

MEDICAL MICROBIOLOGY AND INFECTIOUS DISEASES CODING WORKSHOP

Presents

Introduction to R

INSTRUCTED BY
Grace E. Seo (MSc student)

Email: seog@myumanitoba.ca

Educational series supported by the Department of MMID to promote knowledge translation and dissemination among students and researchers



INFORMATION FOR PARTICIPANTS

All workshops are being recorded and posted to the
[MMID Coding Workshop - YouTube](#)

*Please hold your questions until Q & A session
Question and Answer period will not be recorded.*

INFORMATION FOR PARTICIPANTS

**February 3, 2022 - Thursday Bi-weekly Q & A session
9:00 – 10:30 AM**

Q & A session is open to everyone and does not require a registration.

Zoom meeting Invite Link

<https://umanitoba.zoom.us/j/64916599604?pwd=ci9wL0h1WTlhSHFxQmJvajItQXJmZz09>

**Meeting ID: 649 1659 9604
Passcode: 210551**

LEARNING OBJECTIVES

1. **Describe types of analyses performed using R or BASH.**
2. **Install R and RStudio on your operating system.**
3. **Learn how to customize RStudio.**
4. **Navigate, create, rename, move, copy directory and file in R.**
5. **Install packages, load libraries and write a simple R script.**
6. **Write a simple R markdown.**

Part 1. What is R programming language?

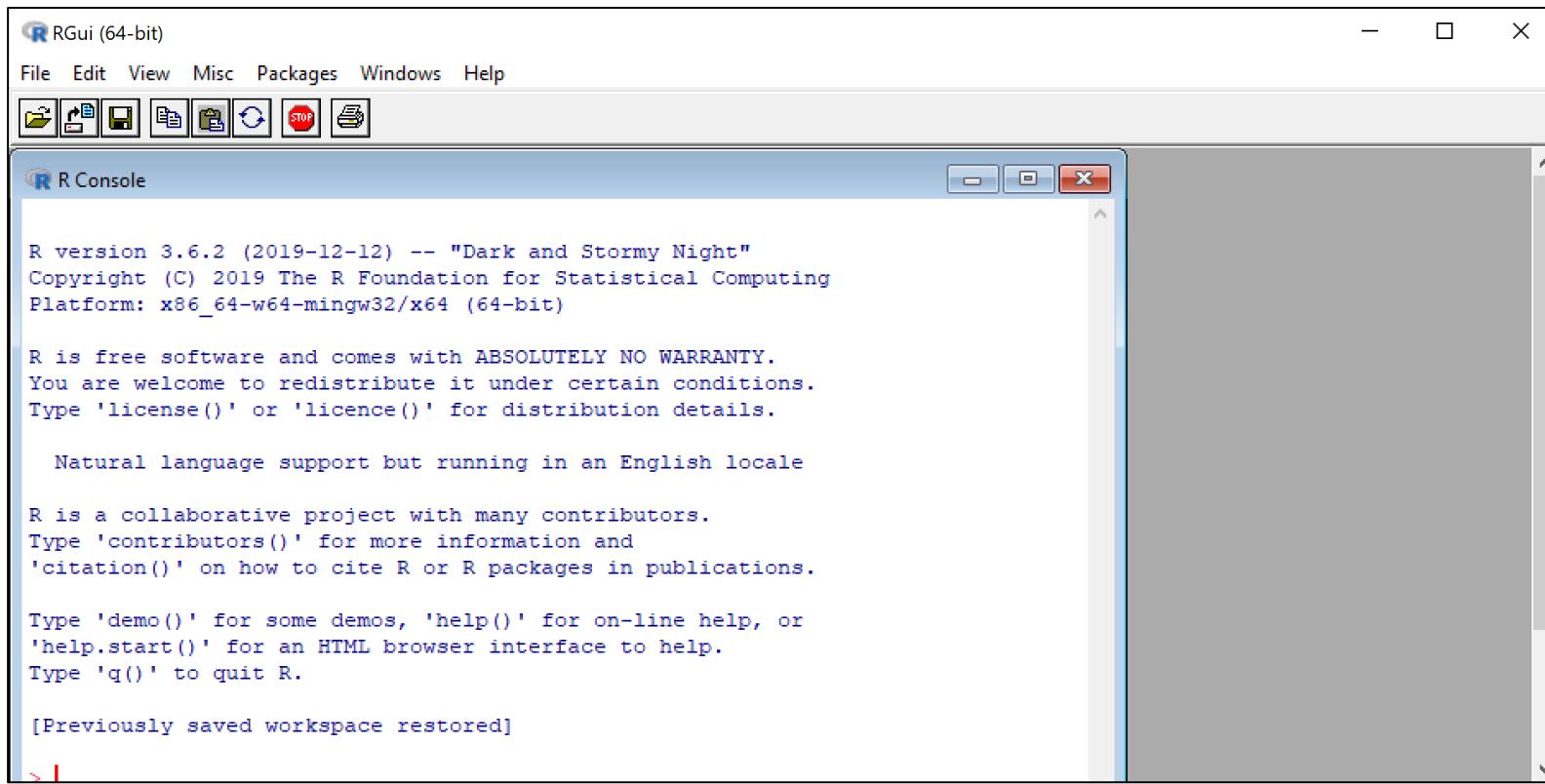
LEARNING OBJECTIVES

- 1. *Describe types of analyses performed using R or BASH.***
- 2. *Install R and RStudio on your operating system.***
- 3. *Learn how to customize RStudio.***
- 4. *Navigate, create, rename, move, copy directory and file in R.***
- 5. *Install packages, load libraries and write a simple R script.***
- 6. *Write a simple R markdown.***



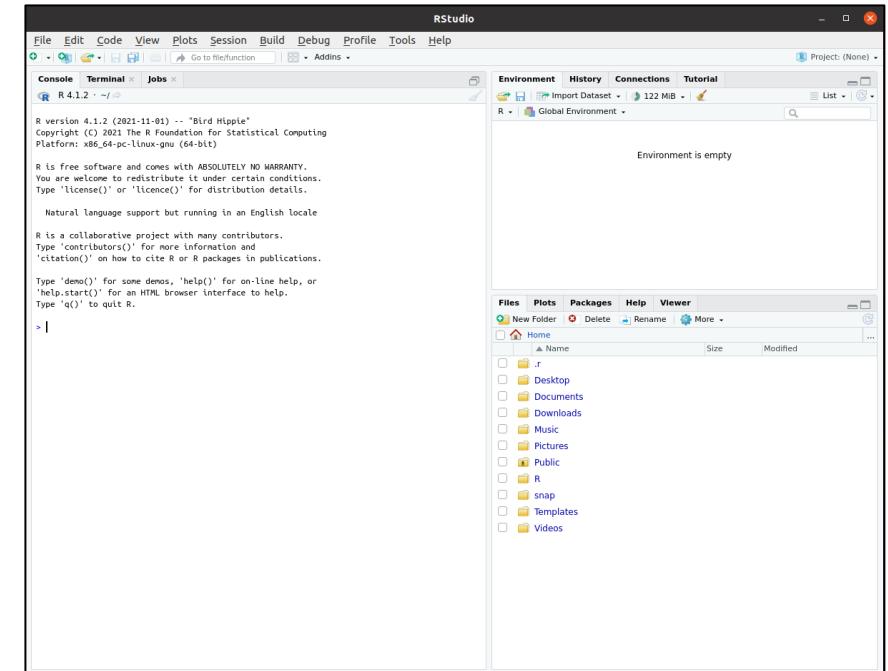
What is R?

- Programming language used for statistical computing and data visualization

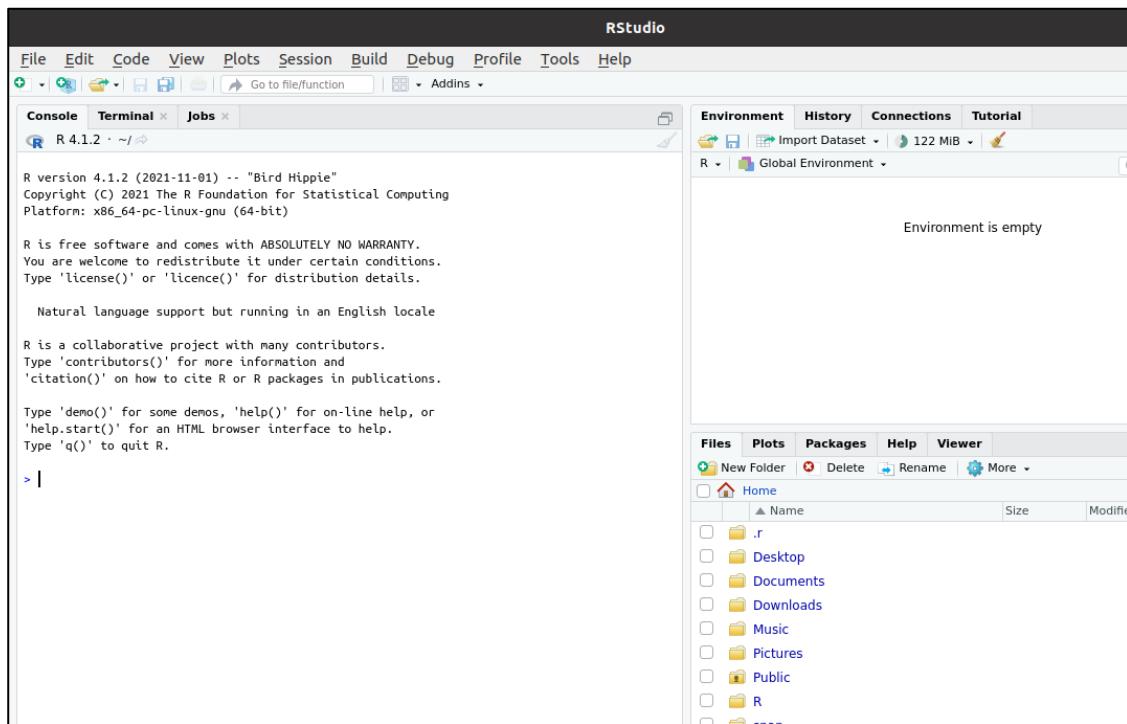


What is RStudio?

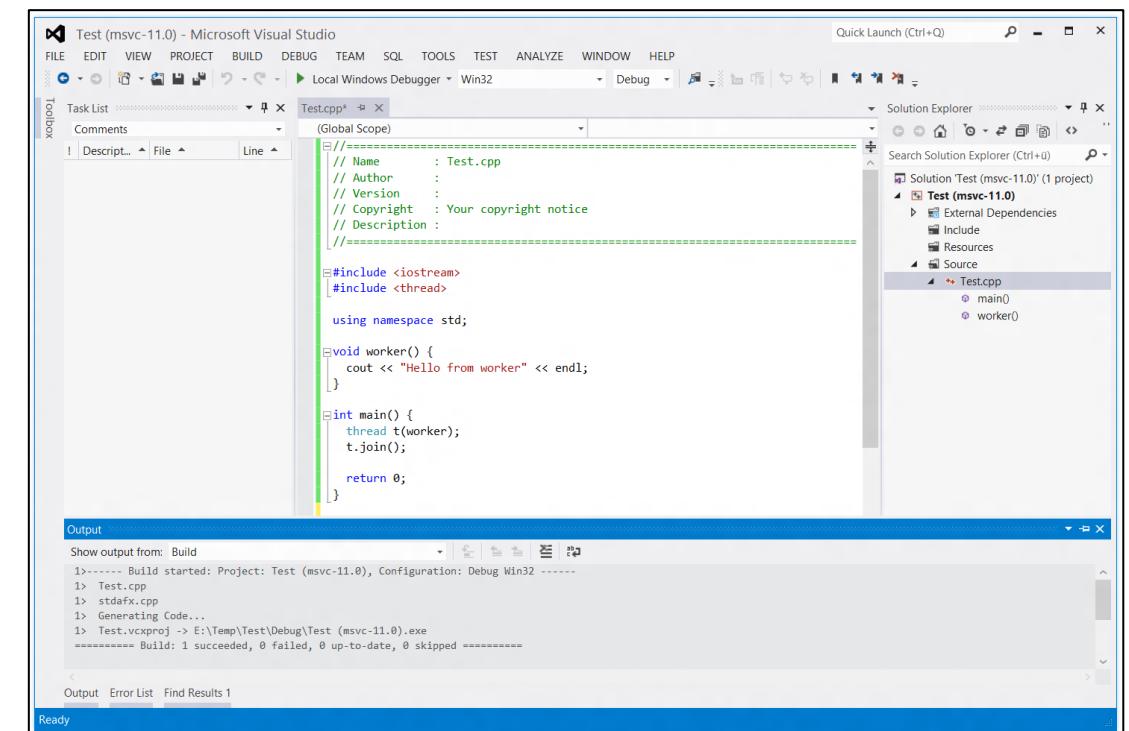
- Integrated Development Environment (IDE) for R
- Uses R language to build applications
- Combines multiple developer tools into a single graphical user interface (GUI)
- Requires R to properly operate



RStudio is similar to Visual Studio



Visual Studio Languages: C/C++, Java, Python etc.



<https://stackoverflow.com/questions/13355656/imitating-the-look-and-feel-of-visual-studio-2012>

Why code R?

- Microsoft Excel or LibreOffice type programs are very slow at handling 100,000+ data points.
- Fancy data editing and visualization programs are often not free.
 - Power BI
 - Tableau
 - Prism - GraphPad
- You can easily perform statistical analyses and make cool plots!
- If you know one programming language, you can easily learn another language (i.e. Python or BASH)

Example analyses (BASH vs. R)



Genome sequencing read processing

- Read size filtering (trimmomatic)
- Primer trimming (cutadapt)
- Adapter / barcode trimming (cutadapt)
- Align reads to a reference (minimap2)

Generate a table of reads mapped

- Total number of ambiguous bases
- Identify total number of mutations



Metadata transformation (tidyverse)

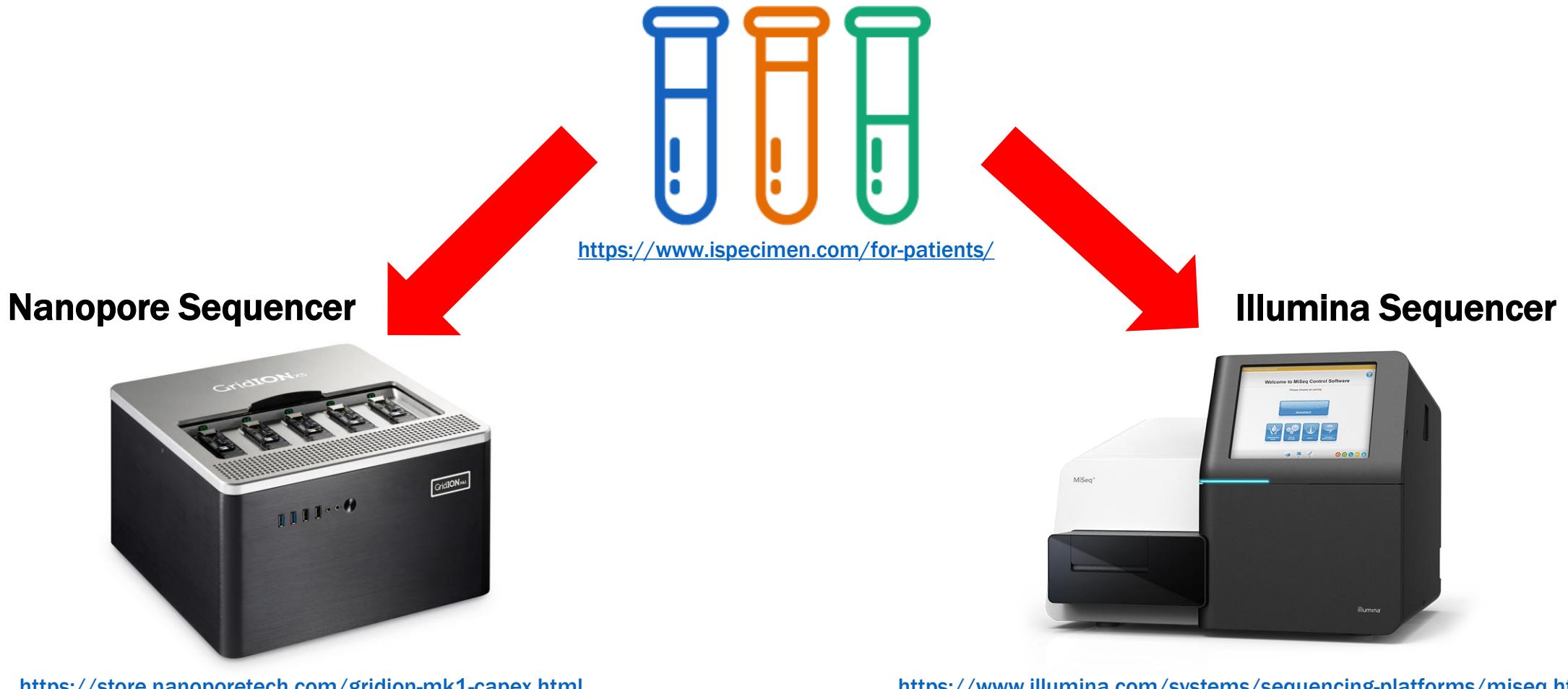
- Group patient by age
 - Group results by collection date
- Combine sequencing results into metadata

Statistical analyses & Visualization

- Perform statistics
- Generate plots to summarize results (ggplot2)

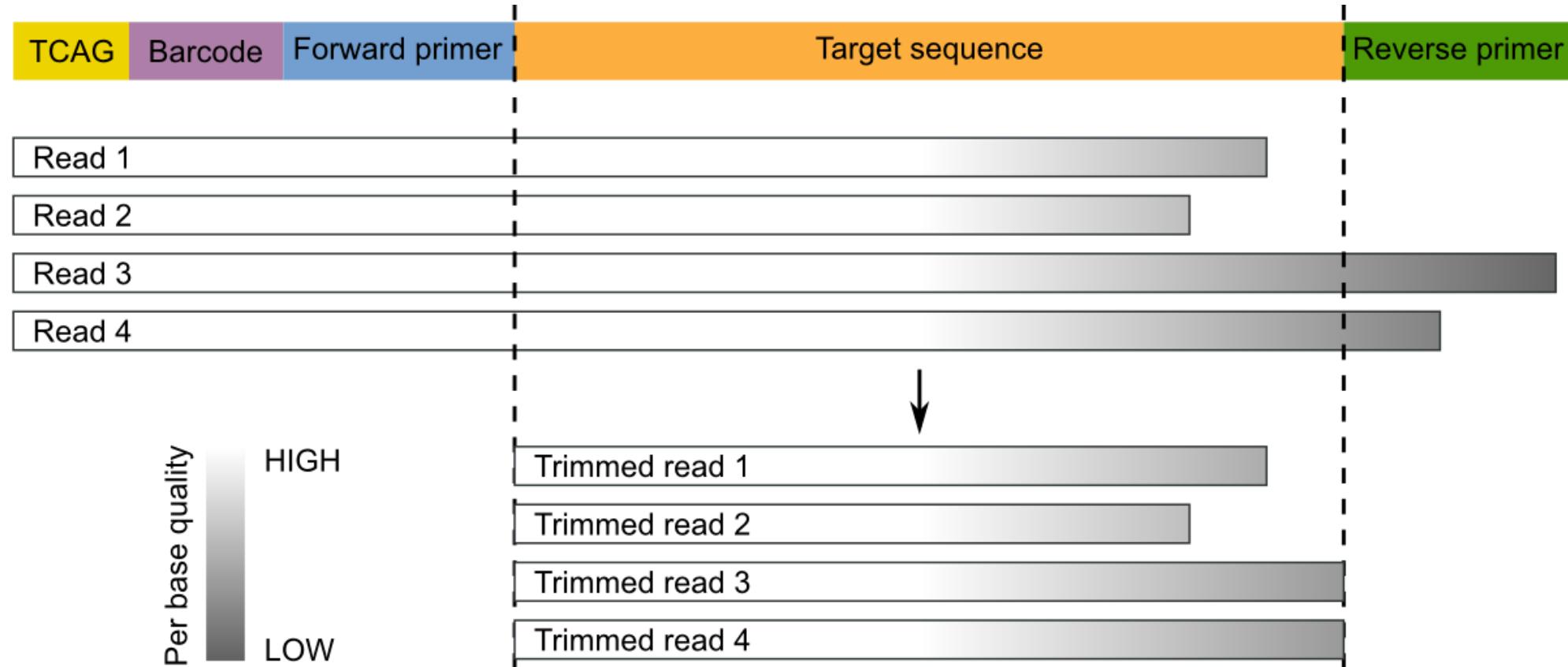
Example bioinformatics workflow:

1. Genome sequencing (Nanopore/Illumina)



Example bioinformatics workflow:

2. Trimming adapter or barcodes (i.e. cutadapt)



<https://micca.readthedocs.io/en/latest/singleend.html>

Example bioinformatics workflow:

