

# **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](mailto: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, 5<sup>th</sup> 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.

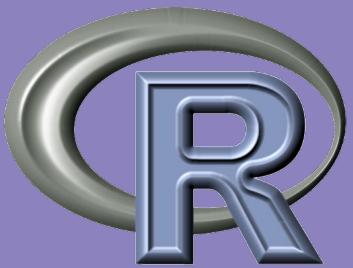


# **Course Objectives**

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



**Blue content: try it out!**



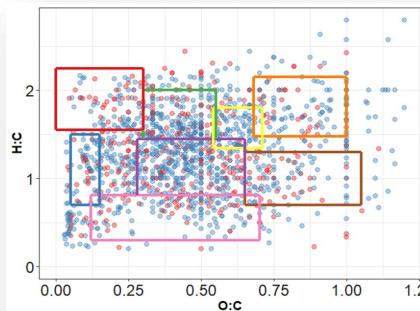
# R is a powerful software

```
> 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)

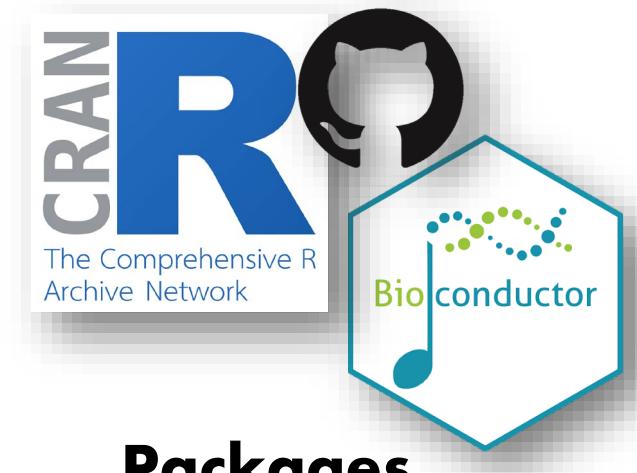
Two Sample t-test

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**

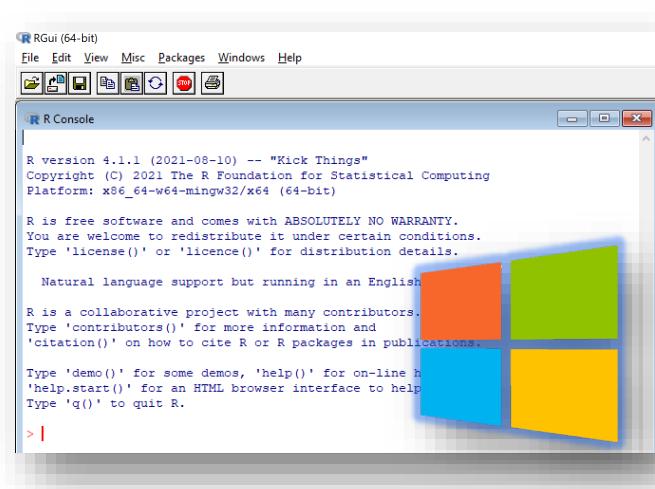


**Packages**

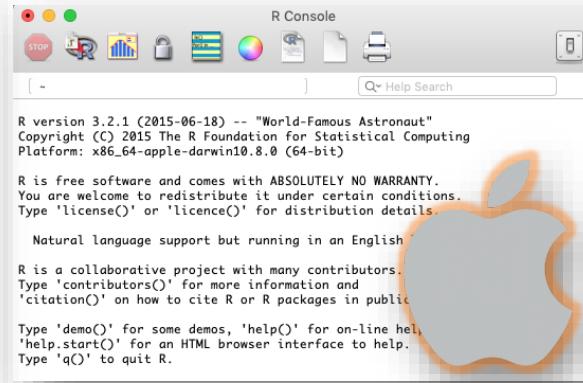


**Community**

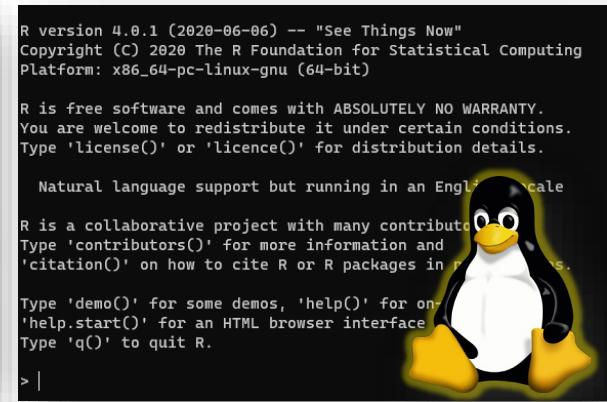
# R runs & compile on most OS



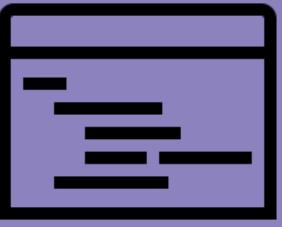
