

# INTRODUCTION TO R PROGRAMMING

## Day 1: R Basics

October 24, 2023



# Who is this workshop for?

- ❖ Those with little or no experience with R
- ❖ Those who would like an R refresher
- ❖ Those with a need or interest for using R in their work

# Who are your instructors?



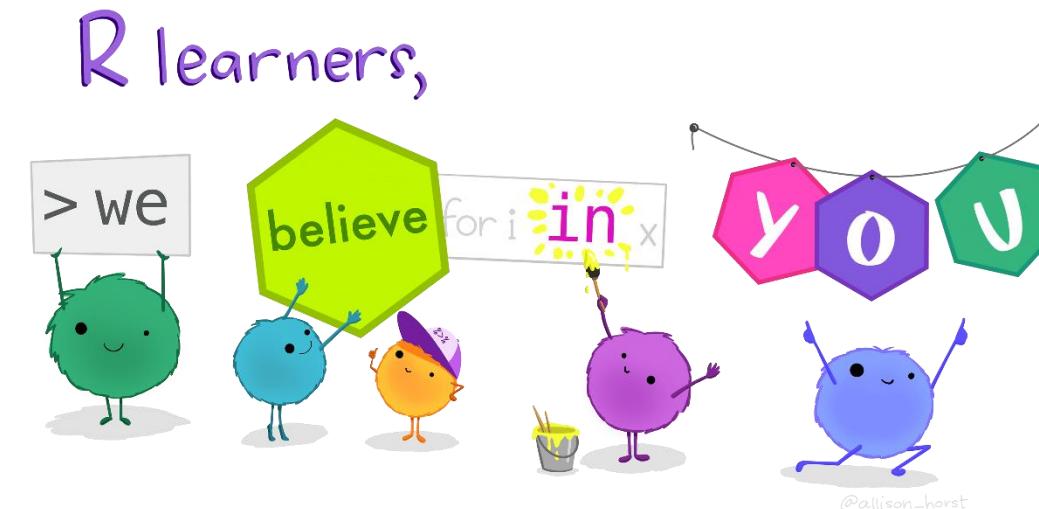
Joy Nyaanga, PhD  
Senior Bioinformatician  
[Day 1 & 3](#)



Stella Karuri, PhD  
Statistician  
[Day 2](#)

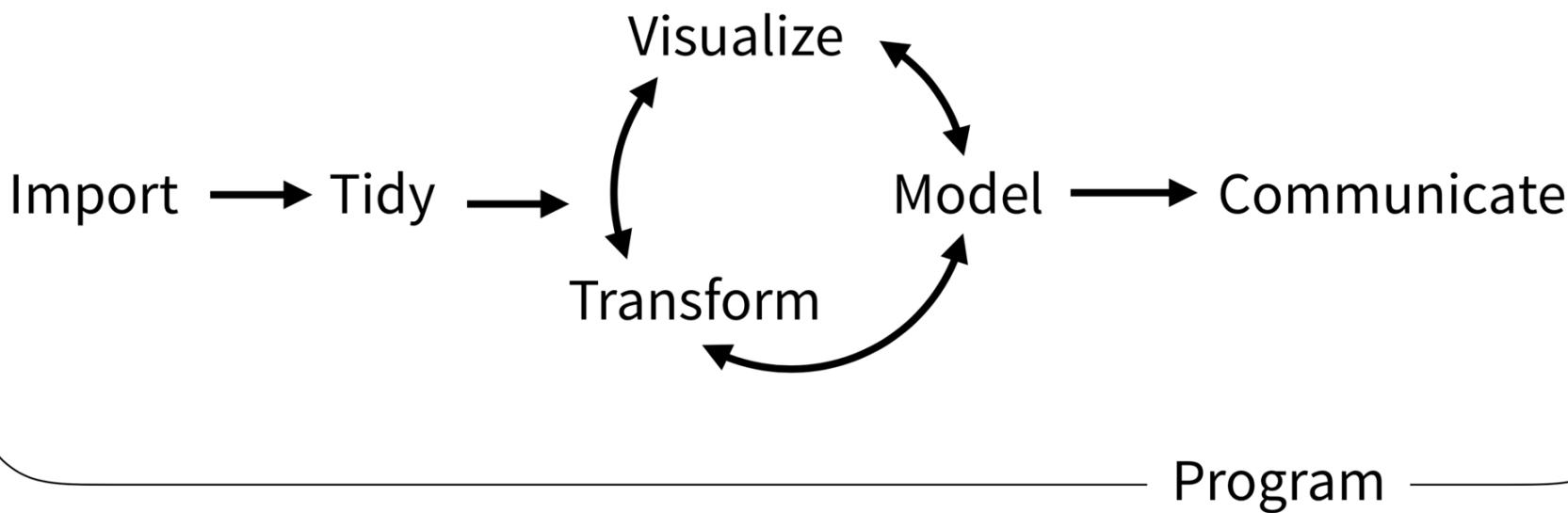
# Why R?

- ❖ Anyone can learn R programming!
- ❖ R provides substantial flexibility and reproducibility over excel
- ❖ Data analysis can be streamlined in R



# Orientation to the workshop

# The data analysis pipeline



# Workshop breakdown

Session 1: Lecture	11:00 a.m. – noon
Break	
Session 2: Interactive	2:00 p.m. – 4:00 p.m.



Course website: <https://stanley-manne-childrens-research.github.io/introR/>



Code repository: <https://github.com/Stanley-Manne-Childrens-Research/introR>

# Day 1: The Basics of R

# Learning objectives

- ❖ Understand basic coding principles
- ❖ Use R as a calculator
- ❖ Be able to load in data
- ❖ Basic operations on data
- ❖ Be able to make a plot
- ❖ Know how to get help



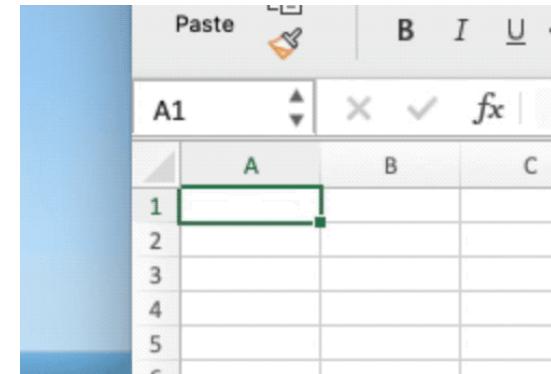
# Why use something other than Excel?

- ❖ Excel can change your data and rename genes

**Scientists rename human genes to stop Microsoft Excel from misreading them as dates** / Sometimes it's easier to rewrite genetics than update Excel

By [James Vincent](#), a senior reporter who has covered AI, robotics, and more for eight years at The Verge.

Aug 6, 2020, 7:44 AM CDT | [0 Comments](#) / [0 New](#)



Human Gene  
Nomenclature  
Committee

Symbols that affect data handling and retrieval. For example, all symbols that autoconverted to dates in Microsoft Excel have been changed (for example, SEPT1 is now SEPTIN1; MARCH1 is now MARCHF1); tRNA synthetase symbols that were also common words have been changed (for example, WARS is now WARS1; CARS is now CARS1).

# Why use something other than Excel?

- ❖ Excel can change your data and rename genes
- ❖ Cut and paste errors are easy to do and can have bad consequences

## Excel snafu costs firm \$24m

Some cleric, some error

 [Drew Cullen](#)

A simple spreadsheet error cost a firm a whopping US\$24m.

“... a cut-and-paste error in an Excel spreadsheet that we did not detect when we did our final sorting and ranking bids prior to submission” – chief executive Steve Snyder

The mistake led to TransAlta, a big Canadian power generator, buying more US power transmission hedging contracts in May at higher prices than it should have.

# Why use something other than Excel?

- ❖ Excel can change your data and rename genes
- ❖ Cut and paste errors are easy to do and can have bad consequences
- ❖ Excel has a relatively low (for some researchers) ceiling on number of available rows

## Analysis

Covid: how Excel may have caused loss of 16,000 test results in England

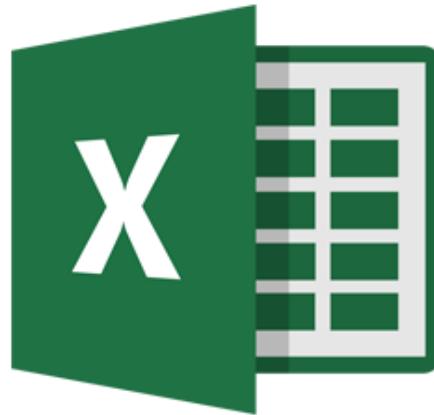
Alex Hern

UK technology editor

Public Health England data error blamed on limitations of Microsoft spreadsheet

- [Coronavirus - latest updates](#)
- [See all our coronavirus coverage](#)

# Point-and-click is not reproducible



# What is R?

- ❖ A programming language
  - A successor of the S language
  - Created in 1991 by 2 statisticians to teach introductory statistics
  - In 1995 the source code for the R system was made accessible to the public
  - In 2000 R version 1.0.0 was officially released!
- ❖ Focus on statistical modeling and data analysis
- ❖ Interfaces with other languages (i.e. python, bash, etc.)