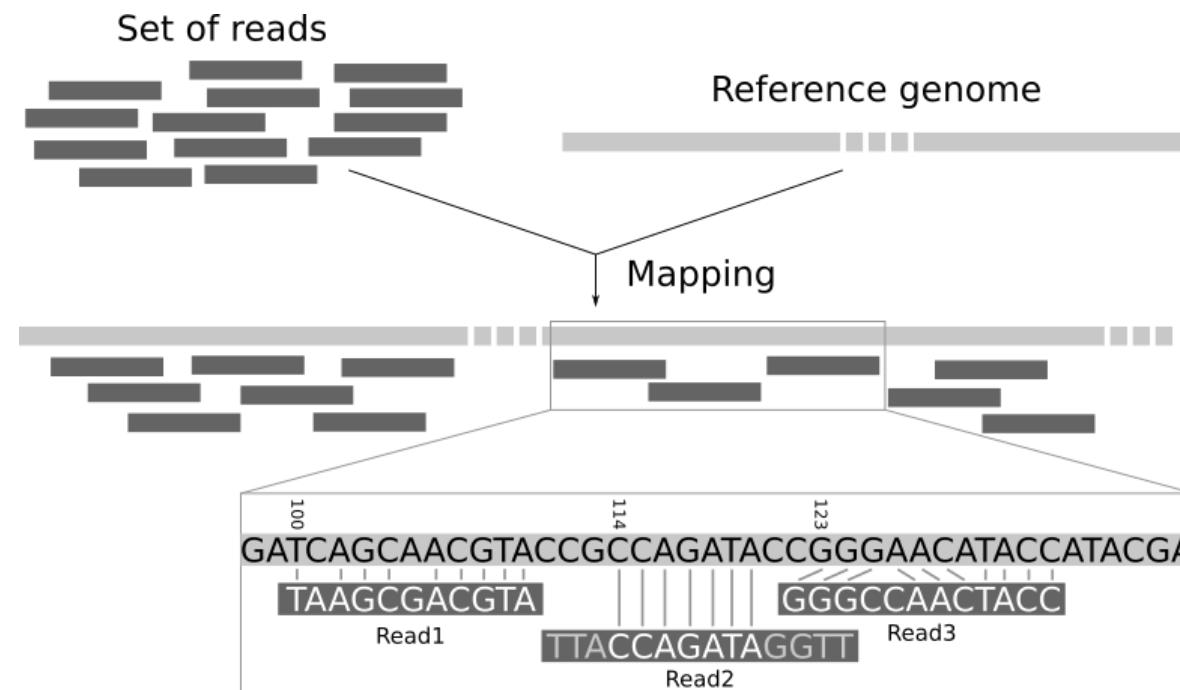
3. Map reads to a reference (i.e. minimap2)

Watch: Jill Rumore's Introduction to CONDA workshop:

<https://www.youtube.com/watch?v=r-wmmhjHFJ4> -

Watch: Aaron Petkau's Downloading and assembling microbial sequence data workshop

<https://www.youtube.com/watch?v=qa0kqE8BIVQ>



<https://training.galaxyproject.org/training-material/topics/sequence-analysis/tutorials/mapping/tutorial.html>

Example bioinformatics workflow:

4. Obtain results

Sample ID	Gross mapping yield (Gb)	Read Depth	Fully called genome fraction	SNP total count	INS total count	DEL total count	SUB total count	Total CNV segment count
A357-N	193.538	66.44	0.971	3492002	295890	299955	95653	246
A357-TT	185.055	64.19	0.961	3486488	272954	274526	98051	330
A357-TA	189.624	66.12	0.956	3457971	277212	280823	101310	360
A355-N	194.519	66.57	0.974	3469773	303858	302061	94328	251
A355-TA	184.075	63.32	0.969	3342011	280658	279064	92470	409
A355-TT	290.237	100.78	0.96	3475951	305251	300437	107964	362
A368-N	187.147	64.78	0.963	3460780	279997	286541	99381	243
A368-TT	296.613	103.31	0.957	3473247	303717	291929	111119	390
A368-TA	173.431	59.97	0.964	3470317	270147	270543	95311	529

https://www.researchgate.net/figure/A-summary-statistics-of-whole-genome-sequencing-results_tbl1_314247140

Example bioinformatics workflow:

5. Combine and transform data (i.e. tidyverse)

February 9 Molly Pratt

Tidy data: combining and transforming data in R

Sample ID	Gross mapping yield (Gb)	Read Depth	Fully called genome fraction	SNP total count	INS total count	DEL total count	SUB total count	Total CNV segment count
A357-N	193.538	66.44	0.971	3492002	295890	299955	95653	246
A357-TT	185.055	64.19	0.961	3486488	272954	274526	98051	330
A357-TA	189.624	66.12	0.956	3457971	277212	280823	101310	360
A355-N	194.519	66.57	0.974	3469773	303858	302061	94328	251
A355-TA	184.075	63.32	0.969	3342011	280658	279064	92470	409
A355-TT	290.237	100.78	0.96	3475951	305251	300437	107964	362
A368-N	187.147	64.78	0.963	3460780	279997	286541	99381	243
A368-TT	296.613	103.31	0.957	3473247	303717	291929	111119	390
A368-TA	173.431	59.97	0.964	3470317	270147	270543	95311	529

https://www.researchgate.net/figure/A-summary-statistics-of-whole-genome-sequencing-results_tbl1_314247140

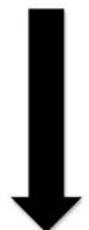
A	B	C	D	E	F	G	H
1 Sample ID	Subject ID	Ethnicity	Gender	DiseaseState	Tissue	DiseaseStage	
2 848978_001	001M	No Info	male	bipolar disorder	dorsolateral prefrontal cortex	active disease	
3 848978_002	002M	Caucasian	male	bipolar disorder	dorsolateral prefrontal cortex	active disease	
4 848978_010	010M	Caucasian	male	bipolar disorder	dorsolateral prefrontal cortex	active disease	
5 848978_021	021M	Caucasian	male	bipolar disorder	dorsolateral prefrontal cortex	active disease	
6 848978_004	004F	Caucasian	female	bipolar disorder	dorsolateral prefrontal cortex	active disease	
7 848978_005	005F	Caucasian	female	bipolar disorder	dorsolateral prefrontal cortex	active disease	
8							
9							
10							

http://www.arrayserver.com/wiki/index.php?title=Sample_MetaData_File



Log data

Patient	Date	Item	quantity
a	2011/10/01	Drug-a	2
a	2011/10/01	Drug-b	3
a	2011/10/02	Drug-a	2
b	2011/10/01	Drug-b	3
b	2011/10/02	Drug-c	4



Patient data

Patient	Drug-a			Drug-b		
	Yes/No	quantity	days	Yes/No	quantity	days
a	Yes	4	2	Yes	3	1
b	No			Yes	3	1

Example bioinformatics workflow:

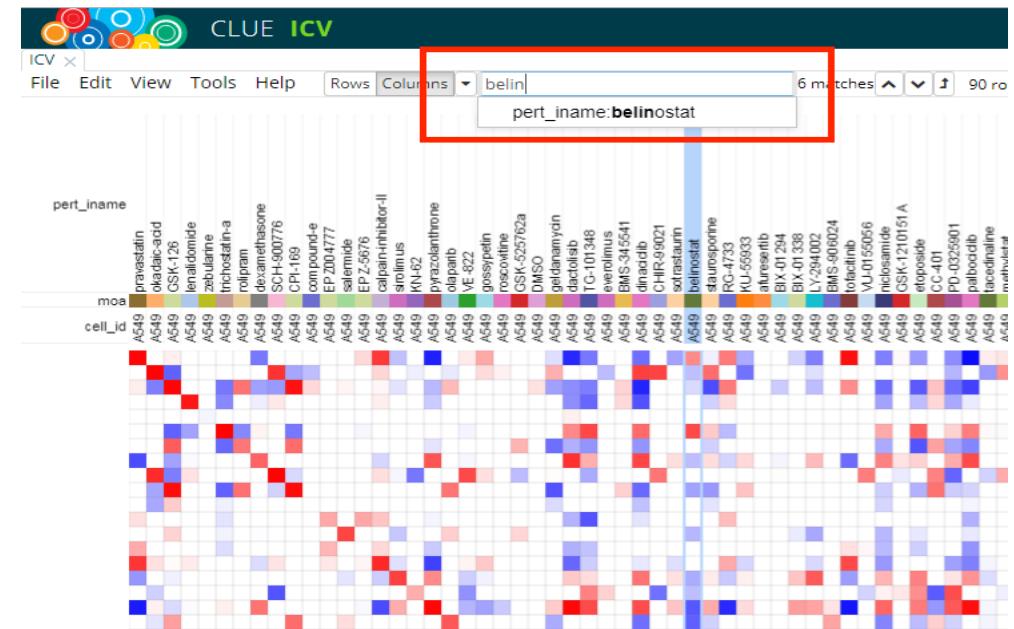
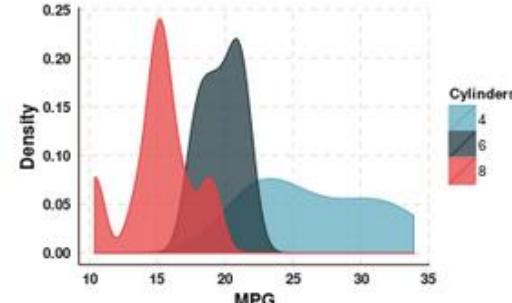
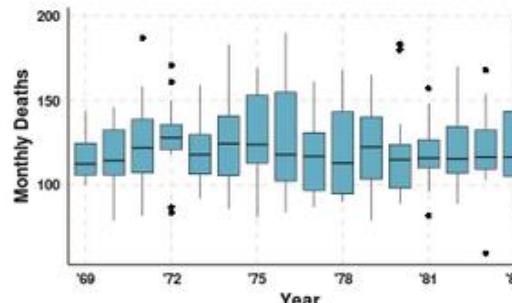
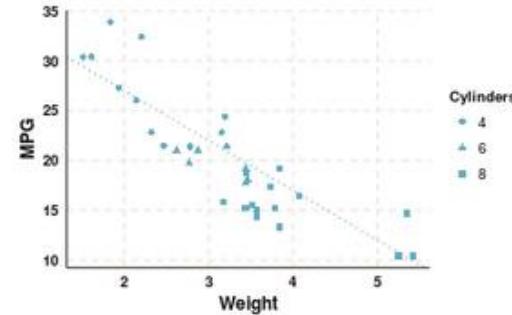
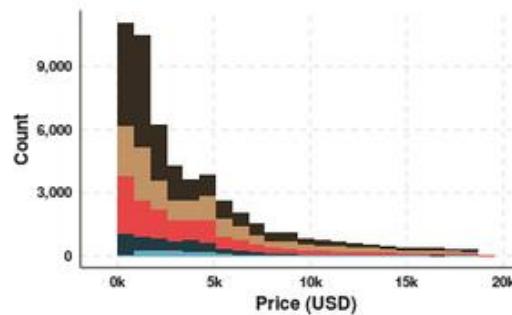
6. Data visualization: ggplot2

February 16 Samantha Lee

Using ggplot to visualize data and statistical results

February 23 Mackenzie Wilke

Data visualization using antiviral drug-repurposing results from CLUE



https://twitter.com/shane_a_lynn/status/499991316122849282

<https://clue.io/proteomics-tutorial>

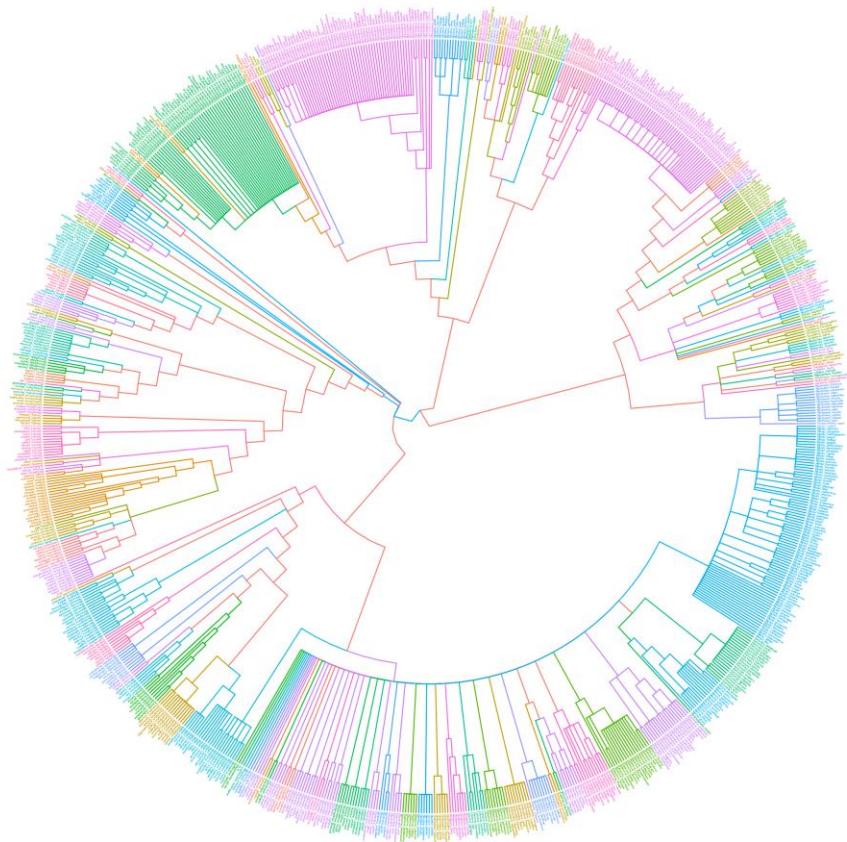
Example bioinformatics workflow:

6. Data visualization: ggtree

March 2

Taylor Davedow

Data visualization using ggtree



List of other visualization packages in R:

<https://mode.com/blog/r-ggplot-extension-packages/>

- gganimate
- ggdendro
- ggthemes
- plotly
- ggridges
- ggcorrplot

<https://www.molecular ecologist.com/2017/02/08/phylogenetic-trees-in-r-using-ggtree/>

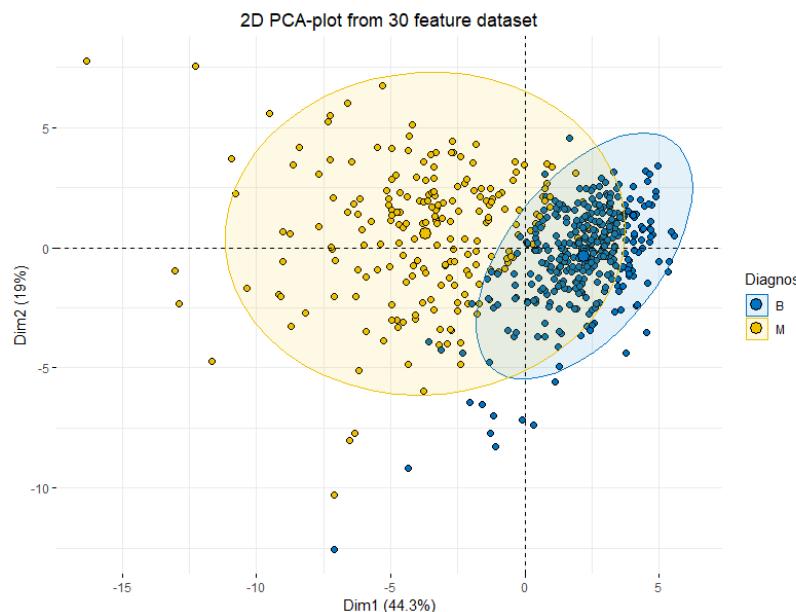
Example bioinformatics workflow:

7. RNA-seq analysis and visualization: DESeq2/edgeR

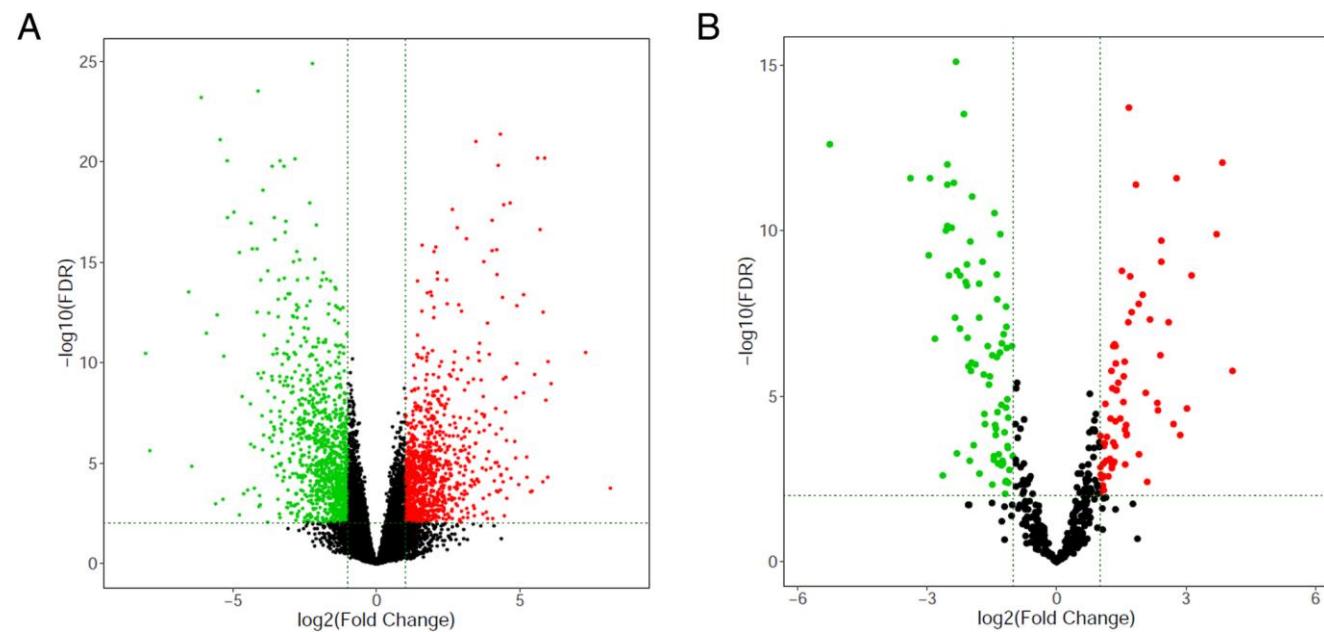
March 9

Jessy Slota

RNA-seq data analysis in R



<https://towardsdatascience.com/principal-component-analysis-pca-101-using-r-361f4c53a9ff>

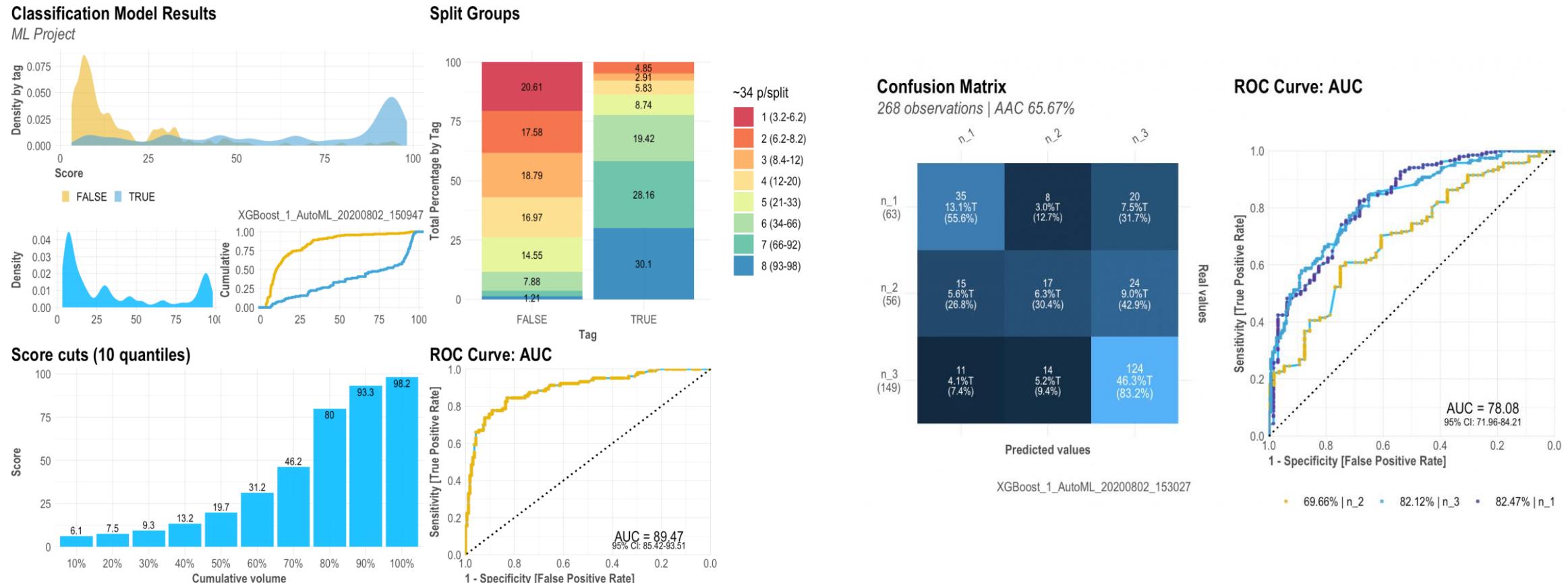


<https://bmccancer.biomedcentral.com/articles/10.1186/s12885-019-5983-8/figures/2>

Example bioinformatics workflow:

7. Machine learning and visualization

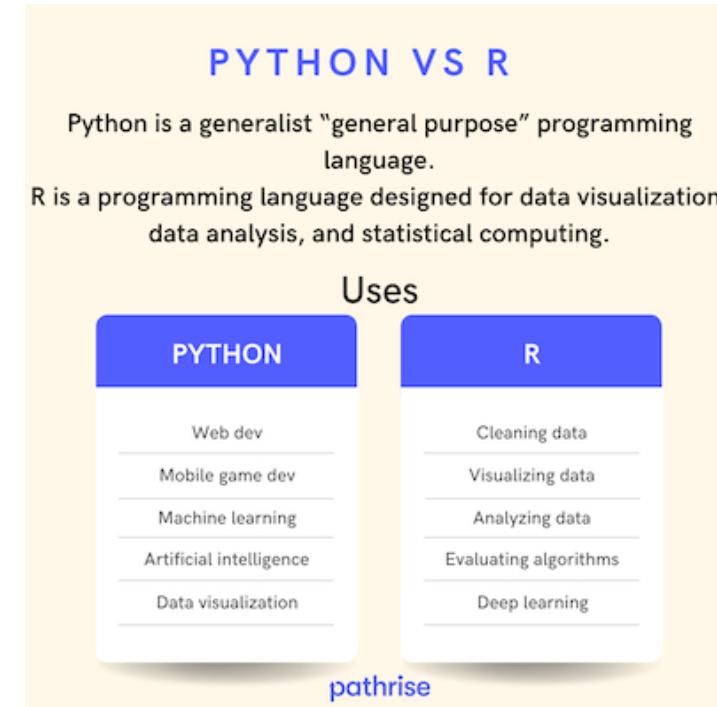
March 16 Vasena Jayamanna Introduction to machine learning in R



<https://datascienceplus.com/machine-learning-results-one-plot-to-rule-them-all/>

Python or R? Your choice!

