Intro to R and Bioconductor

HMS Research Computing

William_Rodriguez@hms.harvard.edu

Please fill out the survey

- Accessible through the Harvard Training Portal
  https://trainingportal.harvard.edu
- Click on “Me” ➡ “Intro to R and Bioconductor”
- Scroll to “Evaluations” and click on the survey

We appreciate any feedback or comments!

Contact Information

Email: rchelp@hms.harvard.edu

Website: https://it.hms.harvard.edu/our-services/research-computing

Wiki: https://wiki.rc.hms.harvard.edu/display/O2

Phone: 617-432-2000 (HMS IT Service Desk, 8a-5p)

Twitter: @hms_rc

Location: Gordon Hall 500, 5th Floor, 25 Shattuck Street

- https://rc.hms.harvard.edu/office-hours/ for Zoom web conferencing during remote work

Office hours: Wednesdays 1-3p for pressing needs, but appointments encouraged.

Course Objectives

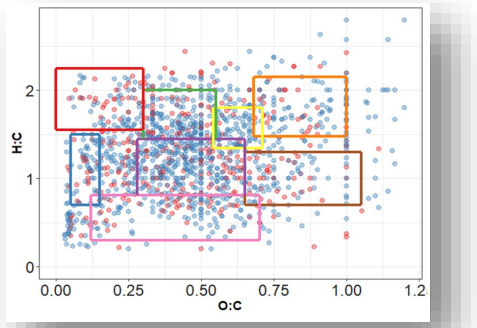
- Gain familiarity with R and Bioconductor
- Learn how to launch RStudio on O2
- Class Exercise

Blue content: try it out!

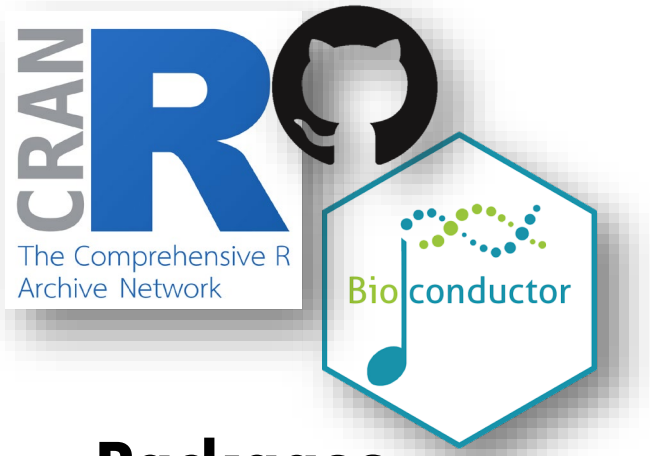
R is a powerful software

```
> set.seed(0)
> Spenders.Cleve <- rnorm(50, mean = 300, sd = 70)
> Spenders.NY <- rnorm(50, mean = 350, sd = 70)
> Amount.Spent <- c(Spenders.Cleve, Spenders.NY)
> city.name <- c(rep("Cleveland", 50), rep("New York", 50))
> t.test(Amount.Spent ~ city.name, var.equal = TRUE)

        Two Sample t-test

data:  Amount.Spent by city.name
t = -4.0115, df = 98, p-value = 0.0001179
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -74.47029 -25.17621
sample estimates:
mean in group Cleveland  mean in group New York
               301.6752                351.4984
```

Statistical Computing



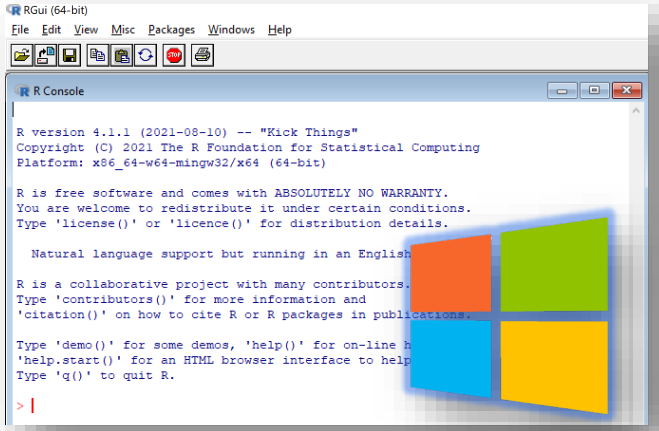
Graphics



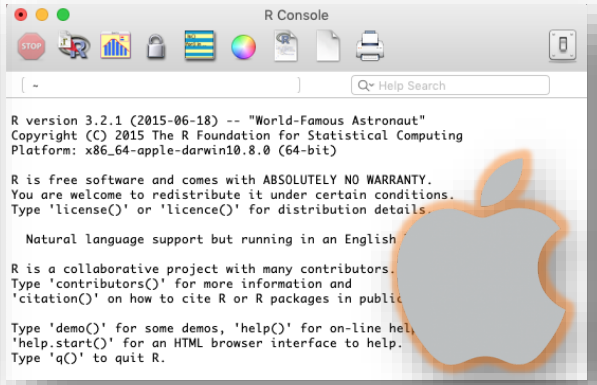
Packages

Community

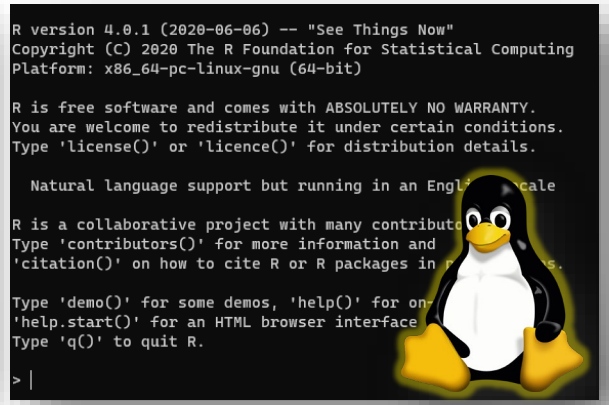
R runs & compile on most OS



Windows

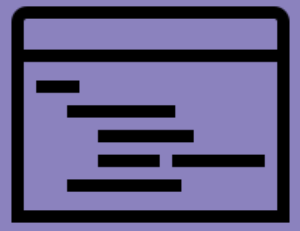


MacOS



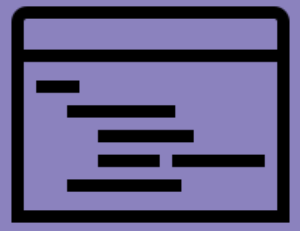
Linux

What are the options?



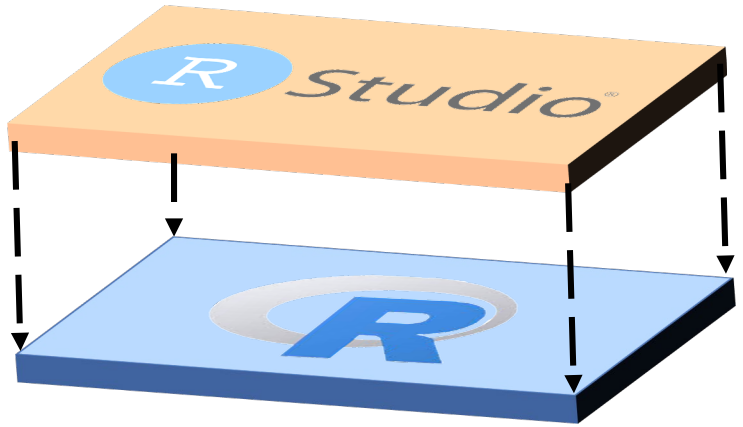
- Personal computer

- O_2
  - Interactive (via srun)
  - Non-interactive (via sbatch)
  - RStudio IDE (via srun + x11)