A programming  
language for  
data analysis

RGui (64-bit)

File Edit View Misc Packages Windows Help



R Console

```
R version 4.2.1 (2022-06-23 ucrt) -- "Funny-Looking Kid"
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.

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.

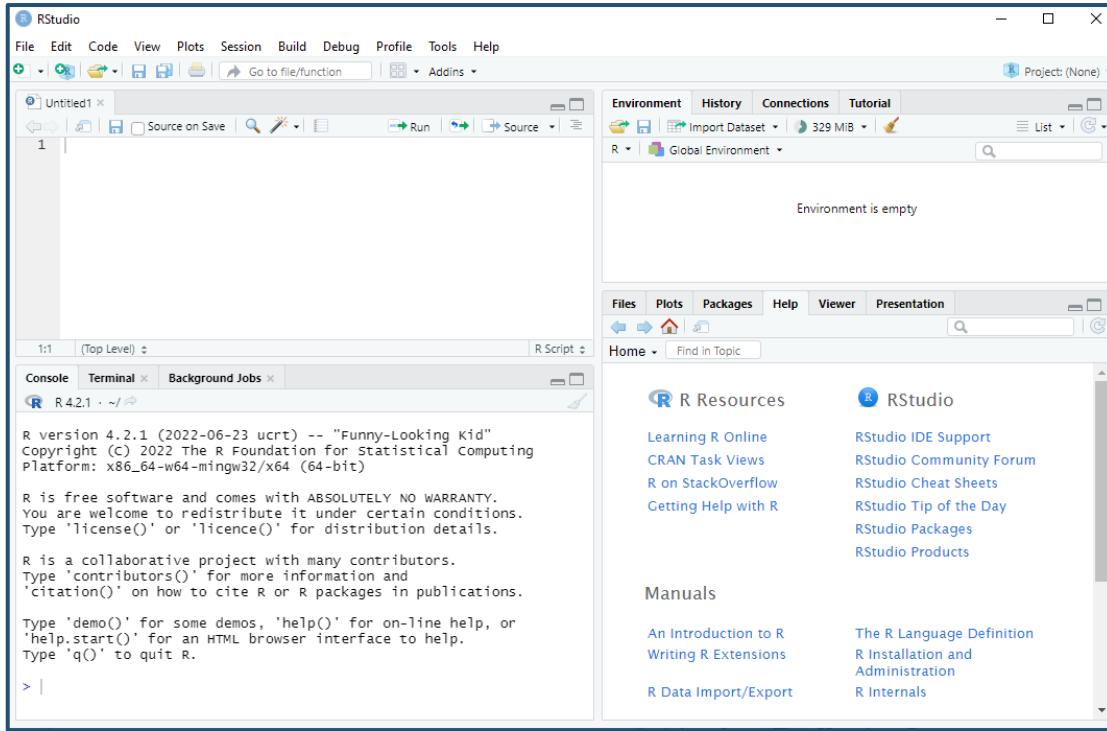
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> |
```



Early developers ran R via the command line console

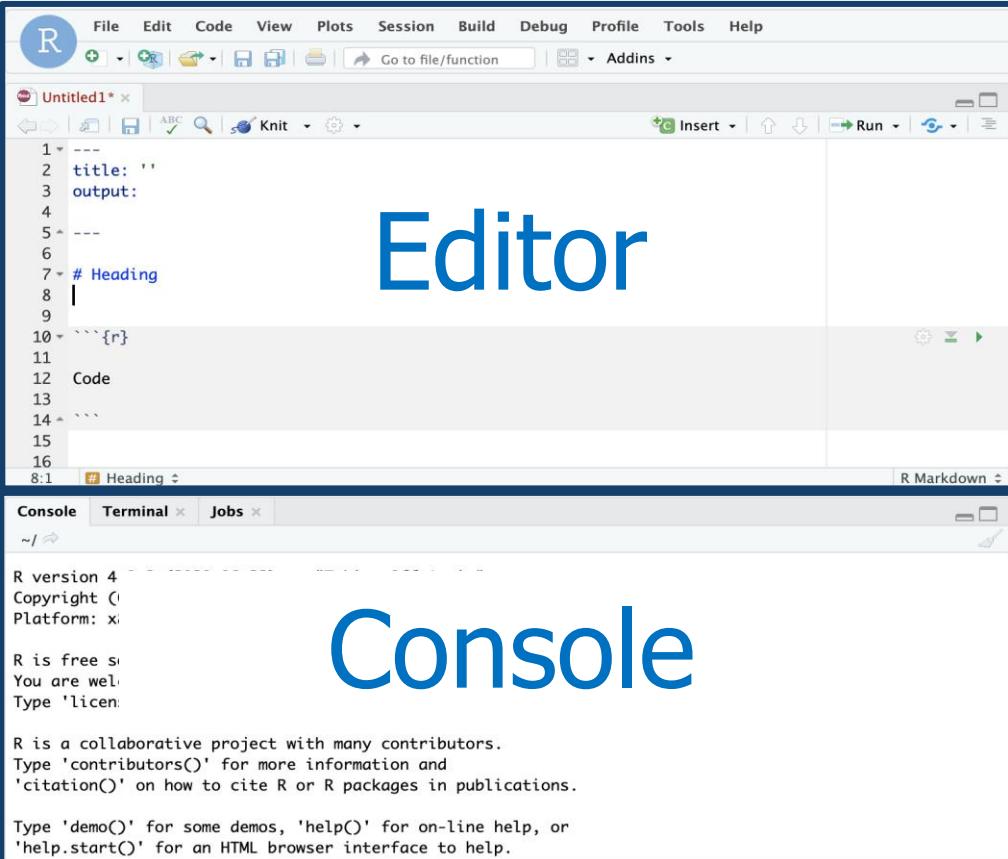
# What is RStudio?



- ❖ An integrated development environment (IDE)
- ❖ Allows you to write, save, and open R code

*"RStudio gives you a way to talk to your computer. R gives you a language to speak in"*

# RStudio anatomy



The RStudio Editor interface is shown. It features a top menu bar with File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. Below the menu is a toolbar with various icons for file operations like Open, Save, and Print. A code editor window displays the following R Markdown code:

```
1 ---  
2 title: ''  
3 output:  
4  
5 ---  
6  
7 # Heading  
8  
9  
10 ````{r}  
11  
12 Code  
13  
14 ````  
15  
16
```

The word "Editor" is overlaid in large blue text across the center of the screenshot.

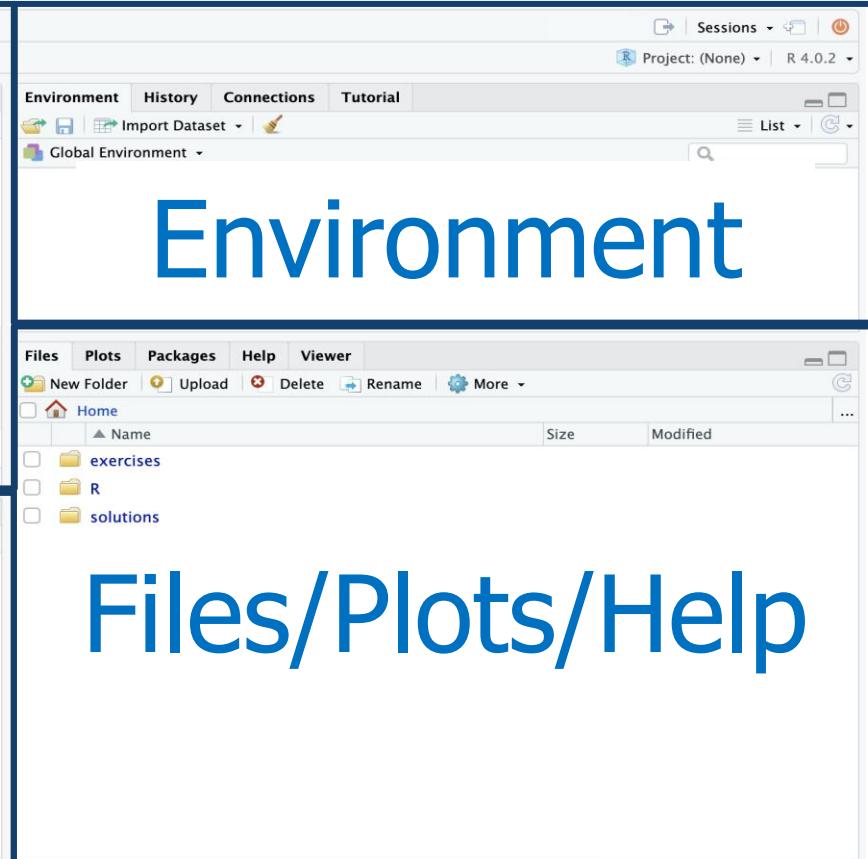
Console Terminal Jobs

R version 4 Copyright (c) 2013 The R Foundation for Statistical Computing  
Platform: x86\_64-w64-mingw32 mingw32

R is free software.  
You are welcome to redistribute it.  
Type 'license()' or 'licence()' for more information about  
the license.

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.

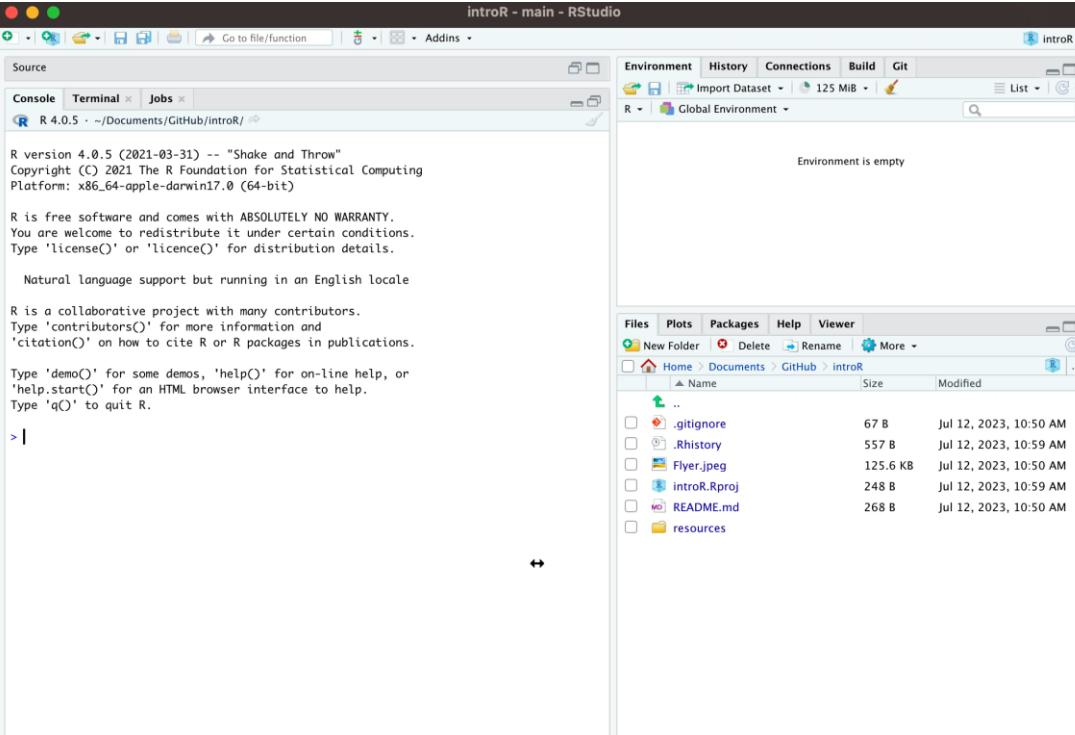
Type 'demo()' for some demos, 'help()' for on-line help, or  
'help.start()' for an HTML browser interface to help.