- Python is a programming language used for software development.
- Both R and Python can perform statistical analyses and data visualization.
- R is more specific for statistical analyses and visualization



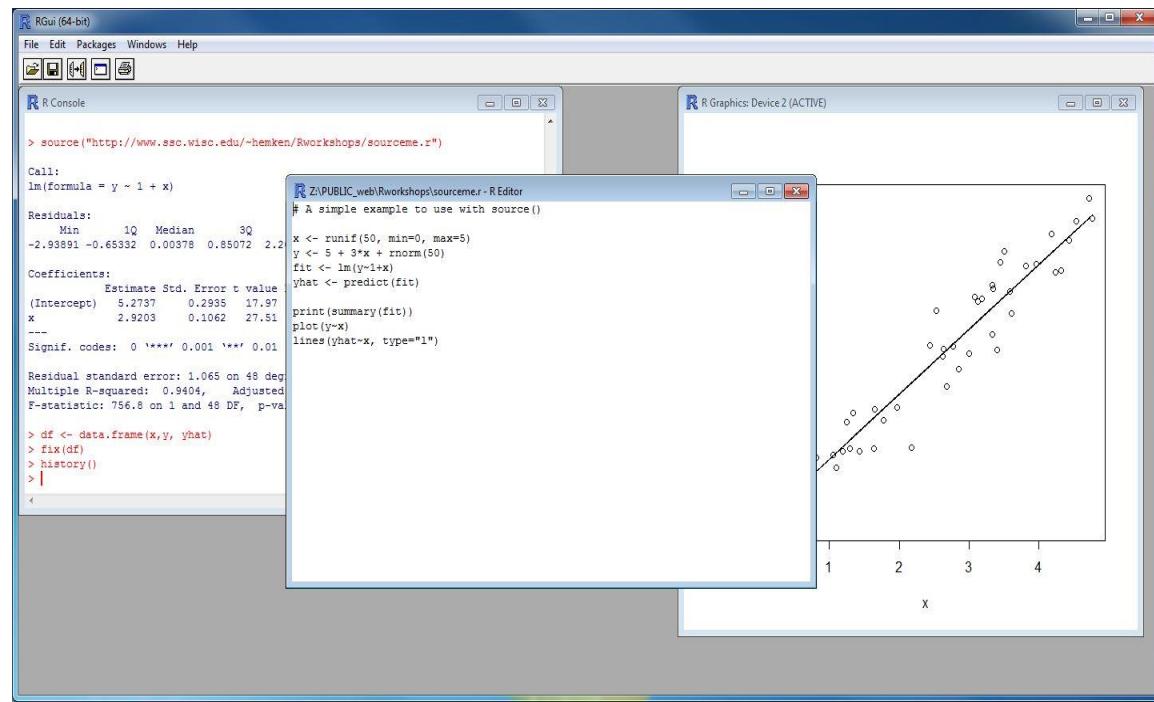
<https://www.pathrise.com/guides/python-vs-r-data-science-languages-to-master/>

Part 2. Installing R and RStudio

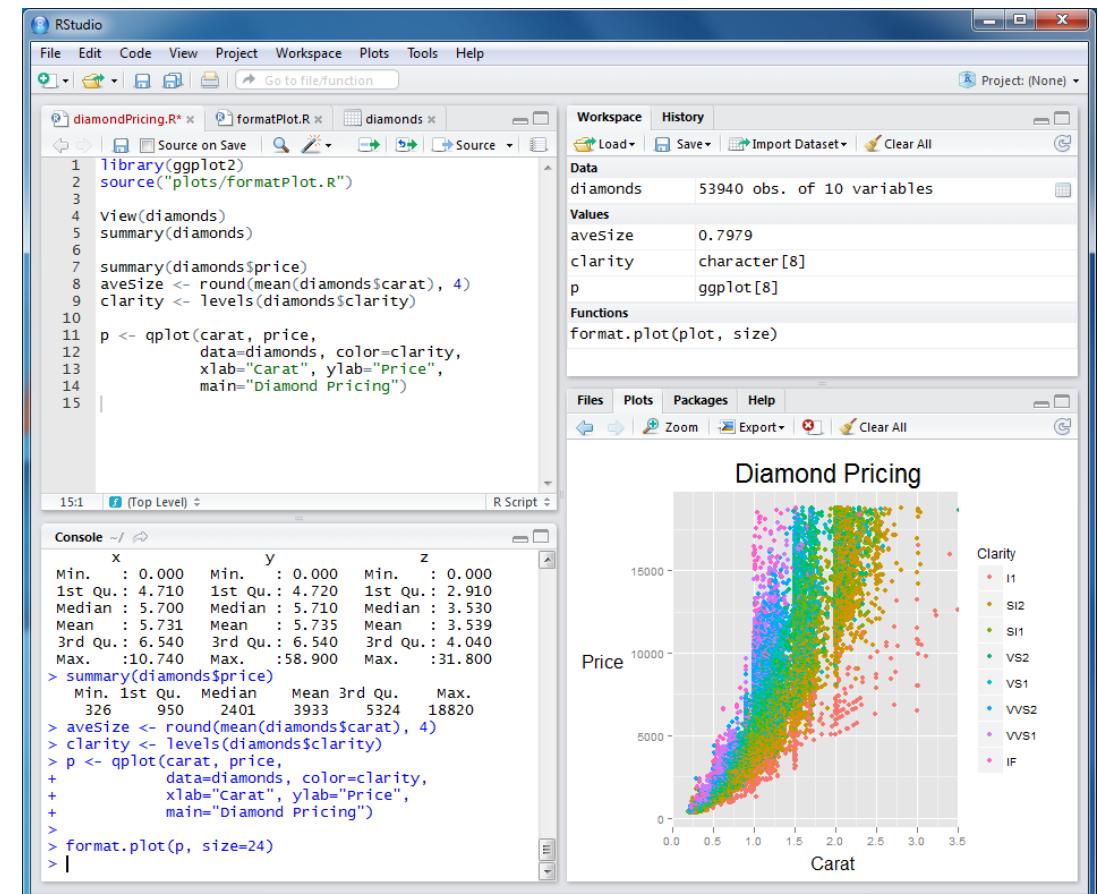
LEARNING OBJECTIVES

- 1. *Describe types of analyses performed using R or BASH.***
- 2. *Install R and RStudio on your operating system.***
- 3. *Learn how to customize RStudio.***
- 4. *Navigate, create, rename, move, copy directory and file in R.***
- 5. *Install packages, load libraries and write a simple R script.***
- 6. *Write a simple R markdown.***

R and RStudio



<https://www.ssc.wisc.edu/~hemken/Rworkshops/interface/RConsole.html>



<http://ds4ps.org/dp4ss-textbook/ch-020-rstudio.html>

Getting started in R

The R Project for Statistical Computing: <https://www.r-project.org/>

RStudio: <https://www.rstudio.com/>

Beginner's online resources to get started in R

- R for Data Science: <https://r4ds.had.co.nz/>
- Introduction to R 2021: https://biocorecrg.github.io/CRG_RIntroduction/index.html
- Introduction to Data Exploration and Analysis with R:
<https://bookdown.org/mikemahoney218/IDEAR/>
- Computational Genomics with R: <https://compgenomr.github.io/book/>

Part 2. Installing R and RStudio on Windows or Mac OS

Guide video for Windows 10:

<https://www.youtube.com/watch?v=NZxSA80lF1I>

Guide video for Mac:

<https://www.youtube.com/watch?v=LanBozXJjOk>

Select CRAN network

1. Select CRAN (Comprehensive R Archive Network) close to you

<https://cran.r-project.org/mirrors.html>

The screenshot shows a web browser displaying the CRAN mirror list at <https://cran.r-project.org/mirrors.html>. The page lists mirrors categorized by country. A red box highlights the 'Canada' section, which contains links to mirrors located at Simon Fraser University, Manitoba Unix User Group, University of Toronto, DigitalOcean, and University of Waterloo. Other sections visible include 'Bulgaria' (with a link to Sofia University), 'Chile' (with a link to Departamento de Ciencias de la Computación, Universidad de Chile), and 'University of Sao Paulo' (with links to São Paulo and Piracicaba). The browser interface includes a back/forward button, address bar, and various icons.

Country	Mirror URL	Description
Bulgaria	https://vps.fmvz.usp.br/CRAN/	University of São Paulo, São Paulo
Bulgaria	https://brieger.esalq.usp.br/CRAN/	University of São Paulo, Piracicaba
Bulgaria	https://ftp.uni-sofia.bg/CRAN/	Sofia University
Canada	https://mirror.rcg.sfu.ca/mirror/CRAN/	Simon Fraser University, Burnaby
Canada	https://muug.ca/mirror/cran/	Manitoba Unix User Group
Canada	https://utstat.toronto.edu/cran/	University of Toronto
Canada	https://cran.pacha.dev/	DigitalOcean
Canada	https://mirror.csclub.uwaterloo.ca/CRAN/	University of Waterloo
Chile	https://cran.dcc.uchile.cl/	Departamento de Ciencias de la Computación, Universidad de Chile
University of Sao Paulo	https://vps.fmvz.usp.br/CRAN/	University of São Paulo, São Paulo
University of Sao Paulo	https://brieger.esalq.usp.br/CRAN/	University of São Paulo, Piracicaba

Download and install base R

2. Click “Download R for ___” the version that matches your operating system.

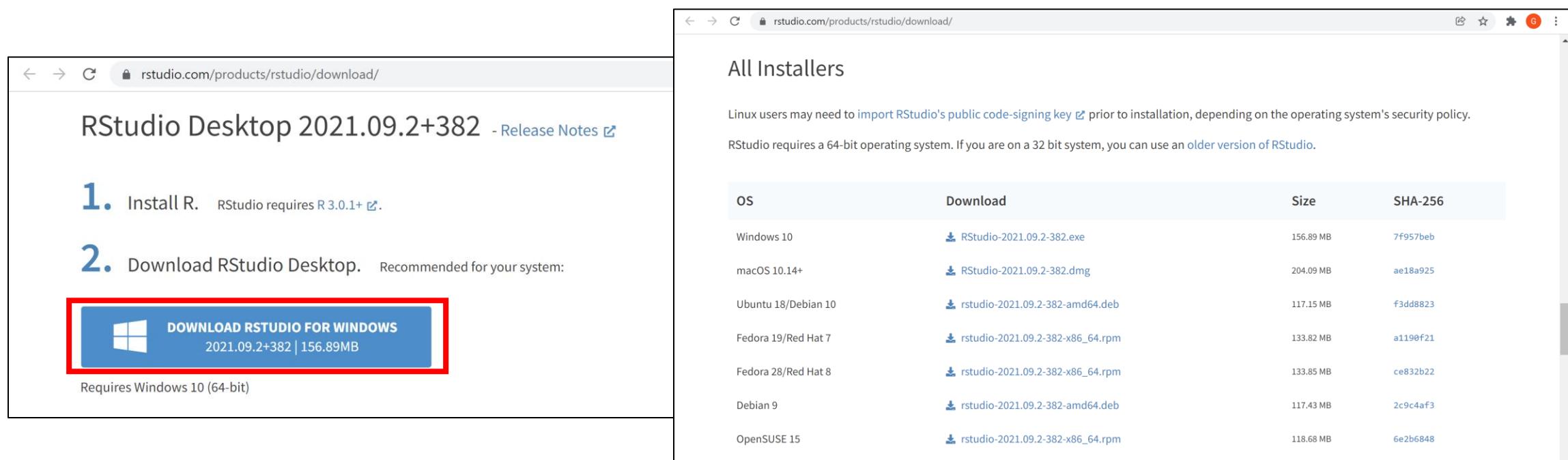
3. After downloading, double-click application and install using default options.

The screenshot shows a web browser displaying the CRAN (Comprehensive R Archive Network) website at muug.ca/mirror/cran/. The page features the R logo and navigation links for CRAN Mirrors, What's new?, Task Views, and Search. A large heading "The Comprehensive R Archive Network" is centered above two main download sections. The top section, "Download and Install R", is highlighted with a red box and contains text about precompiled binary distributions for Windows and Mac users, followed by a bulleted list of download links for Linux (Debian, Fedora/Redhat, Ubuntu), macOS, and Windows. Below this is a note about Linux package management. The bottom section, "Source Code for all Platforms", contains text for Windows and Mac users about downloading precompiled binaries instead of source code, which requires compilation. The browser interface includes a back/forward button, address bar, and various icons.

Download and install RStudio

4. Download RStudio for your operating system and install using default options.
<https://www.rstudio.com/products/rstudio/download/>

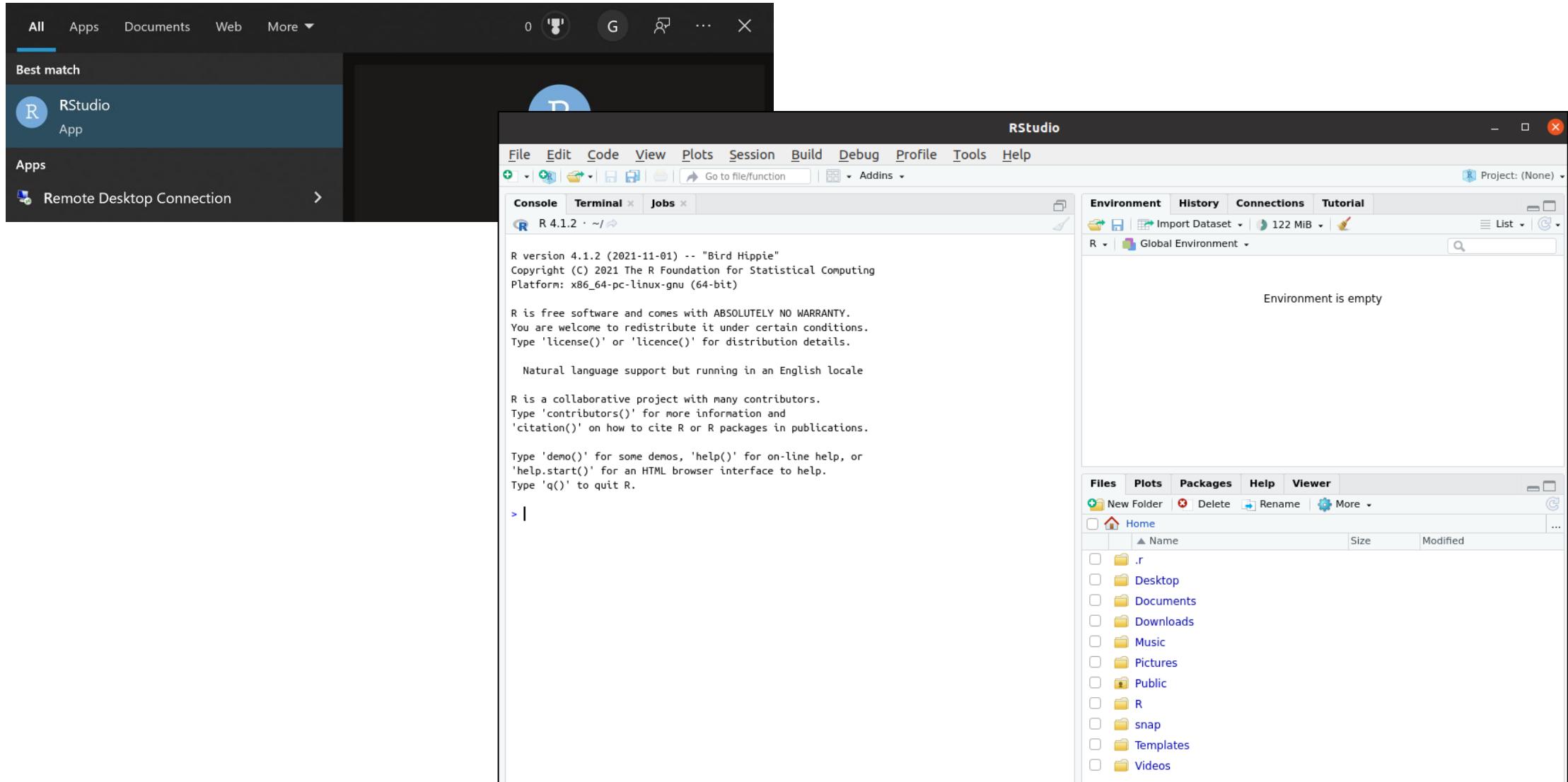
NOTE: if you scroll down, your OS will automatically get recognized for download



The image shows two screenshots of the RStudio download page. The left screenshot shows the main landing page for RStudio Desktop 2021.09.2+382, with step 2 highlighted and the Windows download button boxed in red. The right screenshot shows the 'All Installers' page where the OS column lists various Linux distributions and the 'Download' column lists the corresponding RPM or DEB files.

OS	Download	Size	SHA-256
Windows 10	RStudio-2021.09.2-382.exe	156.89 MB	7f957beb
macOS 10.14+	RStudio-2021.09.2-382.dmg	204.09 MB	ae18a925
Ubuntu 18/Debian 10	rstudio-2021.09.2-382-amd64.deb	117.15 MB	f3dd8823
Fedora 19/Red Hat 7	rstudio-2021.09.2-382-x86_64.rpm	133.82 MB	a1190f21
Fedora 28/Red Hat 8	rstudio-2021.09.2-382-x86_64.rpm	133.85 MB	ce832b22
Debian 9	rstudio-2021.09.2-382-amd64.deb	117.43 MB	2c9c4af3
OpenSUSE 15	rstudio-2021.09.2-382-x86_64.rpm	118.68 MB	6e2b6848

Search and open RStudio application



Part 2. Installing R and RStudio on Linux OS

Guide video for Linux:

<https://www.youtube.com/watch?v=WpNd6j1nvEE>

Install R in terminal

1. Perform overall update and then install r-base

```
$ sudo apt-get update  
$ sudo apt-get -y install r-base
```

OPTIONS

-y → Do not ask for confirmation
-n → Name

Program installation guide: https://github.com/MMID-coding-workshop/2022-01-19-Introduction-to-CONDA/blob/main/MMID_Coding_Workshop-IntroToConda_2022-01-19-PowerPoint.pdf

2. You can now open R in terminal

```
$ R #open R console method 1
```

or

```
$ sudo -i R #open R console method 2
```

```
> #This is R prompt
```

```
> q() #quit R console
```

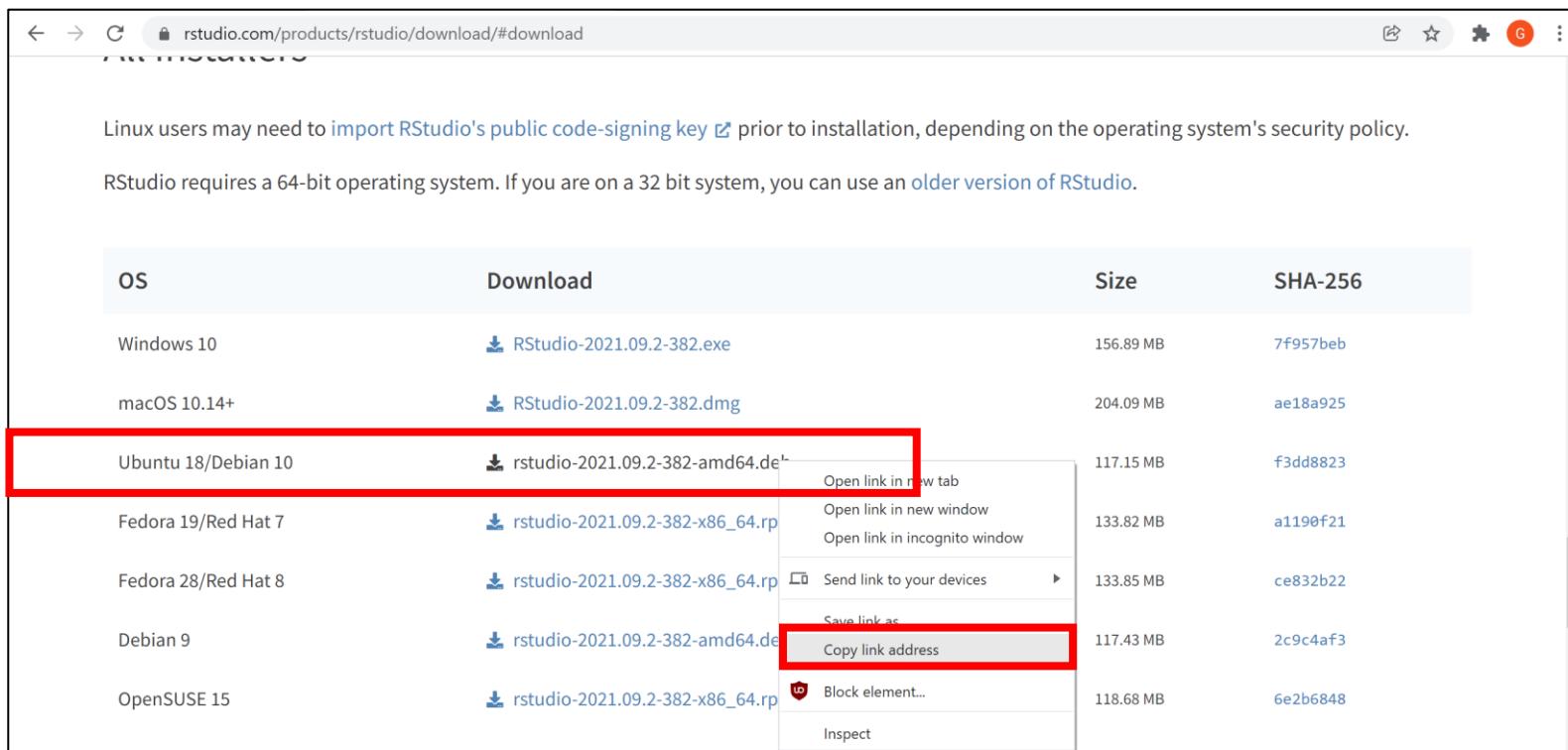
```
seog@seog-VirtualBox:~$ sudo -i R  
  
R version 4.1.2 (2021-11-01) -- "Bird Hippie"  
Copyright (C) 2021 The R Foundation for Statistical Computing  
Platform: x86_64-pc-linux-gnu (64-bit)  
  
R is free software and comes with ABSOLUTELY NO WARRANTY.  
You are welcome to redistribute it under certain 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.  
> q()  
Save workspace image? [y/n/c]: n  
seog@seog-VirtualBox:~$
```

Search for correct RStudio version

From this link, search for the correct OS version of RStudio.