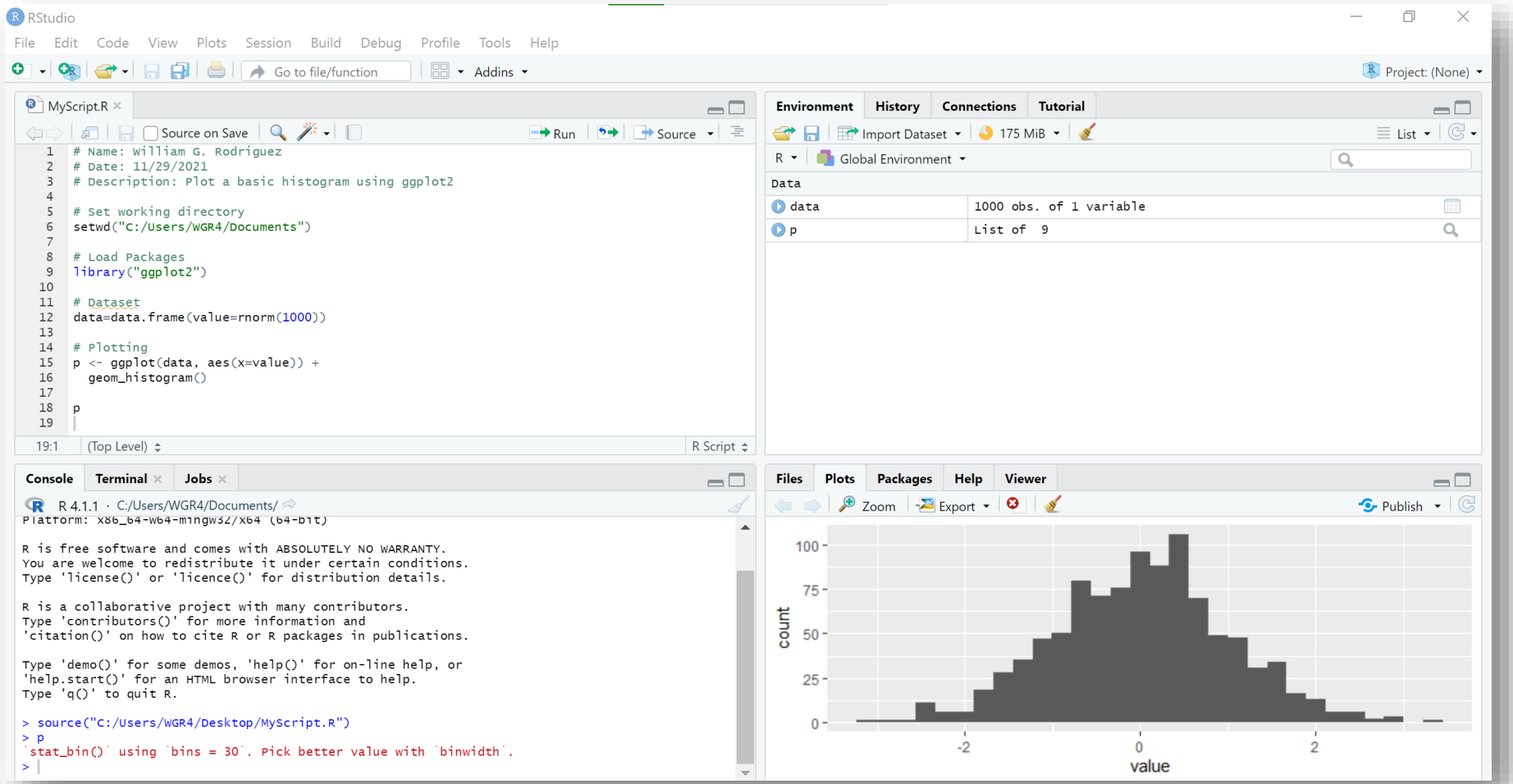


RStudio IDE

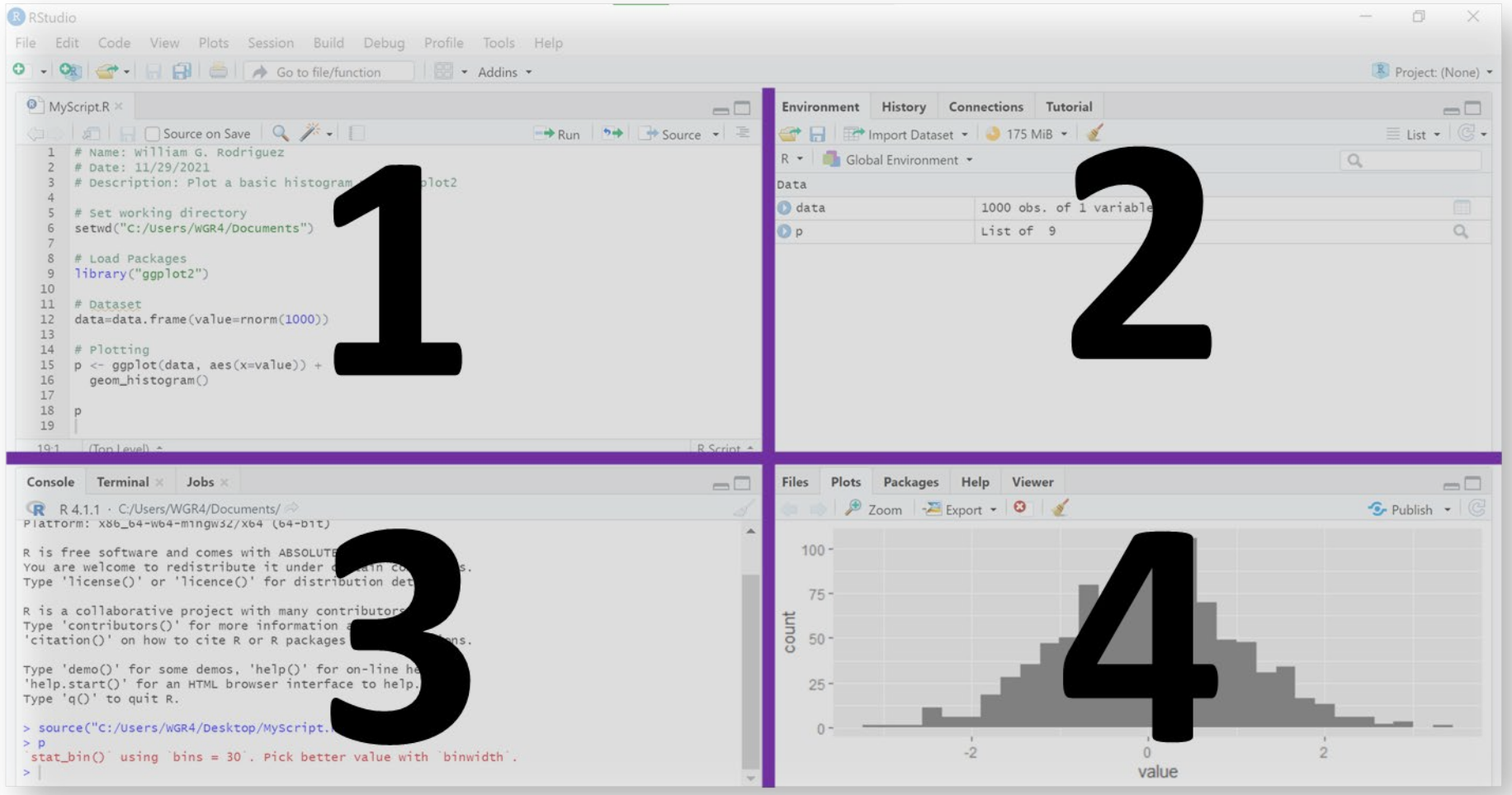
- Clean GUI
- Community Support
- Integrations
- Syntax highlighting
- Autocompletion
- Free*



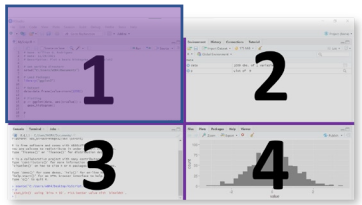
RStudio quadrants



RStudio quadrants



Syntax highlighting



RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins

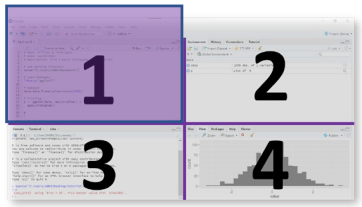
MyScript.R

Source on Save | Run | Source

```
1  # Name: William G. Rodriguez
2  # Date: 11/29/2021
3  # Description: Plot a basic histogram using ggplot2
4
5  # Load Packages
6  library("ggplot2")
7
8  # Dataset
9  data=data.frame(value=rnorm(1000))
10
11 # Plotting
12 p <- ggplot(data, aes(x=value)) +
13   geom_histogram()
14
15 p
```

15:2 (Top Level) R Script

Comments are important



Basic Info

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins

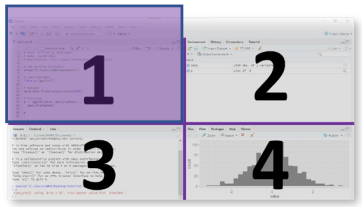
MyScript.R

Source on Save | Run | Source

```
1  # Name: William G. Rodriguez
2  # Date: 11/29/2021
3  # Description: Plot a basic histogram using ggplot2
4
5  # Load Packages
6  library("ggplot2")
7
8  # Dataset
9  data=data.frame(value=rnorm(1000))
10
11 # Plotting
12 p <- ggplot(data, aes(x=value)) +
13   geom_histogram()
14
15 p
```

15:2 (Top Level) R Script

Load packages at the top



Load package(s)

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins

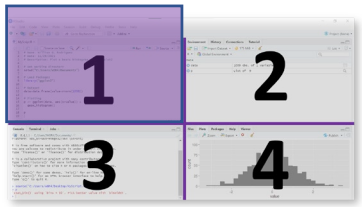
MyScript.R

Source on Save | Run | Source

```
1  # Name: William G. Rodriguez
2  # Date: 11/29/2021
3  # Description: Plot a basic histogram using ggplot2
4
5  # Load Packages
6  library("ggplot2")
7
8  # Dataset
9  data=data.frame(value=rnorm(1000))
10
11 # Plotting
12 p <- ggplot(data, aes(x=value)) +
13   geom_histogram()
14
15 p
```