The RStudio interface is shown with several panes open. The top navigation bar includes Sessions, Addins, and a Project dropdown set to (None). The main tabs are Environment, History, Connections, and Tutorial. The Environment tab shows the Global Environment. The Files tab shows a directory structure with Home, exercises, R, and solutions. The Plots and Packages tabs are also visible.

The words "Environment", "Files/Plots/Help", and "Console" are overlaid in large blue text across the right side of the screenshot.

# Coding in the console



R version 4.0.5 (2021-03-31) -- "Shake and Throw"  
 Copyright (C) 2021 The R Foundation for Statistical Computing  
 Platform: x86\_64-apple-darwin17.0 (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.

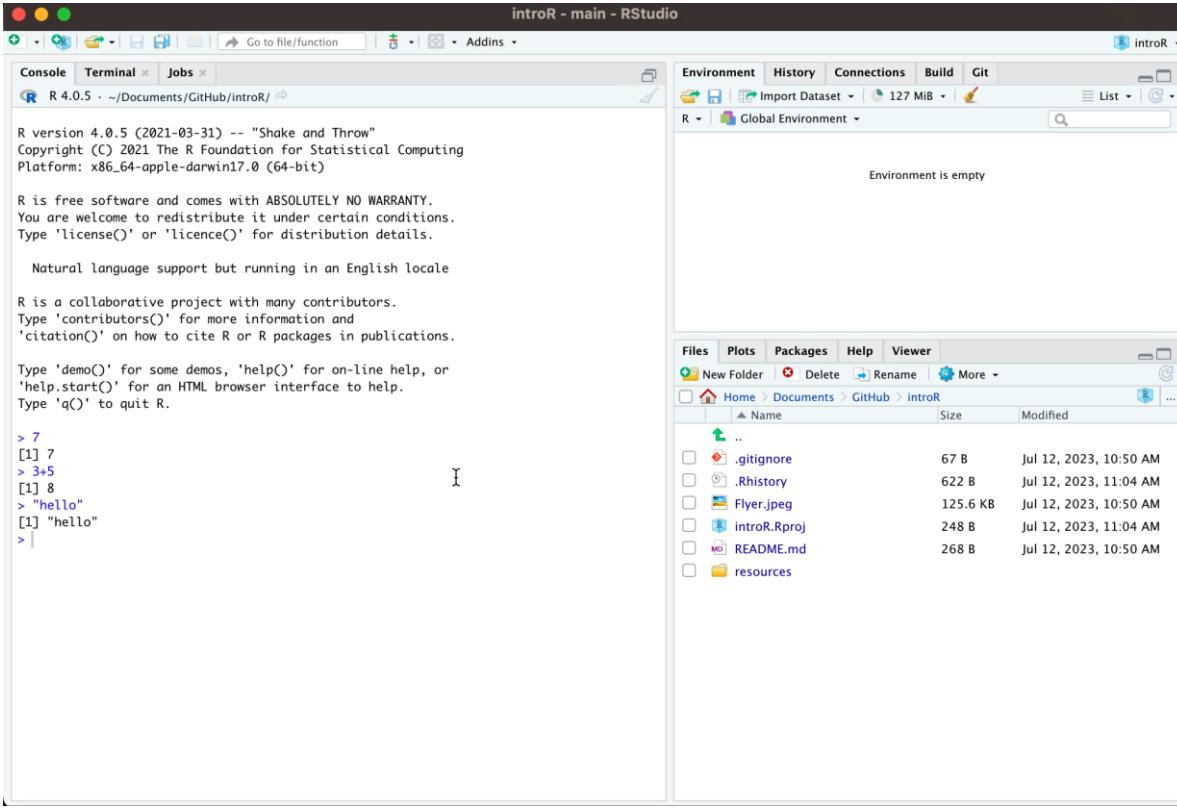
```
> |
```

- ❖ Type code in the console
- ❖ Press **return** to execute code
- ❖ Output shown below

*Coding in the console is not advisable for most situations!*

- ❖ Only recommended for short pieces of code that you don't need to save

# Coding in a script



The screenshot shows the RStudio interface with a project titled "introR". The left pane displays the R console output:

```
R version 4.0.5 (2021-03-31) -- "Shake and Throw"
Copyright (C) 2021 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin17.0 (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.

> 7
[1] 7
> 3+5
[1] 8
> "hello"
[1] "hello"
> |
```

The right pane shows the file browser with the following contents:

Name	Size	Modified
..	67 B	Jul 12, 2023, 10:50 AM
.gitignore	622 B	Jul 12, 2023, 11:04 AM
.Rhistory	125.6 KB	Jul 12, 2023, 10:50 AM
Flyer.jpeg	248 B	Jul 12, 2023, 11:04 AM
introR.Rproj	268 B	Jul 12, 2023, 10:50 AM
README.md		
resources		

Scripts allow you to...

- ❖ Execute code in blocks
- ❖ Save your work
- ❖ Easily share your work

# Coding basics

# Using R as a calculator

```
> 10^2
```

```
[1] 100
```

```
> 6/9
```

```
[1] 0.6666667
```

```
> 9-43
```

```
[1] -34
```

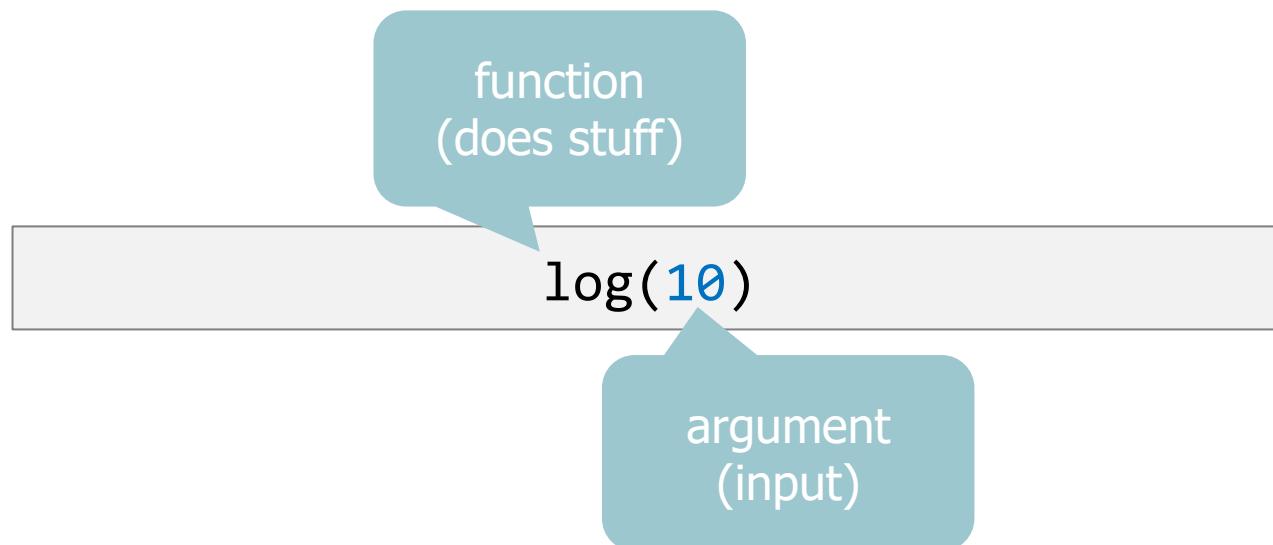
*R follows the rules for order of operations  
and ignores spaces between numbers*

```
> log(10)
```

```
[1] 2.302585
```