<https://www.rstudio.com/products/rstudio/download/#download>

→ Right click on the file → Copy link address



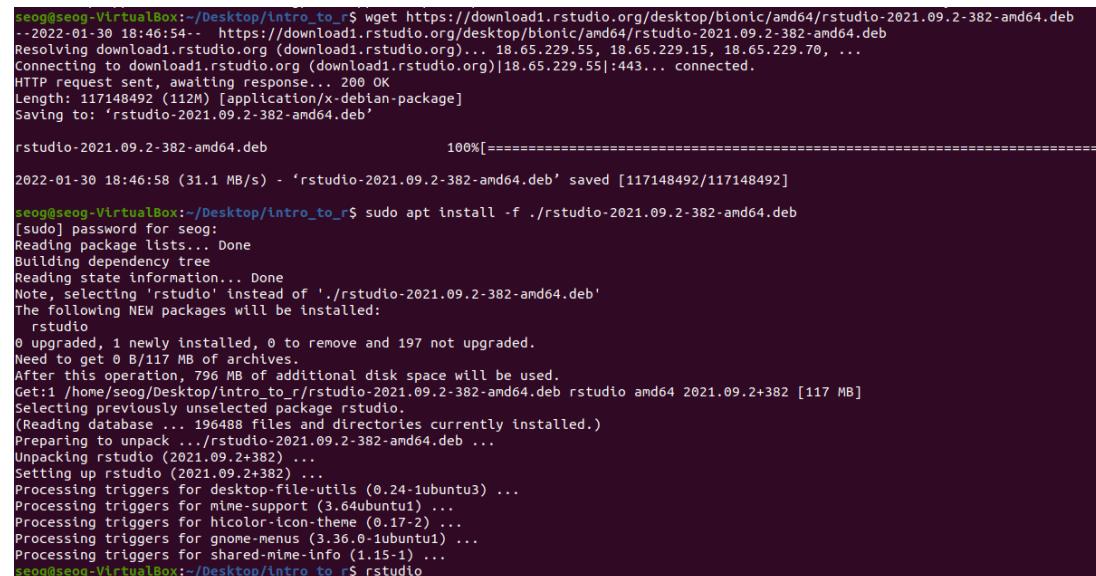
Download and install RStudio in terminal

1. Download RStudio using the copied link

```
$ wget https://download1.rstudio.org/desktop/bionic/amd64/rstudio-2021.09.2-382-amd64.deb
```

2. Install RStudio using the downloaded file in your current directory

```
$ sudo apt install -f ./rstudio-2021.09.2-382-amd64.deb
```



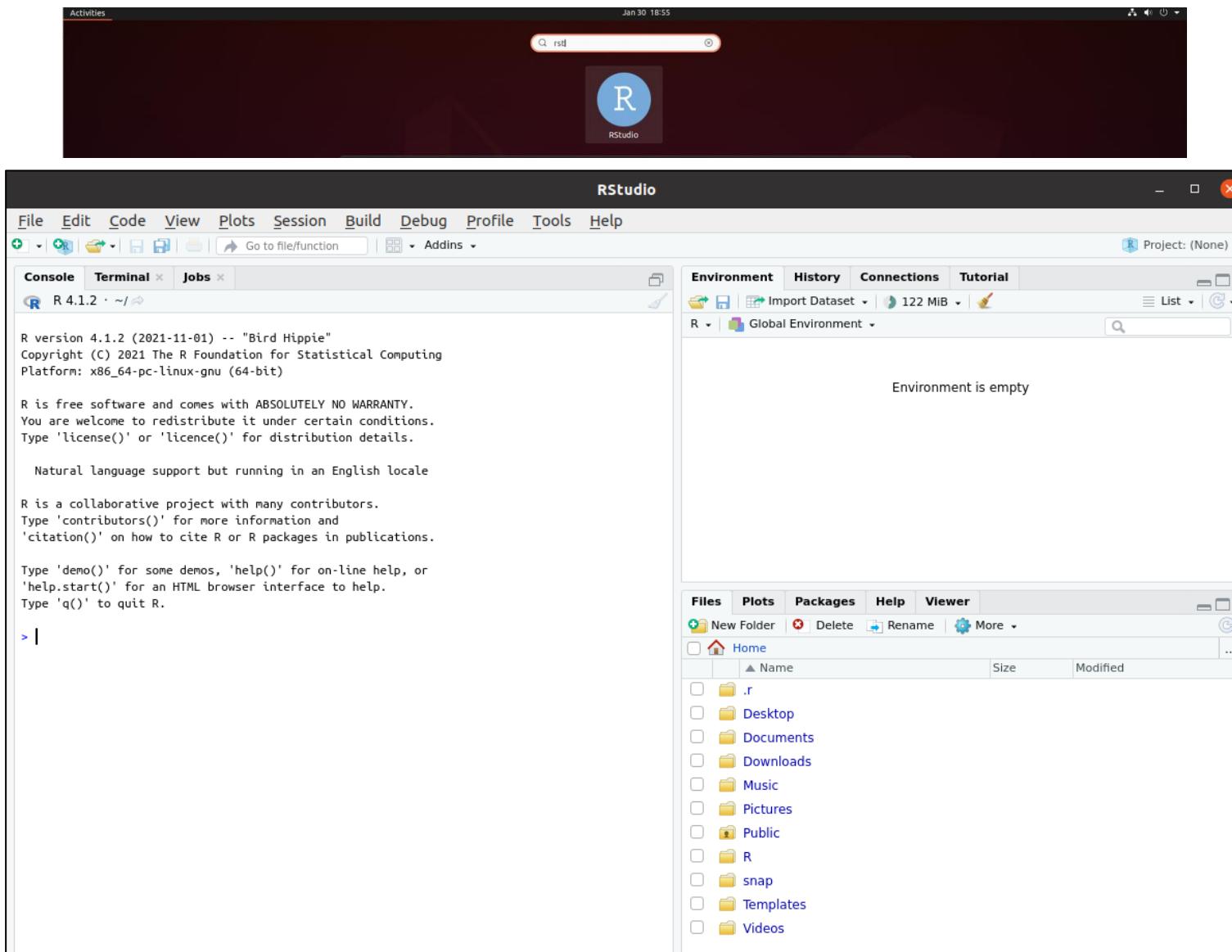
A terminal window showing the download and installation of RStudio. The terminal output is as follows:

```
seog@seog-VirtualBox:~/Desktop/intro_to_r$ wget https://download1.rstudio.org/desktop/bionic/amd64/rstudio-2021.09.2-382-amd64.deb
--2022-01-30 18:46:54-- https://download1.rstudio.org/desktop/bionic/amd64/rstudio-2021.09.2-382-amd64.deb
Resolving download1.rstudio.org (download1.rstudio.org)... 18.65.229.55, 18.65.229.15, 18.65.229.70, ...
Connecting to download1.rstudio.org (download1.rstudio.org)|18.65.229.55|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 117148492 (112M) [application/x-debian-package]
Saving to: 'rstudio-2021.09.2-382-amd64.deb'

rstudio-2021.09.2-382-amd64.deb          100%[=====] 31.1 MB/s   2022-01-30 18:46:58 (31.1 MB/s) - 'rstudio-2021.09.2-382-amd64.deb' saved [117148492/117148492]

seog@seog-VirtualBox:~/Desktop/intro_to_r$ sudo apt install -f ./rstudio-2021.09.2-382-amd64.deb
[sudo] password for seog:
Reading package lists... Done
Building dependency tree
Reading state information... Done
Note, selecting 'rstudio' instead of './rstudio-2021.09.2-382-amd64.deb'
The following NEW packages will be installed:
  rstudio
0 upgraded, 1 newly installed, 0 to remove and 197 not upgraded.
Need to get 0 B/117 MB of archives.
After this operation, 796 MB of additional disk space will be used.
Get:1 /home/seog/Desktop/intro_to_r/rstudio-2021.09.2-382-amd64.deb rstudio amd64 2021.09.2+382 [117 MB]
Selecting previously unselected package rstudio.
(Reading database ... 196488 files and directories currently installed.)
Preparing to unpack .../rstudio-2021.09.2-382-amd64.deb ...
Unpacking rstudio (2021.09.2+382) ...
Setting up rstudio (2021.09.2+382) ...
Processing triggers for desktop-file-utils (0.24-1ubuntu3) ...
Processing triggers for mime-support (3.64ubuntu1) ...
Processing triggers for hicolor-icon-theme (0.17-2) ...
Processing triggers for gnome-menus (3.36.0-1ubuntu1) ...
Processing triggers for shared-mime-info (1.15-1) ...
seog@seog-VirtualBox:~/Desktop/intro_to_r$ rstudio
```

Search and open RStudio application

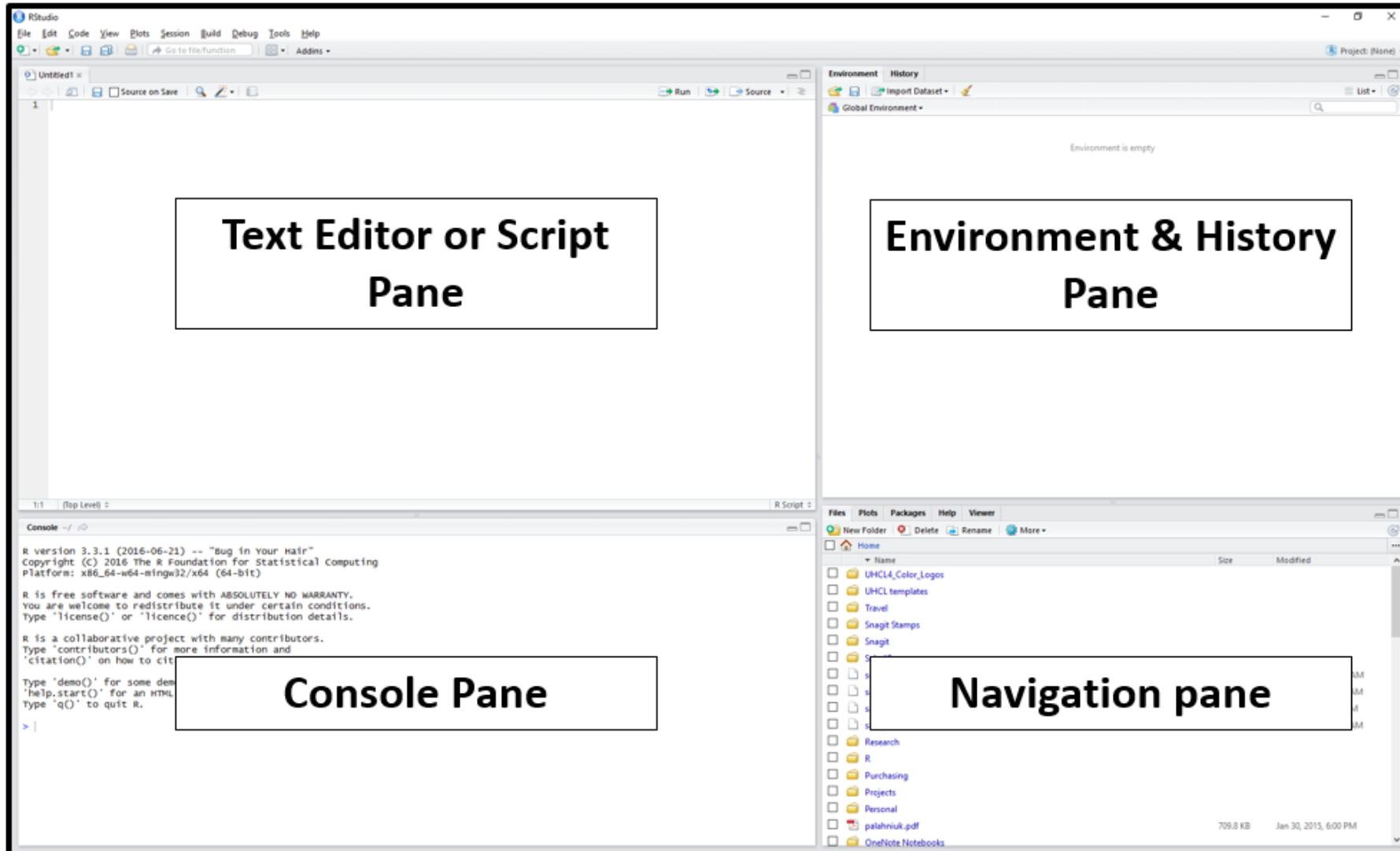


Part 3. Customize RStudio

LEARNING OBJECTIVES

1. *Describe types of analyses performed using R or BASH.*
2. *Install R and RStudio on your operating system.*
3. **Learn how to customize RStudio.**
4. *Navigate, create, rename, move, copy directory and file in R.*
5. *Install packages, load libraries and write a simple R script.*
6. *Write a simple R markdown.*

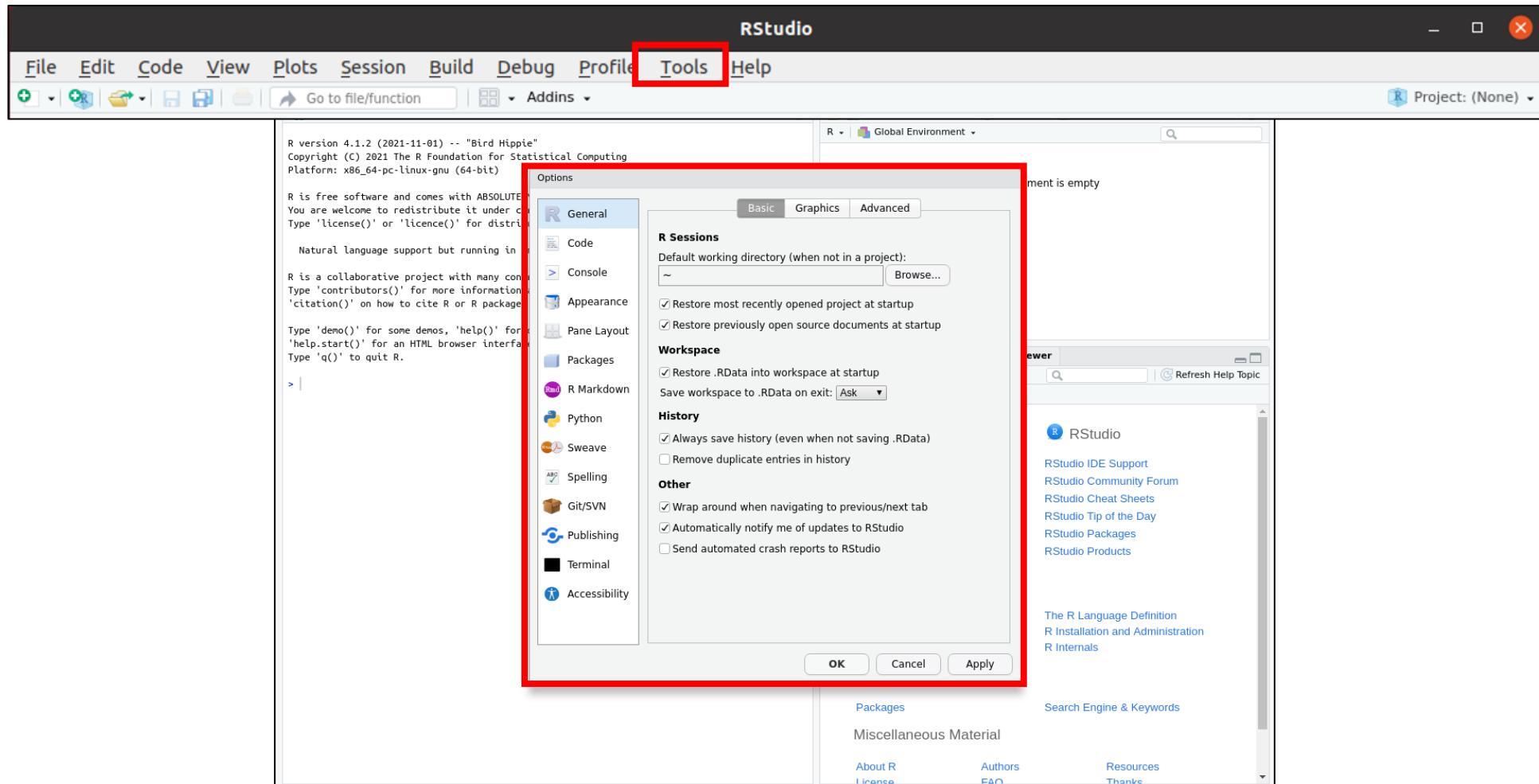
Default RStudio



<https://librarycarpentry.org/lc-r/00-before-we-start/index.html>

R Customization

- From the menu at the top, click → “Tools” → “Global Options...”



R customization

The image shows two side-by-side screenshots of the RStudio Options dialog. Both screenshots have a sidebar on the left listing various sections: General, Code, Console, Appearance, Pane Layout, Packages, R Markdown, Python, Sweave, Spelling, Git/SVN, Publishing, Terminal, and Accessibility. The 'General' section is selected in both.

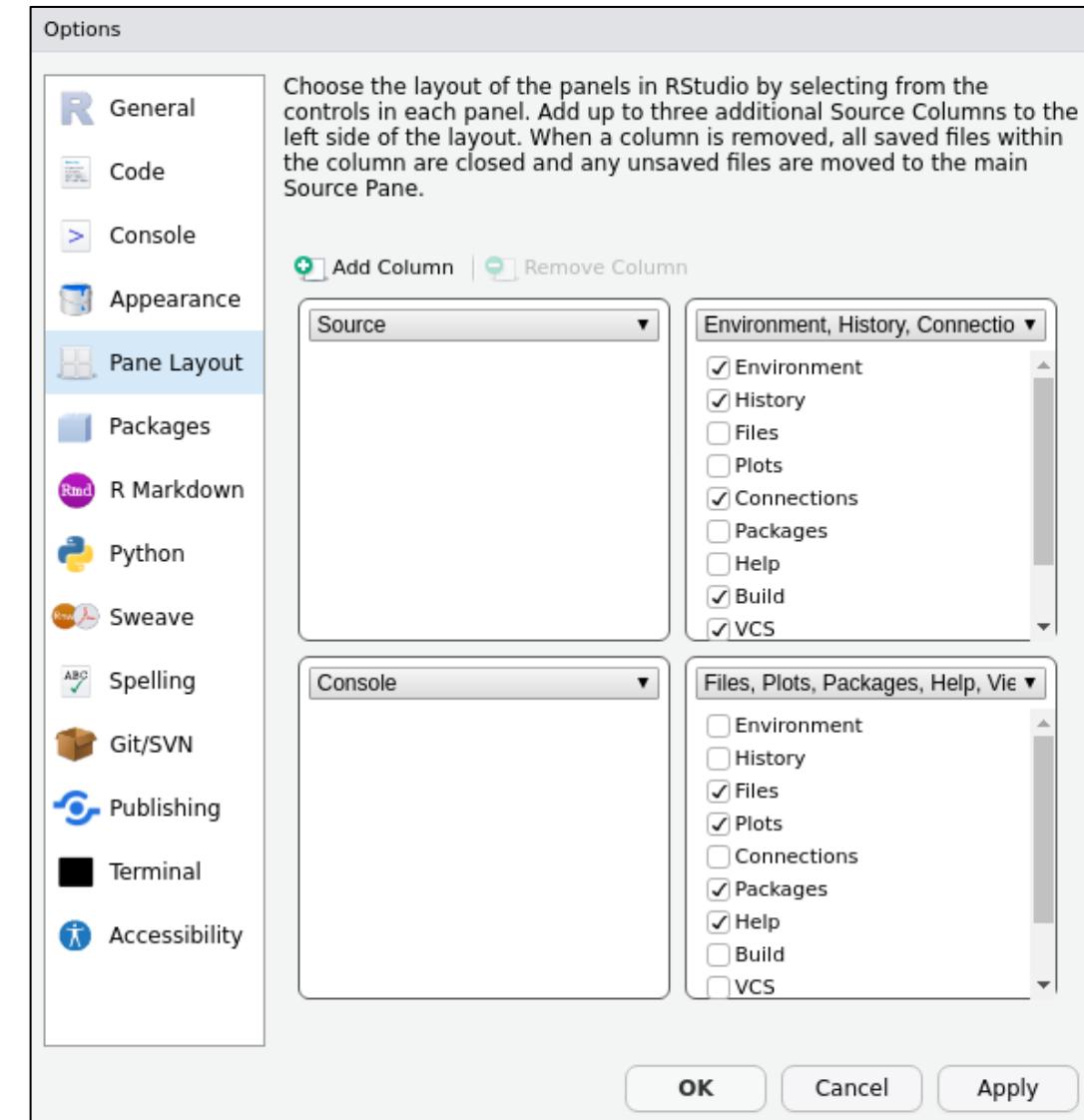
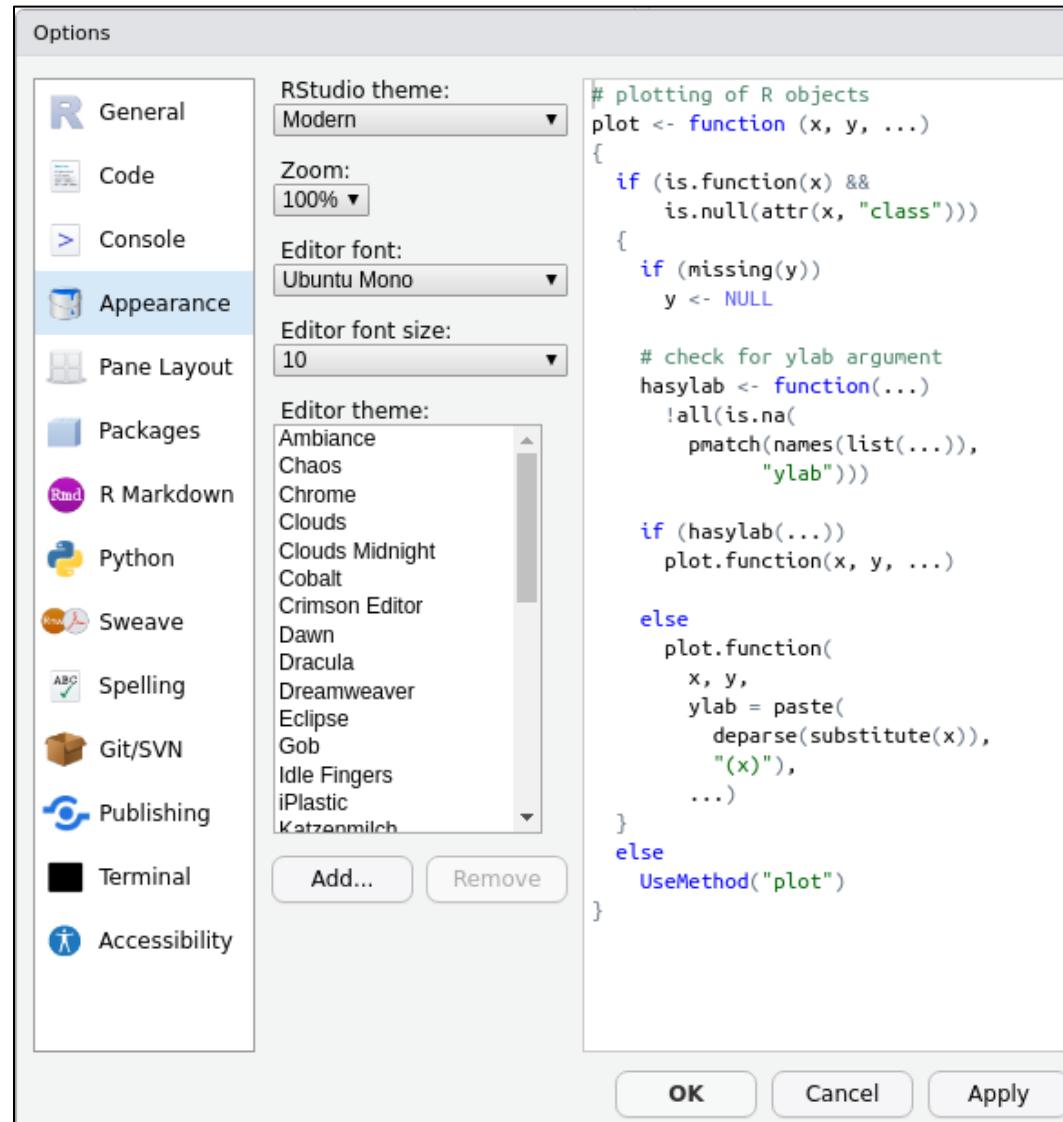
General Tab (Left Screenshot):

