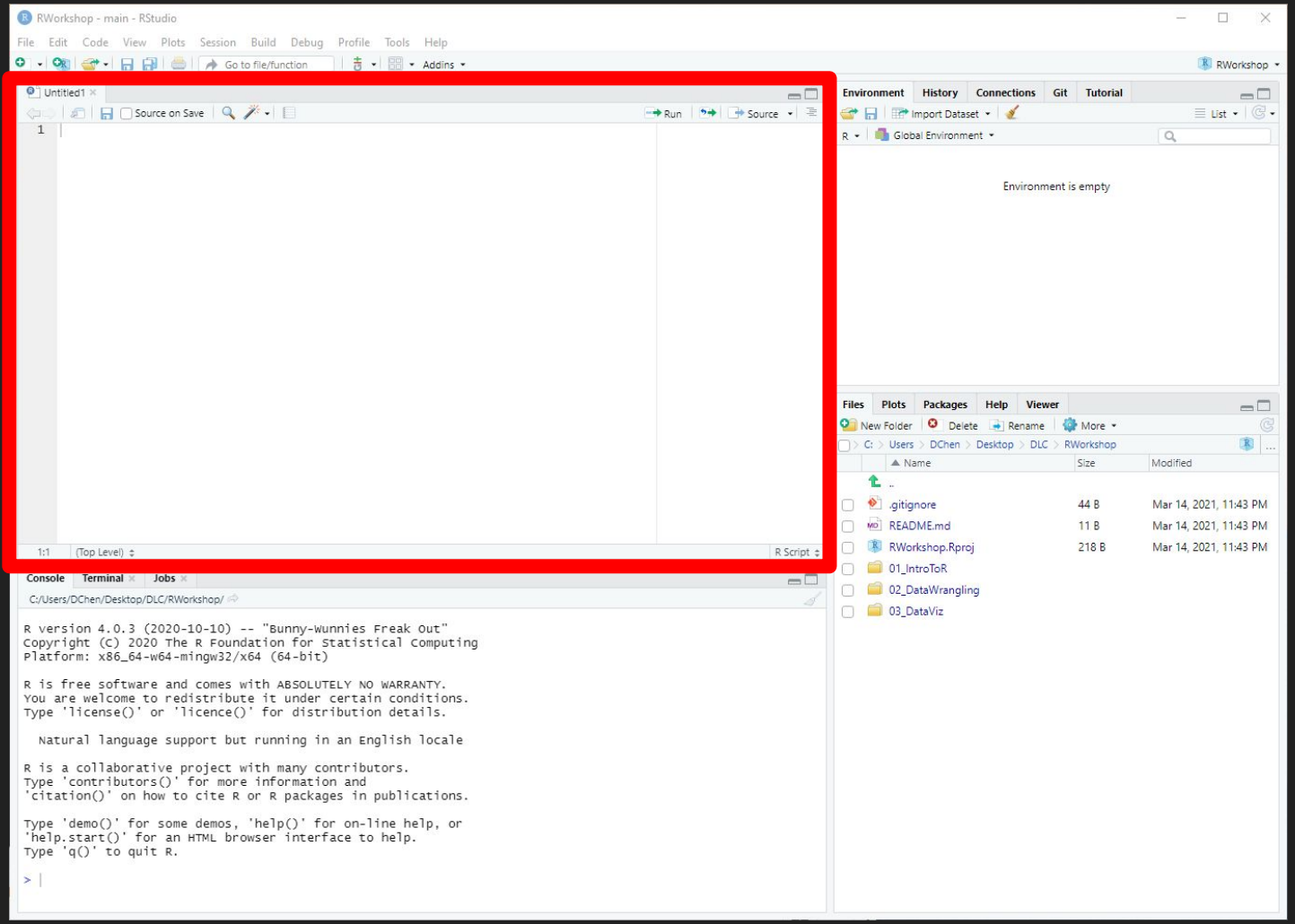


Script Editor

Type code

Run code (output in console)