15:2 (Top Level) R Script

Import your data into R



Load data

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins

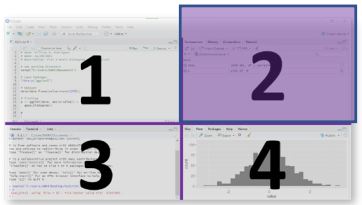
MyScript.R

Source on Save | Run | Source

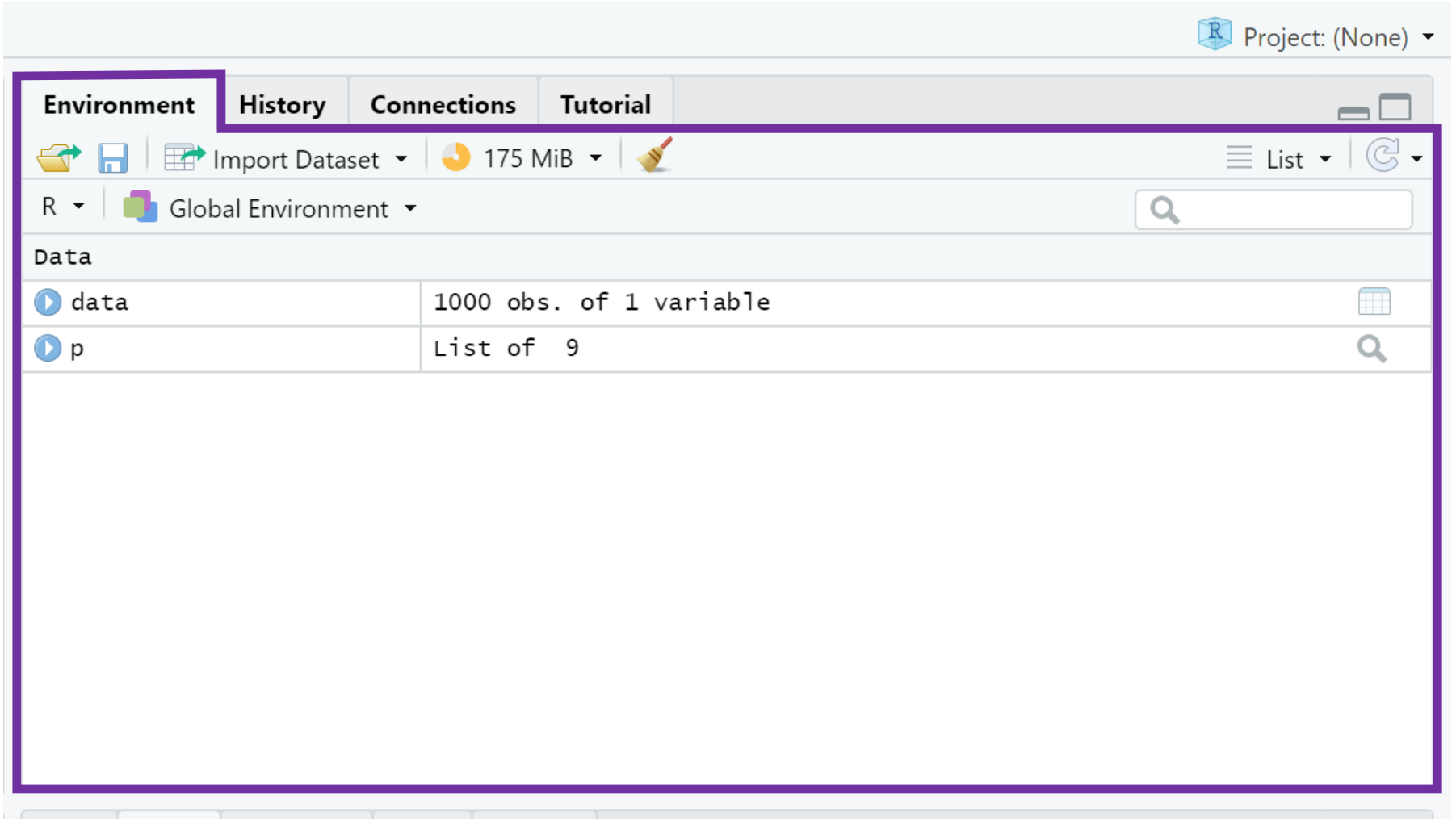
```
1  # Name: William G. Rodriguez
2  # Date: 11/29/2021
3  # Description: Plot a basic histogram using ggplot2
4
5  # Load Packages
6  library("ggplot2")
7
8  # Dataset
9  data=data.frame(value=rnorm(1000))
10
11 # Plotting
12 p <- ggplot(data, aes(x=value)) +
13   geom_histogram()
14
15 p
```

15:2 (Top Level) R Script

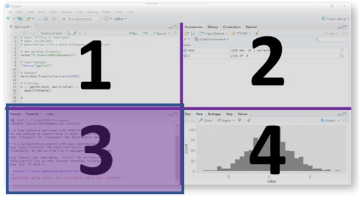
Objects loaded on R workspace



R objects



Display entered commands & errors



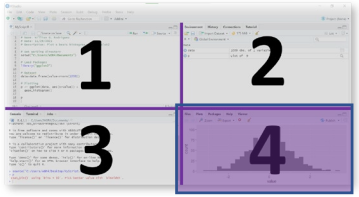
R console

Console | Terminal | Jobs

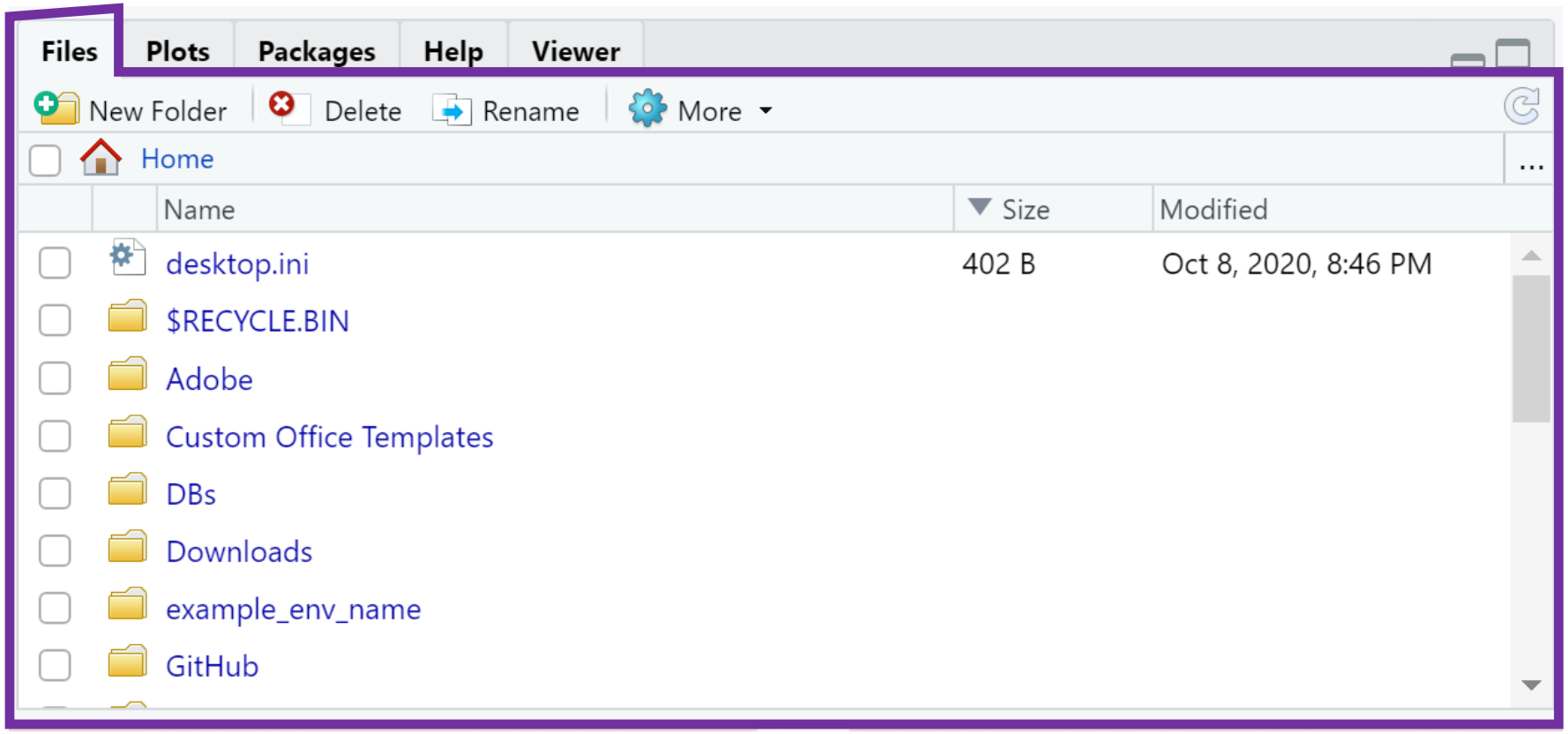
R 4.1.1 · C:/Users/WGR4/Documents/

```
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.

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.

> source("C:/Users/WGR4/Desktop/MyScript.R")
> p
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
>
```