- General**
 - Insert spaces for tab
Tab width
 - Auto-detect code indentation
 - Insert matching parens/quotes
 - Use native pipe operator, |> (requires R 4.1+)
 - Auto-indent code after paste
 - Vertically align arguments in auto-indent
 - Soft-wrap R source files
 - Continue comment when inserting new line
 - Enable hyperlink highlighting in editor
- Surround selection on text insertion:
- Keybindings:
- Execution**
 - Focus console after executing from source
 - Ctrl+Enter executes:
- Snippets**
 - Enable code snippets

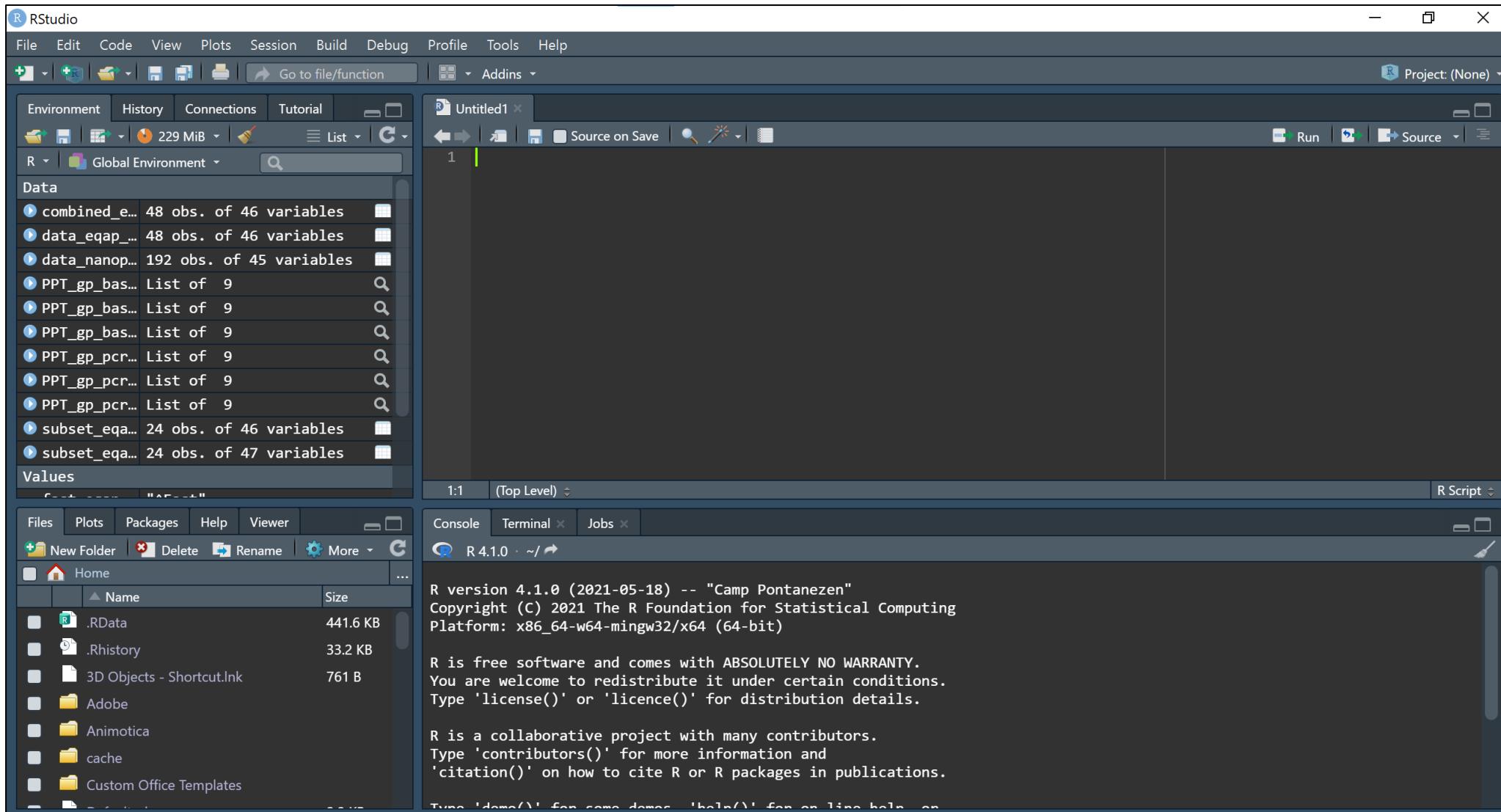
Display Tab (Right Screenshot):

- Display**
 - Show syntax highlighting in console input
 - Different color for error or message output (requires restart)
 - Limit visible console output (requires restart)
- Limit output line length to:
- ANSI Escape Codes:
- Debugging**
 - Automatically expand tracebacks in error inspector
- Other**
 - Double-click to select words

R customization



Example of customization



Part 4. Navigate, create, rename, move, copy directory and file

LEARNING OBJECTIVES

1. *Describe types of analyses performed using R or BASH.*
2. *Install R and RStudio on your operating system.*
3. *Learn how to customize RStudio.*
4. ***Navigate, create, rename, move, copy directory and file in R.***
5. *Install packages, load libraries and write a simple R script.*
6. *Write a simple R markdown.*

Basic R commands (similar to BASH)

NOTE: Just like any programming language, R is also case and space sensitive!

Command explanation	BASH	R
Quit operation in console	Ctrl+C	ESC key
Current working directory	\$ pwd	> getwd()
Change working directory	\$ cd PATH	> setwd()
Create a folder	\$ mkdir FOLDER	> dir.create()
Create a file	\$ touch FILE	> file.create()
List folder contents	\$ ls -l	> list.files() > dir()

Basic R commands (similar to BASH)

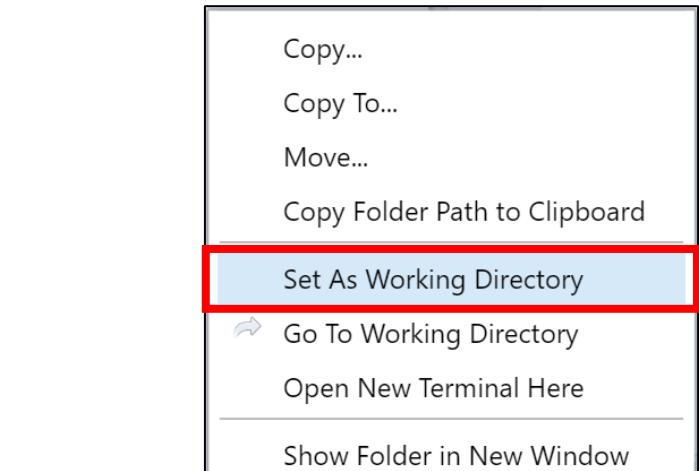
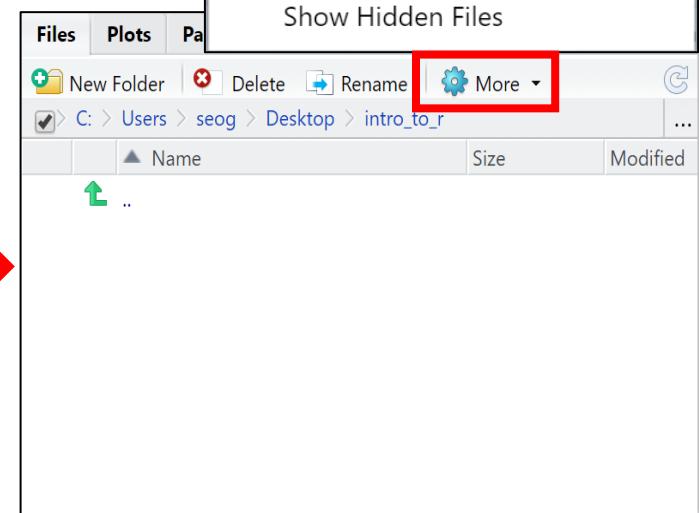
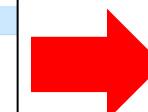
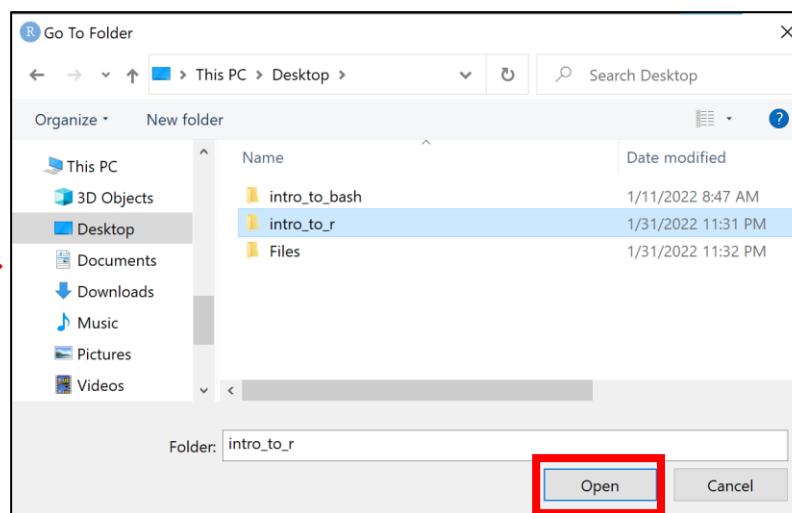
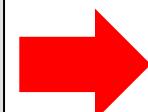
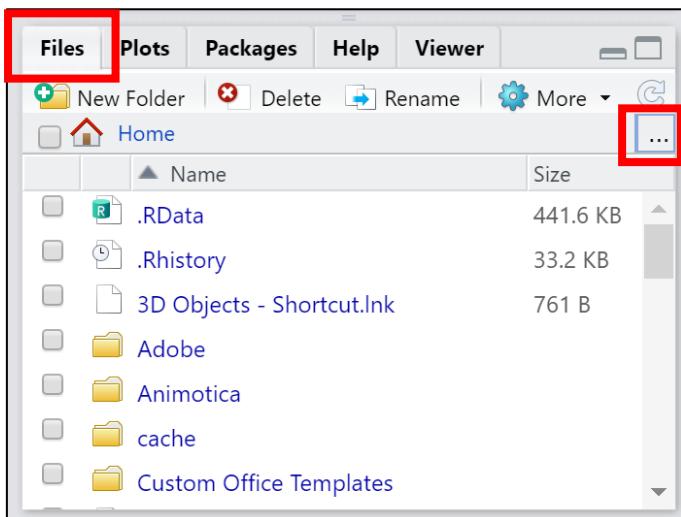
NOTE: Just like any programming language, R is also case and space sensitive!

Command explanation	BASH	R
Copy a file	\$ cp OLD NEW	> <code>file.copy("ITEM", "NEWPATH")</code>
Rename a file or folder	\$ mv OLDNAME NEWNAME	> <code>file.rename("OLDNAME", "NEWNAME")</code>
Move file or folder	\$ mv ITEM NEWPATH	> <code>file.copy("ITEM", "NEWPATH")</code> > <code>file.remove("ITEM")</code>
Remove a file or folder	\$ rm FILE \$ rm -r FOLDER	<code>file.remove("FILE")</code> <code>unlink("FOLDER", recursive = TRUE)</code>
Add text to a file	\$ echo "TEXT" > FILE	> <code>cat(file = "FILE", "MESSAGE", append = TRUE)</code>
Print file content in console	\$ cat FILE	> <code>FILE</code>

Navigate in R

Method 1:

- In “Navigation pane” → Select “Files” tab → Click “...” → Select folder of your choice
- In “Navigation pane” → Click “More” → Click “Set As Working Directory”



R folder path:

(Windows) "C:/Users/seog/Desktop"

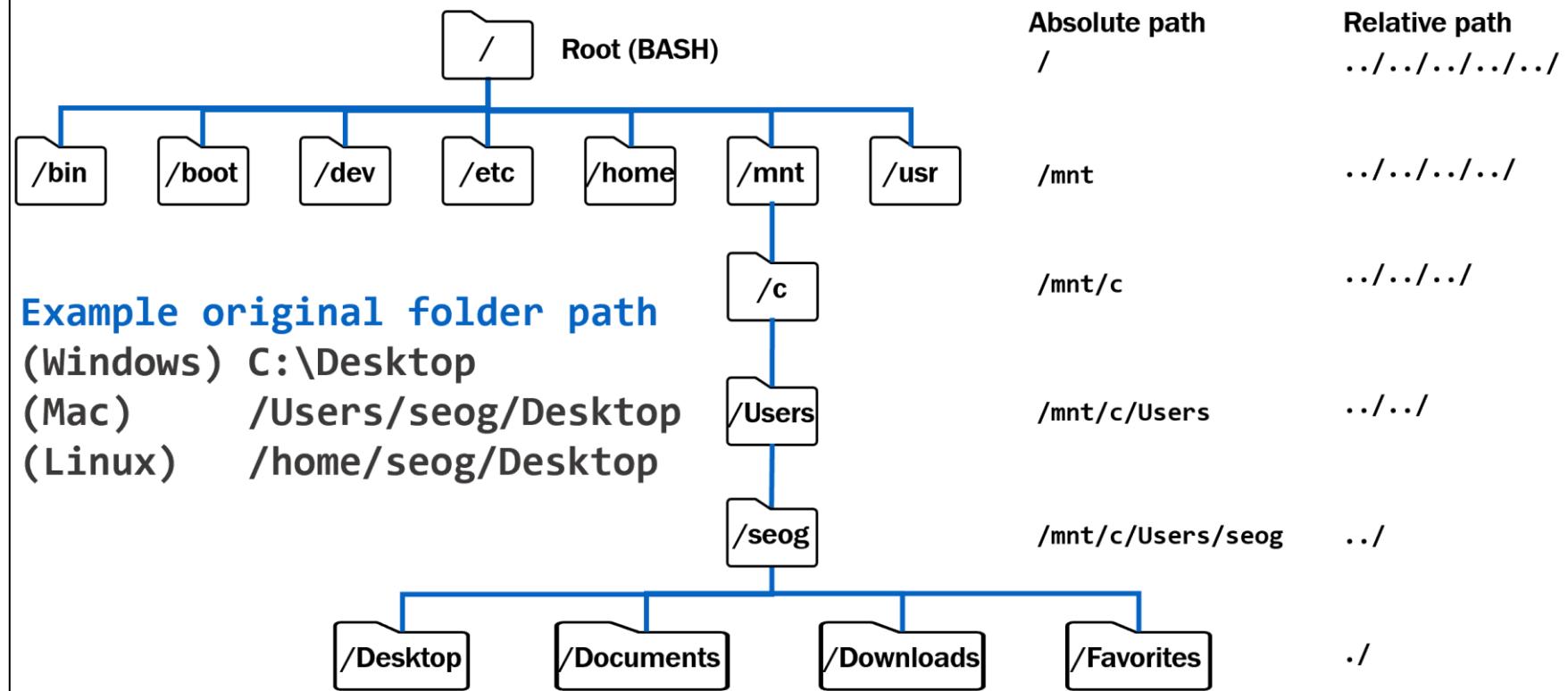
(Mac) "/Users/seog/Desktop"

(Linux) "home/seog/Desktop"

Reference: Intro to BASH

https://www.youtube.com/watch?v=_VUynJ_CBJo

Example Windows Subsystem for Linux BASH folder structure



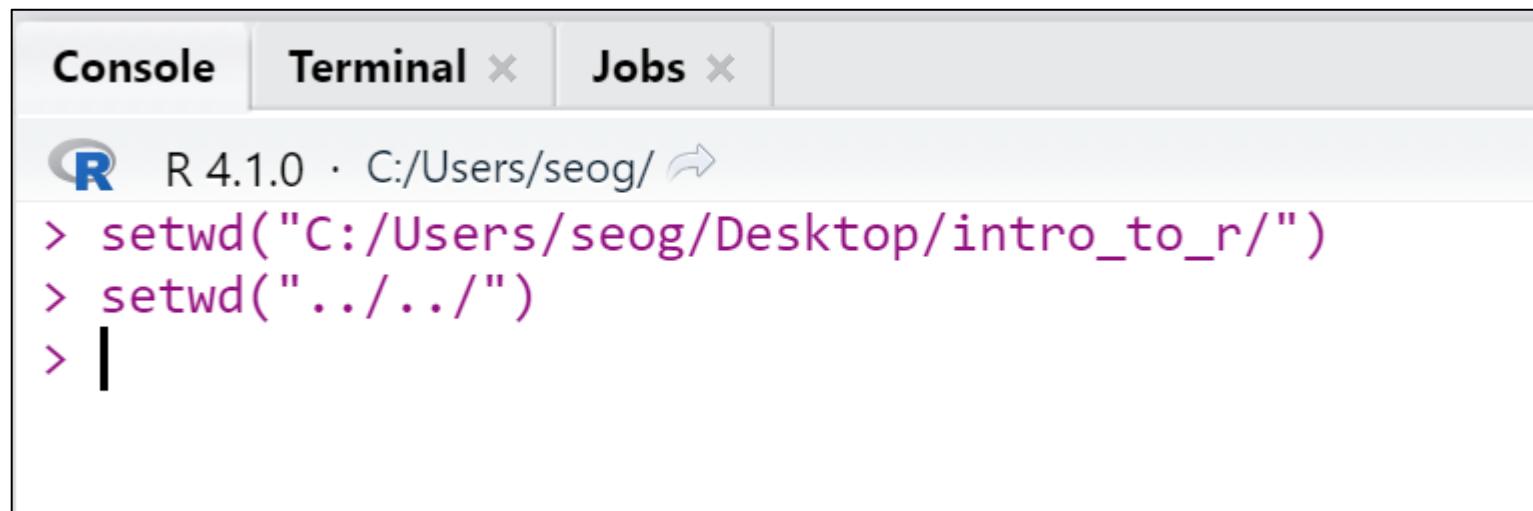
Navigate in R

Method 2: `setwd("PATH")` - change current directory

```
> setwd("C:/PATH")
```

NOTE: you can use relative path as well like in BASH!