Windows



MacOS

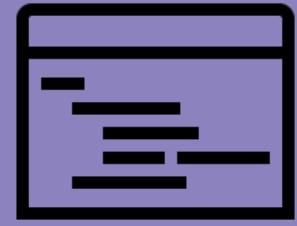


Linux



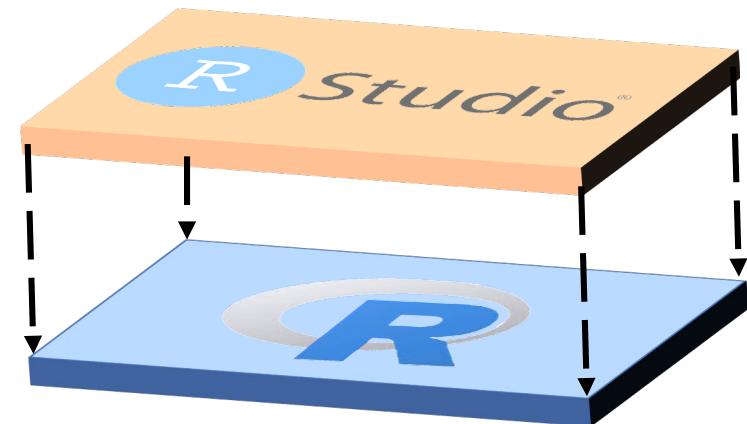
# What are the options?

- Personal computer
- O<sub>2</sub>
  - Interactive (via srun)
  - Non-interactive (via sbatch)
  - RStudio IDE
    - via srun + x11
    - O<sub>2</sub> Portal NEW

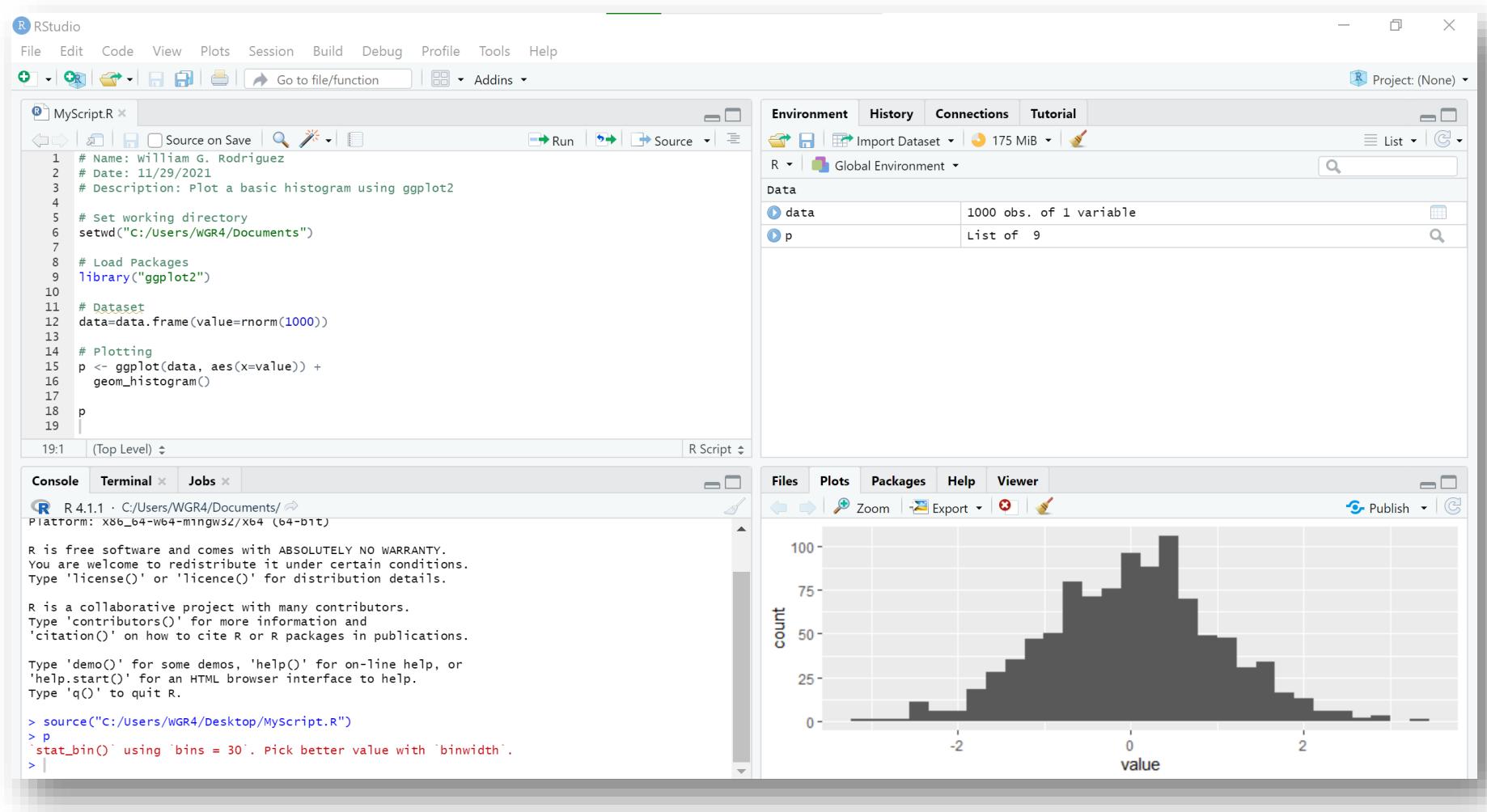


# RStudio IDE

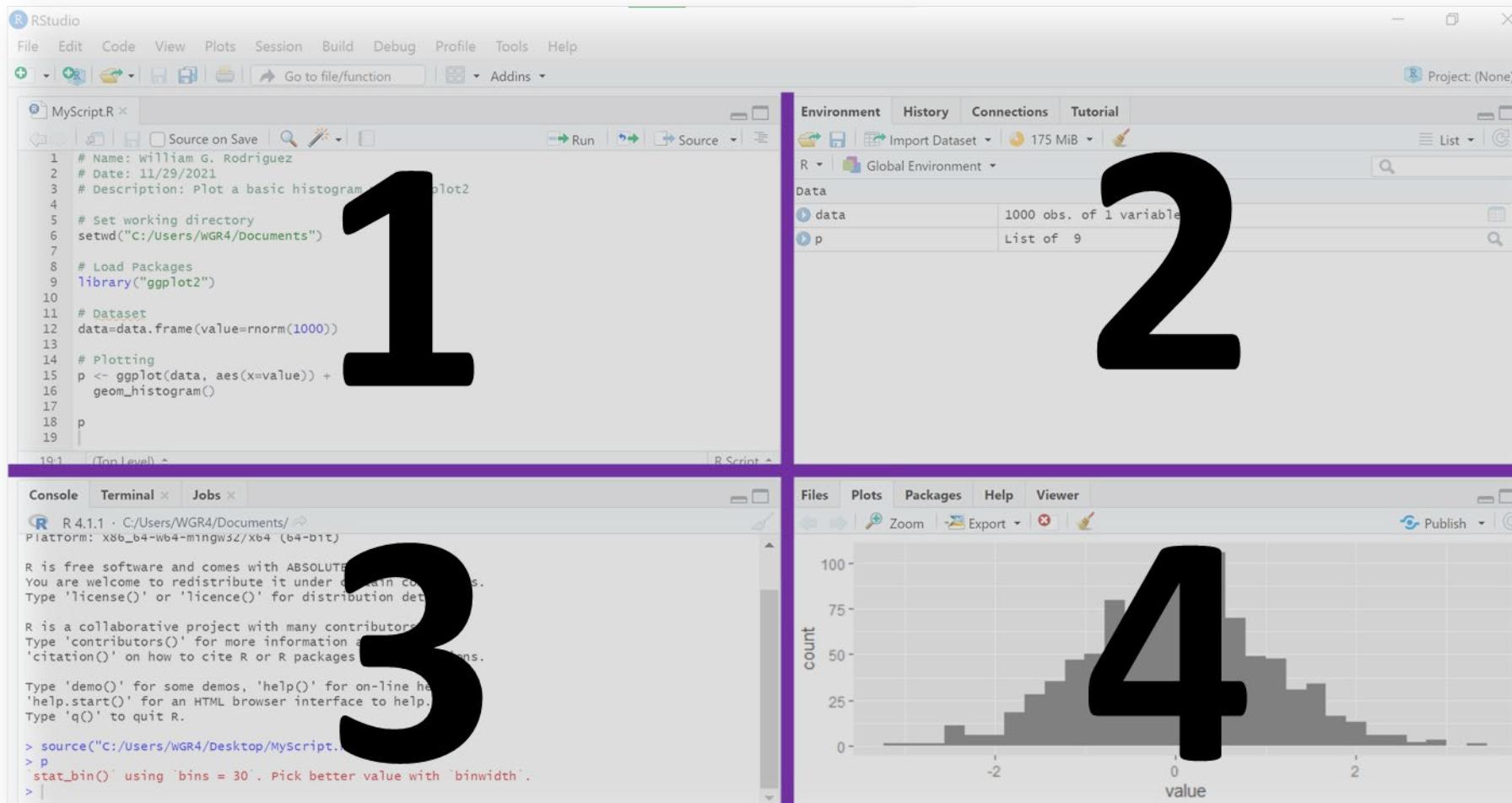
- ✓ **Clean GUI**
- ✓ **Community Support**
- ✓ **Integrations**
- ✓ **Syntax highlighting**
- ✓ **Autocompletion**
- ✓ **Free\***



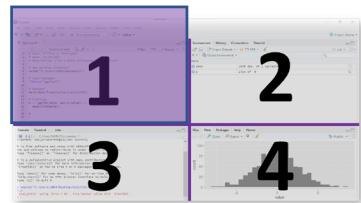
# RStudio quadrants



# RStudio quadrants



# Syntax highlighting



R RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

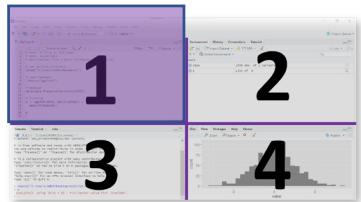
1 2 3 4

MyScript.R