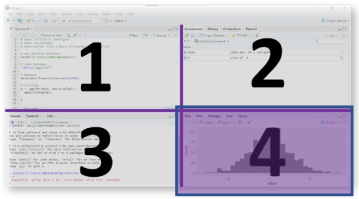
Manage files and folders



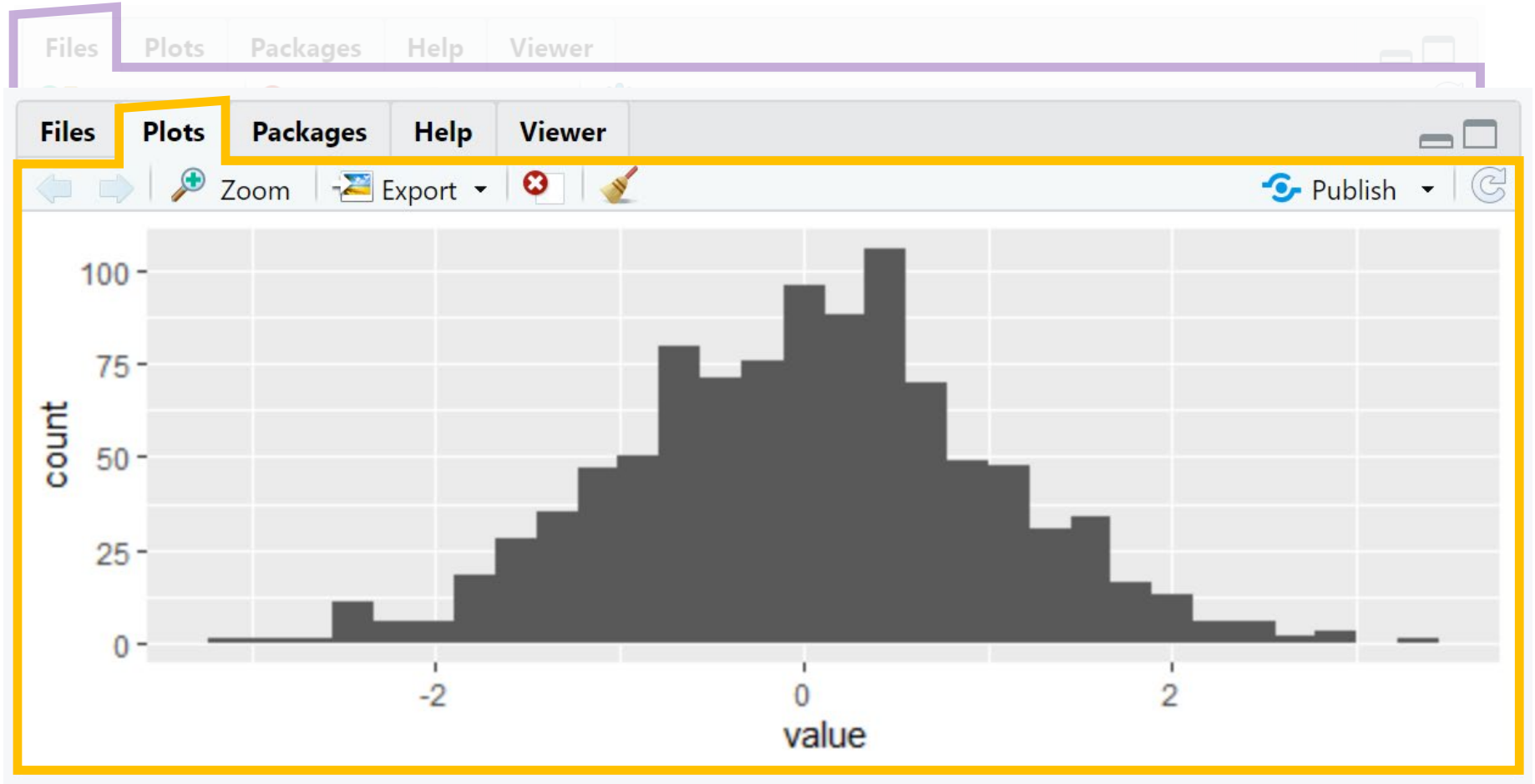
File browser



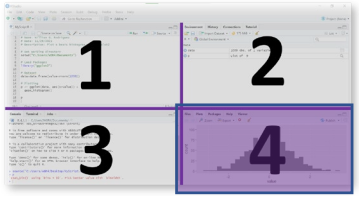
Graphical output from R script



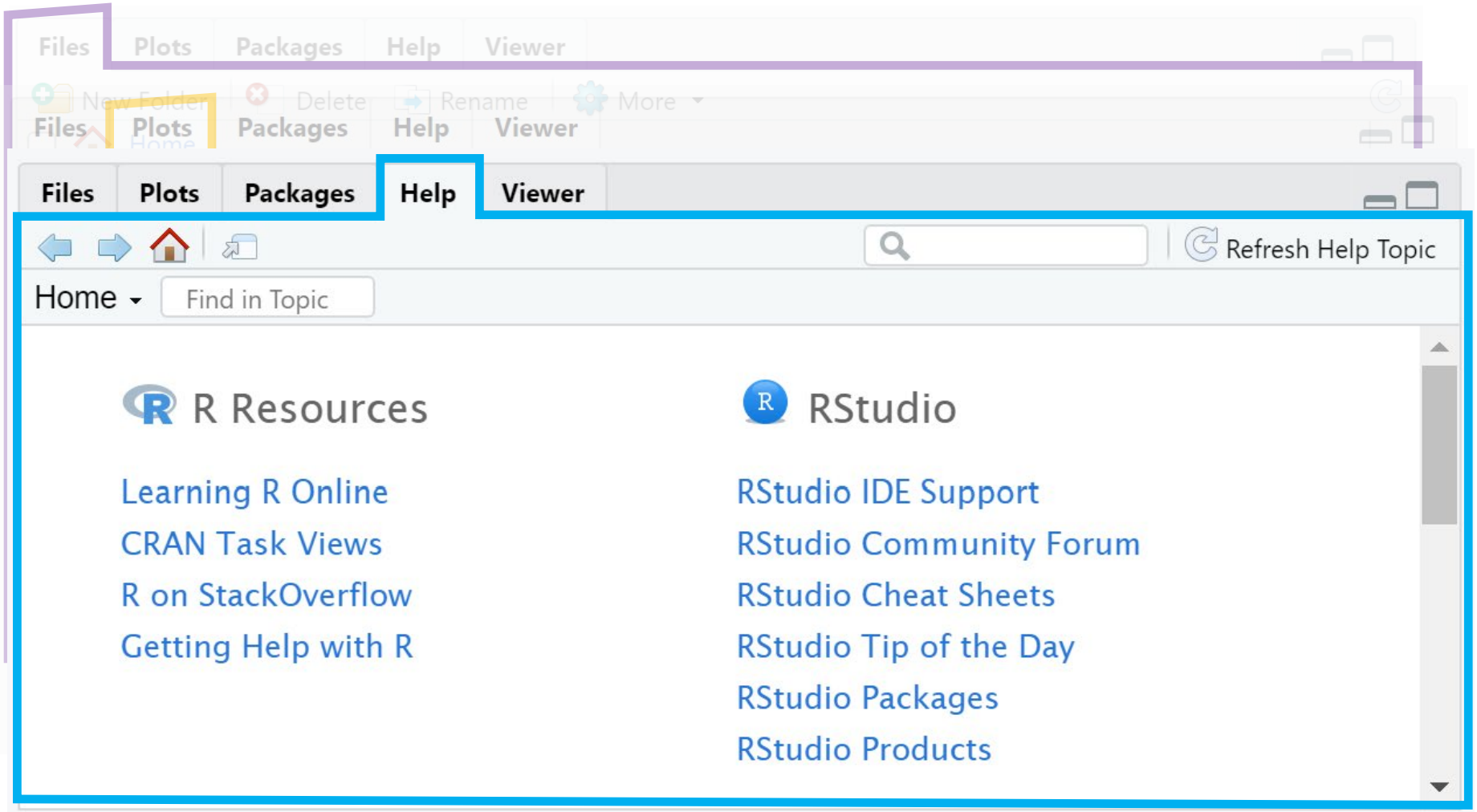
Graph viewer



Documentation for R & RStudio



Resources



Packages

R repository - CRAN

CRAN R
The Comprehensive R Archive Network

Packages -18,993
Default repository

install.packages("ggplot2")

source: https://cran.r-project.org/web/packages/
updated on: 4/19/2022

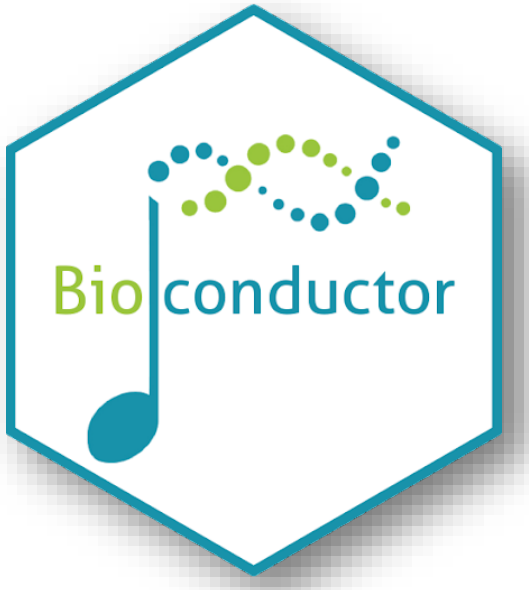
R packages - Source

Compressed file
Old or in-house packages

install.packages("/path/to/file", repos=NULL)