Intro to BASH (see relative path) - https://www.youtube.com/watch?v=_VUynJ_CBJo

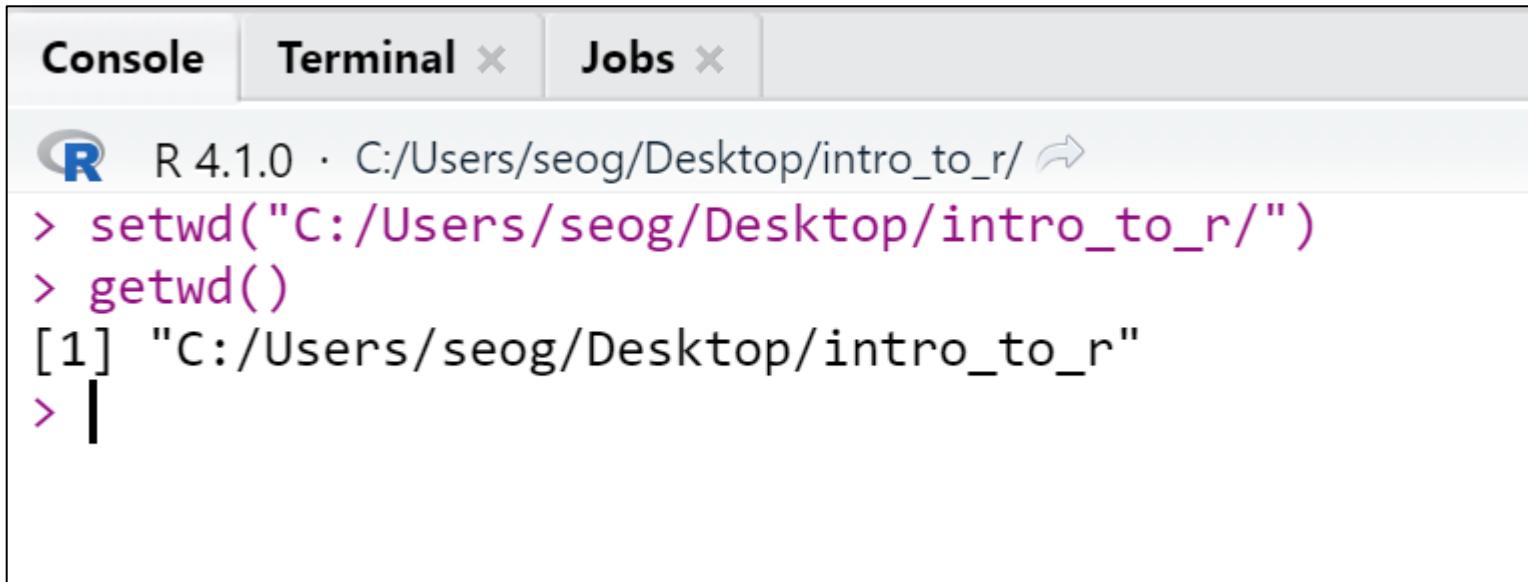


```
R 4.1.0 · C:/Users/seog/ ↵
> setwd("C:/Users/seog/Desktop/intro_to_r/")
> setwd("../..")
> |
```

Navigate in R

getwd() – returns current working directory

```
> getwd()
```



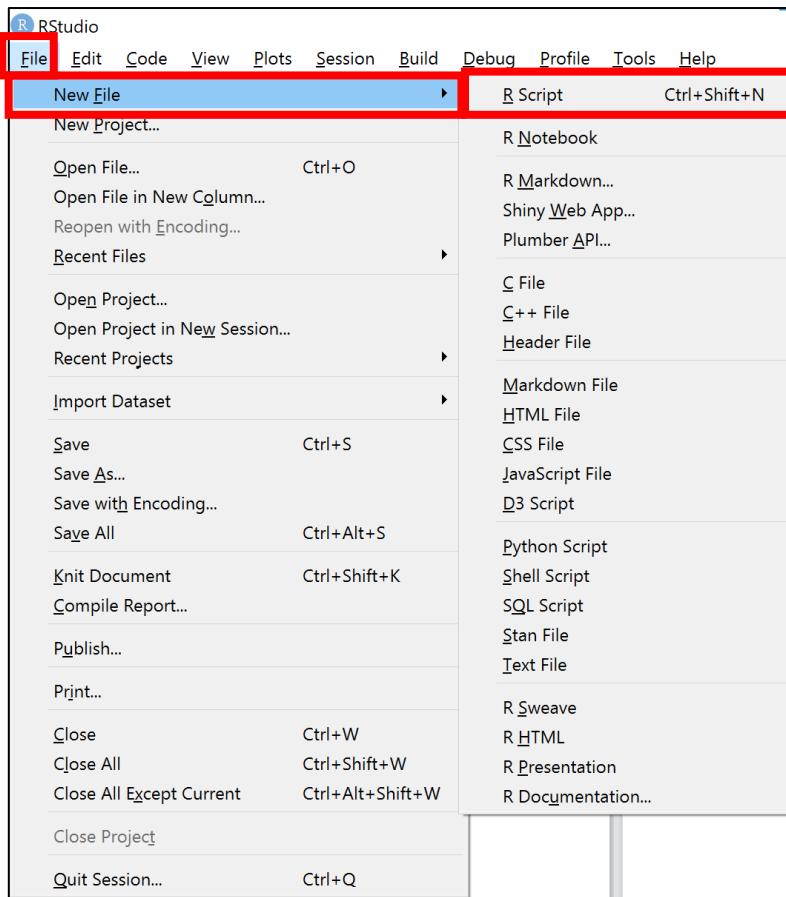
The screenshot shows the RStudio interface with the 'Console' tab selected. The console window displays the following R session:

```
R 4.1.0 · C:/Users/seog/Desktop/intro_to_r/ ↗
> setwd("C:/Users/seog/Desktop/intro_to_r/")
> getwd()
[1] "C:/Users/seog/Desktop/intro_to_r"
> |
```

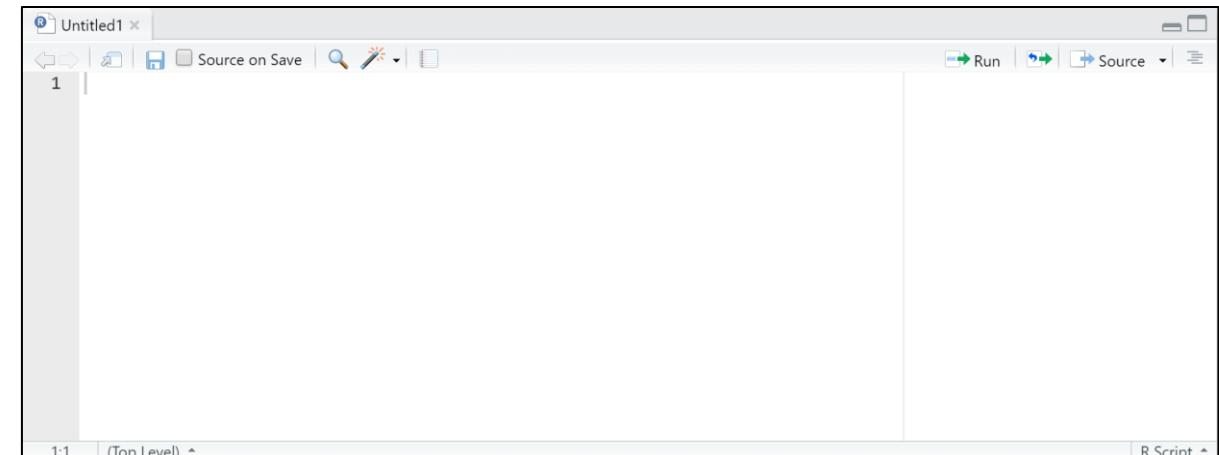
Create a New File i.e. R script

Method 1:

- In Top Menu → “File” tab → “New File” → Select (i.e. R script)



NOTE: file is not saved until you save it!



Create a New File i.e. R script

Method 2: `file.create("NAME.EXTENSION")`

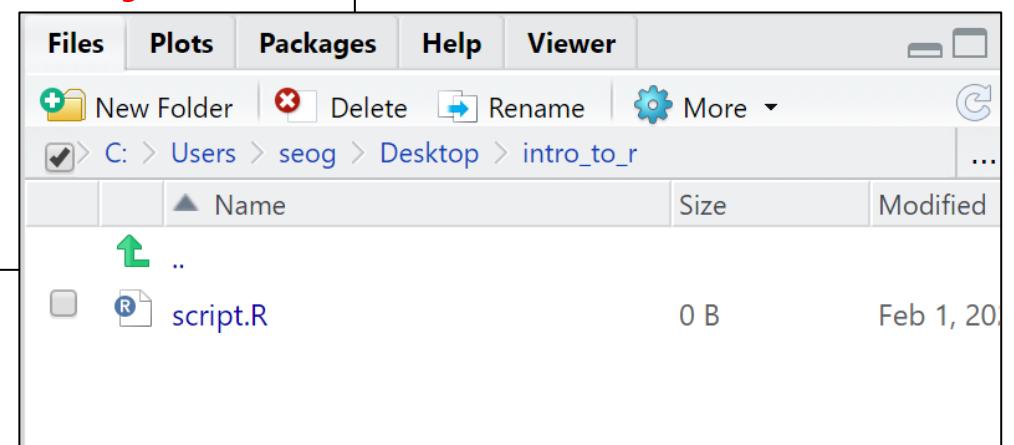
```
> file.create("script.R")
```

Console Terminal × Jobs ×

R 4.1.0 · C:/Users/seog/Desktop/intro_to_r/ ↗

```
> file.create("script.R")
[1] TRUE
> |
```

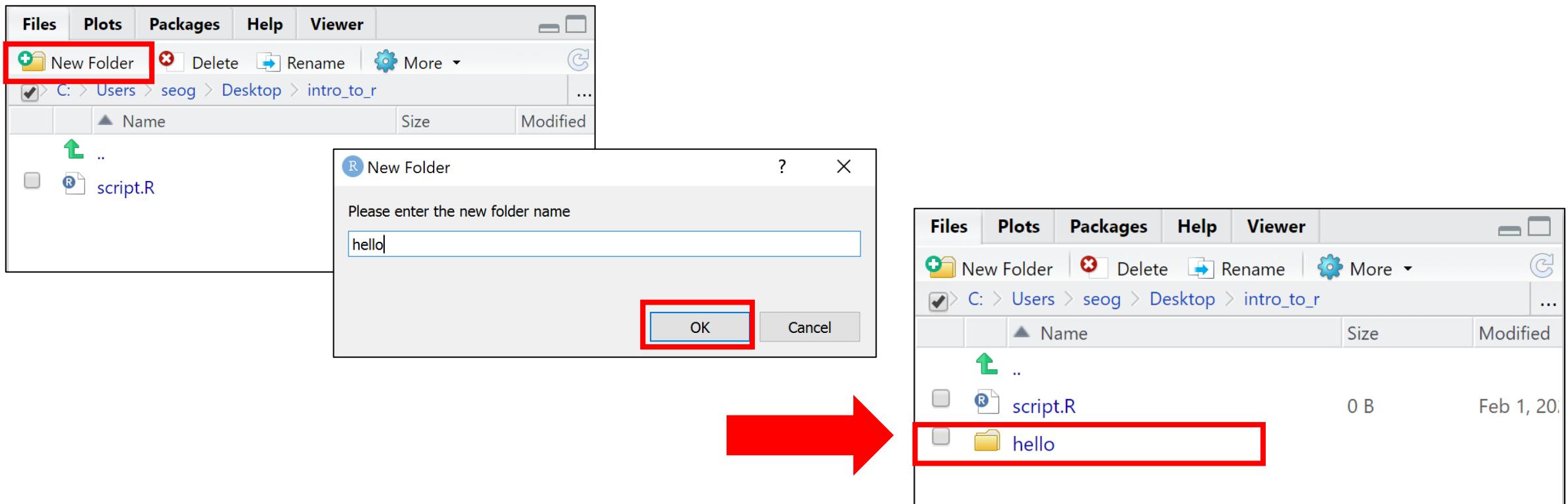
NOTE: you can double-click the file to edit



Create a New Folder

Method 1:

- In “Navigation pane” → “Files” → “New Folder”



Create a New Folder

Method 2: `dir.create("NAME")`

```
> dir.create("hello2")
```

The screenshot shows the RStudio interface. On the left, the Console tab is active, displaying the command `> dir.create("hello2")`. On the right, the File Explorer (Windows) or Finder (Mac) window shows the directory structure. A new folder named "hello2" has been created, along with existing files "script.R" and "hello".

Name	Size	Modified
..		
script.R	0 B	Feb 1, 2022
hello		
hello2		

Rename a file

file.rename("OLDNAME", "NEWNAME")

```
> file.rename("script.R", "newname.R")
```

The screenshot shows the RStudio interface with two panes. The left pane is the Console, and the right pane is the Files browser.

Console:

- Shows the R logo and version: R 4.1.0 · C:/Users/seog/Desktop/intro_to_r/
- Prompted with >
- Entered the command: `file.rename("script.R", "newname.R")`
- Output: [1] TRUE
- Prompted with >

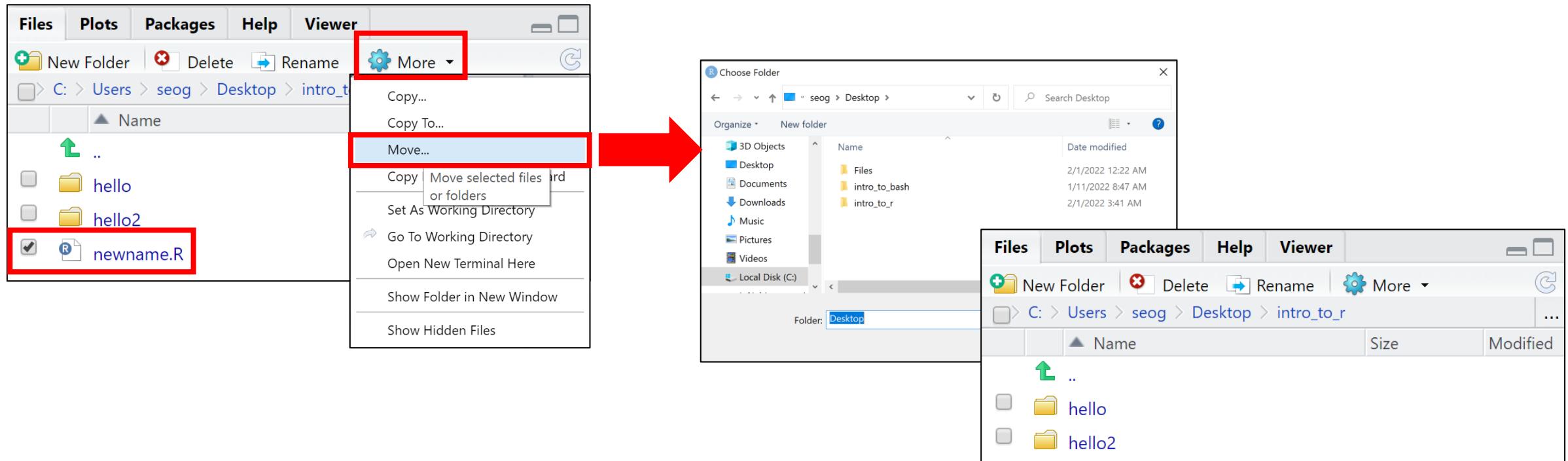
Files Browser:

- Shows the directory structure: C: > Users > seog > Desktop > intro_to_r
- Contains the following files and folders:

Name	Size	Modified
..		
hello		
hello2		
newname.R	0 B	Feb 1, 2022

Move a file

- In “Navigation pane” → “Files” → Select a file → “More” → Select a folder to move

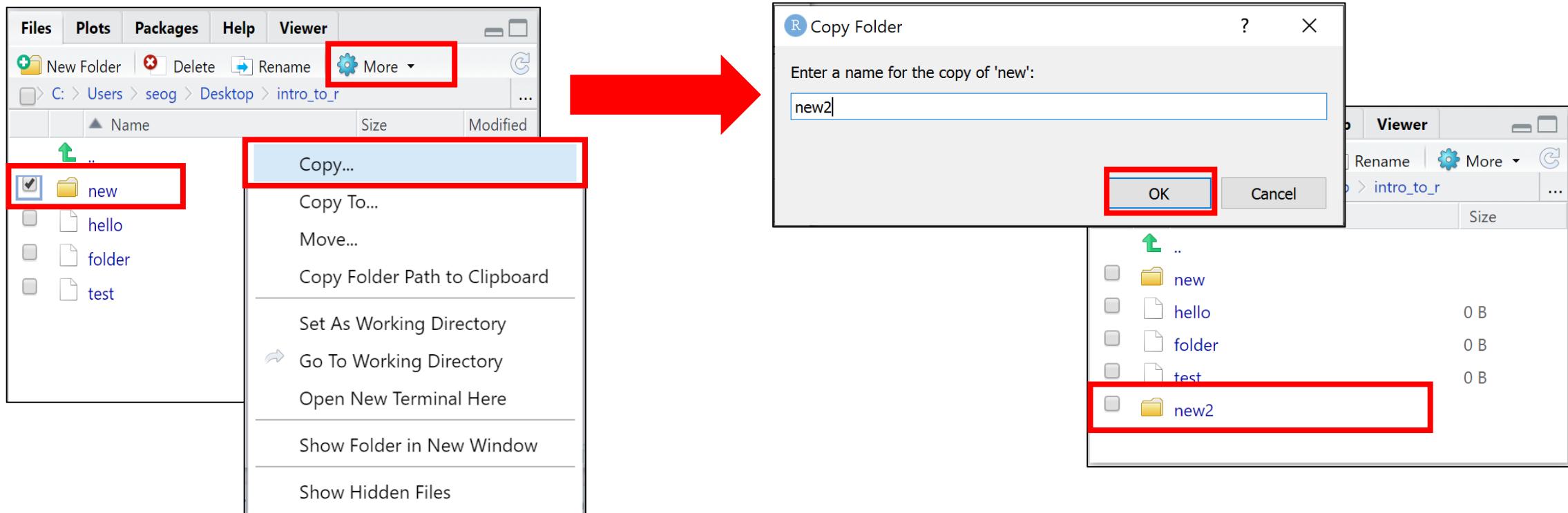


Copy a folder using interface

Method 1:

In “Navigation Pane” → Under “File” tab

→ Select file/folder to copy → click “More” → click “Copy” → enter a name



Copy a file in console

Method 2:

file.copy("ORIGINAL", "NEWCOPY")

```
> file.copy("file", "copied_file")
```

The screenshot shows the RStudio interface with two panes. The left pane is the Console, and the right pane is the Files browser.

Console (Top):

- R 4.1.0 · C:/Users/seog/Desktop/intro_to_r/
- > file.create("file")
[1] TRUE
- > file.copy("file", "copied_file")
[1] TRUE
- >

Files Browser (Bottom):

	Name	Size	Modified
..	..		
	hello		
	hello2		
	file	0 B	Feb 1, 2022
	copied_file	0 B	Feb 1, 2022

Part 5. Install packages, load libraries and write a simple R script

LEARNING OBJECTIVES

1. *Describe types of analyses performed using R or BASH.*
2. *Install R and RStudio on your operating system.*
3. *Learn how to customize RStudio.*
4. *Navigate, create, rename, move, copy directory and file in R.*
5. ***Install packages, load libraries and write a simple R script.***
6. *Write a simple R markdown.*

Packages in R

- Package is a bundle of code, data, documentation, and tests that is easily shared with others.
- Package is similar to a program or conda environments that you can download for bioinformatics analyses in BASH.
 - For example: minimap2, pychopper, trimmomatic etc.
 - Watch “Intro to CONDA” <https://www.youtube.com/watch?v=r-wmmhjHFJ4>
 - Watch “Assembling genome” <https://www.youtube.com/watch?v=qa0kqE8BIVQ>

List of commonly used packages in R

For data transformation

- tidyverse (includes dplyr and tidyr) (<https://www.tidyverse.org/>)
- stringr (<https://stringr.tidyverse.org/>)

For data visualization

- ggplot2 (part of tidyverse) (<https://ggplot2.tidyverse.org/>)
- ggtree
(<http://bioconductor.org/packages/release/bioc/html/ggtree.html>)

For RNA-seq data analyses

- DESeq2
(<https://bioconductor.org/packages/release/bioc/html/DESeq2.html>)
- edgeR
(<https://bioconductor.org/packages/release/bioc/html/edgeR.html>)

For machine learning

- mice (<https://amices.org/mice/>)
- e1071 (<https://cran.r-project.org/web/packages/e1071/e1071.pdf>)
- randomForest (<https://cran.r-project.org/web/packages/randomForest/randomForest.pdf>)
- caret (<https://topepo.github.io/caret/>)

Interactive plots and dashboard

- plotly (<https://plotly.com/r/>)
- shiny (<https://shiny.rstudio.com/>)

R coding basics: R vectors

- **Atomic Vectors:** simple vector of data

```
> object <- c(1, 2, 3, 4, 5)
> object
## 1 2 3 4 5
```

- **Doubles:** stores regular numbers

```
> typeof(object)
## "double"
```

- **Integers:** vector store integers

```
> int <- c(-1L, 2L, 4L) # Need to use L to create an integer
> int
## -1 2 4
```

- **Characters:** stores small pieces of text

```
> text <- c("Hello", "World")
> text
## "Hello" "World"
```

- **Logicals:** stores TRUE and FALSE (Boolean)

```
> 3 >= 4
## FALSE

> logic <- c(TRUE, FALSE, TRUE)
> logic
## TRUE FALSE TRUE
```

- **Factors:** stores categorical information

```
> gender <- factor(c("male", "female", "female"))
> attributes(gender)
## $levels
## [1] "female" "male"
```

Hands-On-Programming with R:

<https://rstudio-education.github.io/hopr/r-objects.html>

Practice Scenario 1

- You are a teaching assistant for Pokémon bioinformatics course.
- Your supervisor provided you with a class metadata and would like you to perform simple analysis as listed below.



<https://www.pokemon.com/us/pokemon-tcg/>

Scenario 1: Script and dataset

For R script and R markdown, go to GitHub page:

<https://github.com/MMID-coding-workshop/2022-02-02-Introduction-to-R>

Raw Pokemon class metadata:

https://github.com/MMID-coding-workshop/2022-02-02-Introduction-to-R/blob/main/class_metadata.csv

Scenario 1: Create R script

To follow along from the beginning,

From top menu,

- Click “File” tab → “New File” → “R Script”
- Save the file first.

