

# Intro to R

Lecture 3

BIOS 6660, Spring 2019

Instructor: Pam Russell



# Homework 1 grading

- Grades and comments will be in Canvas
- Mostly just verifying tool setup
- Will check accuracy of commands for Problem 2
  - Solution available on Canvas later today

# Overall homework grading

First 5 assignments are frequent and worth 6.7% each

Later assignments less frequent

- Homework 6-9: 8.4% each
- Homework 10-12: 11% each

# Yampa/GitHub issue

- Fix is in email from 1/28
- To make the fix permanent, issue this command:
  - `echo "unset SSH_ASKPASS" >> ~/.bashrc`
- Otherwise you will have to type `unset SSH_ASKPASS` every time you want to push to GitHub


## ~/ .bashrc

- The shell is a programming environment
- Bash is a language
- Your commands are Bash commands that modify the shell environment
- ~/ .bashrc is a script that is run every time you start the shell

# Homework 2

- Set up R coding environment
- Write an R script
- Add files to Git and GitHub
- Submit a GitHub URL through Canvas

## Homework 2: Submit GitHub URL

 [pamelarussell / sgxlib](#)

Unwatch ▾ 2

★ Star 1

🍴 Fork 0

<> Code

🔔 Issues 1

🔗 Pull requests 0

📁 Projects 0

📖 Wiki

📊 Insights

⚙️ Settings

Genomics library for Scala

Edit

Manage topics

📦 128 commits

🌿 1 branch

📦 7 releases

🚀 1 environment

👤 1 contributor

📄 MIT

Branch: master ▾


New pull request

Create new file

Upload files

Find file

Clone or download ▾

 [pamelarussell](#) remove allele validation call to htsjdk. also remove allele from vari... ...

Latest commit de6ff15 on Oct 25, 2018

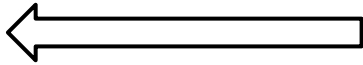
The screenshot shows the GitHub interface for the repository **pamarussell / sgxlib**. The top navigation bar includes links for Code, Issues (1), Pull requests (0), Projects (0), Wiki, Insights, and Settings. The repository is currently on the **master** branch.

The commit history is displayed, with a section for **Commits on Oct 25, 2018**. The first commit, **remove allele validation call to htsjdk. also remove allele from vari...**, is highlighted. This commit was made by **pamarussell** on Oct 25, 2018. The commit hash **de6ff15** is shown, along with a button to view the commit details. A purple box highlights this area, and a tooltip points to the hash, stating: **Browse the repository at this point in the history**.

Below this, another commit **update version number** is shown, also by **pamarussell** on Oct 25, 2018.

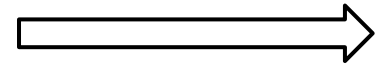
A section for **Commits on Jun 15, 2018** shows a commit **tests for SamReader queryRecords and overlappers** by **Pamela Russell** on Jun 15, 2018.

# Misc recommendation: moving cursor on command line



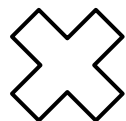
**ctrl + a**

Jump to beginning of line



**ctrl + e**

Jump to end of line



**ctrl + u**

Clear command line



# Misc recommendation: preprints



THE PREPRINT SERVER FOR BIOLOGY



[Advanced Search](#)

## Subject Areas

### All Articles

Animal Behavior and Cognition

Biochemistry

Bioengineering

Bioinformatics

Biophysics

Cancer Biology

Cell Biology

Clinical Trials

Developmental Biology

Ecology

Epidemiology

Evolutionary Biology

Genetics

Genomics

Immunology

Microbiology

Molecular Biology

Neuroscience

Paleontology

Pathology

Pharmacology and Toxicology

Physiology

Plant Biology

Scientific Communication and

Education

Synthetic Biology

Systems Biology

Zoology

# What is R?

- Free and open source
- Programming language and environment
- Statistics and graphics
- Thousands of packages
- Reproducibility with R Markdown
- RStudio: powerful development environment

## R (S) philosophy

“We wanted users to be able to begin in an interactive environment, where they did not consciously think of themselves as programming. Then as their needs became clearer and their sophistication increased, they should be able to slide gradually into programming, when the language and system aspects would become more important.”