Packages

R repository - Bioconductor



Packages – 2,083

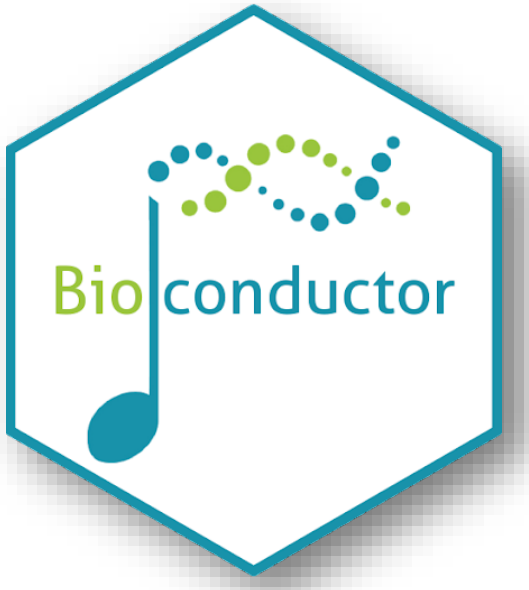
Bioinformactics

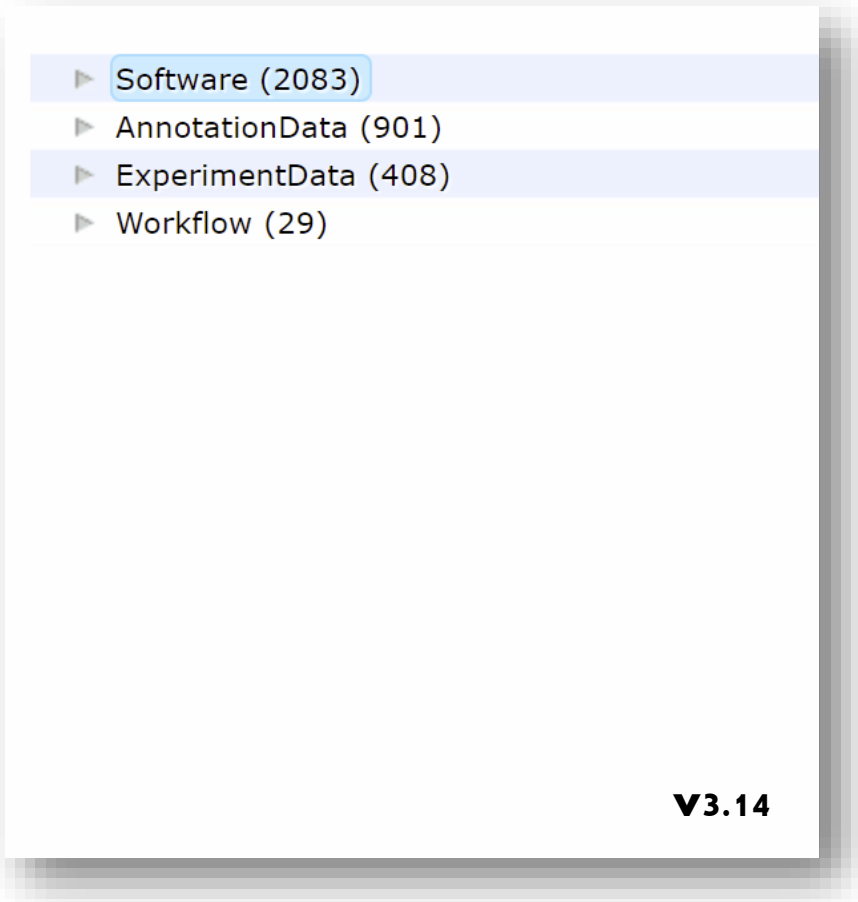
```
install.packages("BiocManager")
BiocManager::install("edgeR")
```

source: https://bioconductor.org/packages/release updated on: 4/19/2022

Packages

Bioconductor Categories





source: https://bioconductor.org/packages/release updated on: 4/19/2022

R repositories - GitHub

GitHub

+

devtools

Dev releases & new features

```
install.packages("devtools")
devtools::install_github("tidyverse/ggplot2@v.3.3.3")
```

x11

Logging Into O2

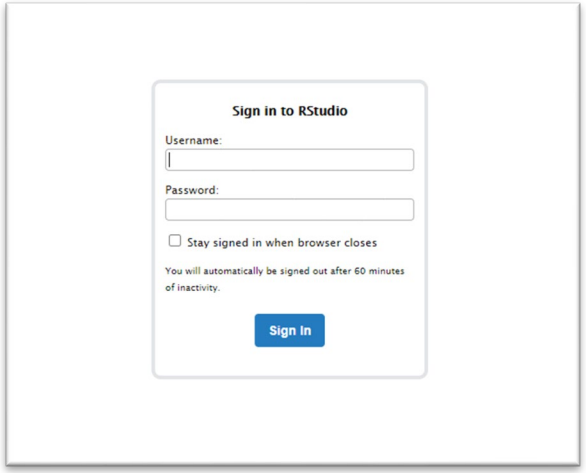
While running XQuartz on the background, open a terminal (type “terminal” on the search bar)

Open a terminal (type “terminal” on the search bar)

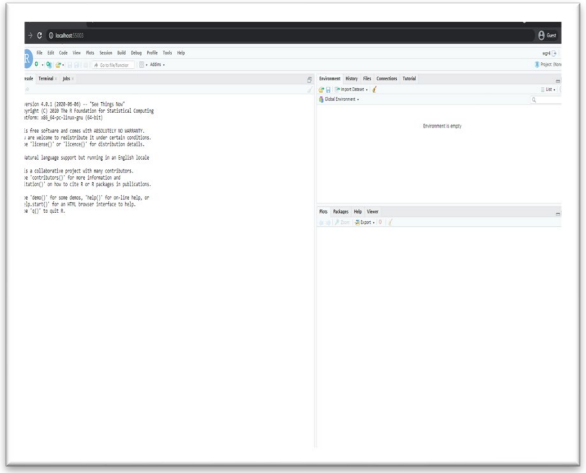
Open MobaXterm

Logging Into O2

Launch RStudio



Authenticate



Open in Browser

Logging Into O2

Launch RStudio

ssh -Y -L PORT:127.0.0.1:PORT <your_HMS_ID>@o2.hms.harvard.edu

module load gcc/6.2.0 R/4.1.1 git/2.9.5

srun -t 0-2:00 --pty -p interactive -c 1 --mem=2G --tunnel PORT:PORT RStudio_launcher.sh PORT

R on O2

- **Available versions**

```
$ module spider R
```

- **Load R module**

```
$ module load gcc/6.2.0 R/version
```

- **How to unload an R module**

```
$ module unload R/version
```

- **Important: start R from an interactive session**

```
$ R
```

Managing R packages on O2

- An R Personal Library is required on O2
- You must create an R Personal Library per version.
- It can be done in two steps

Managing R packages on O2

- An R Personal Library is required on O2
- You must create an R Personal Library per version.
- It can be done in two steps

1) **Create an R Personal Library directory**

```
$ mkdir ~/R-4.1.1
```

2) **Create an .Renviron file**

```
$ echo 'R_LIBS_USER="~/R-4.1.1"' > $HOME/.Renviron'
```

Launch RStudio on O2

```
$ srun -t 0-2:00 --pty -p interactive -c 1 --mem=2G --tunnel PORT:PORT RStudio_launcher.sh PORT
```

SLURM command to obtain a job allocation

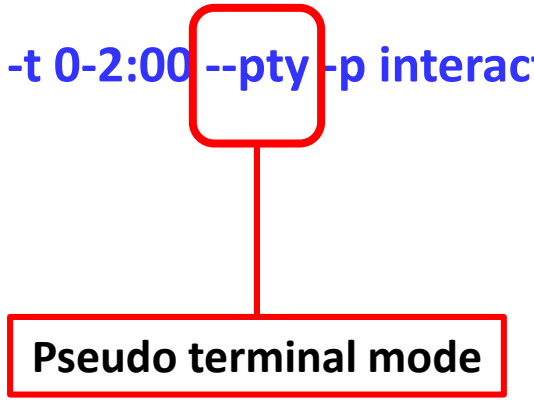
Launch RStudio on O2

```
$ srun -t 0-2:00 --pty -p interactive -c 1 --mem=2G --tunnel PORT:PORT RStudio_launcher.sh PORT
```

Walltime
(DD-HH:MM)

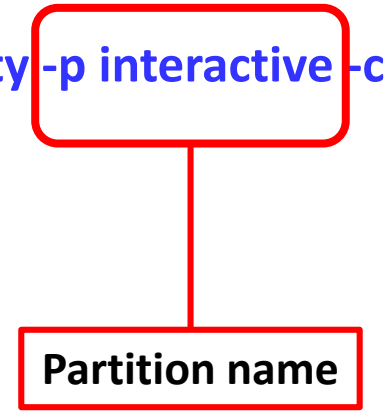
Launch RStudio on O2

```
$ srun -t 0-2:00 --pty -p interactive -c 1 --mem=2G --tunnel PORT:PORT RStudio_launcher.sh PORT
```



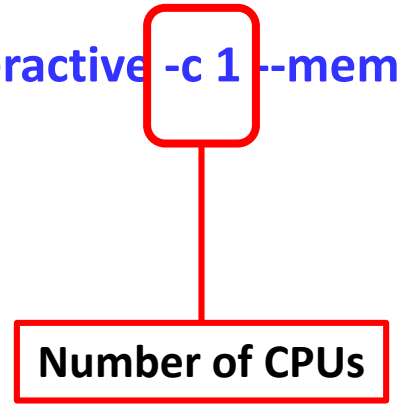
Launch RStudio on O2

```
$ srun -t 0-2:00 --pty -p interactive -c 1 --mem=2G --tunnel PORT:PORT RStudio_launcher.sh PORT
```



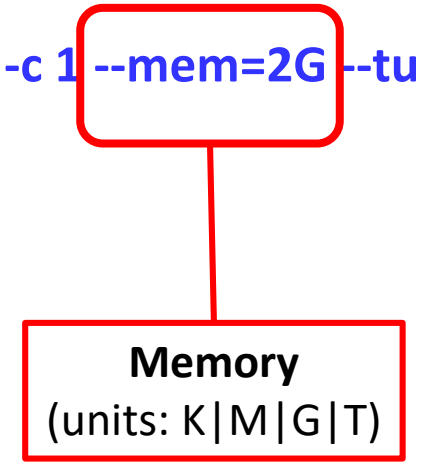
Launch RStudio on O2

```
$ srun -t 0-2:00 --pty -p interactive -c 1 --mem=2G --tunnel PORT:PORT RStudio_launcher.sh PORT
```



Launch RStudio on O2

$ srun -t 0-2:00 --pty -p interactive -c 1 --mem=2G --tunnel PORT:PORT RStudio_launcher.sh PORT