```
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

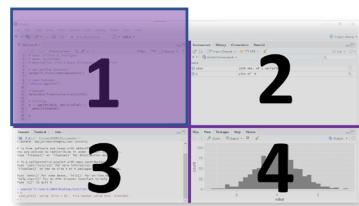
1 2 3 4

MyScript.R

```
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)

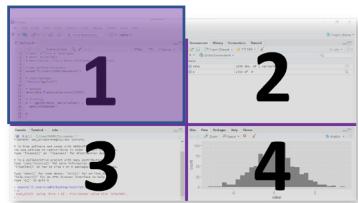
The screenshot shows the RStudio interface with the following details:

- Header:** R Studio
- Menu Bar:** File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help
- Toolbar:** Includes icons for New, Open, Save, Print, Go to file/function, and Addins.
- Code Editor:** The file "MyScript.R" is open. The code is as follows:

```
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
```

The code defines a script to generate a histogram using the ggplot2 package. It starts with metadata, loads the package, creates a dataset with 1000 random normal values, and then plots the histogram.

# Import your data into R



## Load data

The screenshot shows the RStudio environment with the following details:

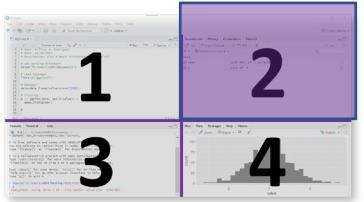
- Title Bar:** RStudio
- Menu Bar:** File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help
- Toolbar:** Includes icons for New, Open, Save, Print, Go to file/function, and Addins.
- Code Editor:** A script named "MyScript.R" is open. The code is as follows:

```
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|
```

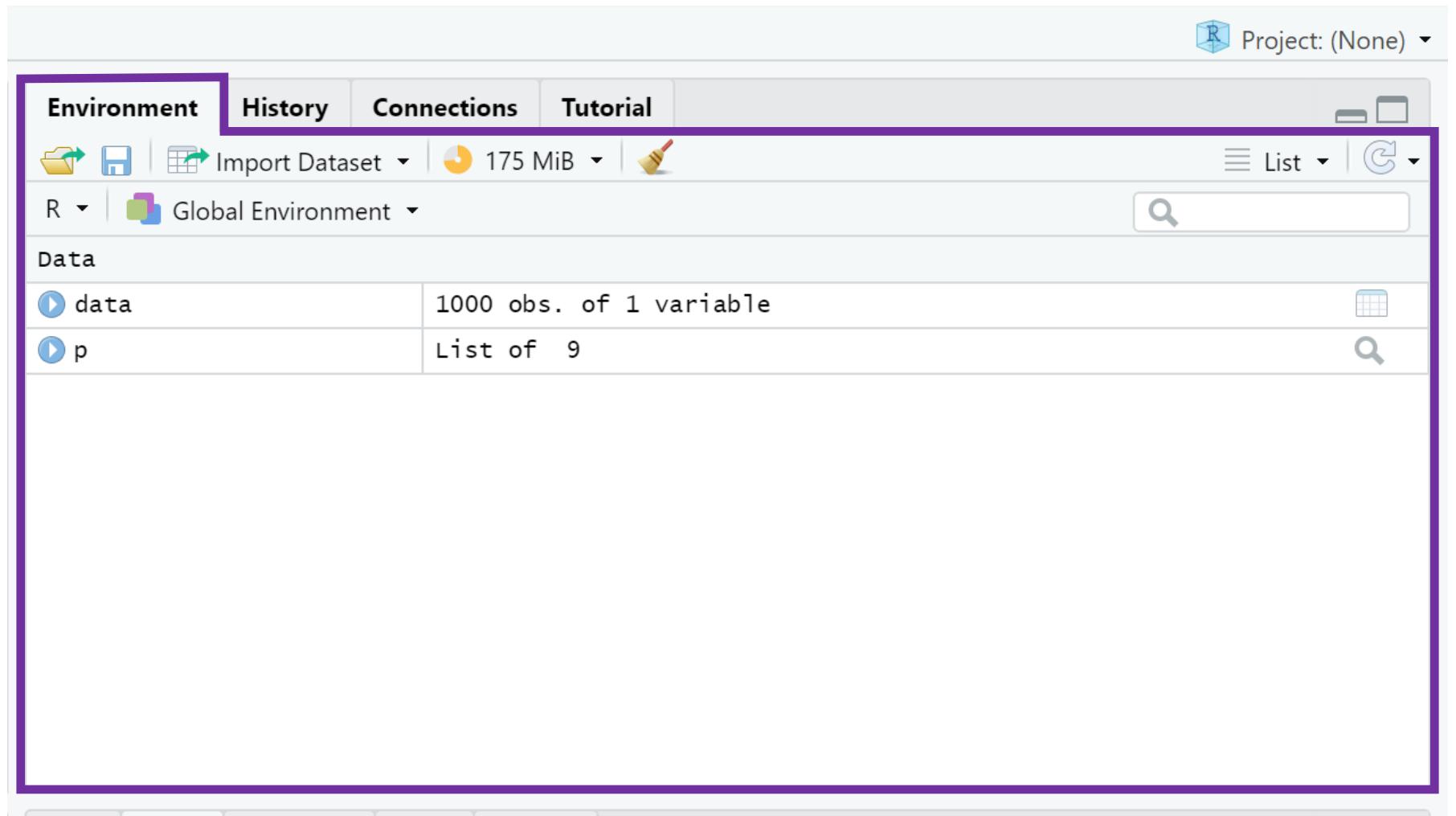
The code imports the "ggplot2" package and creates a histogram from a dataset of 1000 random normal values.

**Status Bar:** 15:2 (Top Level) ▾ R Script ▾

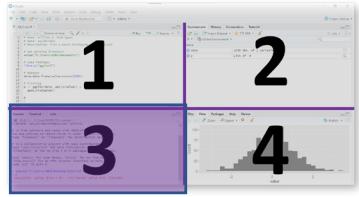
# Objects loaded on R workspace



R objects



# Display entered commands & errors



## R console

```
R 4.1.1 · C:/Users/WGR4/Documents/ ↗
Platform: x86_64-w64-mingw32/x64 (64-bit)

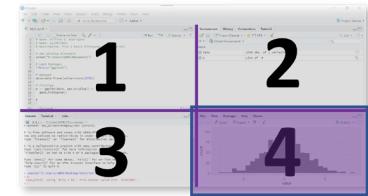
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

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.