- R. John Chambers, one of the creators of S

# For more history and background on R

An excellent [article](#) and [video](#) on the history and many sides of R

(by Roger Peng, Department of Biostatistics,  
Johns Hopkins Bloomberg School of Public  
Health)

# Ways to use R

**Interactive console:** type lines of code at a command prompt and see results in the console

**Scripts:** save code in a file and run it all at once

**R Markdown:** generate reports that mix code, results, graphics, and text explanations

# R console

```
MacBook-Pro-9:~ Pamela$ R

R version 3.4.3 (2017-11-30) -- "Kite-Eating Tree"
Copyright (C) 2017 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin15.6.0 (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.

> 5+7
[1] 12
>
Save workspace image? [y/n/c]: n
MacBook-Pro-9:~ Pamela$
```

Start R console

Type a command  
and press enter

Command output

**Ctrl+D** to exit  
console

# RStudio

A powerful free integrated development environment (IDE) designed specifically for R

## Standard IDE features

- Syntax highlighting
- Code completion
- Smart indentation
- Jump to function definitions

## Special RStudio features

- Integrated console
- Data viewer
- Integrated plotting
- Execute code directly
- Integrated documentation

# R console in Rstudio

The screenshot displays the RStudio application window. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Window, and Help. The status bar at the bottom shows the R version, copyright, and platform information. The main console area on the left contains the R startup message and the user's input commands. The Environment pane on the right shows the Global Environment with a single variable 'x' of type 'numeric'.

**Console** ~/

```
R version 3.4.3 (2017-11-30) -- "Kite-Eating Tree"
Copyright (C) 2017 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin15.6.0 (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.

> sqrt(8)
[1] 2.828427
> ?sqrt
> x <- 9
> |
```

**Environment** History Connections

Global Environment

Values
x 9

**Files** **Plots** **Packages** **Help** **Viewer**

R: Miscellaneous Mathematical Functions Find in Topic

MathFun {base} R Documentation

## Miscellaneous Mathematical Functions

### Description

`abs(x)` computes the absolute value of `x`, `sqrt(x)` computes the (principal) square root of `x`,  $\sqrt{x}$ .

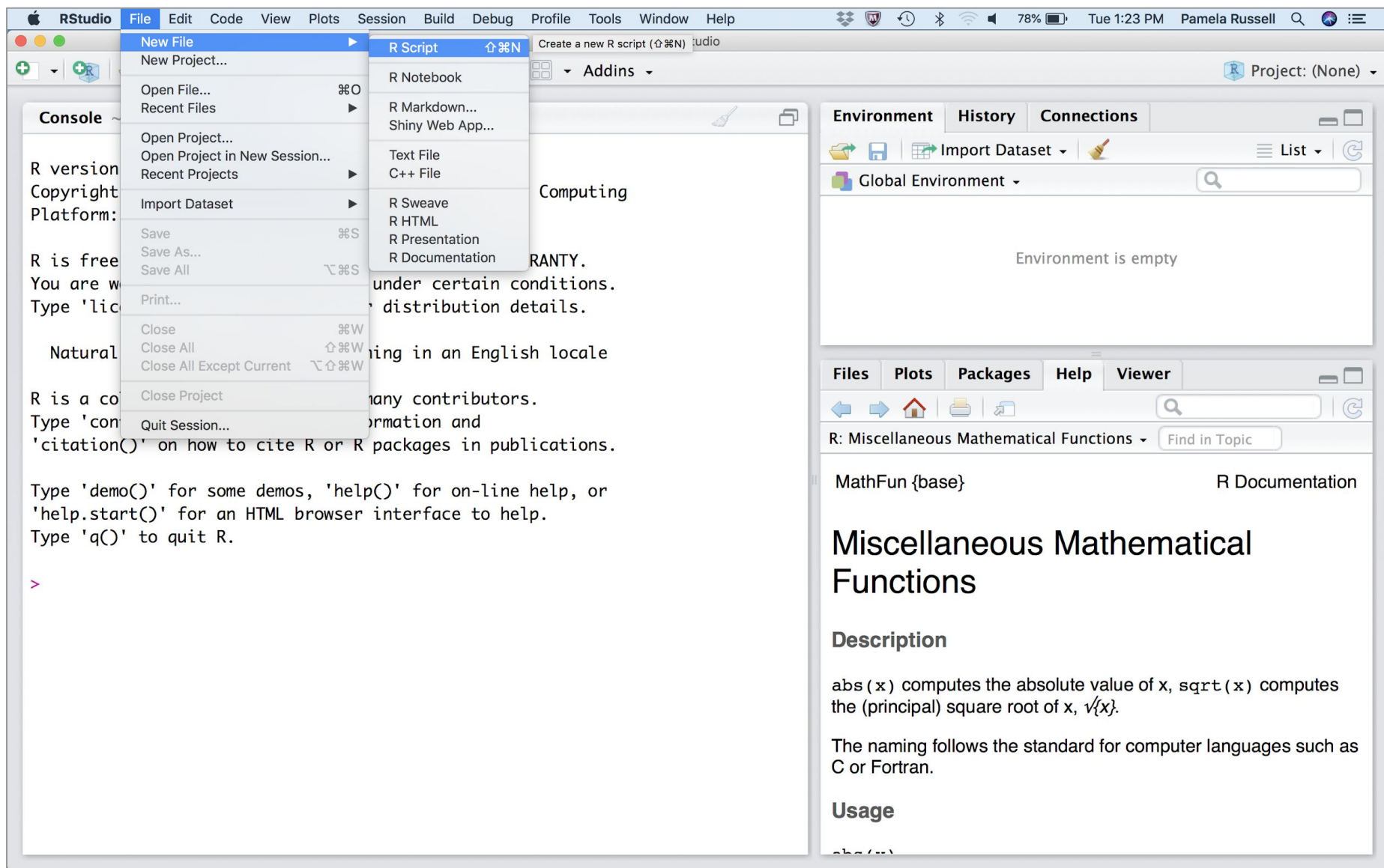
The naming follows the standard for computer languages such as C or Fortran.