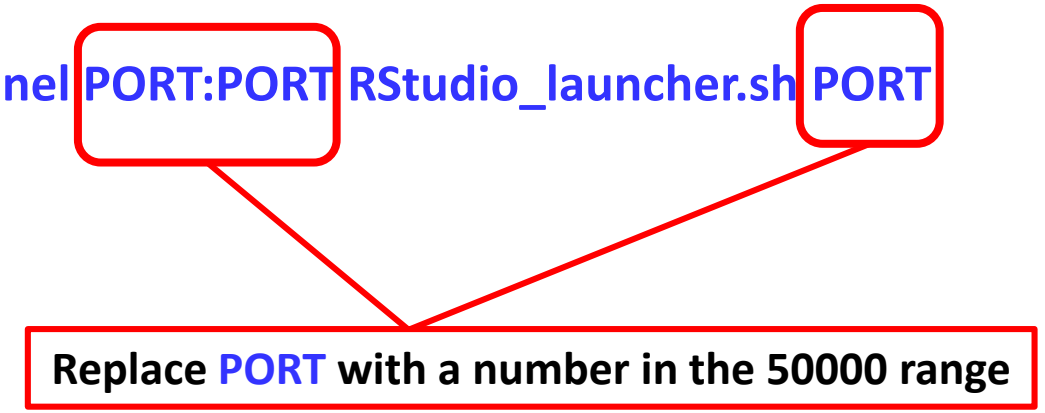
Launch RStudio on O2

```
$ srun -t 0-2:00 --pty -p interactive -c 1 --mem=2G --tunnel PORT:PORT RStudio_launcher.sh PORT
```

Required to execute RStudio launcher

Launch RStudio on O2

```
$ srun -t 0-2:00 --pty -p interactive -c 1 --mem=2G --tunnel PORT:PORT RStudio_launcher.sh PORT
```



Launch RStudio on O2

$ srun -t 0-2:00 --pty -p interactive -c 1 --mem=2G --tunnel PORT:PORT RStudio_launcher.sh PORT

Replace PORT with a number in the 50000 range

```
[wgr4@login06 ~]$ srun -t 0-1:00 --pty -p interactive -c 1 --mem=2G --x11 --tunnel 55003:55003 RStudio_launcher.sh 55003
srun: job 31216484 queued and waiting for resources
srun: job 31216484 has been allocated resources
You can now access RStudio on your local web brower at http://localhost:55003
Login username = wgr4
Password = gqYRIyfRie0y4v9QgaeQ
```

RStudio on O2

1. Connect to O2

```
$ ssh -Y -L $PORT:127.0.0.1:$PORT <your_HMS_ID>@o2.hms.harvard.edu
```

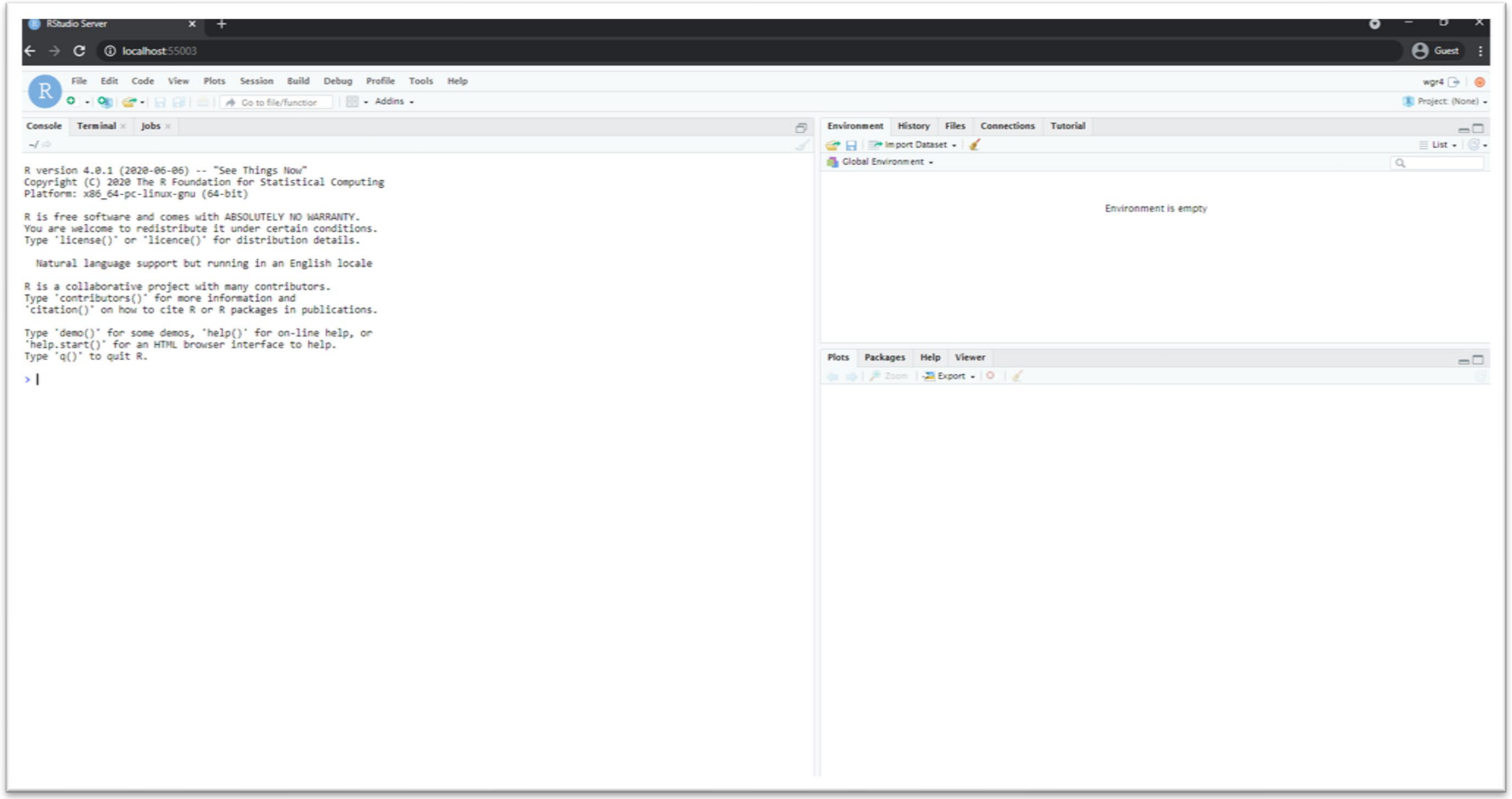
2. Load Modules

```
$ module load rstudio_launcher/1.0
$ module load gcc/6.2.0
$ module load R/4.1.1
```

3. Launch Rstudio

```
$ srun -t 0-2:00 --pty -p interactive -c 1 --mem=2G --tunnel $PORT:$PORT RStudio_launcher.sh $PORT
```