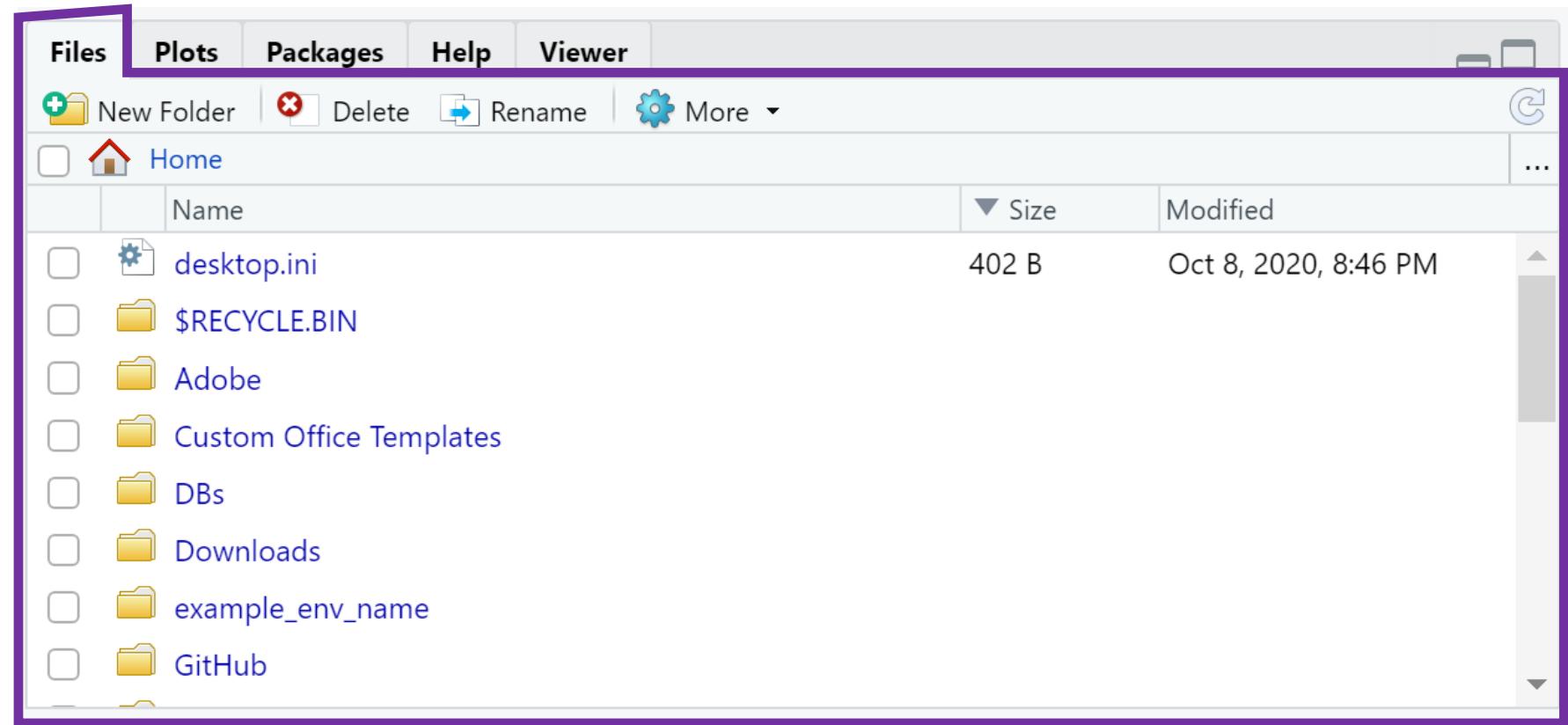
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

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

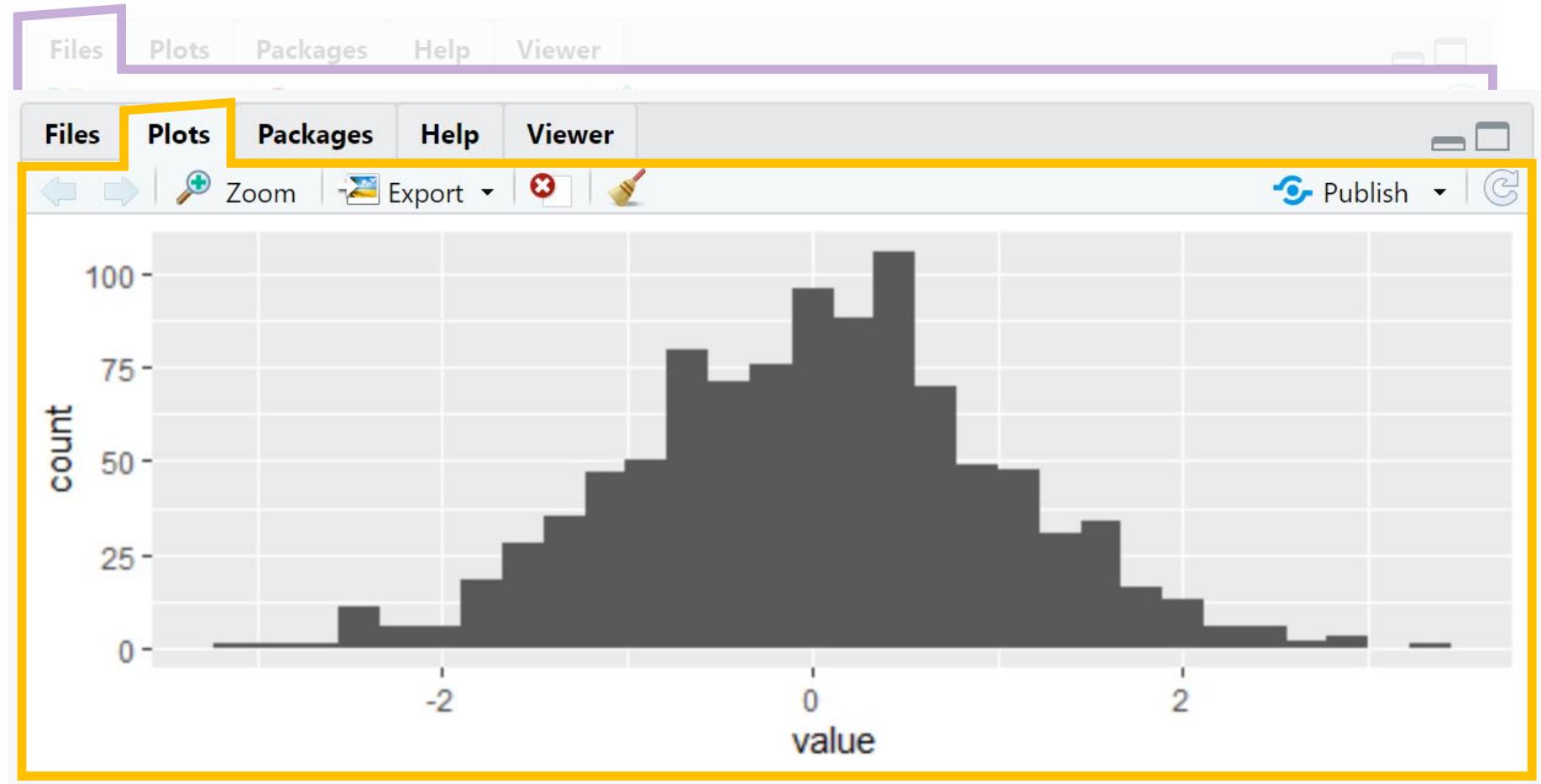
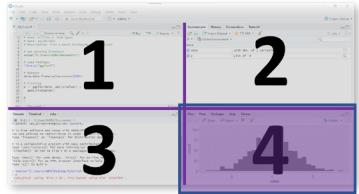
# Manage files and folders