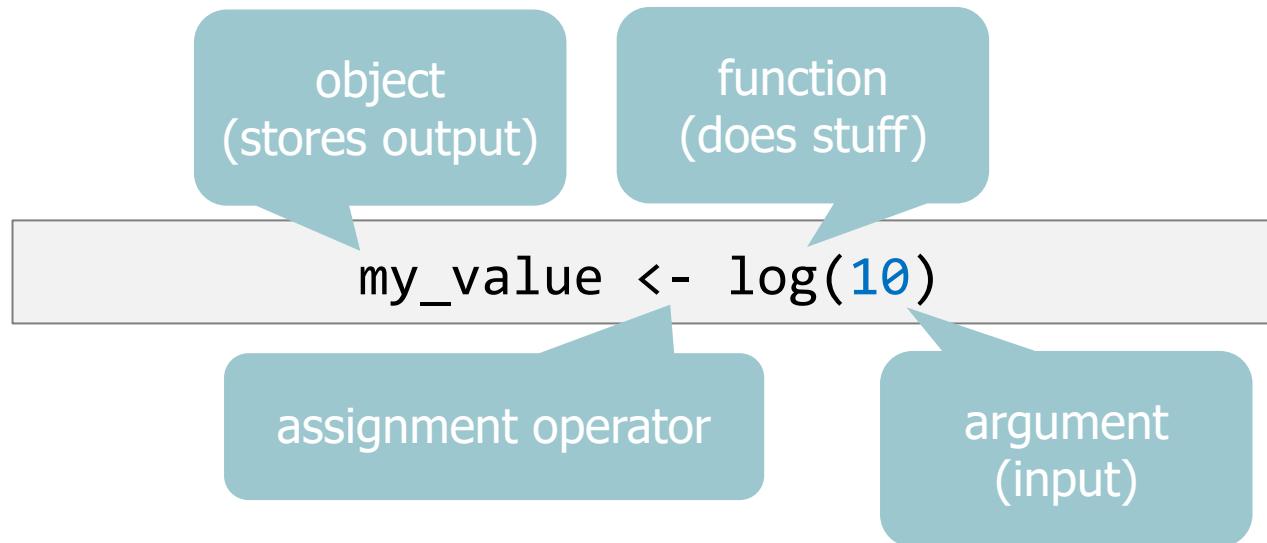
# Functions

- ❖ `log()` is an example of a **function**
- ❖ Functions have “arguments”



# Variables

- ❖ Everything is stored as a variable
- ❖ Variables are assigned using `<-`



# Math with variables

Math using variables with *just one* value

```
> x <- 5  
> x
```

```
[1] 5
```

```
> x + 3
```

```
[1] 8
```

Math using variables with *multiple* values

```
> a <- 3:6  
> a
```

```
[1] 3 4 5 6
```

```
> a+2; a*3
```

```
[1] 5 6 7 8
```

```
[1] 9 12 15 18
```

# Variable types

- ❖ `numeric` – 110, 55.2, 123
- ❖ `integer` – 1L, 55L, 100L (where “L” declares this as an integer)
- ❖ `character` – “hello”, “R is fun”, “5” (sometimes called strings)
- ❖ `logical` – TRUE or FALSE
- ❖ `missing` – NA

We can use the `class()` function to check the data type of a variable

```
class(2023)
```

```
class("example")
```

# Object types

**Vector:** One-dimensional data set of *one* data type

Vector



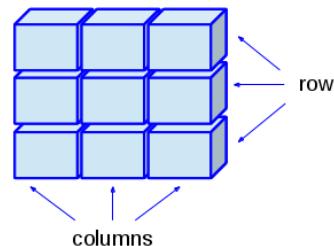
```
c(1, 2, 3)
```

```
c("one", "two", "three")
```

```
c(TRUE, FALSE, FALSE)
```

**Matrix:** Two-dimensional data set of *one* data type

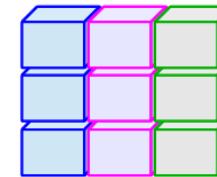
Matrix



	[,1]	[,2]	[,3]
[1,]	1	2	3
[2,]	4	5	6
[3,]	7	8	9

**Data frame:** Two-dimensional data set of *any* data type

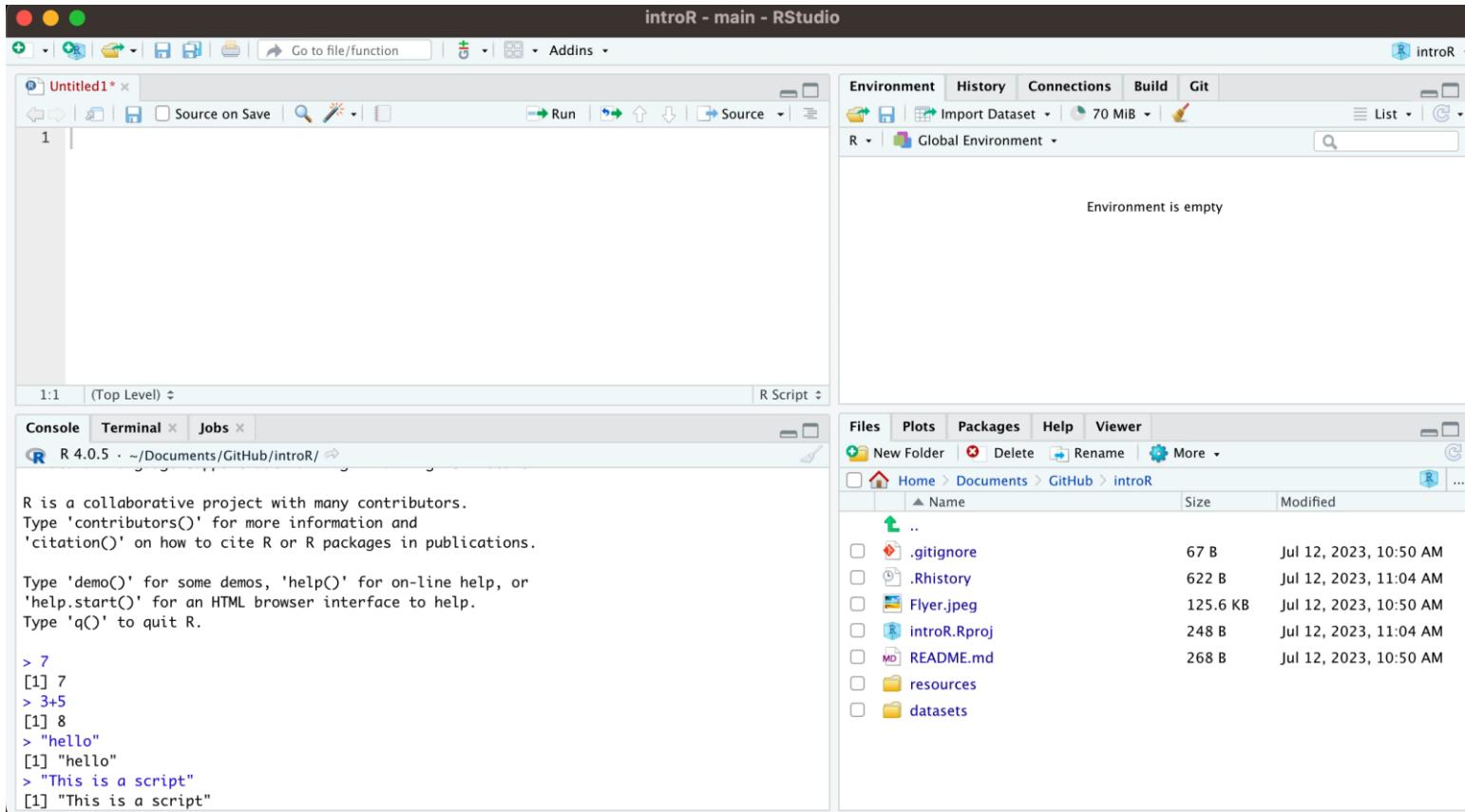
Data Frame  
(Table)



	month	day	year
1	May	1	2023
2	June	12	2023
3	July	10	2023

# Getting data into RStudio

# Importing data using the environment window

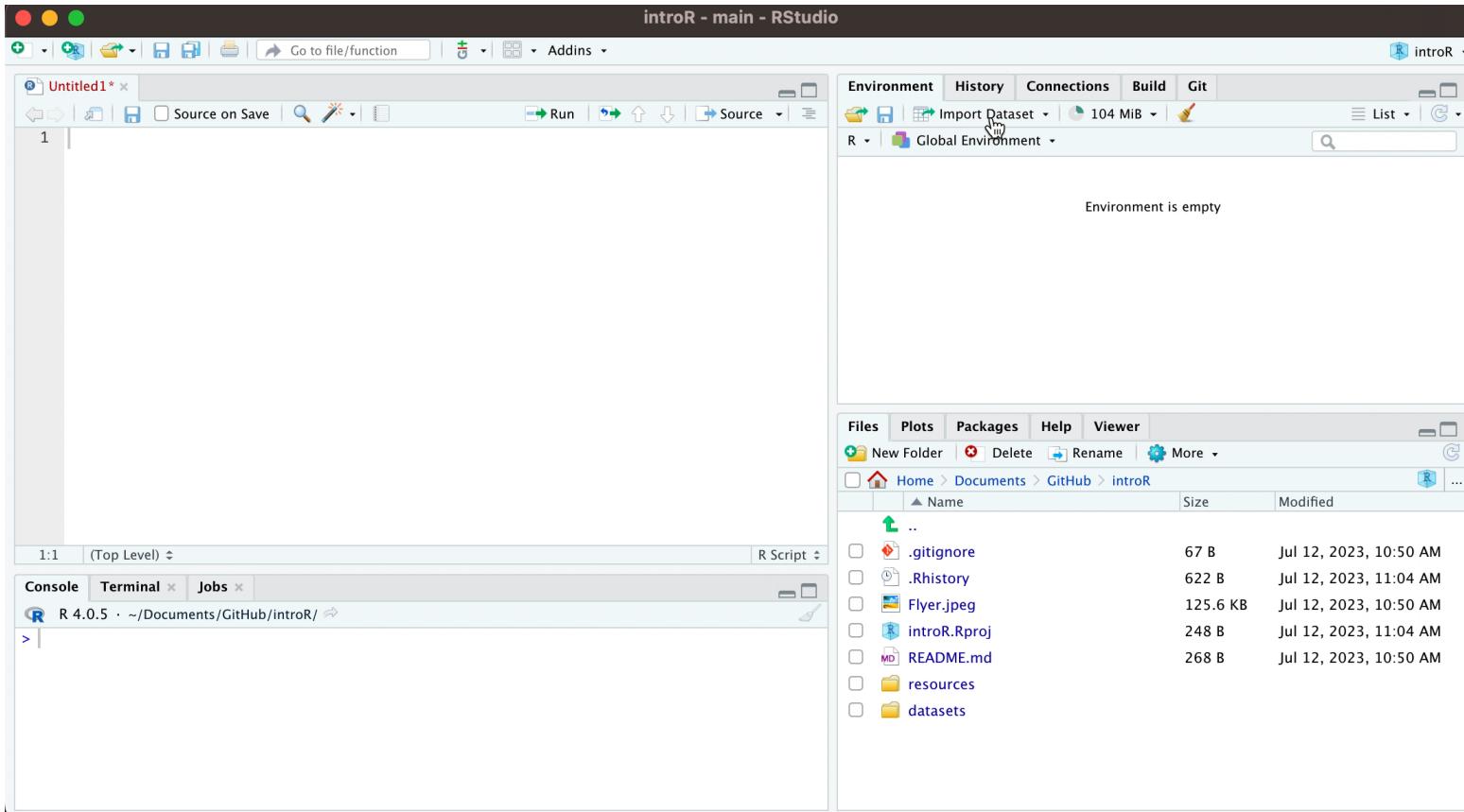


The screenshot shows the RStudio interface with the following components:

- Environment Window:** Shows the "Global Environment" tab with the message "Environment is empty".
- Files Window:** Shows a file tree for the project "introR" located at "/Documents/GitHub/introR". The tree includes files like ".gitignore", ".Rhistory", "Flyer.jpeg", "introR.Rproj", "README.md", "resources", and "datasets".
- Console Window:** Displays R session output:

```
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.  
  
> 7  
[1] 7  
> 3+5  
[1] 8  
> "hello"  
[1] "hello"  
> "This is a script"  
[1] "This is a script"
```

# Importing excel data using the environment window



The screenshot shows the RStudio interface with the following components:

- Top Bar:** "introR - main - RStudio" with standard OS X window controls.
- Toolbar:** Includes "Go to file/function", "Run", "Source", and "Addins".
- Environment Tab:** Shows "Import Dataset" (104 MB) and "Global Environment". The message "Environment is empty" is displayed.
- Files Tab:** Shows the project directory structure:
  - Home > Documents > GitHub > introR
  - Contents:
    - ..
    - .gitignore (67 B, Jul 12, 2023, 10:50 AM)
    - .Rhistory (622 B, Jul 12, 2023, 11:04 AM)
    - Flyer.jpeg (125.6 KB, Jul 12, 2023, 10:50 AM)
    - introR.Rproj (248 B, Jul 12, 2023, 11:04 AM)
    - README.md (268 B, Jul 12, 2023, 10:50 AM)
    - resources
    - datasets
- Console Tab:** Displays "R 4.0.5 · ~/Documents/GitHub/introR/" and a command prompt (>).
- Terminal Tab:** Not visible in the screenshot.
- Jobs Tab:** Not visible in the screenshot.

# Importing data using code

Code Preview:

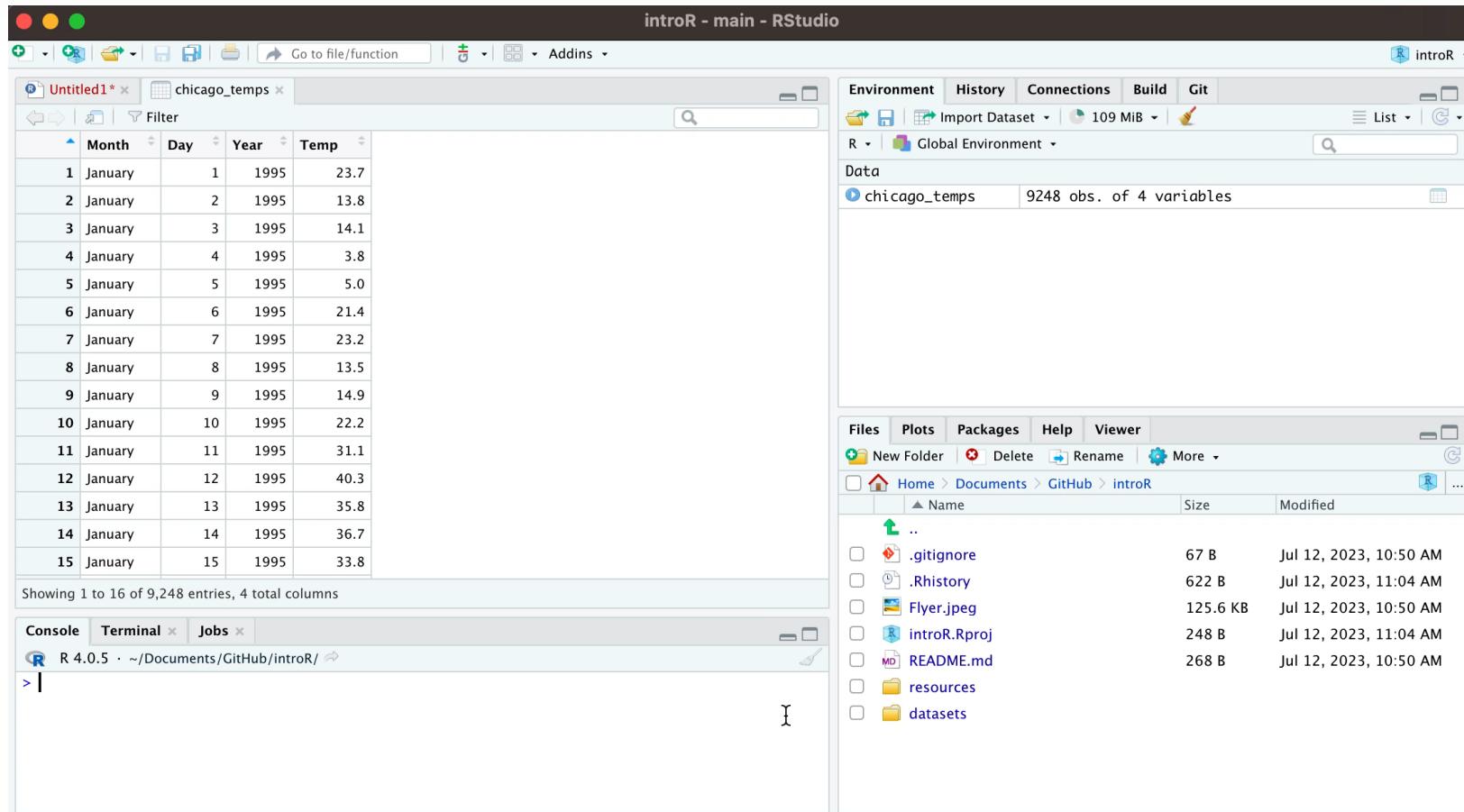
```
library(readxl)
chicago_temps <- read_excel("datasets/chicago_temps
.xlsx")
View(chicago_temps)
```



- ❖ `readxl` is an R package
- ❖ R packages are a collection of data, code, and functions developed by the R user community
- ❖ Packages are stored under a directory called “library”

# Library of packages

introR - main - RStudio



The screenshot shows the RStudio interface with the following components:

- Left Panel (Data View):** Displays a data frame titled "chicago\_temps" with columns: Month, Day, Year, and Temp. The data shows temperatures for January 1995.
- Top Bar:** Includes tabs for Environment, History, Connections, Build, and Git. The Environment tab is active.
- Global Environment:** Shows the "chicago\_temps" dataset with 9248 observations and 4 variables.
- Files View:** Shows the project structure: Home > Documents > GitHub > introR. It lists files and folders including .gitignore, .Rhistory, Flyer.jpeg, introR.Rproj, README.md, resources, and datasets.
- Console View:** Displays the R command "R 4.0.5 · ~/Documents/GitHub/introR/" followed by a prompt ">".

# Working with data

# Basic operations on data: Selecting a column

A data frame named `2021_temp`:

	month	day	year	temp_F
1	January	1	2021	23
2	January	2	2021	14
3	January	3	2021	18
...				
363	December	29	2021	34
364	December	30	2021	36
365	December	31	2021	32

To single out the **month** column

`2021_temp[, 1]`

row

column

	month
1	January
2	January
3	January
...	
363	December
364	December
365	December

# Basic operations on data: Selecting a column

A data frame named `2021_temp`:

	month	day	year	temp_F
1	January	1	2021	23
2	January	2	2021	14
3	January	3	2021	18
...				
363	December	29	2021	34
364	December	30	2021	36
365	December	31	2021	32

To single out the **month** column

`2021_temp$month`

name of column

	month
1	January
2	January
3	January
...	
363	December
364	December
365	December

# Basic operations on data: Selecting multiple columns

A data frame named `2021_temp`:

	month	day	year	temp_F
1	January	1	2021	23
2	January	2	2021	14
3	January	3	2021	18
...				
363	December	29	2021	34
364	December	30	2021	36
365	December	31	2021	32

To single out **month**, **day**,  
and **temp\_F**

`2021_temp[, c(1, 2, 4)]`

`2021_temp[, c(month, day, temp_F)]`

	month	day	temp_F
1	January	1	23
2	January	2	14
3	January	3	18
...			
363	December	29	34
364	December	30	36
365	December	31	32

# Basic operations on data: Creating a new column

A data frame named `2021_temp`:

	month	day	year	temp_F
1	January	1	2021	23
2	January	2	2021	14
3	January	3	2021	18
...				
363	December	29	2021	34
364	December	30	2021	36
365	December	31	2021	32

To create a new column: `temp_C`

```
2021_temp$temp_C <-  
(2021_temp$temp_F - 32)*(5/9)
```

	month	day	year	temp_F	temp_C
1	January	1	2021	23	-5
2	January	2	2021	14	-10
3	January	3	2021	18	-8
...					
363	December	29	2021	34	1
364	December	30	2021	36	2
365	December	31	2021	32	0

# Basic operations on data: Subsetting data

A data frame named `2021_temp`:

	month	day	year	temp_F
1	January	1	2021	23
2	January	2	2021	14
3	January	3	2021	18
...				
363	December	29	2021	34
364	December	30	2021	36
365	December	31	2021	32