Saves all code

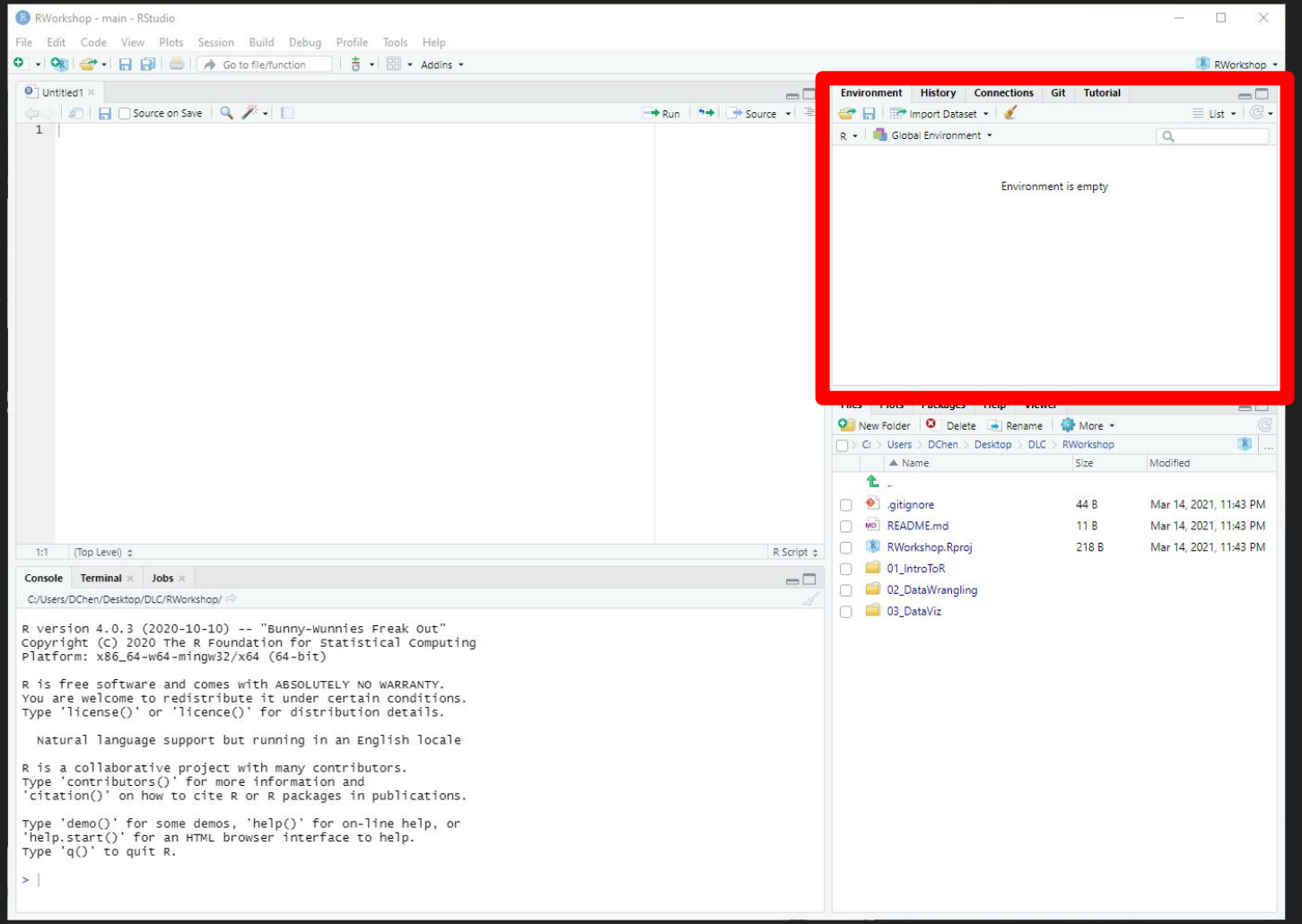


Environment

Import Datasets

Shows all datasets and defined variables

View imported datasets

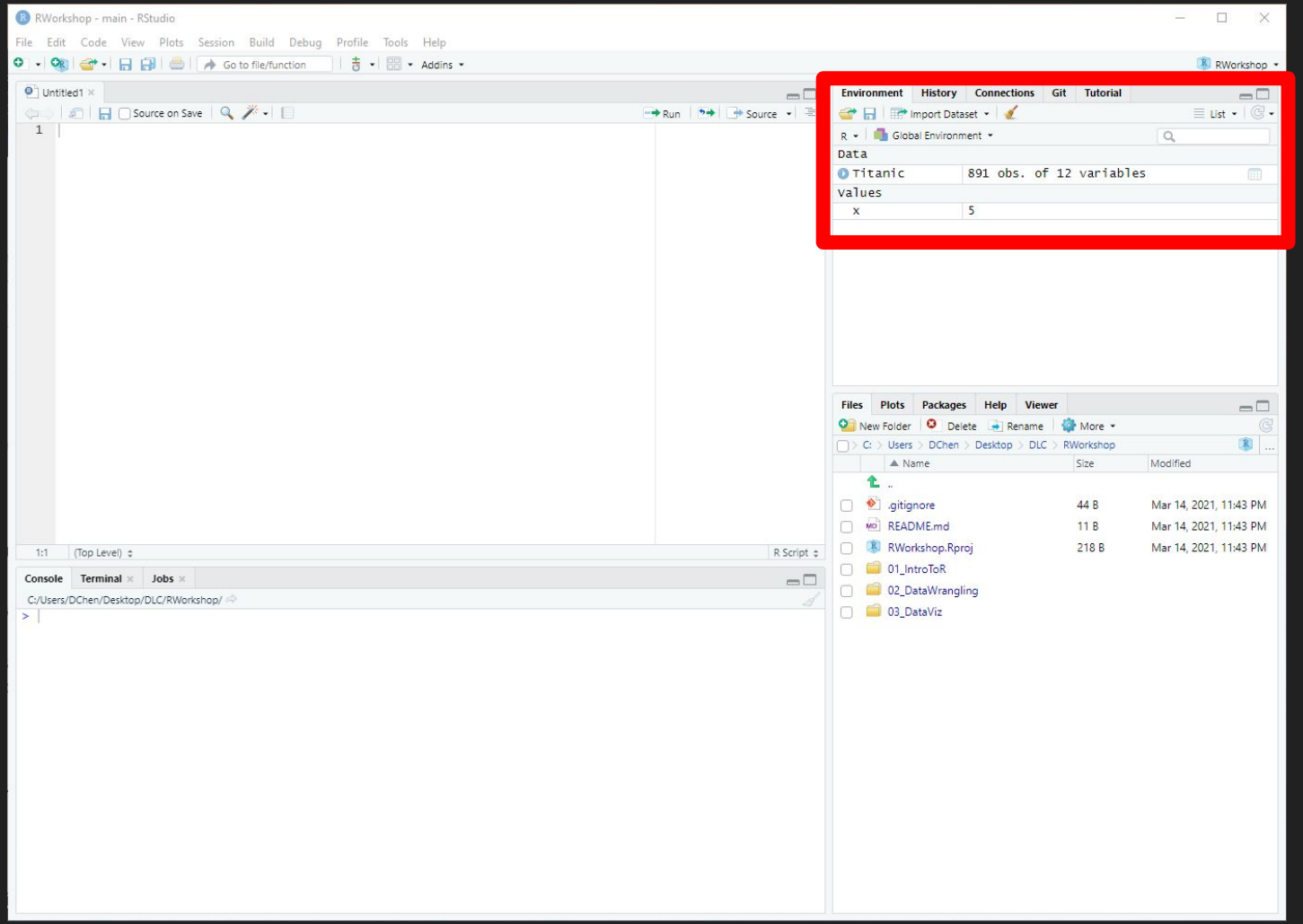


Environment

Import Datasets

Shows all datasets and defined variables

View imported datasets

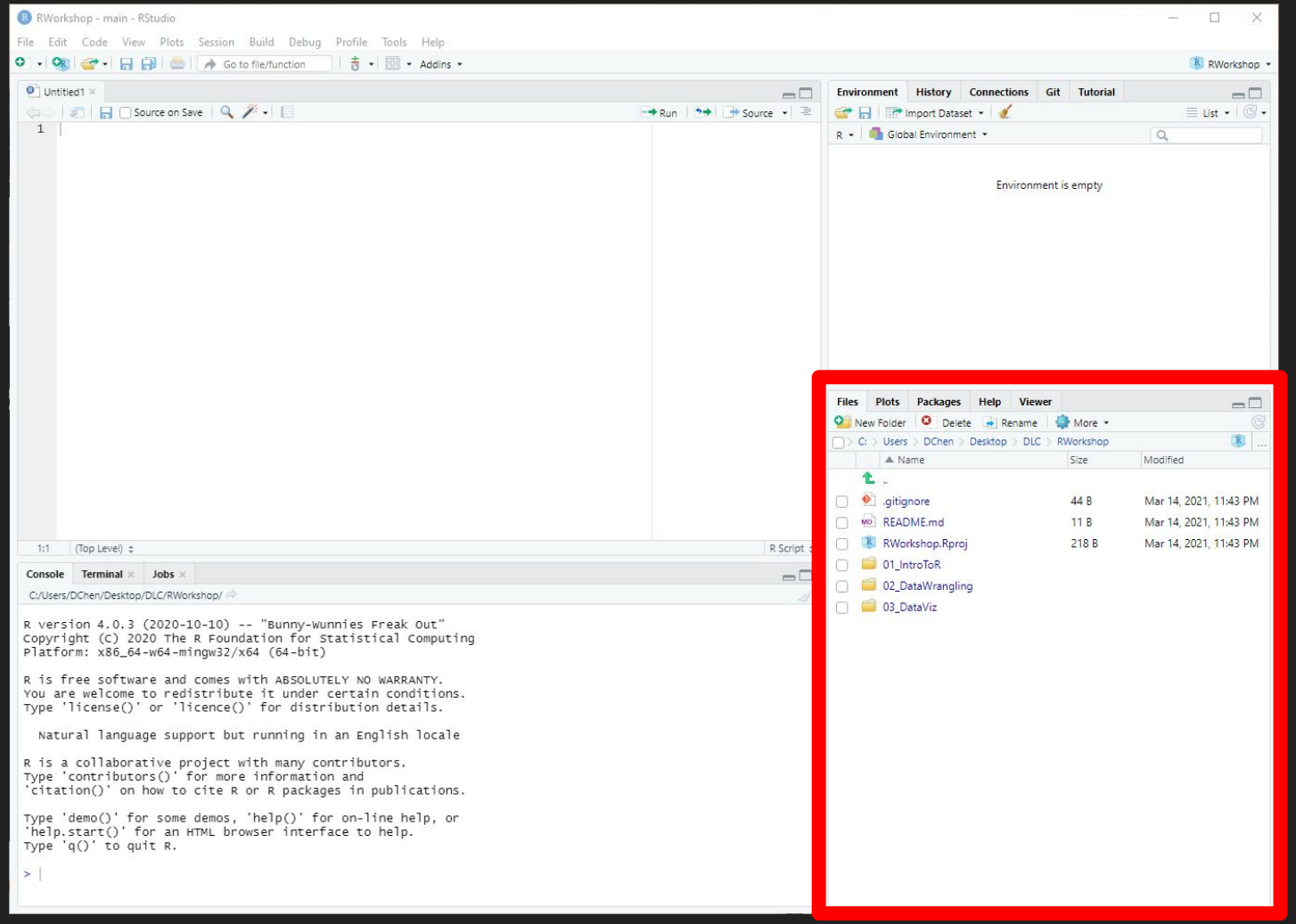


Misc.