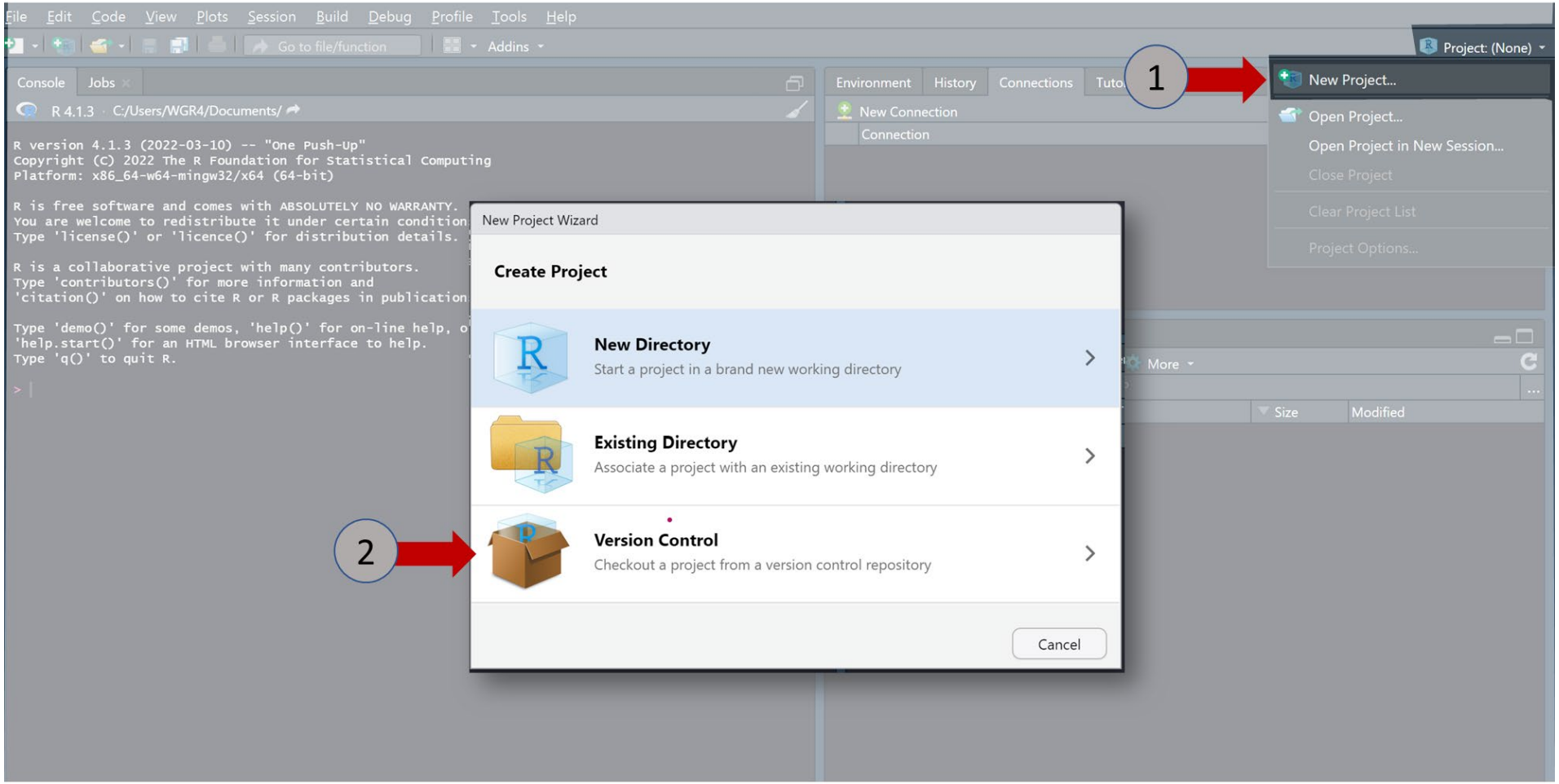
Launch RStudio on O2



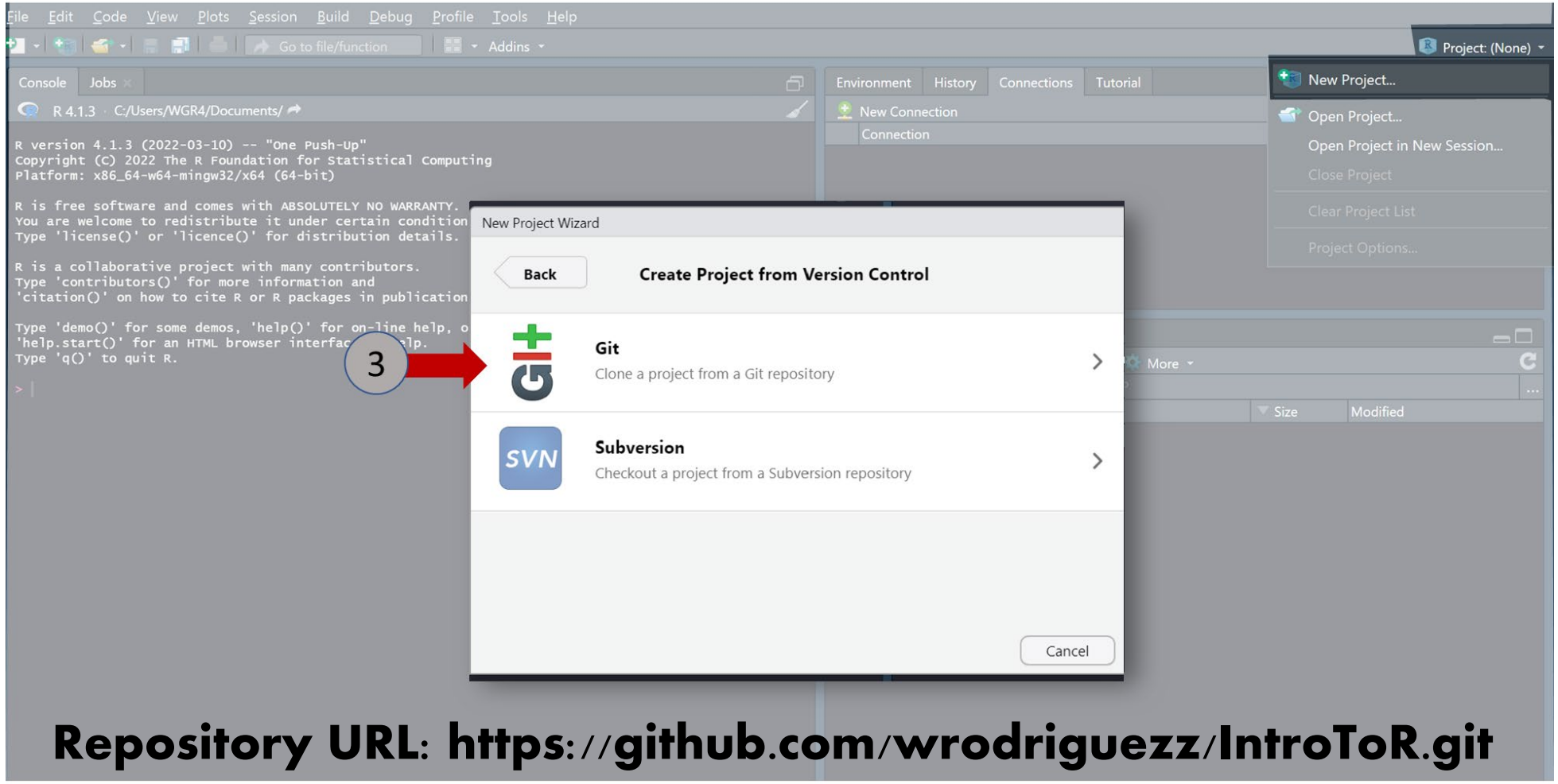
Projects in RStudio

- An easy way to divide your R work
- Projects have a unique working directory

Create an "IntroToR" Project



Create an "IntroToR" Project



Repository URL: https://github.com/wrodriguezz/IntroToR.git

- Data Types
- Data Wrangling
- Operators
- Data Import
- Workspace
- Exercise

Vectors

- Basic way to store data
- Types: numeric, character, & logical

Example:

```
> myvector <- c(3,5,7)
```

Lists

- Like vectors but with mixed data types

Example:

```
> myvector <- c(3,"Tp53",7)
```

Factors

- Mostly use in statistical modeling
- Levels will always be character values

Example:

```
> gender <- c("male", "male", "female")
> gender <- factor(gender)
```

Matrices

- Data must be all the same type
- Columns must have the same length

Example:

```
> mymatrix <- matrix(c(1:6), nrow=3, ncol=2)
```

Data frame

- Like matrices but allowing mixed data types
- Rows and Columns can be named

Example:

```
> mydataframe <- data.frame(L=letters[1:6],N=1:6)
```

Indexing

- **Accessing elements from a vector, matrix, or data frame**

Examples:

`> myvector [2]`	returns 2nd element from a vector
`> mymatrix[1,2]`	returns element in row 1 and column 2
`> mymatrix[1,]`	return all elements in row 1
`> mymatrix[,1]`	return all elements in column 1

Joining rows or columns

- **Add row(s) to a pre-existing data frame or matrix**

```
> mymatrix <- rbind(mymatrix, newrow)
```

- **Add column(s) to a pre-existing data frame or matrix**

```
> mymatrix <- cbind(mymatrix, newcol)
```

Missing values

- **NA: Not Available**
- **NaN: Not a Number**

Example:

```
> is.na(x)
> is.nan(x)
```

Change data type

- **Functions start with "as." followed by the type**

Example:

```
> myvector <- c(3,5,7)
> myvector <- as.character(myvector)
```

Apply function

- Apply a function to a matrix [or data frame] over rows or columns
- Format: apply (to_what, *how*, *function*)

how : "1" apply the *function* over rows

- or -

"2" apply the *function* over columns

For example:

```
> apply(mymatrix, 1, sum) #row sums
> apply(mymatrix, 2, sum) #column sums
```

- Data Types
- Data Wrangling
- Operators
- Data Import
- Workspace
- Exercise

Other useful functions

```
> class(object) #gives object class
> mode(object) #gives object type
> length(vector) #gives length
> head(object) #gives first 6 rows
> tail(object) #gives last 6 rows
> nrow(object) #gives number of rows
> ncol(object) #gives number of columns
> str(object) #gives object structure
```

Arithmetic

Operator	Description
+	Addition
-	Subtraction
*	Multiplication
/	Division
^	Exponent
%%	Modulo (remainder)

Logical

Operator	Description
<	Less than
<=	Less than or equal to
>	Greater than
>=	Greater than or equal to
==	Exactly equal to
!=	Not equal to
|	OR
&	AND

Text file

Example:

```
mydata <- read.table(file=" filename.csv", header=TRUE, sep=",")
```

- "sep=" field separator character
- "header=" logical value
- "row.names=" row names; must be unique

MS Excel

- R package is required (e.g., xlsx)
- Read in the first worksheet from a workbook

```
> mydata <- read.xlsx("myexcel.xlsx", sheetIndex=1)
```

- Read in the worksheet named "mysheet"

```
> mydata <- read.xlsx("myexcel.xlsx", sheetName = "mysheet")
```

Export

Format:

```
> write.table(x=ObjectName, file="FileName.txt", sep="\t")
```

Optional arguments:

```
row.names=FALSE #turn off row names
col.names=FALSE #turn off column names
quote=FALSE #turn off character string quoting
```

"

Data Types

Data Wrangling

Operators

Data Import

Workspace

Exercise

Manage R workspace

- Save R workspace to a file

```
> save.image(file="FileName.RData")
```

- Save R object(s) to a file

```
> save(object list, file="FileName.RData")
```

- Load workspace or R object

```
> load(file="FileName.RData")
```

- Data Types
- Data Wrangling
- Operators
- Data Import
- Workspace
- Exercise

Import Data

- Import the Rcoursetestdata1.csv dataset as a comma separator file. Set the first column as the row.names and account for the headers

```
> head(mydf)
                 TNBC1 TNBC2  TNBC3 Normal1 Normal2 Normal3
ENSG00000008988 15258 15077 144720   12095   43544   46883
ENSG00000009307 14660 20767   8678   13774   23030   18917
ENSG00000019582 50866 55775  15089    6696   13754   86319
ENSG00000026025 21174 47966  26682    6068   21126   12728
ENSG00000034510 25645 31574  56403   29590   25216   37199
ENSG00000044574 23910 27200  13757   13364   10852   12378
```