### Usage

```
abs(x)
```



# Saving commands as a script



# Saving commands as a script

The screenshot displays the RStudio environment. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Window, and Help. The toolbar contains icons for file operations and a 'Go to file/function' search bar. The main editor window, titled 'example\_r\_script.R', contains the following R code:

```
1 x <- 5
2 y <- 7
3 z <- x + y
4
```

The right-hand pane is divided into two sections. The top section, labeled 'Environment', 'History', and 'Connections', shows the 'Global Environment' and indicates that the environment is empty. The bottom section, labeled 'Files', 'Plots', 'Packages', 'Help', and 'Viewer', shows the 'R: Miscellaneous Mathematical Functions' package selected. The 'Help' tab is active, displaying the documentation for the 'MathFun {base}' package, which includes a description of the 'abs(x)' function and its usage.

**Console** ~/

```
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

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

> |
```

**Environment** **History** **Connections**

Global Environment

Environment is empty

**Files** **Plots** **Packages** **Help** **Viewer**

R: Miscellaneous Mathematical Functions Find in Topic

MathFun {base} R Documentation

## Miscellaneous Mathematical Functions

### Description

`abs(x)` computes the absolute value of `x`, `sqrt(x)` computes the (principal) square root of `x`,  $\sqrt{x}$ .

The naming follows the standard for computer languages such as C or Fortran.

### Usage

`abs(x)`

# Running a script from within Rstudio

The screenshot shows the RStudio interface with the 'Code' menu open. The menu options include: Insert Section..., Jump To..., Go To File/Function..., Show Document Outline, Show Diagnostics, Go To Help, Go To Function Definition, Extract Function, Extract Variable, Rename in Scope, Reflow Comment, Comment/Uncomment Lines, Insert Roxygen Skeleton, Reindent Lines, Reformat Code, Run Selected Line(s), Re-Run Previous, Run Region, Send to Terminal, Source, Source with Echo, and Source File... The 'Run Region' option is highlighted, and a sub-menu is visible with options: Run From Beginning To Line, Run From Line to End, Run Function Definition, Run Code Section, and Run All. The 'Run All' option is also highlighted. The console shows the output of the 'demo()' function, and the viewer shows the 'Miscellaneous Mathematical Functions' documentation.

```
1 x <- 5
2 y <- 7
3 z <- x + y
4 |
```

Run All (⌘⌘R)

Run all of the code in the source file (⌘⌘R)

Console

```
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.

> |
```

Viewer

R: Miscellaneous Mathematical Functions

MathFun {base}

R Documentation

## Miscellaneous Mathematical Functions

### Description

`abs(x)` computes the absolute value of `x`, `sqrt(x)` computes the (principal) square root of `x`,  $\sqrt{x}$ .