To select rows where `temp_F < 20`

```
2021_temp[2021_temp$temp_F < 20,]
```

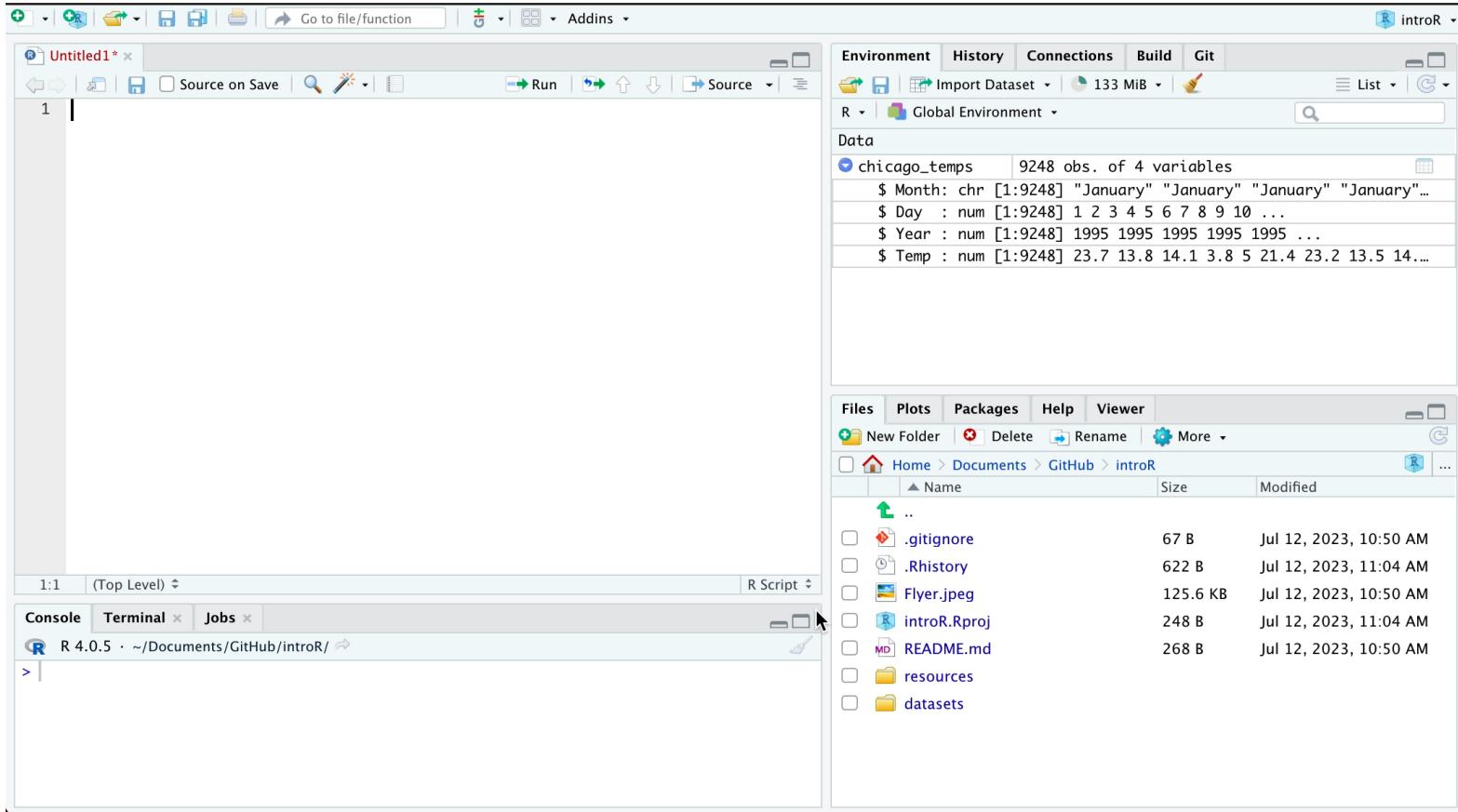
row

column

	month	day	year	temp_F
1	January	2	2021	14
2	January	3	2021	18
3	January	4	2021	19
...				
53	December	17	2021	10
54	December	18	2021	12
55	December	22	2021	15

# Visualizing data

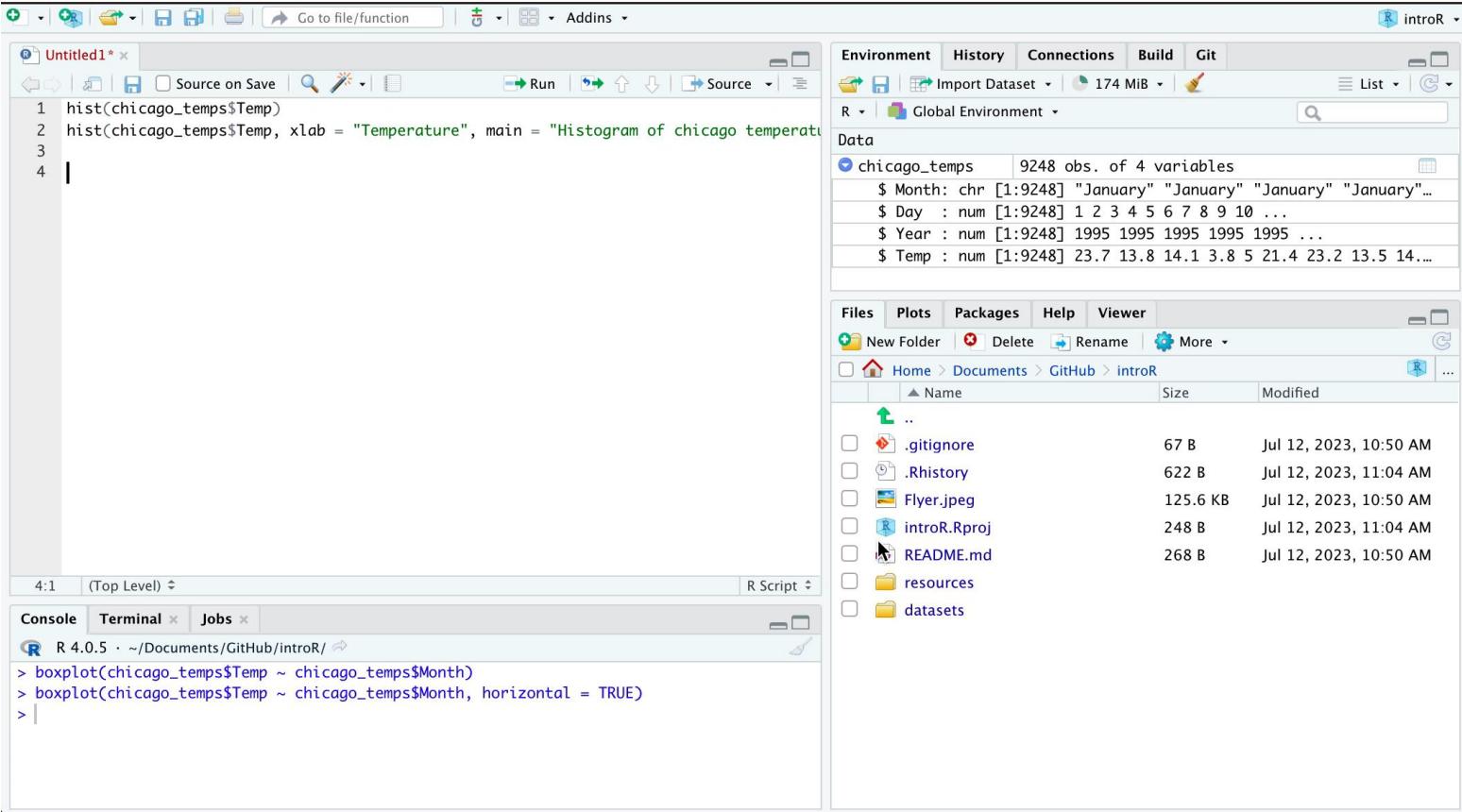
# Basic plots: Histogram



The screenshot shows the RStudio interface with the following components:

- Top Bar:** Includes tabs for Environment, History, Connections, Build, and Git.
- Environment Pane:** Shows the `chicago_temps` dataset with 9248 observations of 4 variables: Month (character), Day (numeric), Year (numeric), and Temp (numeric).
- File Browser:** Shows the directory structure under `introR`, including files like `.gitignore`, `Rhistory`, `Flyer.jpeg`, `introR.Rproj`, `README.md`, `resources`, and `datasets`.
- Console:** Displays the command `R 4.0.5 · ~/Documents/GitHub/introR/` and a prompt `>`.
- Plot Area:** A histogram plot is visible in the background of the main workspace.