- Data Types
- Data Wrangling
- Operators
- Data Import
- Workspace
- Exercise

Import Data

- Import the Rcoursetestdata1.csv dataset as a comma separator file. Set the first column as the row.names and account for the headers

```
> mdata <- "/n/groups/rc-training/introR/Rcoursetestdata1.csv"
> mydf <- read.table(mdata, header=TRUE,  row.names=1, sep=",")
> head(mydf)
```

Basic Statistics

- Return basic statistics

```
> summary(mydf)
```

Basic Statistics

- Return basic statistics

```
> summary(mydf)
```

```
> summary(mydf)
     TNBC1             TNBC2             TNBC3            Normal1
 Min.   :     0   Min.   :    65   Min.   :    31   Min.   :   22
 1st Qu.:  7888   1st Qu.:  9538   1st Qu.:  9324   1st Qu.: 5074
 Median : 13034   Median : 16568   Median : 19108   Median :10869
 Mean   : 18596   Mean   : 26036   Mean   : 25646   Mean   :14746
 3rd Qu.: 23850   3rd Qu.: 28194   3rd Qu.: 30389   3rd Qu.:18866
 Max.   :103007   Max.   :351603   Max.   :272582   Max.   :89837
    Normal2           Normal3
 Min.   :   208   Min.   :    15
 1st Qu.:  7124   1st Qu.:  8944
 Median : 14005   Median : 17710
 Mean   : 19425   Mean   : 25481
 3rd Qu.: 21576   3rd Qu.: 32191
 Max.   :212582   Max.   :244692
```

Transposing Data

- **Need your data to read the other way? Turn it into a matrix, and transpose!**

For example:

```
> mymatrix <- as.matrix(mydf)
> myTmatrix<- t(mymatrix) #t = transpose
> myTdf <- as.data.frame(myTmatrix) #as data frame again
```

Plotting with ggplot2

- To explore later
- Three general components
  - Data set
  - Coordinate system
  - Geoms

Plotting with ggplot2

- To explore later
- Three general components
  - Data set
  - Coordinate system
  - Geoms

```
> library ("ggplot2")

> data("midwest", package = "ggplot2")

> gg <- ggplot(midwest, aes(x=area, y=poptotal)) +
  geom_point(aes(col=state, size=popdensity)) +
  labs(y="Population",
       x="Area",
       title="Scatterplot",
       caption = "Source: midwest")
> gg
```

Plotting with ggplot2

- To explore later
- Three general components
  - Data set
  - Coordinate system
  - Geoms

```
> library ("ggplot2")

> data("midwest", package = "ggplot2")

> gg <- ggplot(midwest, aes(x=area, y=poptotal)) +
  geom_point(aes(col=state, size=popdensity)) +
  labs(y="Population",
       x="Area",
       title="Scatterplot",
       caption = "Source: midwest")
> gg
```

Plotting with ggplot2

- To explore later
- Three general components
  - Data set
  - Coordinate system
  - Geoms

```
> library ("ggplot2")

> data("midwest", package = "ggplot2")

> gg <- ggplot(midwest, aes(x=area, y=poptotal)) +
  geom_point(aes(col=state, size=popdensity)) +
  labs(y="Population",
       x="Area",
       title="Scatterplot",
       caption = "Source: midwest")
> gg
```

Data Types

Data Wrangling

Operators

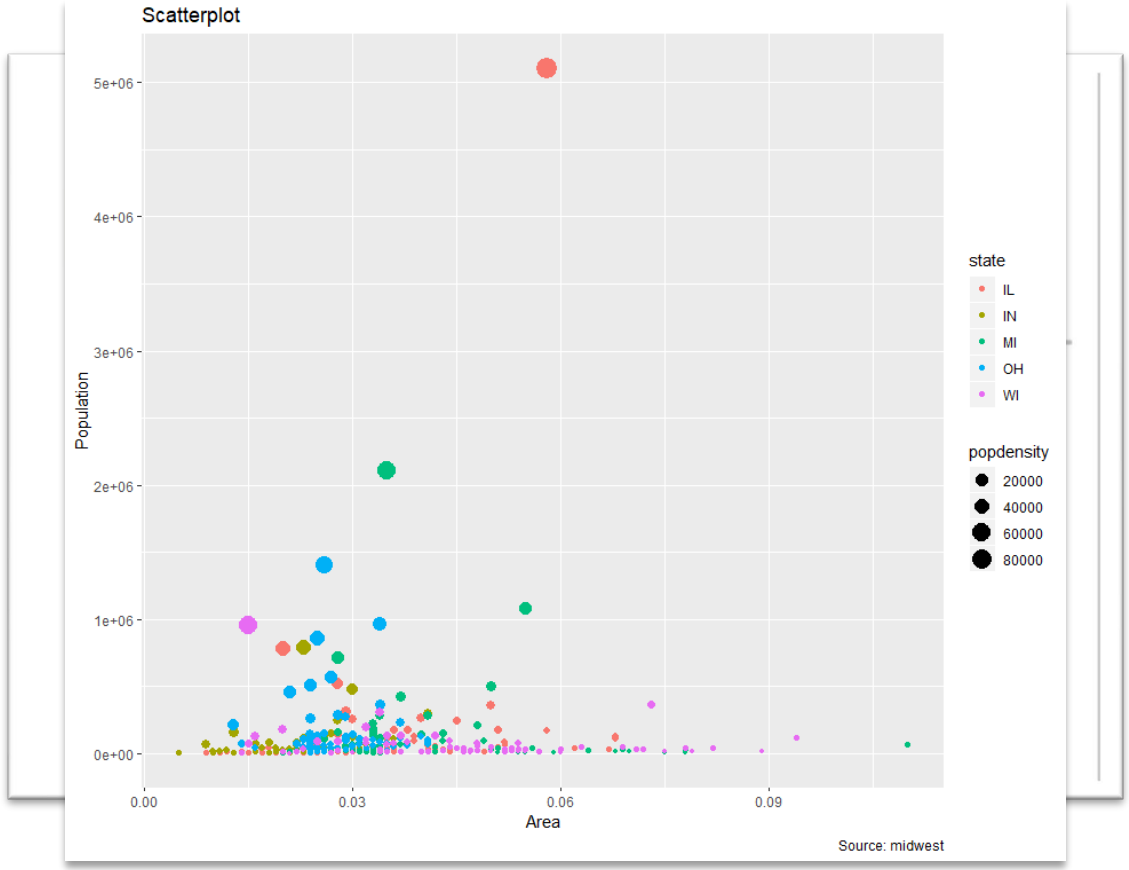
Data Import

Workspace

Exercise

Plotting with ggplot2

- To explore later
- Three general components
  - Data set
  - Coordinate system
  - Geoms



Data Types

Data Wrangling

Operators

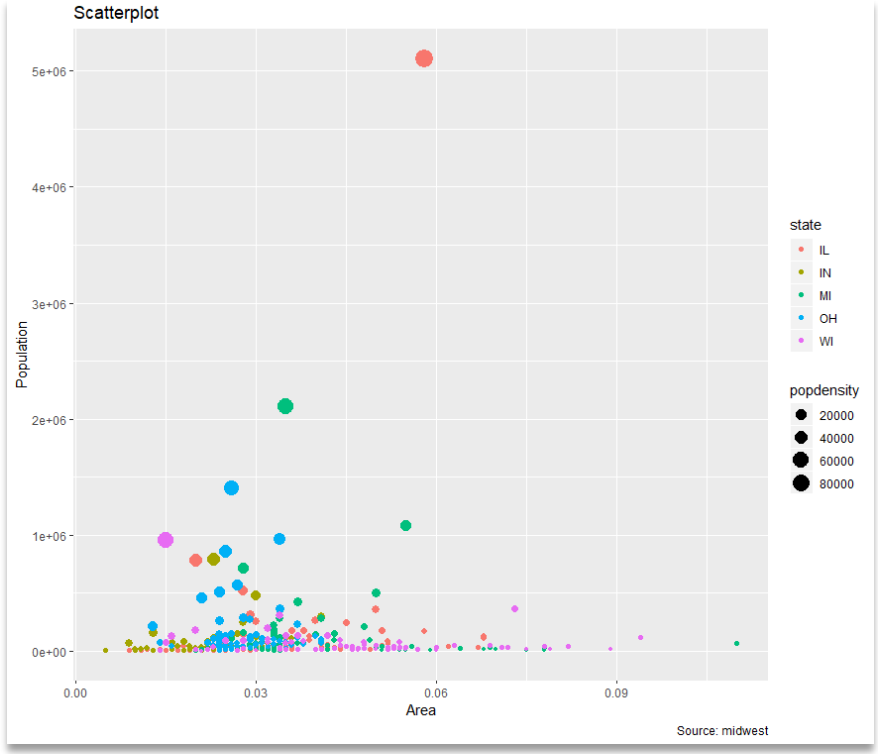
Data Import

Workspace

Exercise

Plotting with ggplot2

- **For more info: Chan Bioinformatics Core**



Contact Information

Email: rchelp@hms.harvard.edu

Website: https://it.hms.harvard.edu/our-services/research-computing

Wiki: https://wiki.rc.hms.harvard.edu/display/O2

Phone: 617-432-2000 (HMS IT Service Desk, 8a-5p)

Twitter: @hms_rc

Location: Gordon Hall 500, 5th Floor, 25 Shattuck Street

- https://rc.hms.harvard.edu/office-hours/ for Zoom web conferencing during remote work

Office hours: Wednesdays 1-3p for pressing needs, but appointments encouraged.