File browser

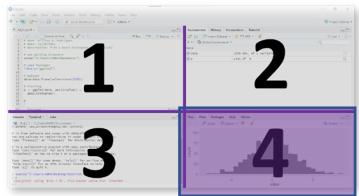


# Graphical output from R script



Graph viewer

# Documentation for R & RStudio



**R Resources**

- [Learning R Online](#)
- [CRAN Task Views](#)
- [R on StackOverflow](#)
- [Getting Help with R](#)

**RStudio**

- [RStudio IDE Support](#)
- [RStudio Community Forum](#)
- [RStudio Cheat Sheets](#)
- [RStudio Tip of the Day](#)
- [RStudio Packages](#)
- [RStudio Products](#)

## Resources

# Packages



# R repository - CRAN



**Packages - 18,391  
Default repository**

**install.packages("ggplot2")**

source: <https://cran.r-project.org/web/packages/>  
updated on: 7/20/2022

# Packages



# R packages - Source



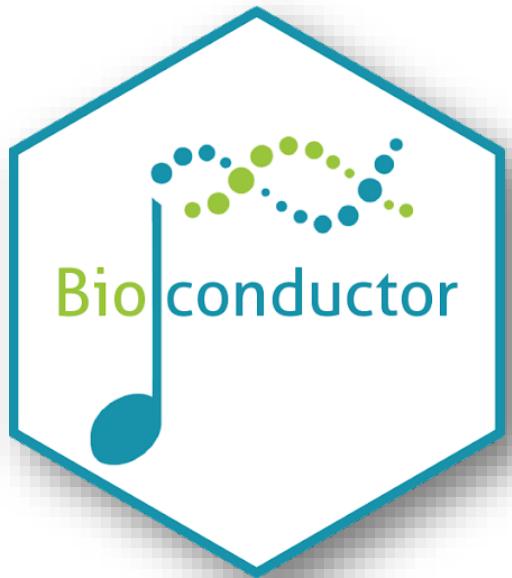
**Compressed file  
Old or in-house packages**

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

# Packages



# R repository - Bioconductor



**Packages – 2,140**  
**Bioinformatics**

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

# Packages



# Bioconductor Categories



Bioconductor version 3.15 (Release)

Autocomplete biocViews search:

- ▶ Software (2140)
- ▶ AnnotationData (910)
- ▶ ExperimentData (410)
- ▶ Workflow (29)

<https://bioconductor.org/packages/release/BiocViews.html>

# Packages



# R repositories - GitHub

**Dev releases & new features**

```
install.packages("devtools")
```

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

x11

# Logging Into O<sub>2</sub>



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



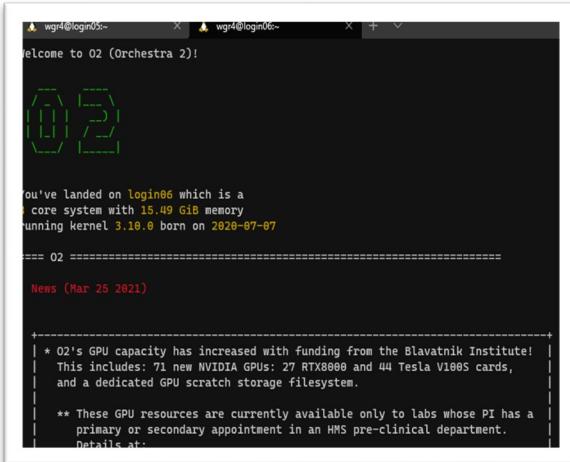
Open a terminal (type “terminal” on the search bar)



Open MobaXterm

# R

# Logging Into O<sub>2</sub>



wgr4@login05:~ wgr4@login06:~

Welcome to O2 (Orchestra 2)!

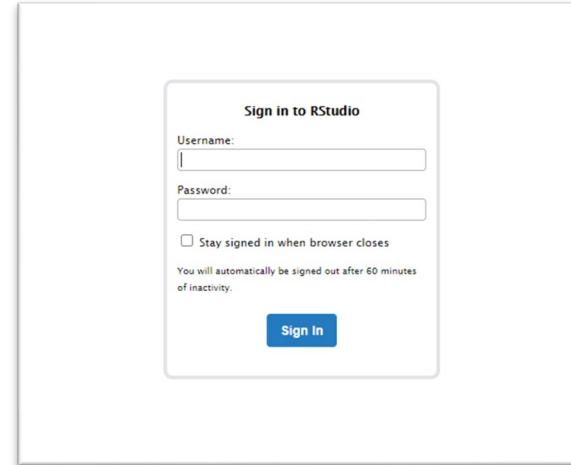
You've landed on login06 which is a core system with 15.49 GiB memory running kernel 3.10.0 born on 2020-07-07

==== O2 =====

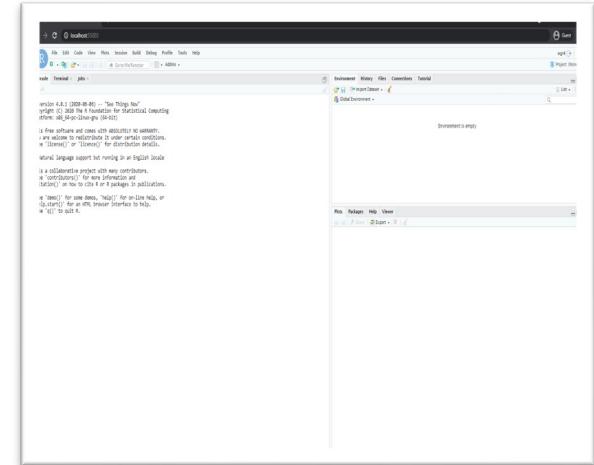
News (Mar 28 2021)