Access Files

View created plots

Read help documentation



Running Code

General Comments

If you see: > print(“Hello World”)

Run print(“Hello World”) in R Script. Do not include the `>`.

Use `#` to write comments - code after # is not run.

> # This is not run

Running Code in the Console

Type the following calculations:

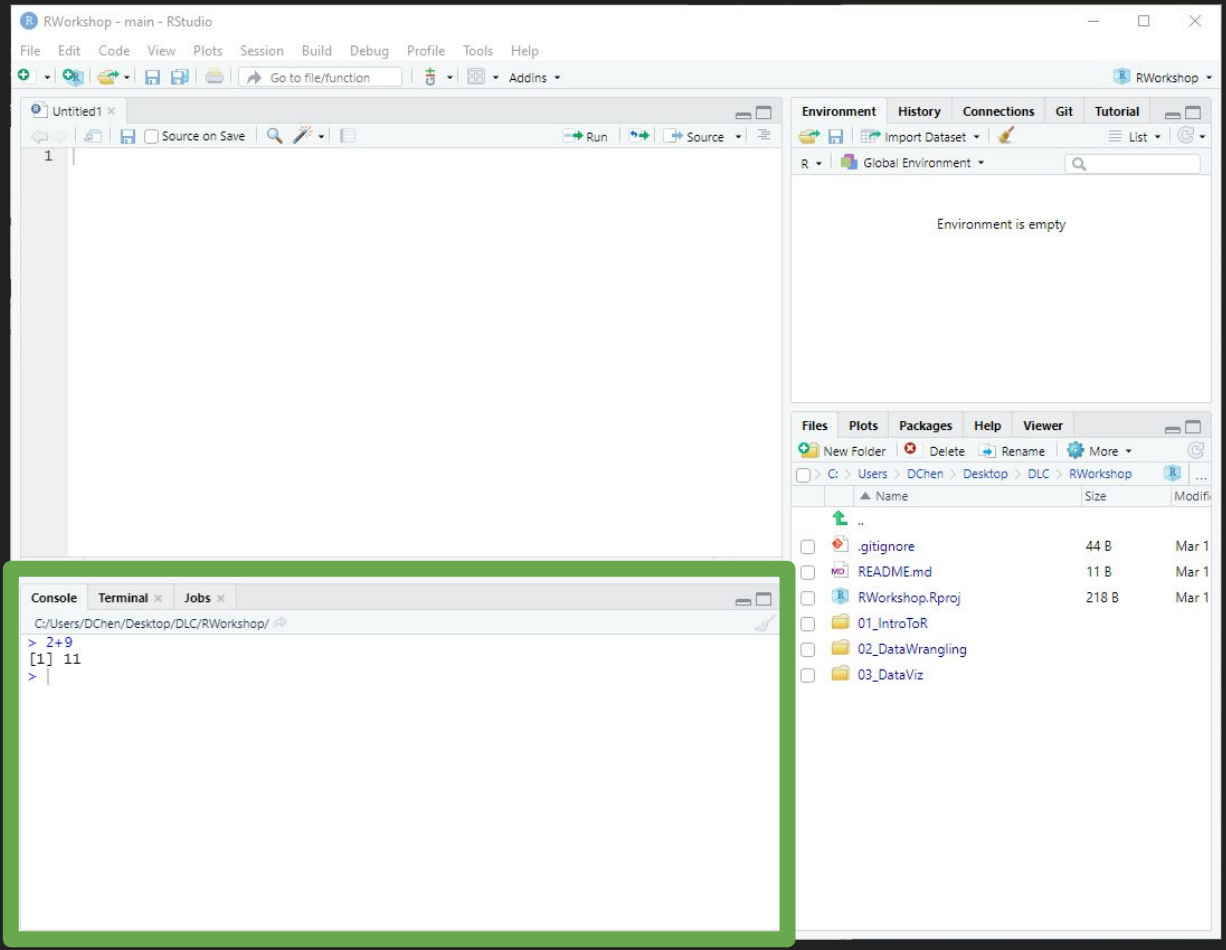
```
> 2+9
```

```
> 15*20
```

```
> 20/5
```

```
> 2^2
```

Press `Enter` to run



Running Code in the Console

Type the following calculations:

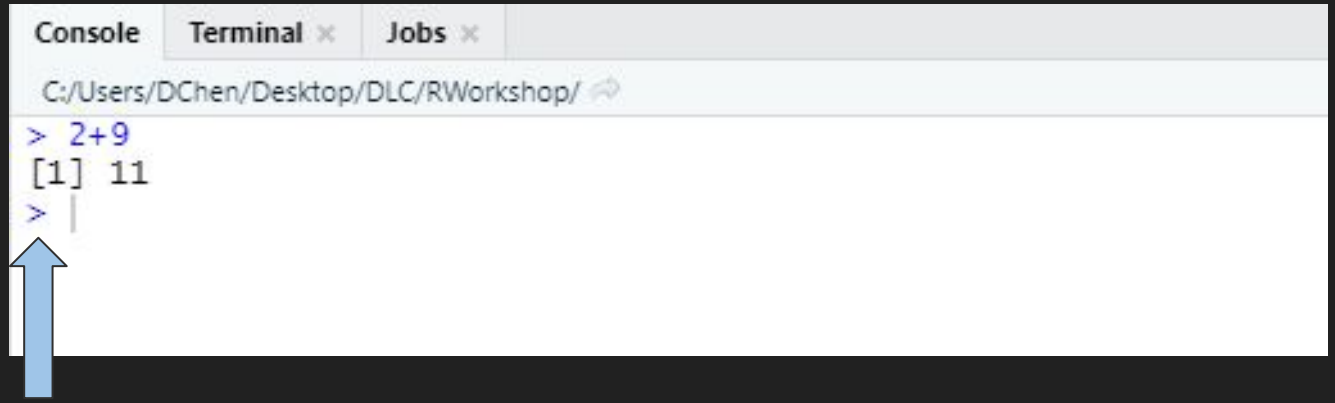
```
> 2+9

> 15*20

> 20/5

> 2^2
```

Press `Enter` to run



`>` means it’s ready for new code

Running Code in the Console

Type the following calculations:

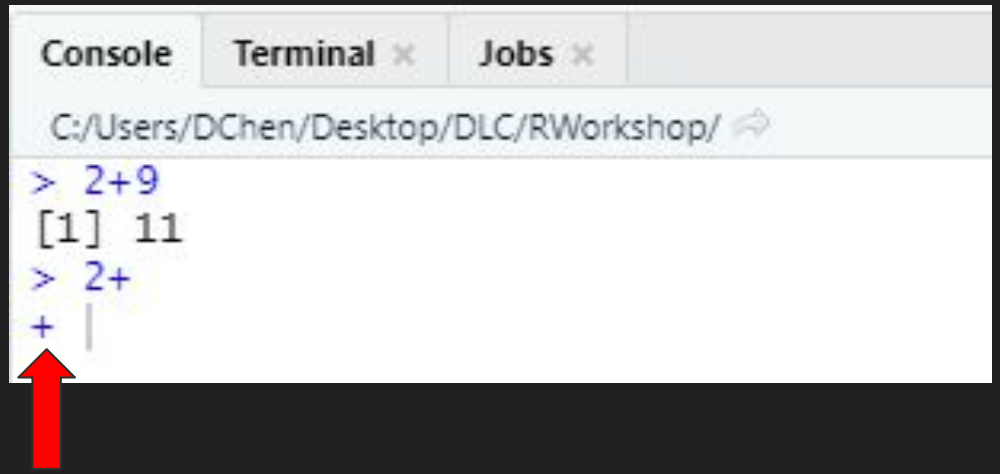
> 2+9

> 15*20

> 20/5

> 2^2

Press `Enter` to run



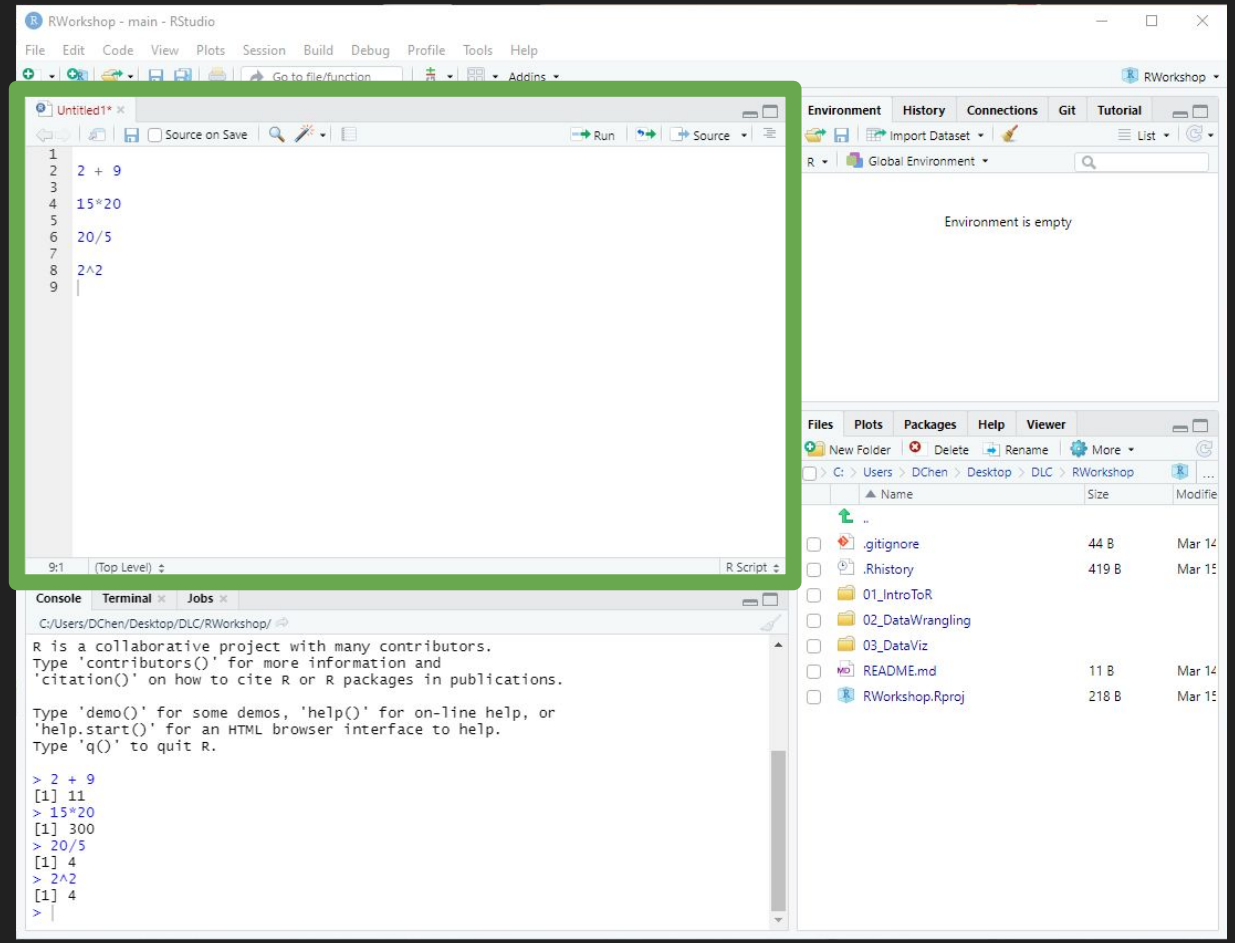
If you see + on the new line:

- Reset with the `Esc` key
- Continue from the previous code

Running Code in the Script

Code in the script can be saved and ran repeatedly

Output shows up in the console

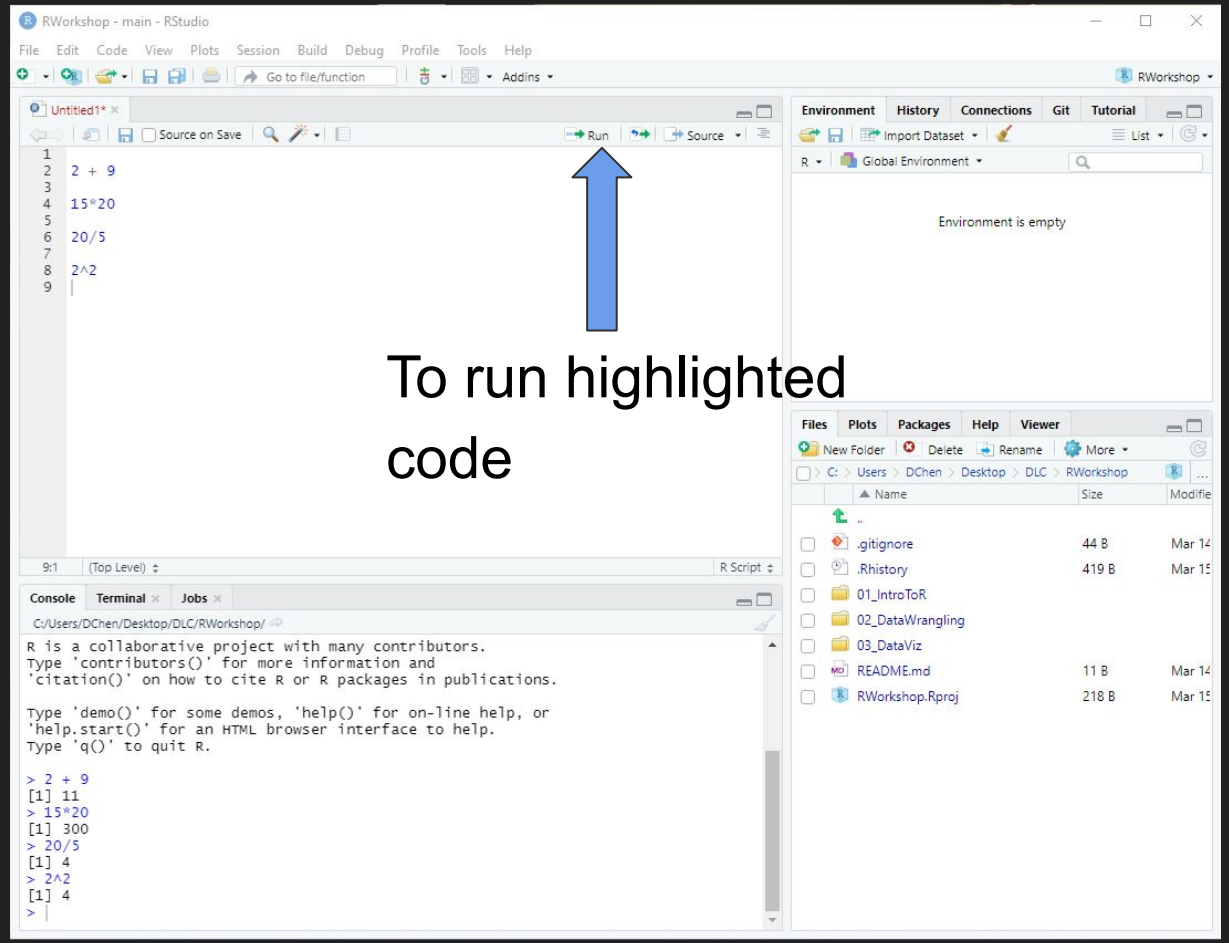


Running Code in the Script

Type code into the R Script

Click or highlight the line of code and

- press `Run`
- press `CTRL` + `ENTER`



Running Code in the Script

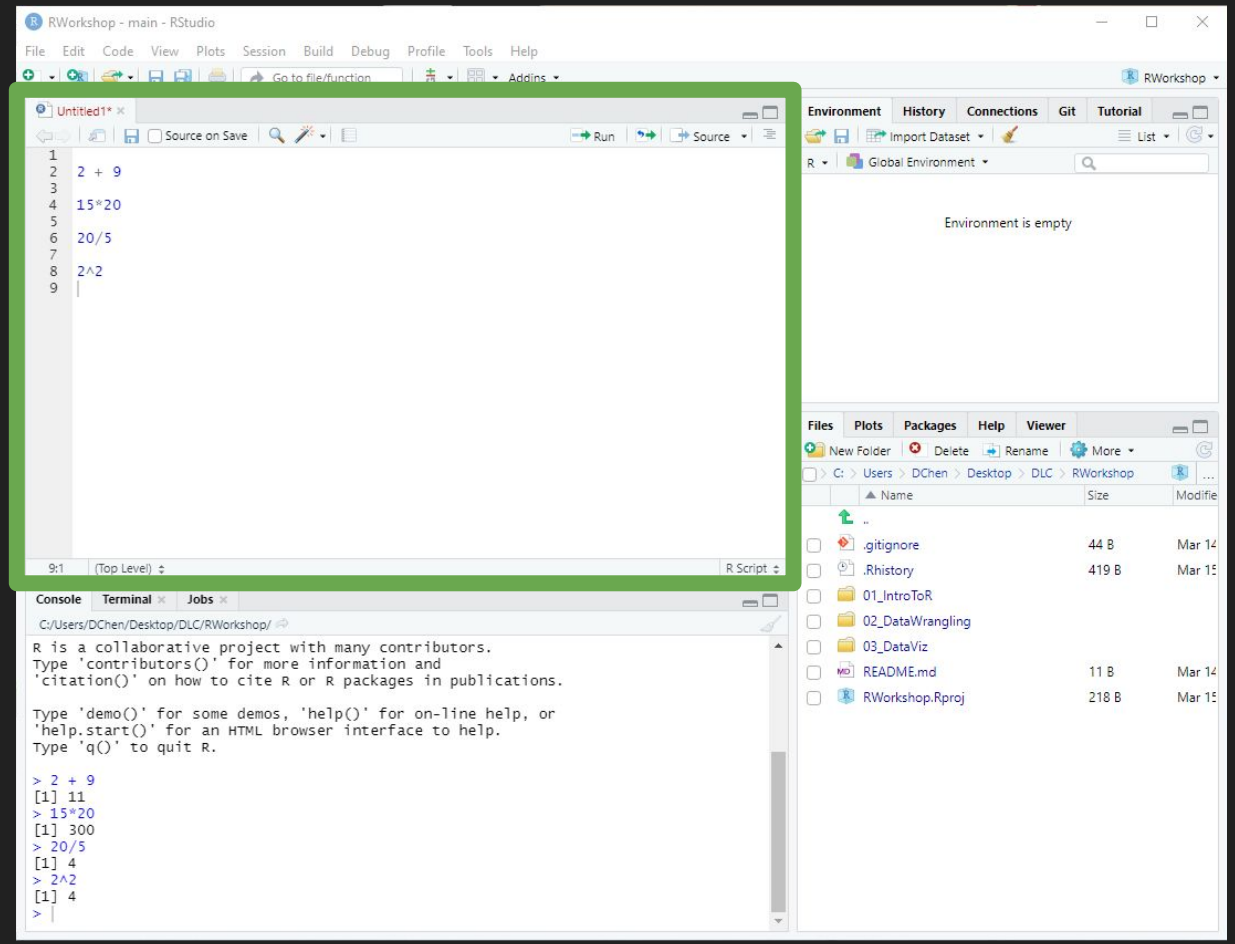
Type the following calculations into the R Script:

```
> 2+9
```

```
> 15*20
```

```
> 20/5
```

```
> 2^2
```



Variable Assignment

variable <- value

Assignment Operator

(Can also be `=`)

Variable Assignment

variable <- value

Replaces values

```
> x <- 5
```

```
> x + 10
```

Variable Assignment

variable <- value

Replaces values

```
> x <- 5
```

```
> x + 10
```

Redefine over and over

```
> y <- 5
```

```
> y <- 3
```

```
> y
```

Variable Assignment

variable <- value

Replaces values

```
> x <- 5
> x + 10
```

Redefine over and over

```
> y <- 5
> y <- 3
> y
```

Multiple variables

```
> x <- 1
> y <- 2
> x * y
```

Variable Assignment - Comments

R is case sensitive

```
> value <- 5
```

```
> VALUE + 5
```

Variable Assignment - Comments

R is case sensitive

```
> value <- 5
> VALUE + 5
```

Variable names must be one line and start with a character

```
> 1x <- 3      # not valid
> x y <- 3     # not valid
```

Functions

function_name(input)

```
> sqrt(4)
```

```
> print(“Hello World”)
```

Functions

What happens when you try:

```
> sqrt(“word”)
```

Data Types

Common Data Types

Data Type	Examples
Numeric	-5, 1, 3.33, 100, pi

(HBC Source)

Common Data Types

Data Type	Examples
Numeric	-5, 1, 3.33, 100, pi
Integer	-5L, 1L, 100L

(HBC Source)

Common Data Types

Data Type	Examples
Numeric	-5, 1, 3.33, 100, pi
Integer	-5L, 1L, 100L
Character	‘words’, “3.33”, ‘TRUE’, “1L”

(HBC Source)

Common Data Types

Data Type	Examples
Numeric	-5, 1, 3.33, 100, pi
Integer	-5L, 1L, 100L
Character	‘words’, “3.33”, ‘TRUE’, “1L”
Boolean/Logical	TRUE, FALSE, T, F

(HBC Source)

Common Data Types

Data Type	Examples
Numeric	-5, 1, 3.33, 100, pi
Integer	-5L, 1L, 100L
Character	‘words’, “3.33”, ‘TRUE’, “1L”
Boolean/Logical	TRUE, FALSE, T, F

(HBC Source)

Common Data Types - Boolean

Operator	Description
<	Less than
<=	Less than or equal to
>	Greater than
>=	Greater than or equal to
==	Equal to
!=	Not equal to

Common Data Types - Boolean

Examples

TRUE

- 5 > 3
- 5 != 3

FALSE

- 5 <= 3
- 8 == 10

Operator	Description
<	Less than
<=	Less than or equal to
>	Greater than
>=	Greater than or equal to
==	Equal to
!=	Not equal to

Common Data Types - Boolean

Try it yourself!

```
> 5 > 3
```

```
> 10 == 10
```

Applies to variables!

```
> x <- 5
```

```
> x == 5
```

```
> x > 10
```

Operator	Description
<	Less than
<=	Less than or equal to
>	Greater than
>=	Greater than or equal to
==	Equal to
!=	Not equal to

Common Data Types

Check the type of a variable/object with `class()`:

```
> class("4")
```

```
> class(4)
```

Certain functions require specific data types:

```
> sqrt("4")
```

Common Data Types - Common Mistakes

If a number is in quotations, think of it as the word instead of the number.