NOTE: to run the contents in the script, highlight codes to run → press Ctrl+Enter

Scenario 1: load dataset

1. Load data into an object in R

Two methods to load dataset:

- URL (use URL for RAW file instead of HTML if using GitHub)
- Download and load from local folder

Method 1 (loading from URL):

```
OBJECTNAME <- read.csv("URL/FILE.csv", header = TRUE)
```

Method 2 (load from local folder):

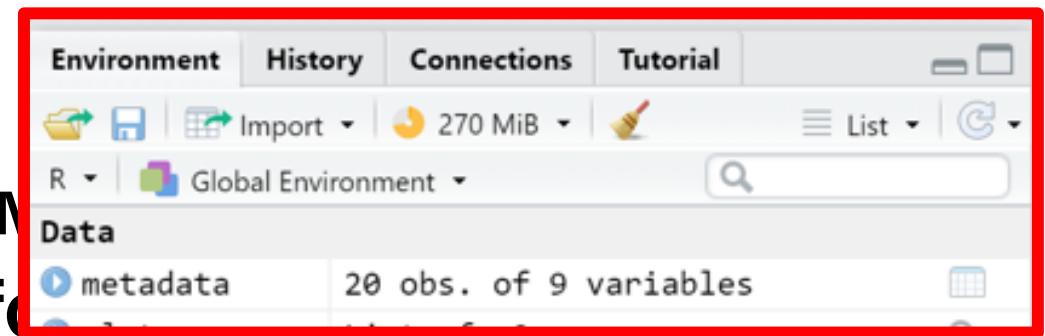
```
OBJECTNAME <- read.csv("FOLDERPATH/FILE.csv", header = TRUE)
```

Scenario 1: load dataset example

1. Load data into an object in R (example)

Two methods to load dataset:

- URL (use URL for RAW file instead of HTML)
- Download and load from local folder



Method 1 (loading from URL):

```
metadata <- read.csv("https://raw.githubusercontent.com/MMID-coding-workshop/2022-02-02-Introduction-to-R/main/class_metadata.csv", header = TRUE)
```

Method 2 (load from local folder):

```
Metadata <- read.csv("C:/Users/seog/Desktop/intro_to_r/class_metadata.csv", header = TRUE)
```

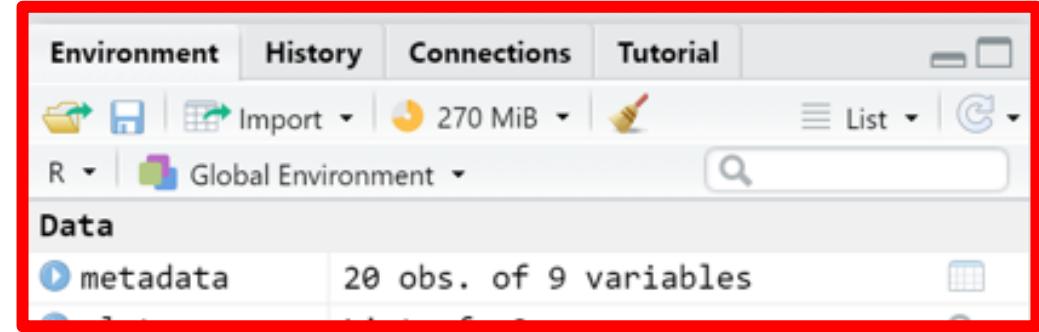
Scenario 1: view loaded dataset

View loaded data

You can:

- Click the object in “Environment pane” or
- Use View() function

View(metadata)



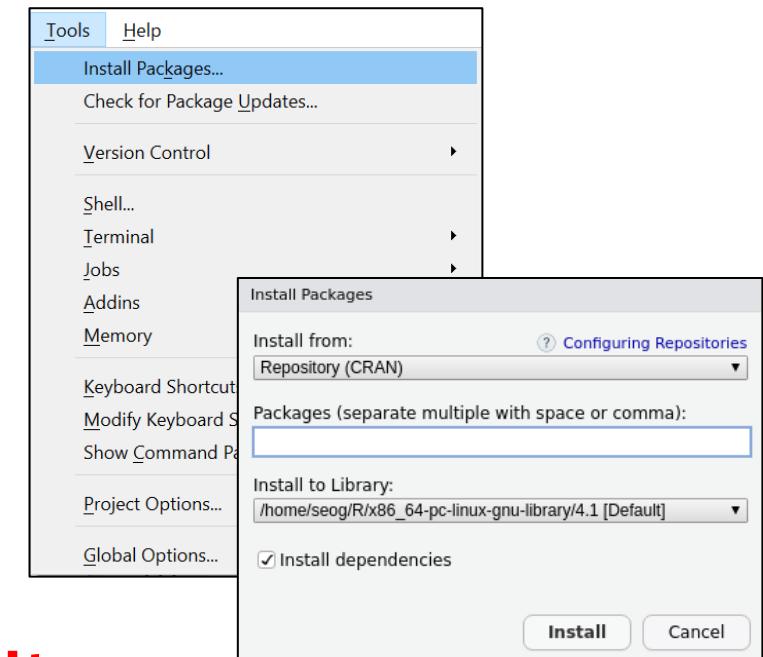
The screenshot shows the RStudio interface. The Environment pane at the top has a red border around it. Inside, there is a list of objects: 'metadata' is highlighted with a blue circle and labeled '20 obs. of 9 variables'. Below the Environment pane is a Data grid window titled 'metadata'. The grid shows 7 rows of data with columns: Number, Student_name, Student_number, Age, Year_of_study, Phone, and Address. The data is as follows:

	Number	Student_name	Student_number	Age	Year_of_study	Phone	Address
1	1	Mew	7991102	28		4	111-111-1111
2	2	Garchomp	7980475	18		5	111-111-1112
3	3	Kanto Starters	7779385	19		3	111-111-1113
4	4	Metagross	7620583	24		3	111-111-1114
5	5	Zoroark	8103955	25		3	111-111-1115
6	6	Tyranitar	8502931	21		4	111-111-1116
7	7	Eevee	8110291	29		4	111-111-1117

Scenario 1: Install and load library

For scenario 1, all you need is **ggplot2**.
tidyverse contains many packages like:

- dplyr
- tidyr
- ggplot2 and more.



2. Install tidyverse only if you don't have it.

You can install by code or by going to “Tools” → “Install Packages”

```
install.packages("tidyverse") # skip if you have it  
library("tidyverse") # load library
```

Scenario 1: nrow()

3. Determine how many students are in your class.

HINT: total row number matches total number of students.

```
nrow(metadata) # count rows in the object
```

ANSWER: 20

Scenario 1: ncol()

4. Determine how many columns are present in your metadata.

```
ncol(metadata) # count columns in the object
```

ANSWER: 9

Scenario 1: colnames()

5. Show the list of column headers in the metadata.

```
colnames(metadata) # print all column headers
```

Number	Student_name	Student_number	Age	Year_of_study	Phone	Address	Address_city	Major
1	1	Mew	7991102	28	4	111-111-1111	111 Pokemon Street	Pallet Town
2	2	Garchomp	7980475	18	5	111-111-1112	112 Pokemon Street	Pallet Town

ANSWER:

```
[1] "Number"          "Student_name"    "Student_number" "Age"           "Year_of_study"   "Phone"  
[7] "Address"         "Address_city"    "Major"
```

Scenario 1: OBJECT\$COLNAME

6. Print out row contents of only column “Student_name”

\$ operator in R allows user to access data frame column

```
metadata$Student_name # prints 20 student names
```

ANSWER:

[1] "Mew"	"Garchomp"	"Kanto Starters"	"Metagross"	"Zoroark"	"Tyranitar"
[7] "Eevee"	"Snorlax"	"Volcarona"	"Mewtwo"	"Charizard"	"Milotic"
[13] "Kyurem"	"Haxorus"	"Arceus"	"Ninetales"	"Espeon"	"Golurk"
[19] "Lapras"	"Pikachu"				

Scenario 1: mean()

7. Determine average age of all students in this class.

HINT: use \$ operator to access “Age” column

```
mean(metadata$Age) # calculate average student age
```

ANSWER: 24

Scenario 1

8. Determine Pikachu's student number.

HINT1: use \$ operator to access “Student_number” column
HINT2: use == to specify value you are looking for

```
metadata$Student_number[metadata$Student_name=="Pikachu")
# OBJECT$COL_INTEREST[OBJECT$COL_KNOWN_VALUE=="VALUE"]
```

This code tells R to go into “metadata” and search the row that has “Student_name” column value equal to “Pikachu”. Then within the same row, go to column “Student_number” and return “Pikachu’s” student number.

ANSWER: 7094583

Scenario 1: ggplot(), geom_point()

9. Draw a scatter plot by student's Year_of_study vs. Age and create a figure legend by student's study Major

HINT: use ggplot in tidyverse package.

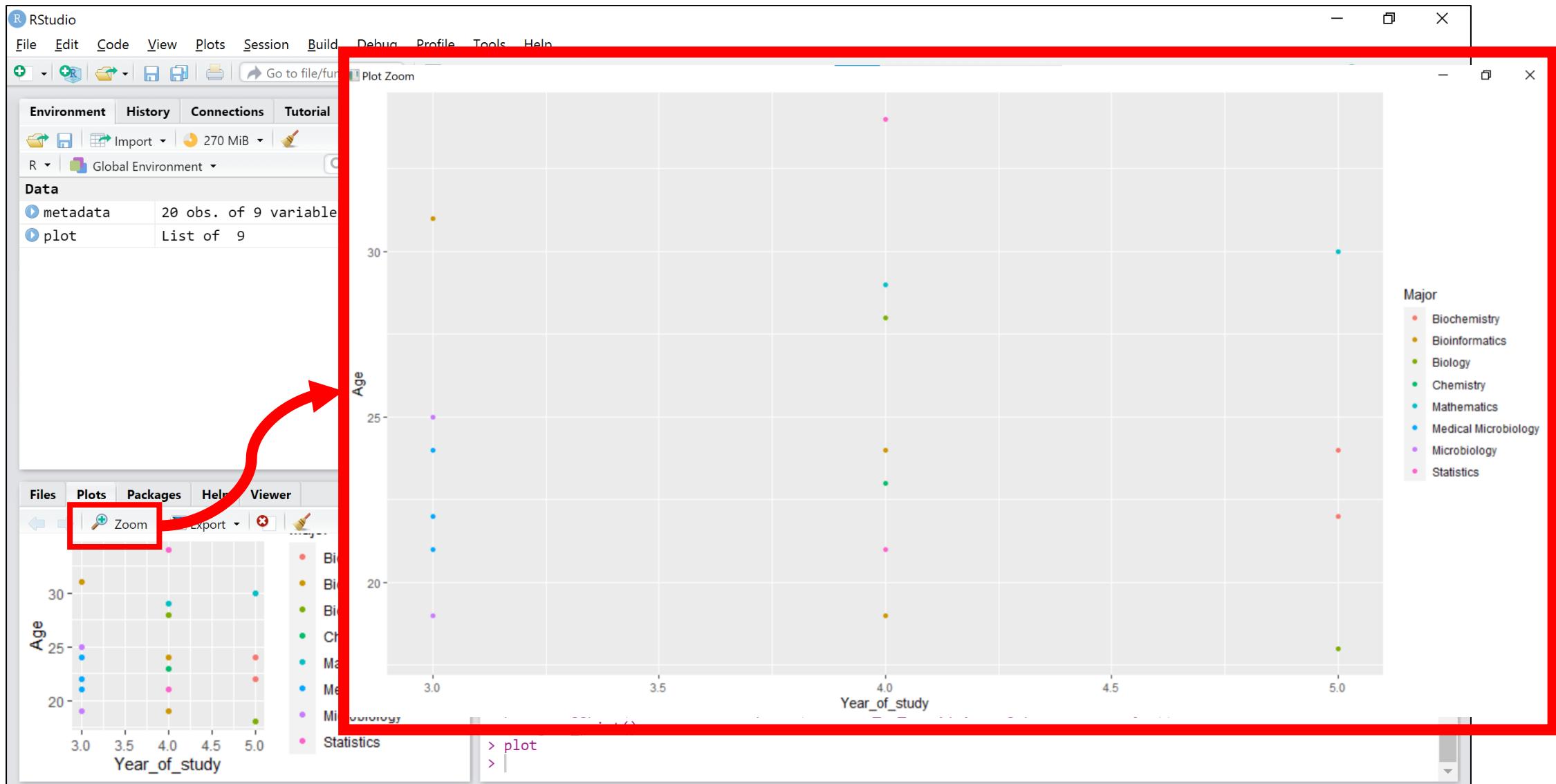
```
plot <- ggplot(data = metadata, # use metadata to plot  
                 aes(x = Year_of_study, y = Age, colour = Major)) + # set x, y  
                 geom_point() # group by "Major" and create scatter plot  
  
plot # View graph stored in "plot" object
```

Scenario 1: ggplot(), geom_point()

The screenshot shows the RStudio interface with the following components:

- File Menu:** File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help.
- Project Bar:** Project: (None).
- Environment Tab:** Shows Global Environment (20 obs. of 9 variables) and plot (List of 9).
- Code Editor:** Displays the R script `scenario_1_r_script.R*` containing code for creating a scatter plot.
- Console:** Shows the R session output, including package loading and an error message about the `metadata` function.
- Plots Tab:** Contains a scatter plot titled "Age" vs "Year_of_study". The plot shows data points colored by Major. A legend on the right lists the majors: Biochemistry (red), Bioinformatics (orange), Biology (green), Chemistry (teal), Mathematics (blue), Medical Microbiology (light blue), Microbiology (purple), and Statistics (pink). The x-axis ranges from 3.0 to 5.0, and the y-axis ranges from 20 to 30.

Scenario 1: Plots → Zoom



Scenario 1: ggplot(), geom_point()

10. Scatter plot is too small to see. Increase the size of each scatter point, remove grey background and rename the x-axis to "Year of Study".

HINT: use size = in geom_point()

```
plot_2 <- ggplot(data = metadata, aes(x = Year_of_study, y = Age, colour = Major)) +  
  geom_point(size = 5) +  
  xlab("Year of Study") + # change x-axis label  
  theme_bw() # use dark-on-light ggplot2 theme  
  
Plot_2 # View graph stored in “plot” object
```

Scenario 1: ggplot(), geom_point()

The screenshot shows the RStudio interface with the following components:

- Environment Tab:** Shows the Global Environment with objects: metadata (20 obs. of 9 variables), plot (List of 9), and plot_2 (List of 9).
- Code Editor:** Displays the R script `scenario_1_r_script.R`. The code creates a scatter plot where points are colored by major. A portion of the code is highlighted with a red box.
- Console:** Shows the R session output. It starts with the R version and path, then attempts to run the ggplot code. It fails because the `metadata` function is not found. The user then tries running the code again, which succeeds.
- Plots Tab:** Contains a scatter plot titled "Age" vs "Year of Study". The x-axis ranges from 3.0 to 5.0, and the y-axis ranges from 20 to 30. Points are colored according to their major: Biochemistry (red), Bioinformatics (orange), Biology (green), Chemistry (teal), Mathematics (blue), Medical Microbiology (light blue), Microbiology (purple), and Statistics (pink). A legend on the right side lists these majors with their corresponding colors.

```
## 10. Scatter plot is too small to see.  
## Increase the size of each scatter point, remove grey background  
## and rename the x-axis to "Year of Study".  
#=====  
plot_2 <- ggplot(data = metadata, aes(x = Year_of_study, y = Age, colour = Major)) +  
  geom_point(size = 5) +  
  xlab("Year of Study") +  
  theme_bw()  
plot_2 # view the plot
```

```
R 4.1.0 · C:/Users/seog/Desktop/intro_to_r/  
> plot <- ggplot(data = metadata(aes(x = Year_of_study, y = Age, colour = Major))) +  
+   geom_point()  
Error in metadata(aes(x = Year_of_study, y = Age, colour = Major)) :  
  could not find function "metadata"  
> plot <- ggplot(data = metadata, aes(x = Year_of_study, y = Age, colour = Major)) +  
+   geom_point()  
> plot  
> plot_2 <- ggplot(data = metadata, aes(x = Year_of_study, y = Age, colour = Major)) +  
+   geom_point(size = 5) +  
+   xlab("Year of Study") +  
+   theme_bw()  
> plot_2 # view the plot  
>
```

Scenario 1: Plots → Zoom



Part 6. Write a simple R markdown

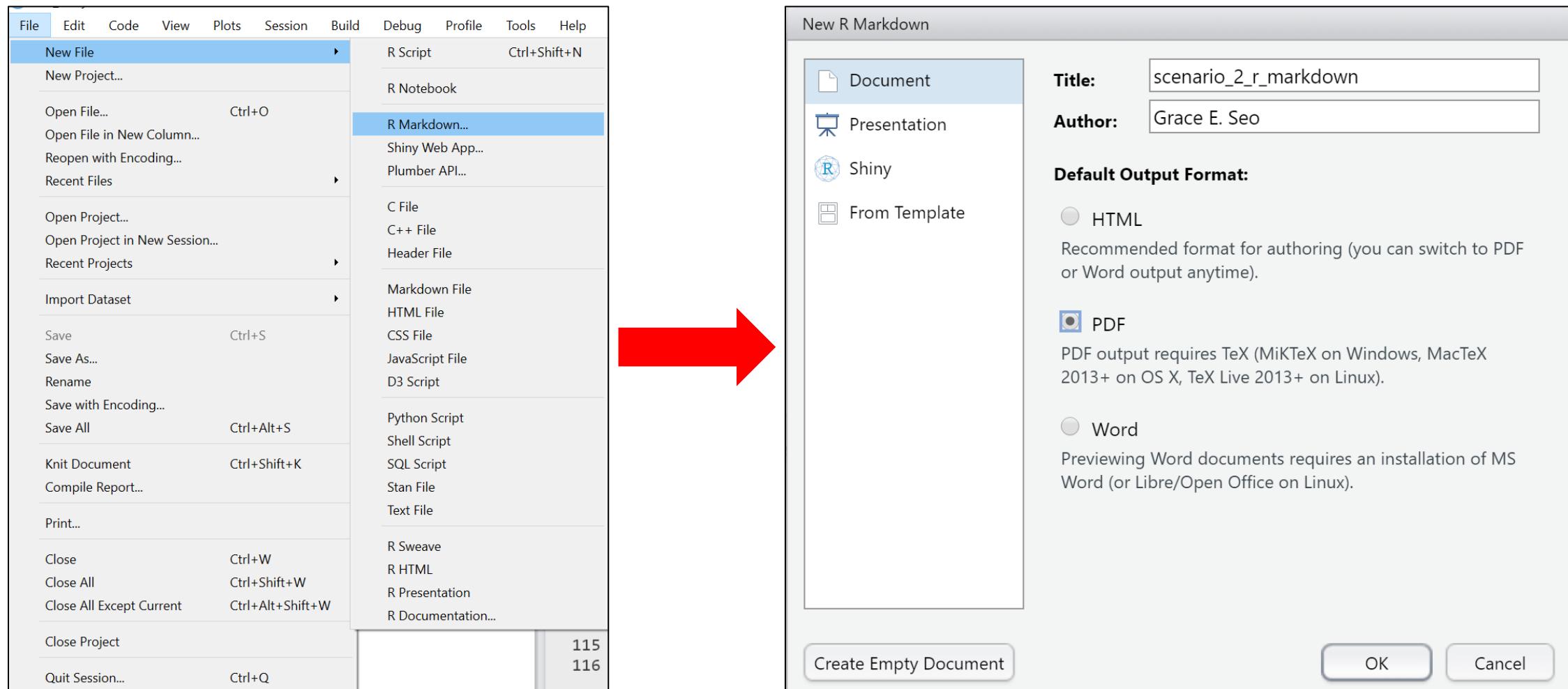
LEARNING OBJECTIVES

- 1. *Describe types of analyses performed using R or BASH.***
- 2. *Install R and RStudio on your operating system.***
- 3. *Learn how to customize RStudio.***
- 4. *Navigate, create, rename, move, copy directory and file in R.***
- 5. *Install packages, load libraries and write a simple R script.***
- 6. *Write a simple R markdown.***

Practice Scenario 2

- Now, to provide the result you generated from scenario 1 to your supervisor, you need to create a PDF output using R markdown.
- Copy and paste your codes from scenario 1 and create a beautiful PDF output using R markdown.

Scenario 2: create R markdown file



Scenario 2: markdown basics

Once the file is created, by default these components will be present.

```
---
```

```
title: "scenario_2_r_markdown"
author: "Grace E. Seo"
date: "2/1/2022"
output: pdf_document
```

```
---
```

```
```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```
```

Scenario 2: markdown basics

Anything written in white space becomes a text in the output (PDF or HTML)

```
---
```

```
title: "scenario_2_r_markdown"
author: "Grace E. Seo"
date: "2/1/2022"
output: pdf_document
```

```
---
```

Becomes plain text message

```
```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```
```

Scenario 2: markdown basics

All the codes must be wrapped in

```{r} Start of code chunk  
... End of code chunk

```

```

```
title: "scenario_2_r_markdown"
author: "Grace E. Seo"
date: "2/1/2022"
output: pdf_document

```

```
82
83 ### View dataset
84
85 ```{r}
86
87 # CODE: View("OBJECTNAME")
88
89 View(metadata)
90
91 ## NOTE: new window will pop-up beside this script. You can scroll through the content.
92
93 metadata # this will print data into console.
94
95 ...
96
```

Becomes plain text message

{  
```{r setup, include=FALSE}  
knitr::opts_chunk\$set(echo = TRUE)
```

## Scenario 2: markdown basics

To create a bookmark / table of contents with headers, use #

# Header 1

## Header 2

### Header 3

```
```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```
```

```
Header 1
Header 2
Header 3
```

scenario\_2\_r\_markdown

Grace E. Seo

2/1/2022

### Scenario 2

In Scenario 1, you performed some analysis and created a scatter plot of Pokemon bioinformatics class using student metadata.

To provide the result to your supervisor, you need to create a PDF output using R markdown.

Before you begin, let's learn how R markdown works briefly.

If you want to know more about R markdown, here is a great resource: <https://bookdown.org/yihui/rmarkdown-cookbook/>

In R markdown, you can create a Bookmark/table of contents by using #.

For example:

Header 1

Header 2

Header 3

Bullet points

# Scenario 2: markdown basics

To create a bullet point, use – and space

- Like this!

**-If you don't put a space in front of dash, this sentence won't be recognized as bullet**

```
```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```
Bullet points
```

If you want to include bullet points,

- Like this! Make sure you put a space before the dash!

Header 1

Header 2

Header 3

Bullet points

If you want to include bullet points, use “–” in front of your sentence.

- Like this! Make sure you put a space before the dash!
- If you want to leave some blank lines between sentences, include a white space between two sentences by pressing “Enter”.

# Scenario 2: markdown basics

To include a timestamp on your document,

In the white space, wrap your time/date code in `` (red text)

```
```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```
```

### Include a timestamp on this document

To know when this PDF file was generated, include a timestamp.

date: "Last compiled on `r format(Sys.time(), '%d %B, %Y, at %H:%M:%S')`"

Include a timestamp on this document

To know when this PDF file was generated, include a timestamp.

date: "Last compiled on 01 February, 2022, at 06:27:09"

Now, let's include all the codes from scenario 1 as well include outputs directly in this PDF file.