+-----+  
| \* O2's GPU capacity has increased with funding from the Blavatnik Institute!  
| This includes: 71 new NVIDIA GPUs: 27 RTX8000 and 44 Tesla V100S cards,  
| and a dedicated GPU scratch storage filesystem.  
  
| \*\* These GPU resources are currently available only to labs whose PI has a  
| primary or secondary appointment in an HMS pre-clinical department.  
| Details at: https://go.hms.harvard.edu/.../O2-GPU-Resources.html  
+-----+

**Launch RStudio**



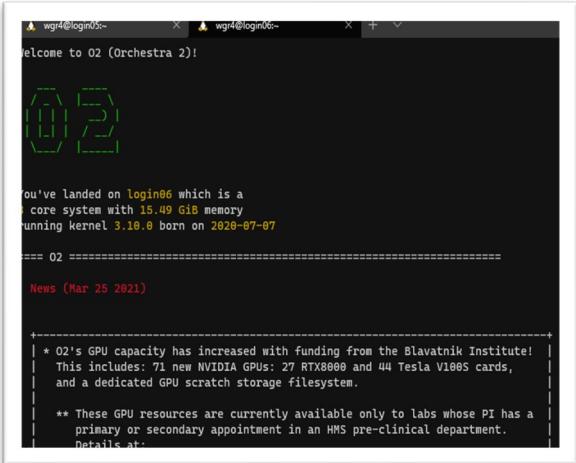
**Authenticate**



**Open in Browser**

R

# 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 O<sub>2</sub>

- 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 O<sub>2</sub>**

- **An R Personal Library is required on O<sub>2</sub>**
- **You must create an R Personal Library per version.**
- **It can be done in two steps**

# Managing R packages on O<sub>2</sub>

- An R Personal Library is required on O<sub>2</sub>
- 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
```

Pseudo terminal mode

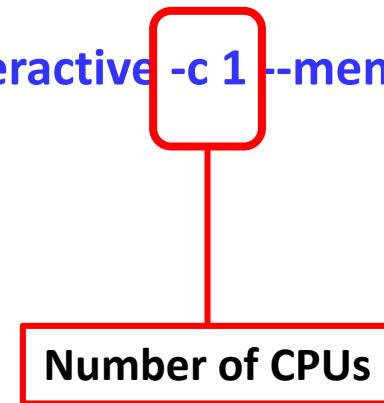
# Launch RStudio on O2

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

Partition name

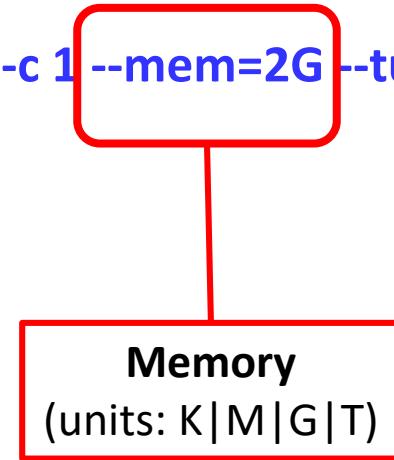
# Launch RStudio on O2

```
$ 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
```

Replace **PORT** with a number in the 50000 range

# 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
```