- “5” != 5

Common Data Types - Common Mistakes

If a number is in quotations, think of it as the word instead of the number.

- “5” != 5

TRUE/FALSE are actually encoded as 1/0

- TRUE == 1 ; FALSE == 0

Common Data Types - Common Mistakes

If a number is in quotations, think of it as the word instead of the number.

- “5” != 5

TRUE/FALSE are actually encoded as 1/0

- TRUE == 1 ; FALSE == 0

Characters without quotation marks are variables!

- Hi <- 5

Common Data Types - Common Mistakes

If a number is in quotations, think of it as the word instead of the number.

- “5” != 5

TRUE/FALSE are actually encoded as 1/0

- TRUE == 1 ; FALSE == 0

Characters without quotation marks are variables!

- Hi <- 5

Be careful of single `=`; outside of functions, it is assignment!

- 5 = 5 --> error!
- x = 5 is equivalent to x <- 5, but not recommended

Data Structures

Vectors

1	50	9	42

“A”	“B”	“C”

TRUE	F

A column in excel

Any length from 1 onwards

All values have to be same type (all numeric, boolean, character, etc.)

One value in a vector is called an **element**

(HBC Source)

Vectors

Create vectors with function `c()`

```
> lengths <- c(50, 100, 150, 200)
```

```
> color <- c("blue", "white", "purple")
```

Bonus: What happens if you try to add a different type?

```
> bonus <- c("Yes", TRUE, 5)
```

Factors

A special vector with values assigned to each factor level (category)

(HBC Source)

Data Frame

"A"	102	"Hela"	TRUE
"B"	40	"BHK"	F
"C"	12	"hESC"	T

Most common data format; equivalent to an excel sheet.

Multiple vectors combined together

Columns must be same type

All columns must have equal number of rows

(HBC Source)

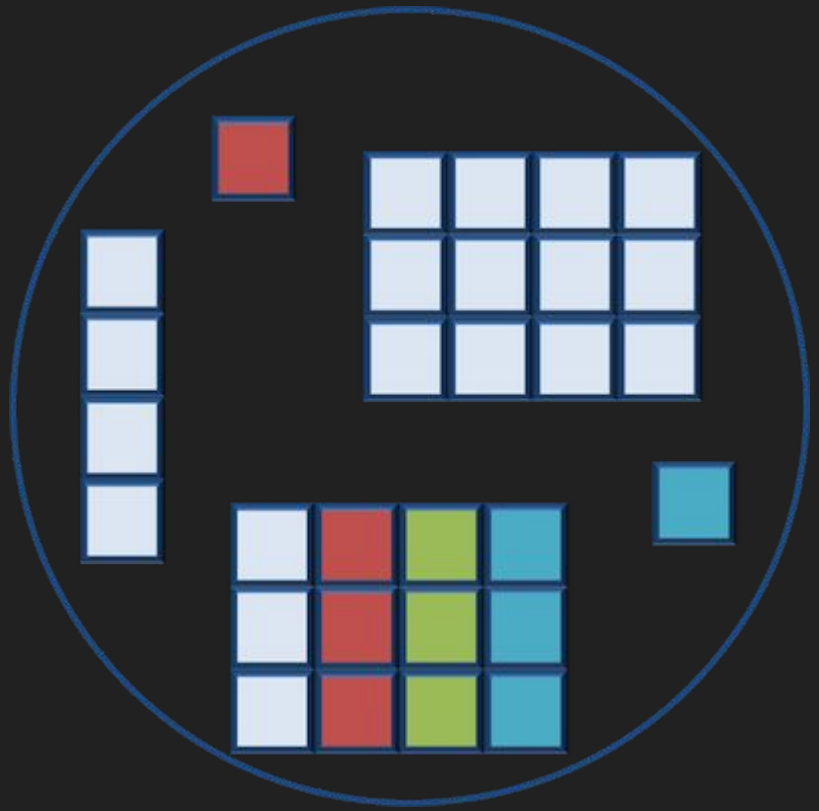
Matrix

90	5	137	9
87	40	2	52
4	102	32	41

Data frame that requires same type and length

Used in some statistical/mathematical functions

Lists



Holds any number of data structures or types

Think of it as the excel file, where it can contain multiple sheets

(HBC Source)

Reading in Data

Titanic Dataset: https://web.stanford.edu/class/archive/cs/cs109/cs109.1166/stuff/titanic.csv

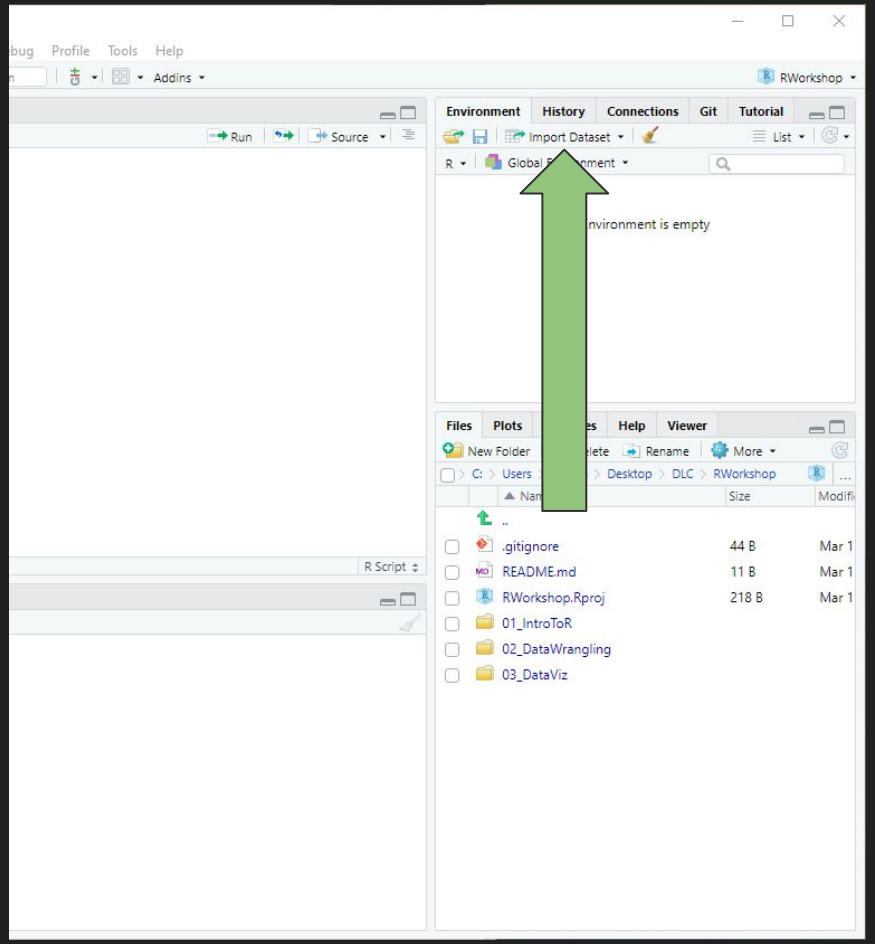
Reading in Datasets

Data type	Extension	Function	Package
Comma separated values	csv	`read.csv()`	utils (default)
		`read_csv()`	readr (tidyverse)
Tab separated values	tsv	`read_tsv()`	readr
Other delimited formats	txt	`read.table()`	utils
Stata version 7-12	dta	`read.dta()`	foreign
SPSS	sav	`read.spss()`	foreign
SAS	sas7bdat	`read.sas7bdat()`	sas7bdat
Excel	xlsx, xls	`read_excel()`	readxl (tidyverse)