The naming follows the standard for computer languages such as C or Fortran.

### Usage

```
abs(x)
```

# Running a script from within Rstudio

The screenshot shows the RStudio interface with the following components:

- Source Editor:** Contains a script named `example_r_script.R` with the following code:

```
1 x <- 5
2 y <- 7
3 z <- x + y
4
```
- Environment Panel:** Shows the **Global Environment** with the following values:

Variable	Value
x	5
y	7
z	12
- Console:** Shows the execution of the script:

```
> source('~/.Desktop/example_r_script.R', echo=TRUE)
> x <- 5
> y <- 7
> z <- x + y
>
```
- Viewer Panel:** Displays the **R Documentation** for the **MathFun** package, specifically the **Miscellaneous Mathematical Functions** section. The description states: `abs(x)` computes the absolute value of x, `sqrt(x)` computes the (principal) square root of x,  $\sqrt{x}$ . The usage section is also visible.

# R session environment

The screenshot displays the RStudio application window. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Window, and Help. The toolbar below the menu bar contains icons for file operations and a search bar. The main editor window shows a script file named 'example\_r\_script.R' with the following code:

```
1 x <- 5
2 y <- 7
3 z <- x + y
4
```

The Environment pane on the right, highlighted with an orange box, shows the 'Global Environment' with the following values:

Values	
x	5
y	7
z	12

The Console pane at the bottom left shows the execution of the script:

```
> source('~/.Desktop/example_r_script.R', echo=TRUE)
> x <- 5
> y <- 7
> z <- x + y
>
```

The Viewer pane on the bottom right displays the R documentation for 'Miscellaneous Mathematical Functions', including the title 'MathFun {base}', the source 'R Documentation', and the section 'Miscellaneous Mathematical Functions'.



# Working directory

The screenshot displays the RStudio interface. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Window, and Help. The toolbar contains icons for file operations and a 'Go to file/function' search bar. The main editor window shows a script named 'example\_r\_script.R' with the following code:

```
1 x <- 5
2 y <- 7
3 z <- x + y
4
```

The right-hand pane is divided into three tabs: Environment, History, and Connections. The Environment tab is active, showing the 'Global Environment' with the following values:

Variable	Value
x	5
y	7
z	12

Below the Environment tab is the 'Files' pane, which shows a file explorer view. A context menu is open over the file explorer, listing the following options:

- Copy...
- Copy To...
- Move...
- Set As Working Directory**
- Go To Working Directory
- Show Folder in New Window

The bottom pane is the Console, which shows the execution of the script:

```
> source('~/.Desktop/example_r_script.R', echo=TRUE)
> x <- 5
> y <- 7
> z <- x + y
> getwd()
[1] "/Users/Pamela"
```

# Working directory

The screenshot shows the RStudio interface with the following components:

- Source Editor:** Contains a script named `example_r_script.R` with the following code:

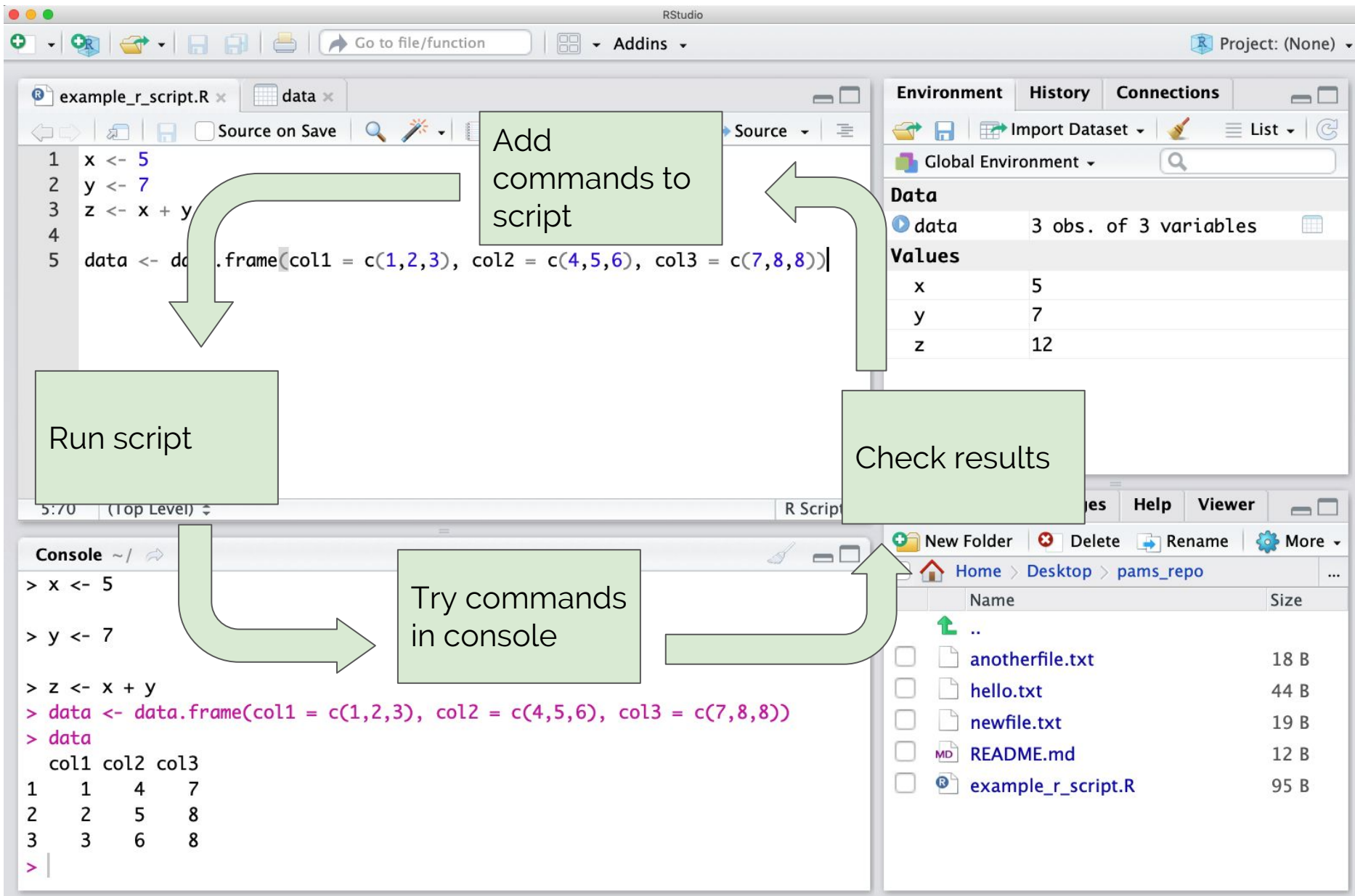
```
1 x <- 5
2 y <- 7
3 z <- x + y
4
```
- Console:** Shows the execution of the script, with the current working directory set to `~/Desktop/pams_repo/`. The output of `list.files()` is:

```
[1] "hello.txt" "newfile.txt" "README.md" "thirdfile.txt"
```
- Environment:** Displays the global environment with the following values:

Variable	Value
x	5
y	7
z	12
- Files Panel:** Shows the contents of the `pams_repo` directory:

Name	Size	Modified
..		
.gitignore	25 B	Sep 13, 2024
thirdfile.txt	21 B	Sep 13, 2024
README.md	12 B	Sep 13, 2024
hello.txt	44 B	Sep 13, 2024
newfile.txt	19 B	Sep 13, 2024

# Development cycle





# Viewing variable values

The image shows the RStudio interface with several components highlighted by orange boxes to illustrate different ways to view variable values:

- Environment Pane:** The 'data' object is listed under the 'Global Environment' with a description '3 obs. of 3 variables'. The 'Values' section below it shows the values for variables x (5), y (7), and z (12).
- Console:** The output of the commands `y`, `z`, and `data` is shown. The `data` command outputs a data frame with columns col1, col2, and col3. The command `View(data)` is also shown.
- Viewer Pane:** A data frame with 3 rows and 3 columns is displayed. The columns are labeled col1, col2, and col3. The values are: Row 1: 1, 1, 4; Row 2: 2, 2, 5; Row 3: 3, 3, 6.