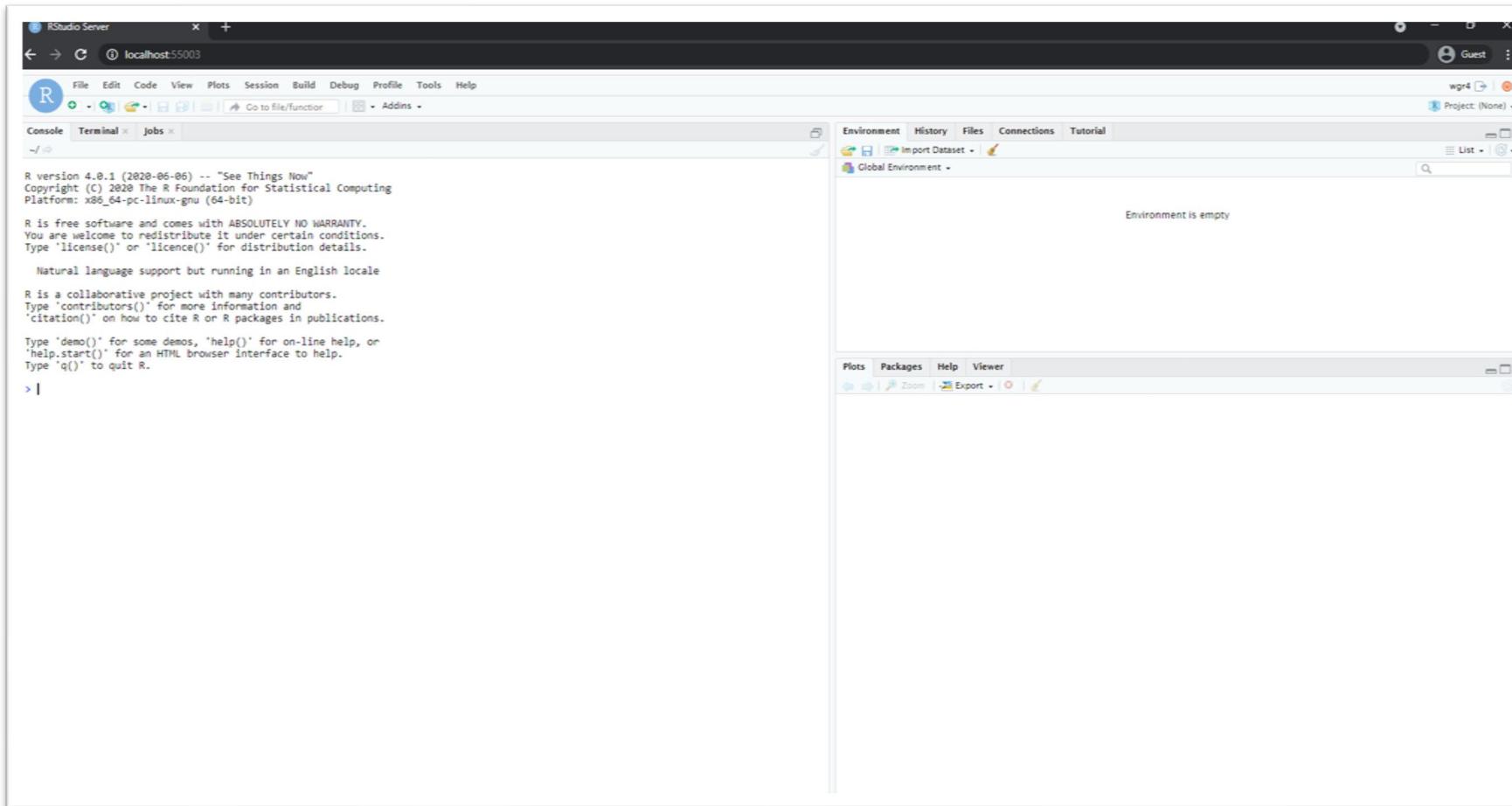
```
$ module load gcc/6.2.0
```

```
$ 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



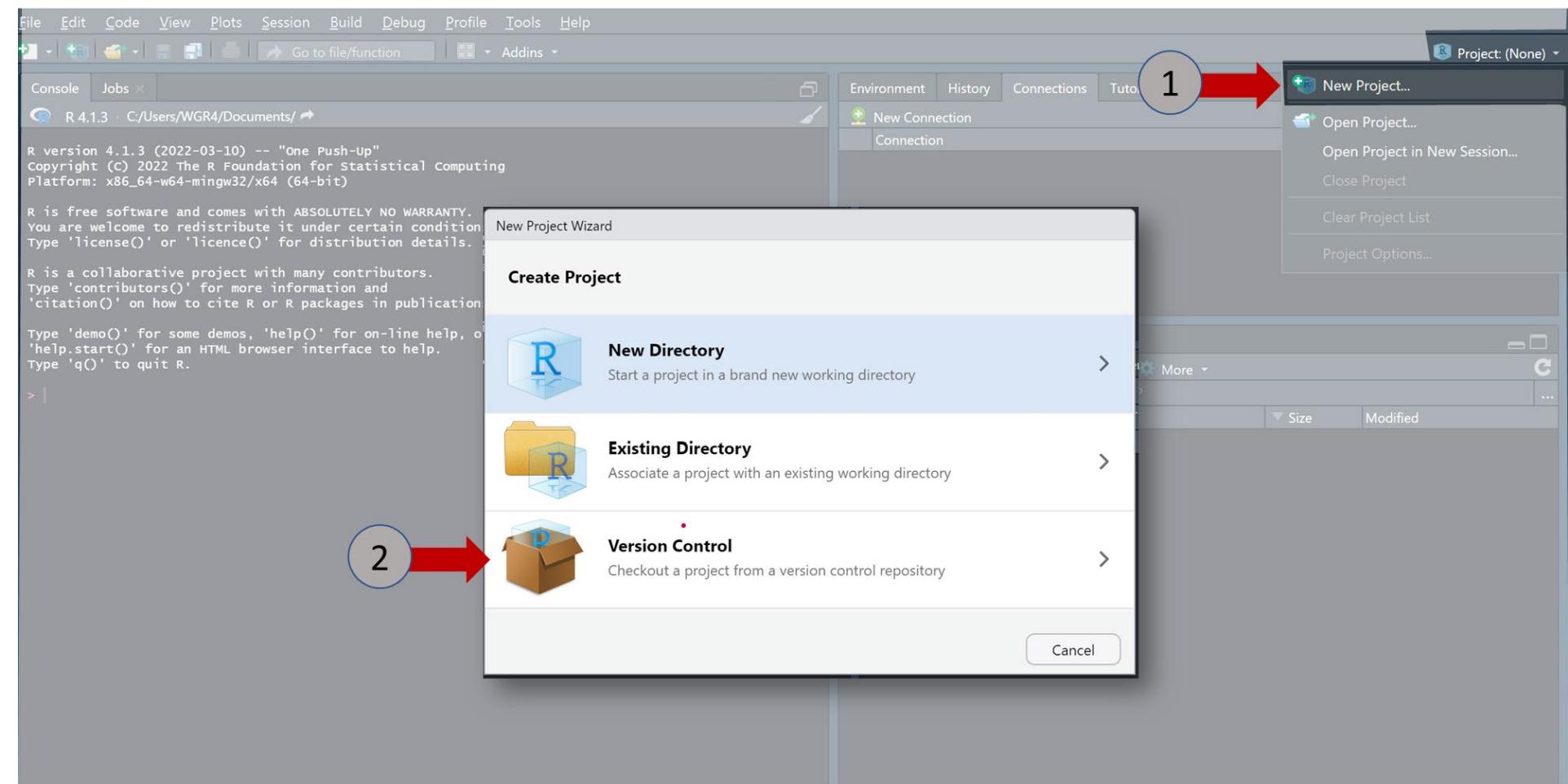


# Projects in RStudio



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

# Create an “IntroToR” Project



# Create an “IntroToR” Project



The screenshot shows the RStudio interface with the following details:

- File Menu:** File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help.
- Toolbar:** Go to file/function, Addins.
- Console:** R 4.1.3 - C:/Users/WGR4/Documents/  
R version 4.1.3 (2022-03-10) -- "One Push-Up"  
Copyright (C) 2022 The R Foundation for Statistical Computing  
Platform: x86\_64-w64-mingw32/x64 (64-bit)  
  
R is free software and comes with ABSOLUTELY NO WARRANTY.  
You are welcome to redistribute it under certain conditions.  
Type 'license()' or 'licence()' for distribution details.  
  
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.
- Project Bar:** Environment, History, Connections, Tutorial. Project: (None).
- Project Options:** New Project..., Open Project..., Open Project in New Session..., Close Project, Clear Project List, Project Options... (highlighted).
- New Project Wizard:** Create Project from Version Control (highlighted).
  - Git:** Clone a project from a Git repository.
  - SVN:** Checkout a project from a Subversion repository.
- Red Arrow:** Points to the '3' icon inside a blue circle on the 'Git' section of the dialog.
- Text at the bottom:** Repository URL: <https://github.com/wrodriguezz/IntroToR.git>

# Vectors

Data Types

Data Wrangling

Operators

Data Import

Workspace

Exercise

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

**Example:**

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

# Lists

Data Types

Data Wrangling

Operators

Data Import

Workspace

Exercise

- Like vectors but with mixed data types

## Example:

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

# Factors

Data Types

Data Wrangling

Operators

Data Import

Workspace

Exercise

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

## Example:

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

# Matrices

Data Types

Data Wrangling

Operators

Data Import

Workspace

Exercise

- **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

Data Types

Data Wrangling

Operators

Data Import

Workspace

Exercise

- 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 2<sup>nd</sup> element from a vector

> **mymatrix[ 1,2 ]**

returns element in row1 and column 2

> **mymatrix[ 1, ]**

return all elements in row 1

> **mymatrix[,1]**

return all elements in column 1

Data Types

Data Wrangling

Operators

Data Import

Workspace

Exercise

# Joining rows or columns

Data Types

Data Wrangling

Operators

Data Import

Workspace

Exercise

- 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

Data Types

Data Wrangling

Operators

Data Import

Workspace

Exercise

- **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)
```

Data Types

Data Wrangling

Operators

Data Import

Workspace

Exercise

# Apply function

Data Types

Data Wrangling

Operators

Data Import

Workspace

Exercise

- **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
```

# Other useful functions

Data Types

Data Wrangling

Operators

Data Import

Workspace

Exercise

- > **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

Data Types

Data Wrangling

Operators

Data Import

Workspace

Exercise

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

# Logical

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

Data Types

Data Wrangling

Operators

Data Import

Workspace

Exercise



# Text file

Data Types

Data Wrangling

Operators

Data Import

Workspace

Exercise

## 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

Data Types

Data Wrangling

Operators

Data Import

Workspace

Exercise

- 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

Data Types

Data Wrangling

Operators

Data Import

Workspace

Exercise

## 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**

# Manage R workspace

Data Types

Data Wrangling

Operators

Data Import

Workspace

Exercise

- **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")
```