(HBC Source)

Reading in Datasets

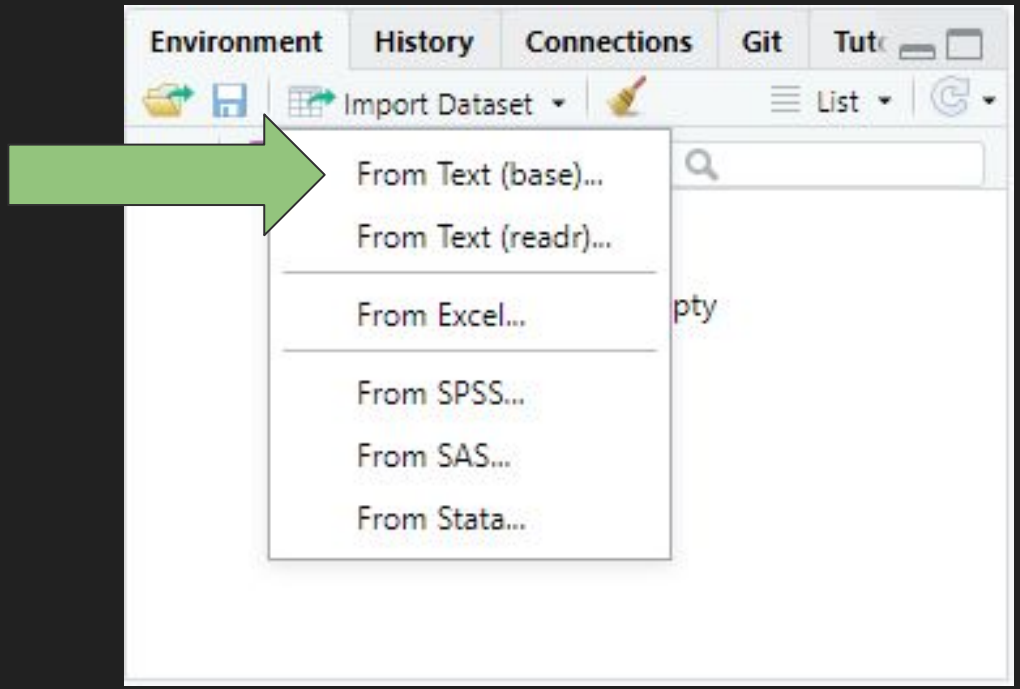
`Import Dataset` in the environment will automatically generate the code for you!



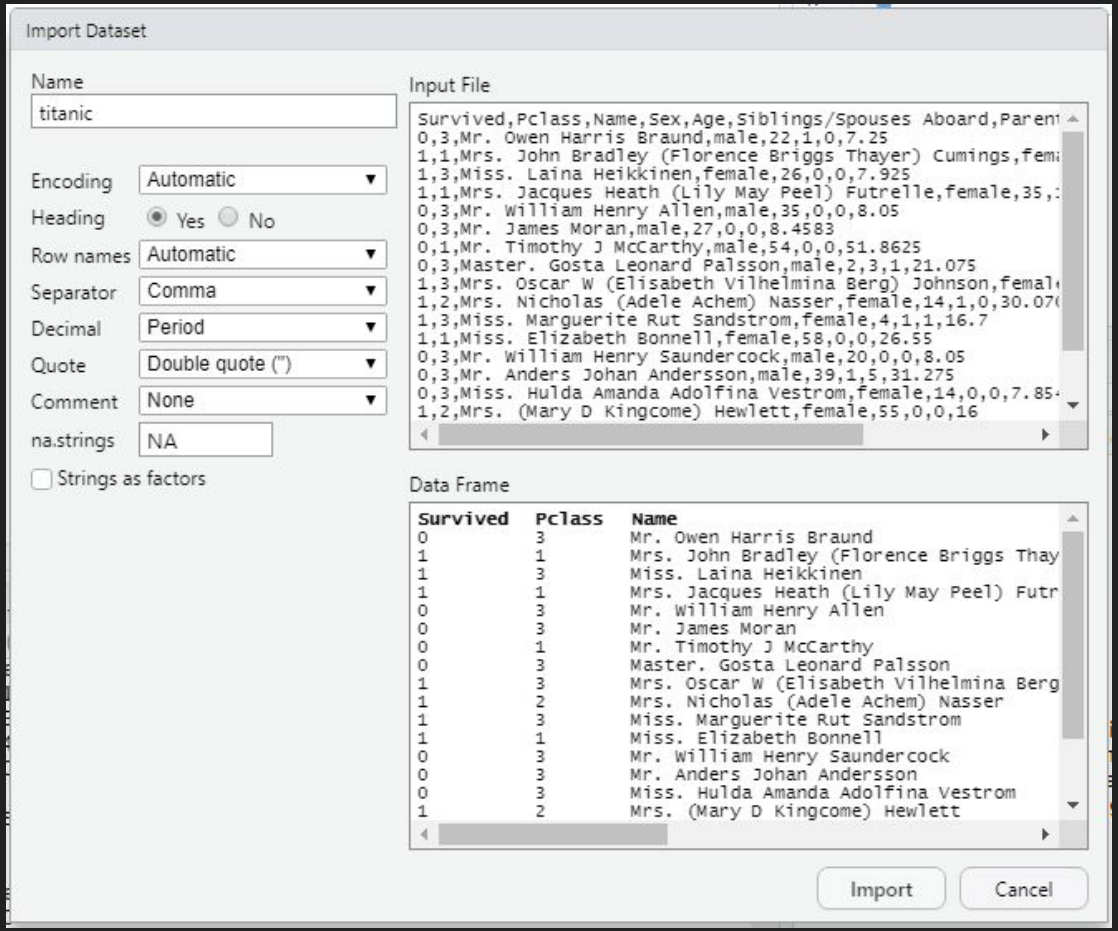
Reading in Datasets

From Text works for almost all basic data files (.csv, .txt, etc.)