# Basic plots: Boxplot



The screenshot shows the RStudio interface with the following components:

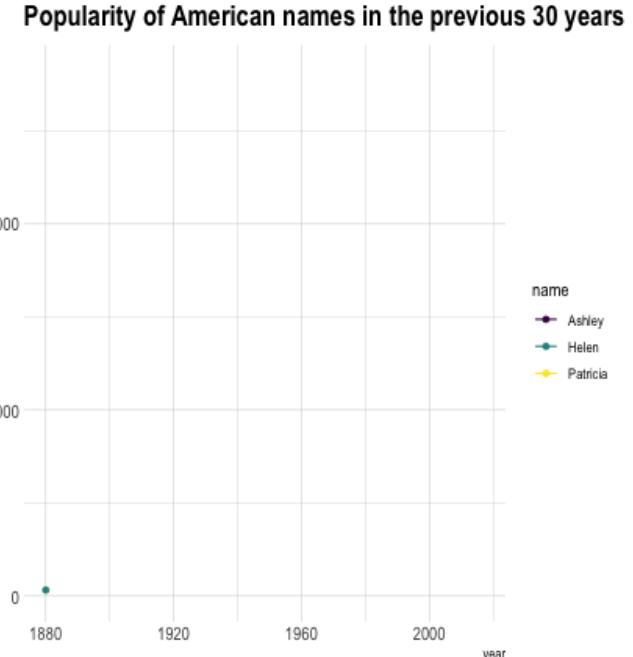
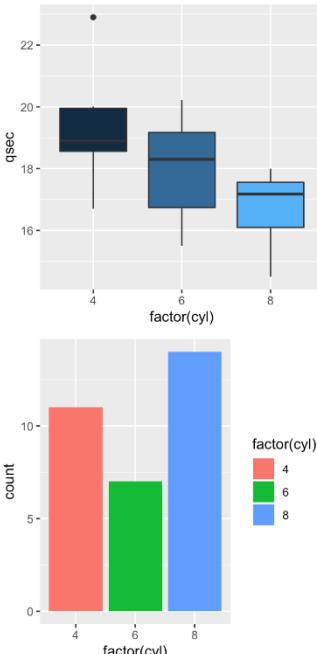
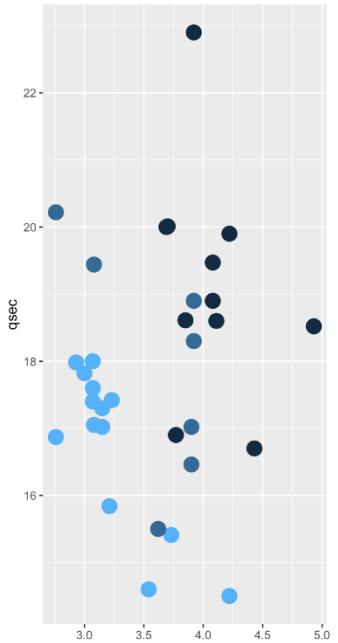
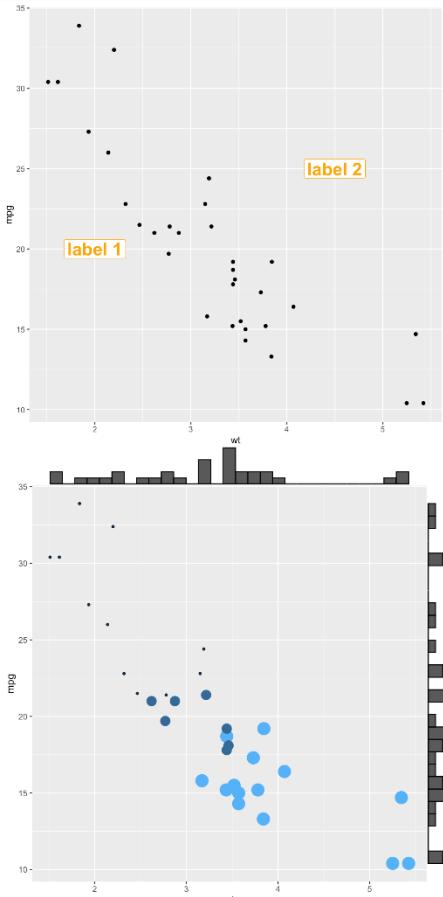
- Code Editor:** An untitled R script containing the following code:
 

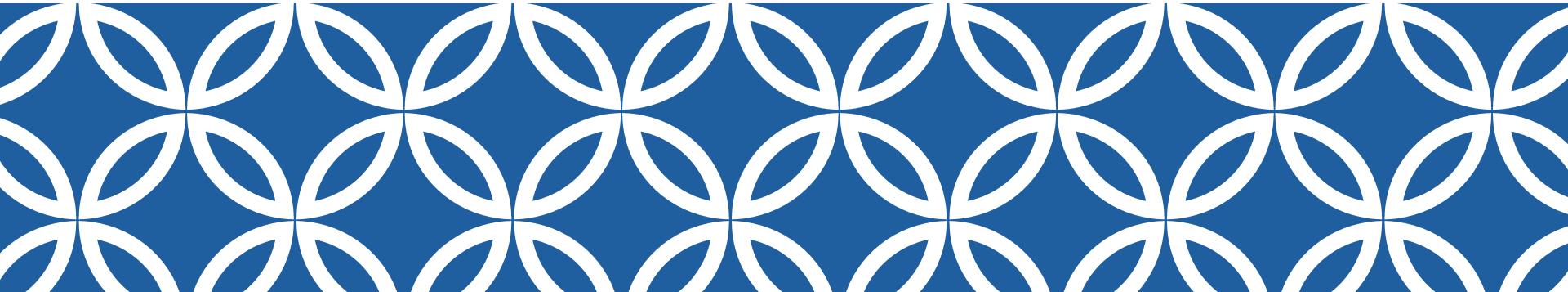
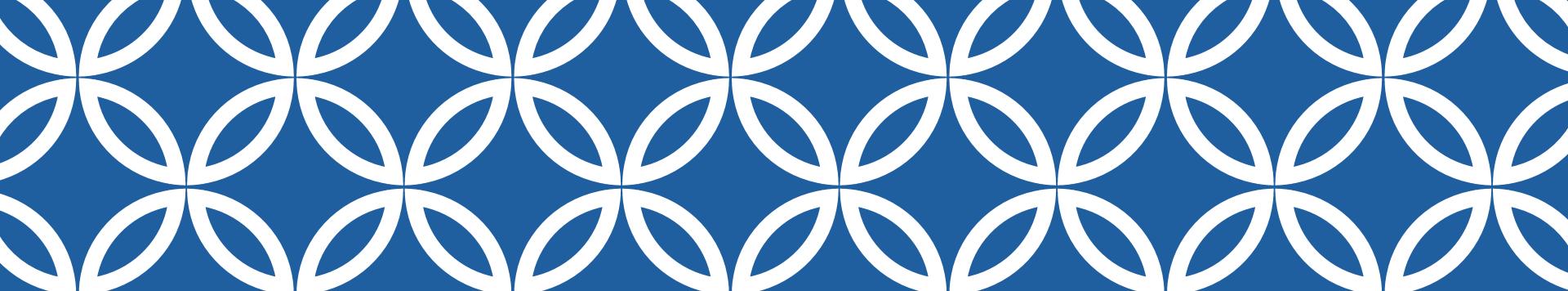
```
1 hist(chicago_temps$Temp)
2 hist(chicago_temps$Temp, xlab = "Temperature", main = "Histogram of chicago temperature")
3
4
```
- Environment View:** Shows the `chicago_temps` dataset with 9248 observations of 4 variables:
  - `$ Month`: chr [1:9248] "January" "January" "January" "January" ...
  - `$ Day`: num [1:9248] 1 2 3 4 5 6 7 8 9 10 ...
  - `$ Year`: num [1:9248] 1995 1995 1995 1995 1995 ...
  - `$ Temp`: num [1:9248] 23.7 13.8 14.1 3.8 5 21.4 23.2 13.5 14. ...
- Files View:** A file browser showing the project structure:
 

Name	Size	Modified
..		
.gitignore	67 B	Jul 12, 2023, 10:50 AM
.Rhistory	622 B	Jul 12, 2023, 11:04 AM
Flyer.jpeg	125.6 KB	Jul 12, 2023, 10:50 AM
introR.Rproj	248 B	Jul 12, 2023, 11:04 AM
README.md	268 B	Jul 12, 2023, 10:50 AM
resources		
datasets		
- Console View:** Displays the R command history:
 

```
R 4.0.5 · ~/Documents/GitHub/introR/
> boxplot(chicago_temps$Temp ~ chicago_temps$Month)
> boxplot(chicago_temps$Temp ~ chicago_temps$Month, horizontal = TRUE)
>
```

# Endless plotting possibilities

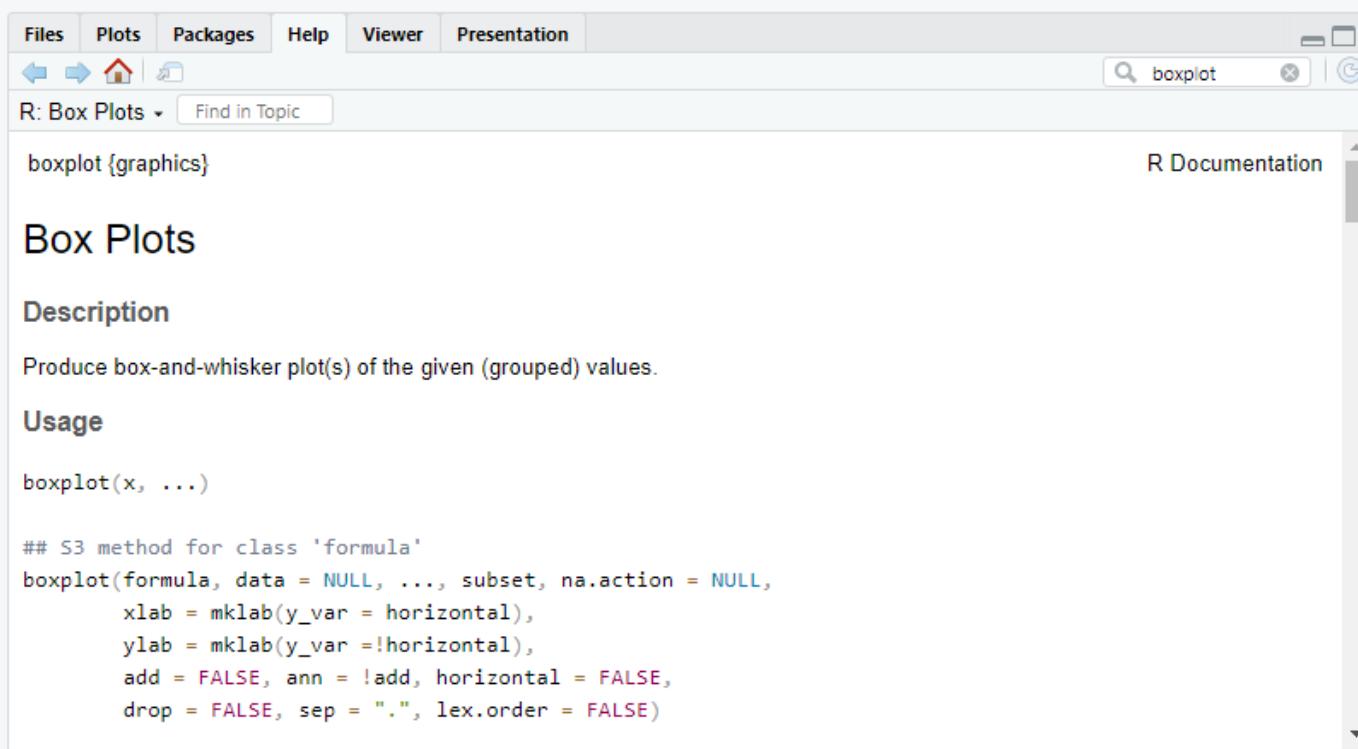




The more you know...