# Import Data

Data Types

Data Wrangling

Operators

Data Import

Workspace

Exercise

- 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
```

# Import Data

Data Types

Data Wrangling

Operators

Data Import

Workspace

Exercise

- 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

Data Types

Data Wrangling

Operators

Data Import

Workspace

Exercise

- 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
```

Data Types

Data Wrangling

Operators

Data Import

Workspace

Exercise

# Transposing Data

Data Types

Data Wrangling

Operators

Data Import

Workspace

Exercise

- 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

Data Types

Data Wrangling

Operators

Data Import

Workspace

Exercise

# Plotting with ggplot2

Data Types

Data Wrangling

Operators

Data Import

Workspace

Exercise

- 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

Data Types

Data Wrangling

Operators

Data Import

Workspace

Exercise

- 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

Data Types

Data Wrangling

Operators

Data Import

Workspace

Exercise

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

```
> library("ggplot2")
```

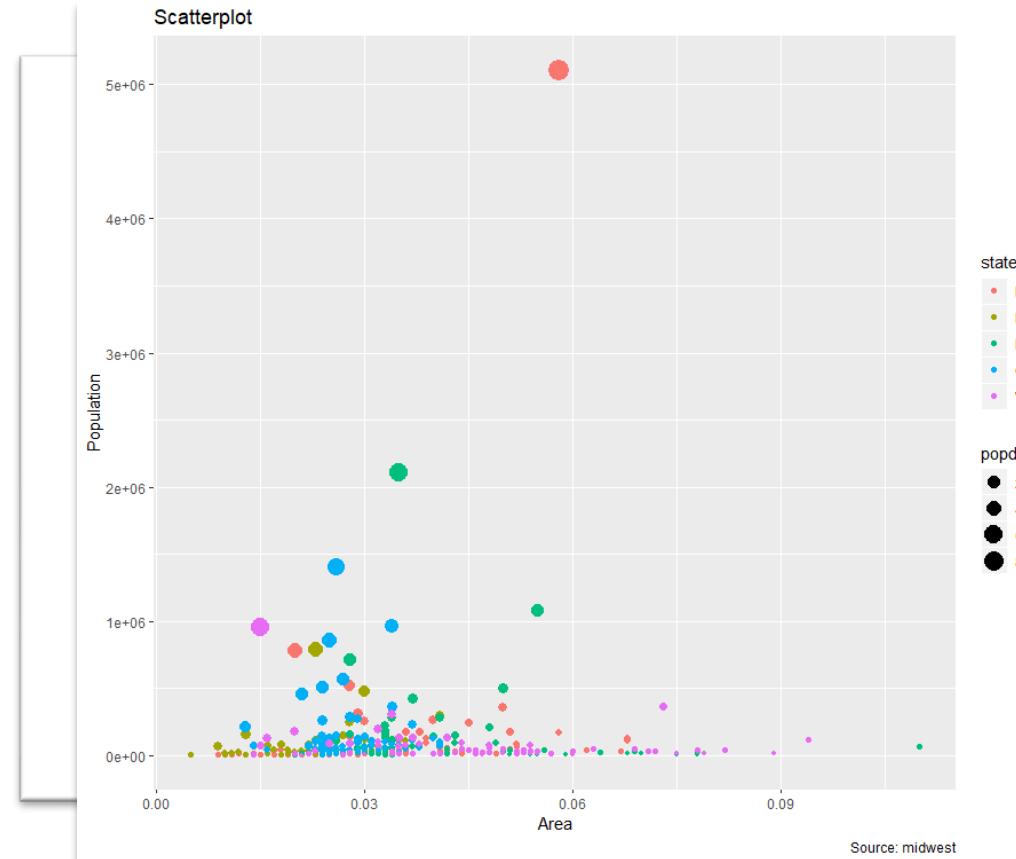
```
> data("midwest", package = "ggplot2")
```

```
> 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



Data Types

Data Wrangling

Operators

Data Import

Workspace

Exercise

# Plotting with ggplot2

Data Types

Data Wrangling

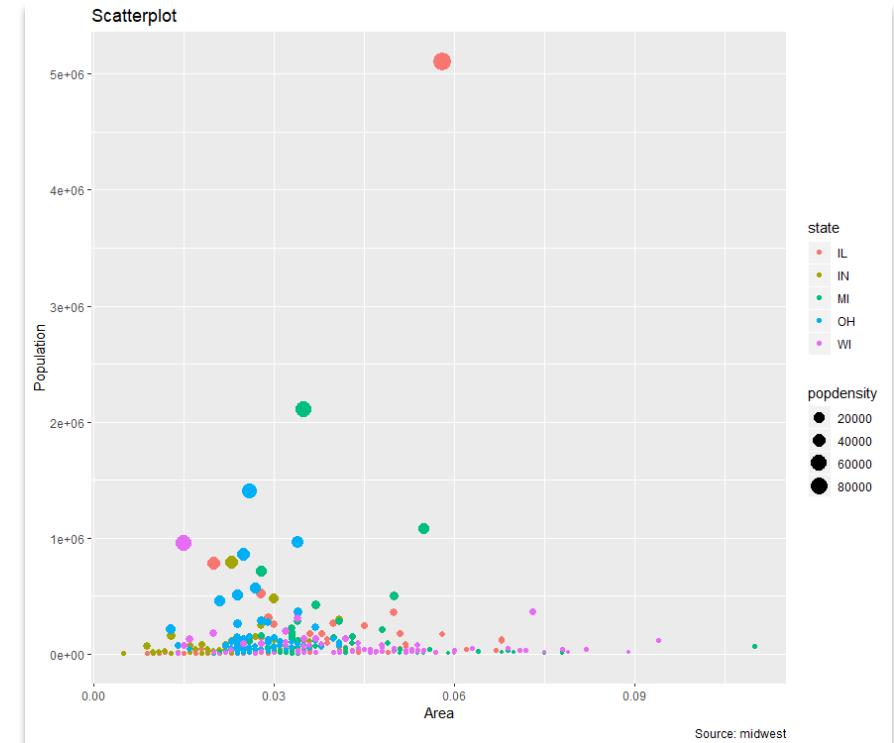
Operators

Data Import

Workspace

Exercise

- For more info: [Chan Bioinformatics Core](#)



# Important Links

- O2 Portal - <https://o2portal.rc.hms.harvard.edu/>
- RStudio on O2 Portal - <https://harvardmed.atlassian.net/l/cp/0t2poMVS>
- Other R pkgs on O2 - <https://harvardmed.atlassian.net/l/cp/nNA900j3>

# Contact Information



**Email:** [rchelp@hms.harvard.edu](mailto: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, 5<sup>th</sup> 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.