Showing 1 to 3 of 3 entries

```
[1] 5
> y
[1] 7
> z
[1] 12
> data
  col1 col2 col3
1     1     4     7
2     2     5     8
3     3     6     8
> View(data)
>
```

# Function documentation

The screenshot displays the RStudio interface with the following components:

- Source Editor:** Contains R code for creating a data frame and a function call to `cor.test`.

```
1 x <- 5
2 y <- 7
3 z <- x + y
4
5 data <- data.frame(col1 = c(1,2,3), col2 = c(4,5,6), col3 = c(7,8,8))
```
- Environment:** Shows the `data` object with 3 observations and 3 variables.
- Console:** Shows the execution of the code, resulting in a data frame with 3 rows and 3 columns. The last command, `?cor.test`, is highlighted with an orange box.

```
> y <- 7
> z <- x + y
> data <- data.frame(col1 = c(1,2,3), col2 = c(4,5,6), col3 = c(7,8,8))
> data
  col1 col2 col3
1     1     4     7
2     2     5     8
3     3     6     8
> ?cor.test
```
- Documentation Panel:** Displays the R documentation for `cor.test`, including the title "Test for Association/Correlation Between Paired Samples", a description of the test, and the usage syntax.

**Test for Association/Correlation Between Paired Samples**

**Description**

Test for association between paired samples, using one of Pearson's product moment correlation coefficient, Kendall's *tau* or Spearman's *rho*.

**Usage**

```
cor.test(x, ...)
```

## Default S3 method:

```
cor.test(x, y,
         alternative = c("two.sided", "less", "greater"),
         method = c("pearson", "kendall", "spearman"),
         exact = NULL, conf.level = 0.95, continuity = FALSE, ...)
```

# **Global environment persists throughout R session**

RStudio always has an active R session in progress with its own global environment (variable values)

When running script within RStudio, watch out for variables left over in the environment from previous runs/commands.

# Global environment persists throughout R session

The screenshot shows the RStudio interface with the following components:

- Source Editor:** Contains the script `example_r_script.R` with the following code:

```
1 x <- 5
2 #y <- 7
3 z <- x + y
4
5 data <- data.frame(col1 = c(1,2,3), col2 = c(4,5,6), col3 = c(7,8,8))
```

The line `#y <- 7` is highlighted with an orange box.
- Environment Panel:** Shows the **Global Environment** with a table of data:

Data	
	3 obs. of 3 variables
x	5
y	7
z	12
- Console:** Shows the execution history:

```
> x <- 5
> #y <- 7
> z <- x + y
> data <- data.frame(col1 = c(1,2,3), col2 = c(4,5,6), col3 = c(7,8,8))
>
```

The last four lines are highlighted with an orange box.
- Files Panel:** Shows the file explorer with a list of files in the `pams_repo` directory:

Name	Size
..	
anotherfile.txt	18 B
hello.txt	44 B
newfile.txt	19 B
README.md	12 B
example_r_script.R	96 B

An orange box highlights the text: "Code still runs and uses the value of y that was in the environment from before".

# **Use Rstudio locally; use Git/GitHub to move scripts back and forth to Yampa**

Last lecture: how to use Git and GitHub to work on the same files from multiple computers

For R scripts that you want to run on Yampa: you should still write them on your computer with Rstudio.

Add scripts to GitHub from your computer, then pull them down from Yampa.

# Running a script from the command line

A screenshot of a macOS terminal window. The title bar at the top shows three colored window control buttons (red, yellow, green) on the left, and a folder icon followed by the text "Desktop — -bash — 91x32" on the right. The terminal area has a light blue background. It displays three lines of text: the first line is "MacBook-Pro-9:~ Pamela\$ cd ~/Desktop/" followed by a right-pointing cursor; the second line is "MacBook-Pro-9:Desktop Pamela\$ Rscript example\_r\_script.R" followed by a right-pointing cursor; and the third line is "MacBook-Pro-9:Desktop Pamela\$" followed by a solid black cursor block.

```
MacBook-Pro-9:~ Pamela$ cd ~/Desktop/
MacBook-Pro-9:Desktop Pamela$ Rscript example_r_script.R
MacBook-Pro-9:Desktop Pamela$
```

# Running a script from the console

```
MacBook-Pro-9:Desktop Pamela$ R

R version 3.4.3 (2017-11-30) -- "Kite-Eating Tree"
Copyright (C) 2017 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin15.6.0 (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.

> setwd("~/Desktop")
> source("example_r_script.R")
> ls()
[1] "x" "y" "z"
> x
[1] 5
> y
[1] 7
> z
[1] 12
>
```