## Scenario 2: markdown basics

To include a figure caption under your figures,

In the code chunk header ```{r }`, type `fig.cap = "FIGURE TITLE"`

```{r, fig.cap = "FIGURE TITLE"}`

code

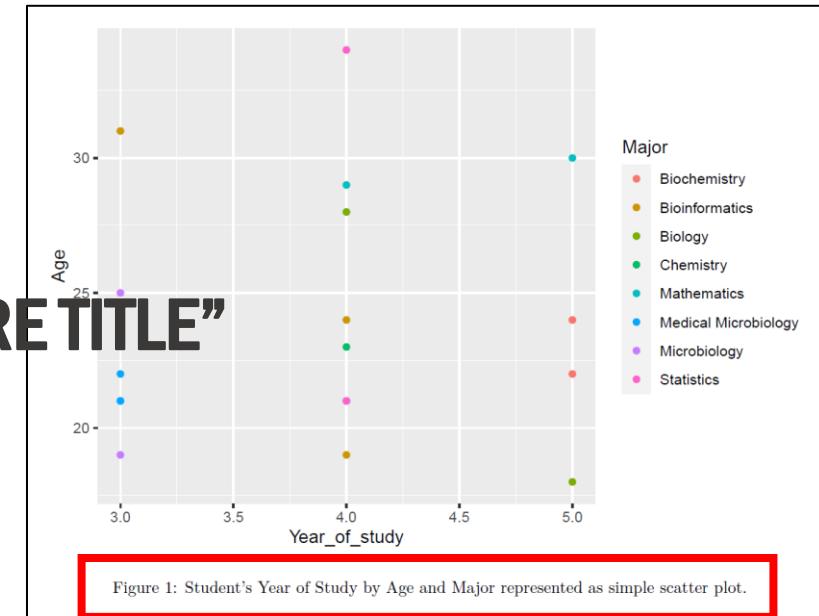
```

```
```{r, fig.cap = "Student's Year of Study by Age and Major represented as simple scatter plot."}
```

```
plot <- ggplot(data = metadata, aes(x = Year_of_study, y = Age, colour = Major)) +  
  geom_point()
```

```
plot # view the plot
```

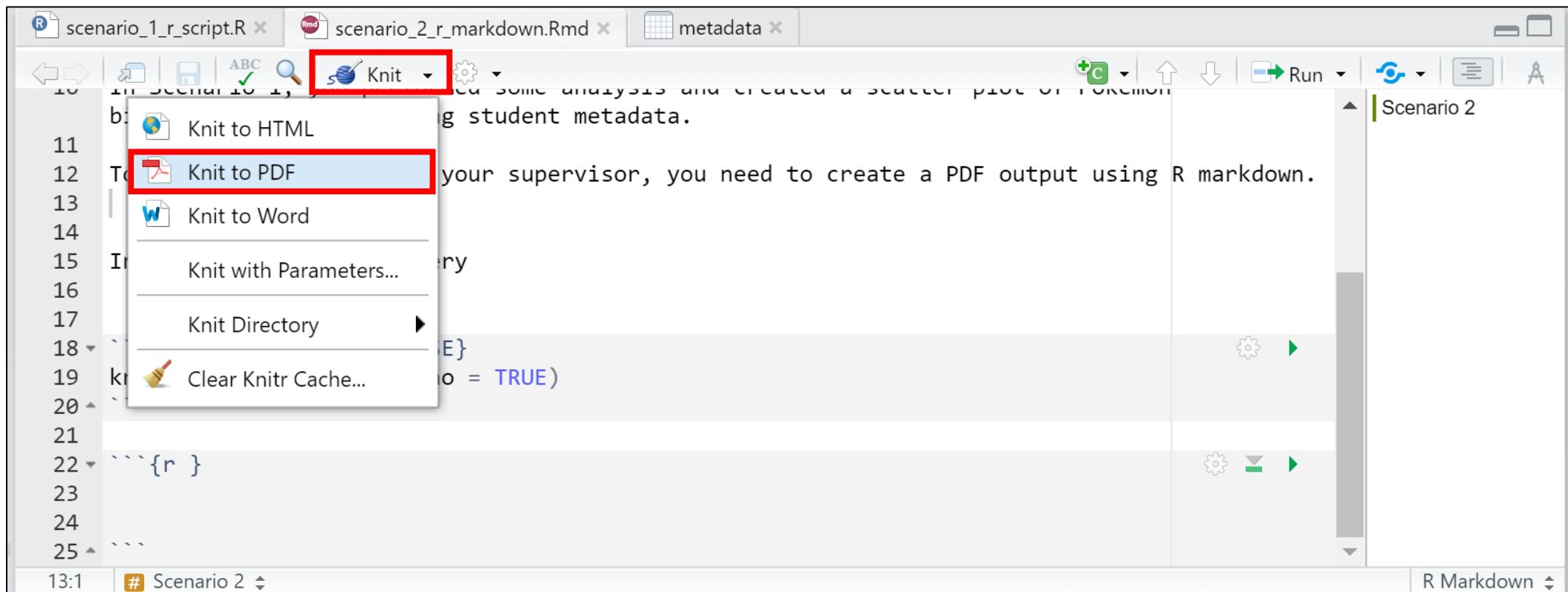
```



# Scenario 2: knit your markdown → PDF

At the top of your markdown → Click “Knit” → “Knit to PDF”

- Choose file type that you prefer for reporting.



# Scenario 2: PDF output

- You should get a prompt of your dataset (from View(metadata) code)
- scenario\_2\_r\_markdown.pdf

The screenshot shows the RStudio interface. On the left, a Data frame titled "Data: metadata" is displayed with the following data:

|    | Number | Student_name   | Student_number | Age | Year_of_study | Phone        |
|----|--------|----------------|----------------|-----|---------------|--------------|
| 1  | 1      | Mew            | 7991102        | 28  | 4             | 111-111-1111 |
| 2  | 2      | Garchomp       | 7980475        | 18  | 5             | 111-111-1112 |
| 3  | 3      | Kanto Starters | 7779385        | 19  | 3             | 111-111-1113 |
| 4  | 4      | Metagross      | 7620583        | 24  | 3             | 111-111-1114 |
| 5  | 5      | Zoroark        | 8103955        | 25  | 3             | 111-111-1115 |
| 6  | 6      | Tyranitar      | 8502931        | 21  | 4             | 111-111-1116 |
| 7  | 7      | Eevee          | 8110291        | 29  | 4             | 111-111-1117 |
| 8  | 8      | Snorlax        | 8110429        | 30  | 5             | 111-111-1118 |
| 9  | 9      | Volcarona      | 7889281        | 34  | 4             | 111-111-1119 |
| 10 | 10     | Mewtwo         | 8719237        | 21  | 3             | 111-111-1120 |
| 11 | 11     | Charizard      | 7817241        | 22  | 3             | 111-111-1121 |
| 12 | 12     | Moltot         |                |     |               |              |
| 13 | 13     | Kyurem         |                |     |               |              |
| 14 | 14     | Haxorus        |                |     |               |              |
| 15 | 15     | Arceus         |                |     |               |              |
| 16 | 16     | Ninetales      |                |     |               |              |
| 17 | 17     | Espeon         |                |     |               |              |
| 18 | 18     | Golurk         |                |     |               |              |
| 19 | 19     | Lapras         |                |     |               |              |

Below the Data frame, the file browser shows the following files:

- scenario\_1\_r\_script.R
- scenario\_2\_r\_markdown.pdf
- scenario\_2\_r\_markdown.Rmd

The screenshot shows a PDF viewer window with the title "scenario\_2\_r\_markdown.pdf". The PDF content includes:

- A header section with "Scenario 2", "Header 1", "Header 2", "Header 3", and "Bullet points".
- A timestamp section with "Include a timestamp" and "2/1/2022".
- A main content section titled "Scenario 2" containing numbered steps from 1 to 10, each followed by a brief description.
- An "Header 1" section with "Header 2" and "Header 3" examples.
- An "Header 2" section with "Header 3" examples.
- An "Header 3" section with "Bullet points" examples.
- A "References" section.
- A note about bullet points: "If you want to include bullet points, use “-” in front of your sentence." with examples.
- A "Include a timestamp on this document" section with a timestamp example.
- A note at the bottom: "Now, let's include all the codes from scenario 1 as well include outputs directly in this PDF file."

# LEARNING OBJECTIVES

- 1. *Describe types of analyses performed using R or BASH.***
- 2. *Install R and RStudio on your operating system.***
- 3. *Learn how to customize RStudio.***
- 4. *Navigate, create, rename, move, copy directory and file in R.***
- 5. *Install packages, load libraries and write a simple R script.***
- 6. *Write a simple R markdown.***

# Great practice for beginners: swirl()

- **Swirl teaches you R programming and data science interactively in R console!**

<https://swirlstats.com/>

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

> library("swirl")
> swirl()
```

The swirl R package makes it fun and easy to learn R programming and data science. If you are new to R, have no fear. On this page, we'll walk you through each of the steps required to begin using swirl today!

### Step 1: Get R

In order to run swirl, you must have R 3.1.0 or later installed on your computer. If you are on a Linux operating system, please visit our [Installing swirl on Linux](#) page.

If you need to install R, you can do so [here](#).

For help installing R, check out one of the following videos (courtesy of Roger Peng at Johns Hopkins Biostatistics):

- [Installing R on Windows](#)
- [Installing R on Mac](#)

### Step 2 (recommended): Get RStudio

In addition to R, it's highly recommended that you install RStudio, which will make your experience with R much more enjoyable.

If you need to install RStudio, you can do so [here](#). Select the appropriate installer for your operating system.

### Step 3: Install swirl

Open RStudio (or just plain R if you don't have RStudio) and type the following into the console:

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

Note that the > symbol at the beginning of the line is R's prompt for you type something into the console. We include it here so you know that this command is to be typed into the console and not elsewhere. The part you type begins after >.

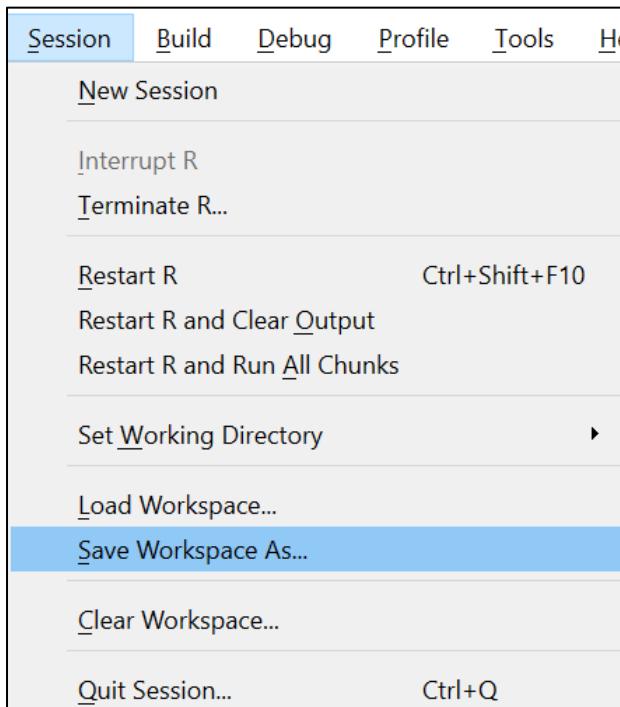
### Step 4: Start swirl

This is the only step that you will repeat every time you want to run swirl. First, you will load the package using the `library()` function. Then you will call the function that starts the magic! Type the following, pressing Enter after each line:

```
> library("swirl")
> swirl()
```

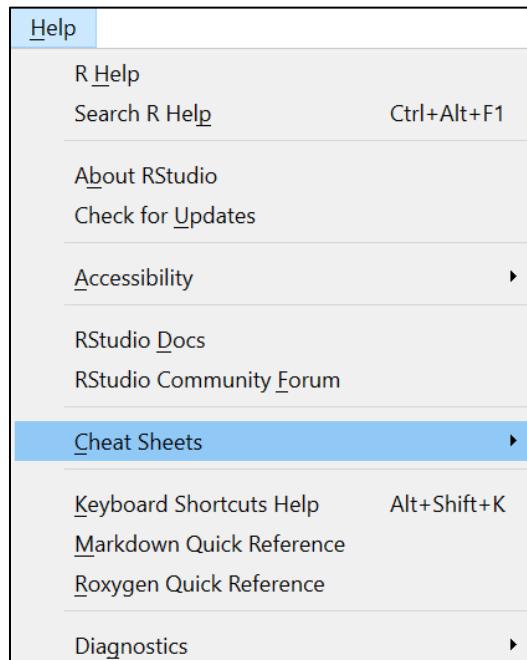
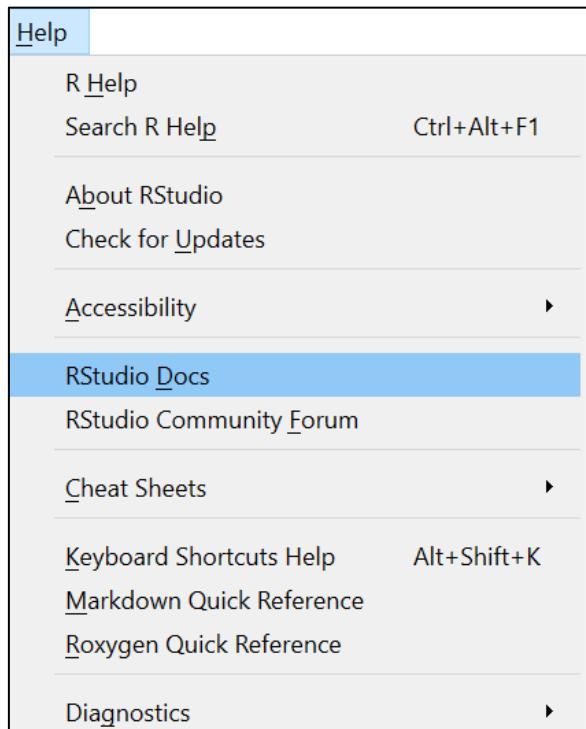
# Saving your R session

- Not really recommended if you already have a reproducible script.
- Save your session only if you want to keep all of your objects and current setting.



# R Documents and Cheat sheets

- Find resources in RStudio



The screenshot shows the RStudio Cheatsheets page at https://www.rstudio.com/resources/cheatsheets/. The page has a header with the R Studio logo and navigation links for Products, Solutions, Customers, Resources, About, and Pricing. Below the header, there's a section titled 'RStudio Cheatsheets' with a note about adding new cheatsheets and a 'SUBSCRIBE TO CHEATSHEET UPDATES' button. There are three main sections displayed: 'Data visualization with ggplot2 cheatsheet', 'RStudio IDE Cheat Sheet', and 'with dplyr'. Each section includes a thumbnail image of the cheat sheet and a brief description. At the bottom, there's a 'Browse Cheat Sheets...' button.

# Free healthcare dataset for practice

1. **Canada COVID-19 cases.** <https://health-infobase.canada.ca/covid-19/epidemiological-summary-covid-19-cases.html>
2. **Canada's SARS-CoV-2 virus sequences.** <https://virusseq-dataportal.ca/>
3. **BC Data Catalogue.** <https://catalogue.data.gov.bc.ca/>
4. **HealthData.gov.** <https://healthdata.gov/browse?limitTo=datasets&sortBy=newest&utf8=%E2%9C%93>
5. **Data.world.** <https://data.world/datasets/health>

The collage consists of five screenshots arranged in a grid-like layout:

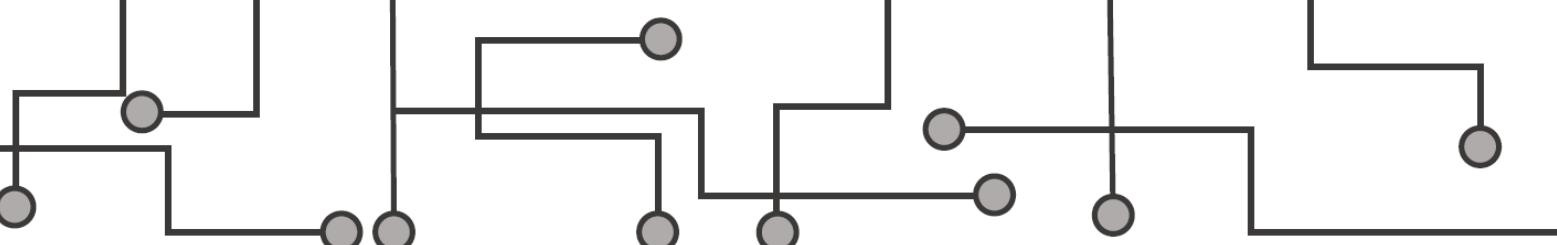
- Screenshot 1:** Canadian VirusSeq Data Portal. A yellow banner at the top says "Scheduled Maintenance". Below it is a large image of a scientist in a lab coat and mask holding a petri dish. Text on the page includes "CanCOGeN + VirusSeq", "About VirusSeq Data Portal", "Explore VirusSeq Data", "Meet the Team", and "Canadian VirusSeq Data Portal".
- Screenshot 2:** HealthInfobase COVID-19 epidemiological summary. It shows a map of Canada with case counts by province/territory and a bar chart titled "Count of cases (last 7 days) of COVID-19 in Yukon was 147 as of January 31, 2022".
- Screenshot 3:** BC Data Catalogue. It features a blue header with the BC logo and "Data Catalogue". Below the header are three buttons: "Browse thousands of government datasets", "Access data in a wide variety of file formats", and "Login with IDIR and gain access to even more data".
- Screenshot 4:** HealthData.gov COVID 20220110 FAQ Template v10. This is a template for daily data reports on testing, capacity and utilization, and patient flows. It includes fields for "Authority" (Official, Community), "Categories" (Hospital), and "Dataset" details like "Created January 6, 2022", "Views 3,105", and "Tags covid-19, hospital".
- Screenshot 5:** Data.world homepage. It features a main message "There are 3413 health datasets available on data.world.", a section for "TOP OPEN DATA TOPICS" (e.g., "Cancer Rates by U.S. State", "active (3006)", "transportation (3924)", etc.), and a sidebar with "How data.world helps you." and "Find fantastic data."

# Helpful resources to get started

1. **Swirl.** <https://swirlstats.com/students.html>
2. **R for Data Science.** <https://r4ds.had.co.nz/>
3. **R Basics:** <http://www.sthda.com/english/wiki/r-basics-quick-and-easy>
4. **Introduction to R 2021.** [https://biocorecrg.github.io/CRG\\_RIntroduction/](https://biocorecrg.github.io/CRG_RIntroduction/)
5. **Introduction to Data Exploration and Analysis with R.**  
<https://bookdown.org/mikemahoney218/IDEAR/>
6. **Hands-On-Programming with R.** <https://rstudio-education.github.io/hopr/>
7. **Computational Genomics with R.** <https://compgenomr.github.io/book/>

## R Markdown

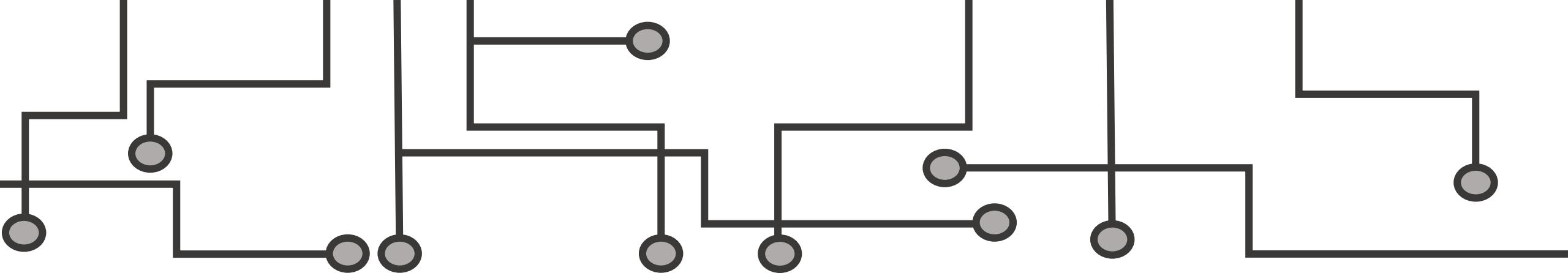
1. **R Markdown.** <https://rmarkdown.rstudio.com/lesson-1.html>
2. **RMarkdown for Scientists.** <https://rmd4sci.njtierney.com/>
3. **Getting Used to R, RStudio, and R Markdown.** <https://ismayc.github.io/rbasics-book/index.html>



# MMID CODING WORKSHOP

## SCHEDULE

| DATE        | INSTRUCTOR       | TOPIC                                                                 |
|-------------|------------------|-----------------------------------------------------------------------|
| January 12  | Grace E. Seo     | Introduction to BASH                                                  |
| January 19  | Jill Rumore      | Introduction to CONDA                                                 |
| January 26  | Aaron Petkau     | Downloading and assembling microbial sequence data                    |
| February 2  | Grace E. Seo     | Introduction to R                                                     |
| February 9  | Molly Pratt      | Tidy data: combining and transforming data in R                       |
| February 16 | Samantha Lee     | Using ggplot to visualize data and statistical results                |
| February 23 | Mackenzie Wilke  | Data visualization using antiviral drug-repurposing results from CLUE |
| March 2     | Taylor Davedow   | Data visualization using ggtree                                       |
| March 9     | Jessy Slota      | RNA-seq data analysis in R                                            |
| March 16    | Vasena Jayamanna | Introduction to machine learning in R                                 |



**THANK YOU FOR ATTENDING!**  
***The Q&A Session will now begin.***

**Please make sure to fill out the [Exit Survey](#)**  
**We value your feedback!**

**More questions? Please email us at  
[mmid.coding.workshop@gmail.com](mailto:mmid.coding.workshop@gmail.com) or post them to the workshop slack channel**