Find and select the dataset



Reading in Datasets

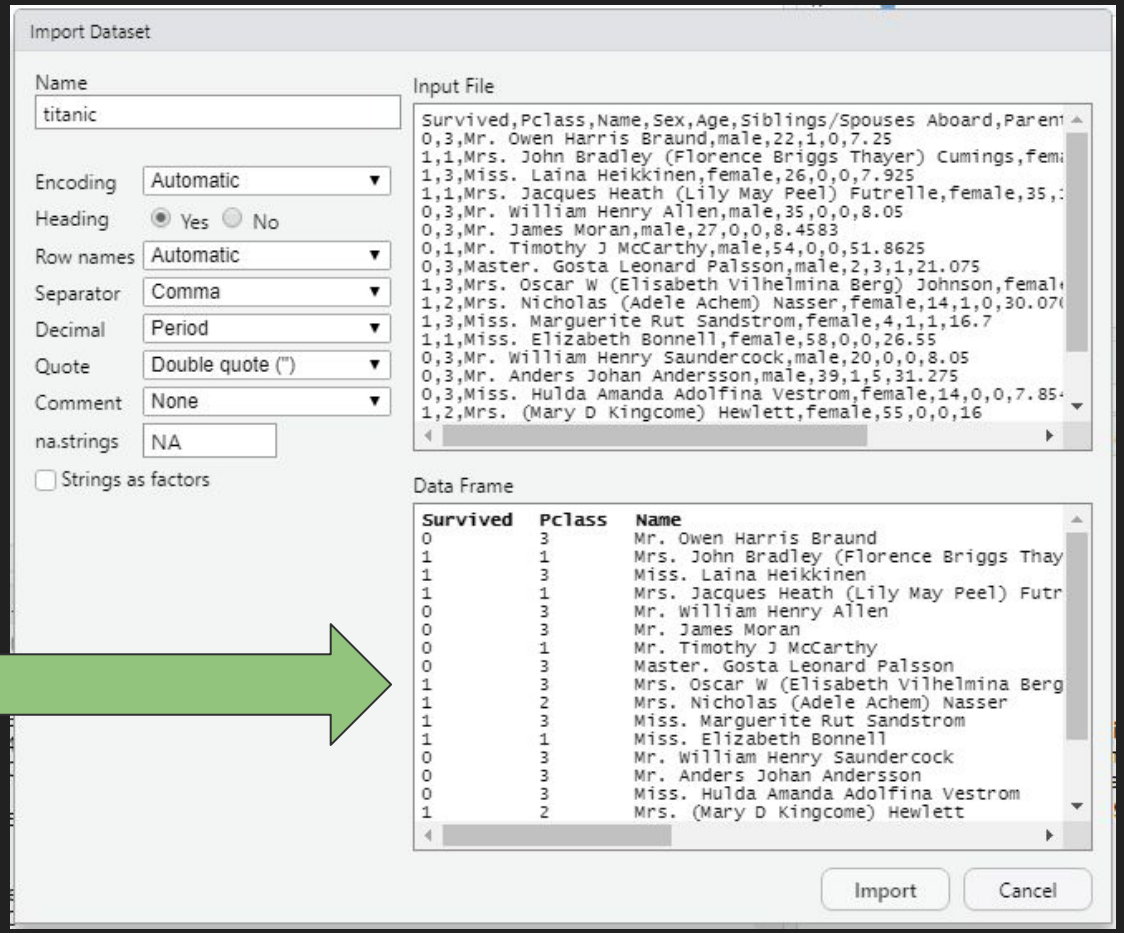


Reading in Datasets

Make sure the data is being imported correctly!

Check:

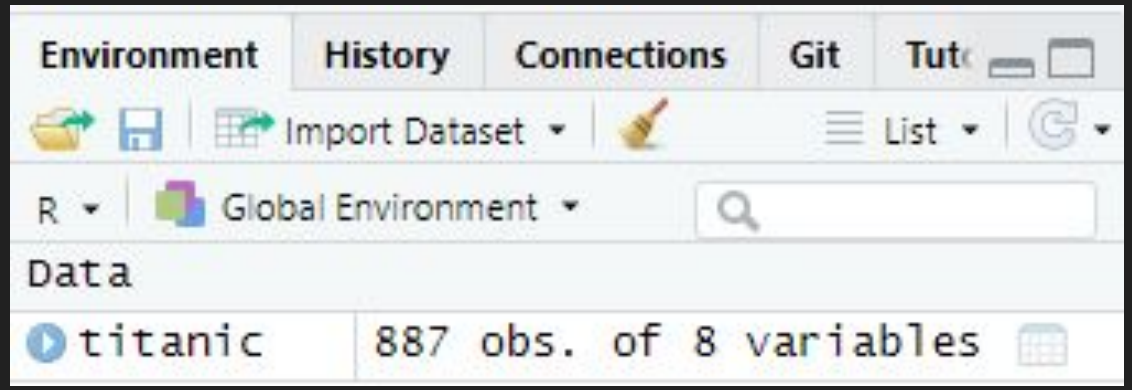
- Headers
- Columns
- Values



Viewing the Datasets

The data is now in the environment!

Click the name (titanic) to view the data!



Viewing the Datasets

The data view after clicking the name in the environment

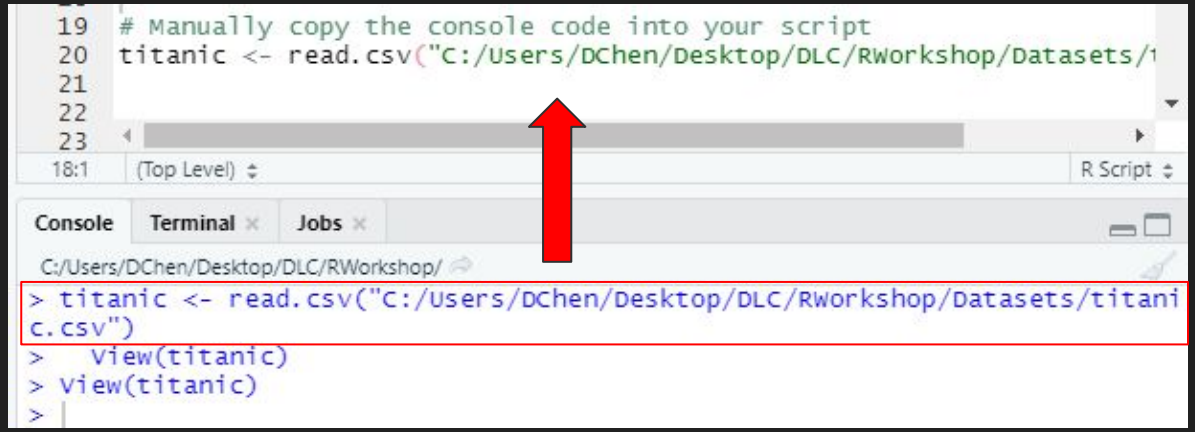
Untitled1* titanic

Filter

	Survived	Pclass	Name	Sex	Age	Siblings
1	0	3	Mr. Owen Harris Braund	male	22.00	
2	1	1	Mrs. John Bradley (Florence Briggs Thayer) Cumings	female	38.00	
3	1	3	Miss. Laina Heikkinen	female	26.00	
4	1	1	Mrs. Jacques Heath (Lily May Peel) Futrelle	female	35.00	
5	0	3	Mr. William Henry Allen	male	35.00	
6	0	3	Mr. James Moran	male	27.00	
7	0	1	Mr. Timothy J McCarthy	male	54.00	
8	0	3	Master. Gosta Leonard Palsson	male	2.00	
9	1	3	Mrs. Oscar W (Elisabeth Vilhelmina Berg) Johnson	female	27.00	
10	1	2	Mrs. Nicholas (Adele Achem) Nasser	female	14.00	
11	1	3	Miss. Marguerite Rut Sandstrom	female	4.00	
12	1	1	Miss. Elizabeth Bonnell	female	58.00	

After Importing the Data

Copy the generated code into the R Script.



In the future, you will only need to run the code.

Do not include the `>` or the `View()` functions.

Examine the Data - Try These Functions!

Functions - Add `titanic` to ()	
str()	dim()
summary()	nrow()
head()	ncol()
tail()	colnames()

```
> str(titanic)
```

```
> head(titanic)
```

```
> colnames(titanic)
```

Next Workshop
Data Processing

```
> install.packages("tidyverse")
```