Start R console

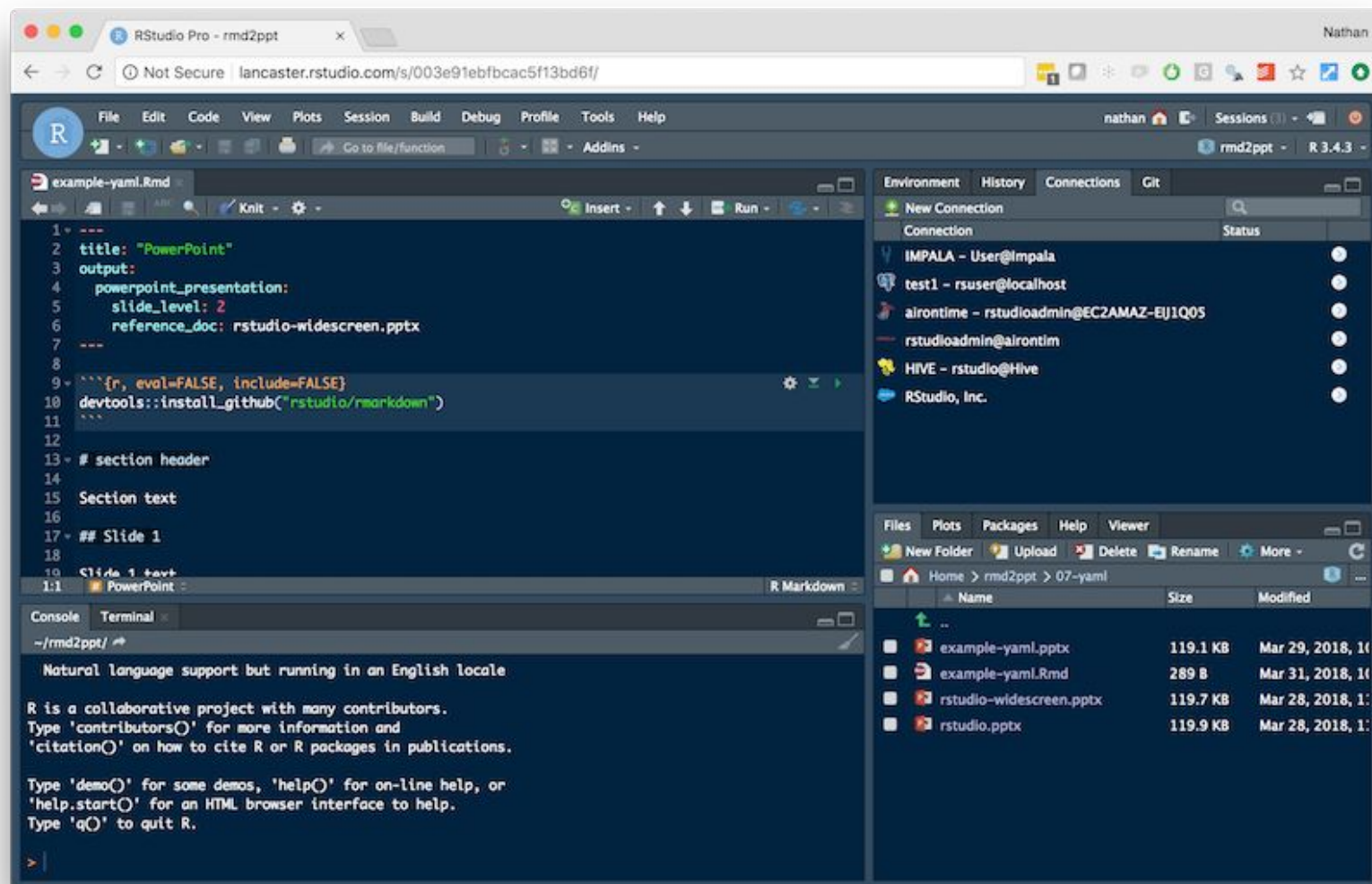
Set working directory to the directory containing the script

The `source` command loads a script and runs all of the code

The `ls` command lists variables in the environment. The code in the script has added variables to our environment.



# RStudio Server: RStudio on Linux server accessible by web browser





# R variables and handling data

Slides:

[L3\\_intro\\_to\\_R.html](#)