# How to get help

```
> ?boxplot
```

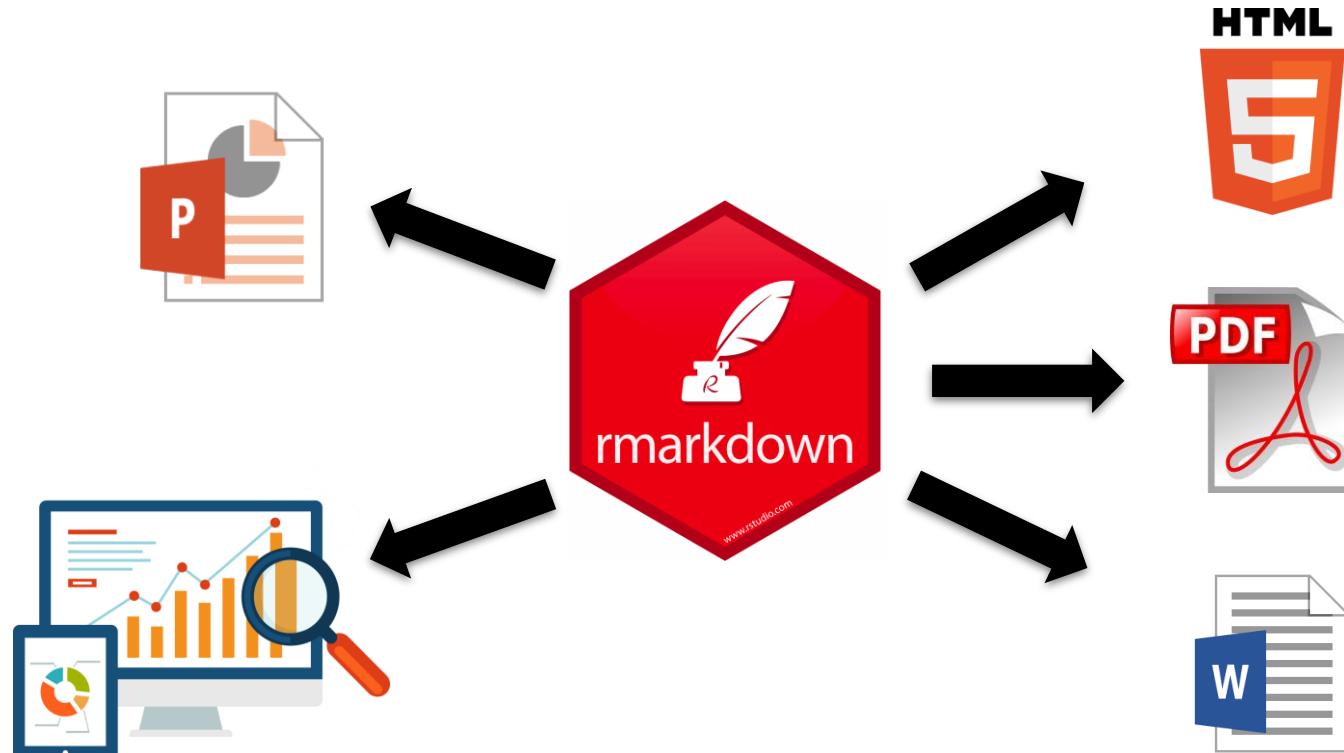


The screenshot shows the R documentation viewer interface. The title bar includes tabs for Files, Plots, Packages, Help, Viewer, and Presentation. A search bar at the top right contains the text "boxplot". The main content area displays the help page for the `boxplot` function. The title "Box Plots" is visible, along with sections for "Description" and "Usage". The "Description" section states: "Produce box-and-whisker plot(s) of the given (grouped) values." The "Usage" section shows the function signature `boxplot(x, ...)` followed by its S3 method implementation for the `formula` class.

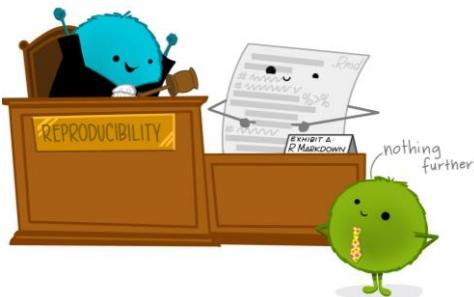
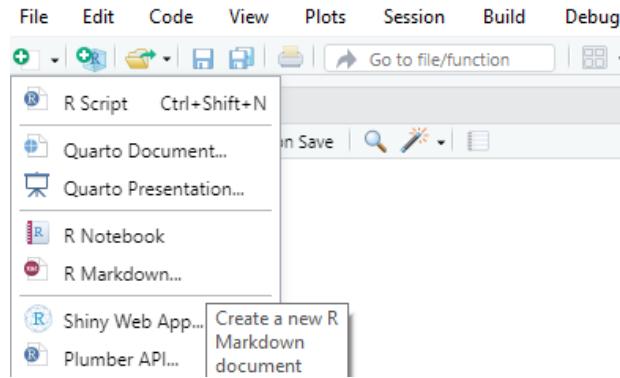
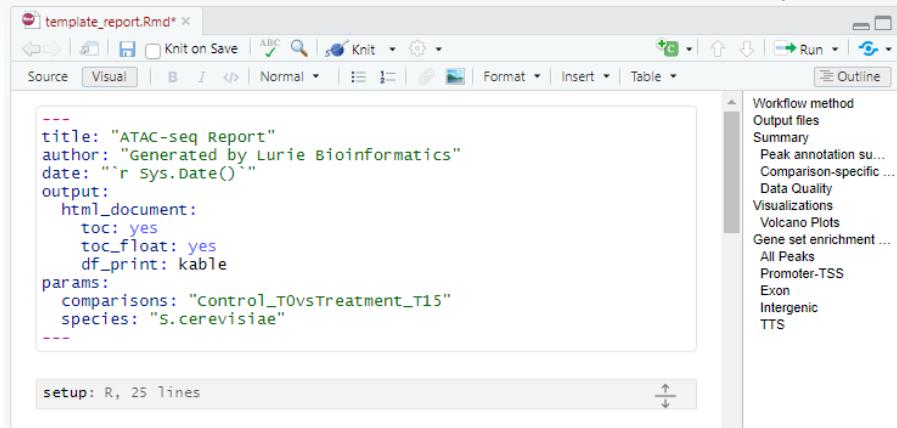
```
boxplot(x, ...)

## S3 method for class 'formula'
boxplot(formula, data = NULL, ..., subset, na.action = NULL,
        xlab = mklab(y_var = horizontal),
        ylab = mklab(y_var = !horizontal),
        add = FALSE, ann = !add, horizontal = FALSE,
        drop = FALSE, sep = ".", lex.order = FALSE)
```

# Reproducible reports with Rmarkdown



# Rmarkdown reports

The screenshot shows an RStudio session with an Rmd file named 'template\_report.Rmd' open. The code in the file is:

```

---
title: "ATAC-seq Report"
author: "Generated by Lurie Bioinformatics"
date: "`r Sys.Date()`"
output:
  html_document:
    toc: yes
    toc_float: yes
    df_print: kable
params:
  comparisons: "Control_T0vsTreatment_T15"
  species: "S.cerevisiae"
---

```

Below the code, it says 'setup: R, 25 lines'. On the right side of the screen, there is a sidebar titled 'Workflow method' with a tree view of the project structure:

- Workflow method
  - Output files
  - Summary
  - Peak annotation su...
  - Comparison-specific ...
  - Data Quality
  - Visualizations
  - Volcano Plots
  - Gene set enrichment ...
  - All Peaks
  - Promoter-TSS
  - Exon
  - Intergenic
  - TTS

## Workflow method

Our workflow relies on the [nf-core ATAC-seq pipeline v1.2.2](#).

Briefly, reads were mapped to the reference genome using `bwa` and peaks were called for each sample independently using `MACS2`. Sample-specific peaks were combined into a consensus peak-set with `bedtools`. `featurecounts` was used to count the number of reads relative to the consensus peak-set across all samples. The resulting raw count matrix served as input to `DESeq2` for differential accessibility analysis. Downstream analyses, including plotting and GSEA, were performed in `R`.

A detailed results description can be found under [pipeline\\_info/results\\_description.html](#). A full list of software versions can be found at [pipeline\\_info/software\\_versions.csv](#).

## Output files

The workflow generates a directory of files: [atacseq-results](#).

# Interactive reports with Rmarkdown

Workflow method
Output files
Summary
Visualizations
Gene set enrichment analysis

## ATAC-seq Report

Generated by Lurie Bioinformatics

2023-06-02

### Workflow method

Our workflow relies on the nf-core ATAC-seq pipeline v1.2.2.

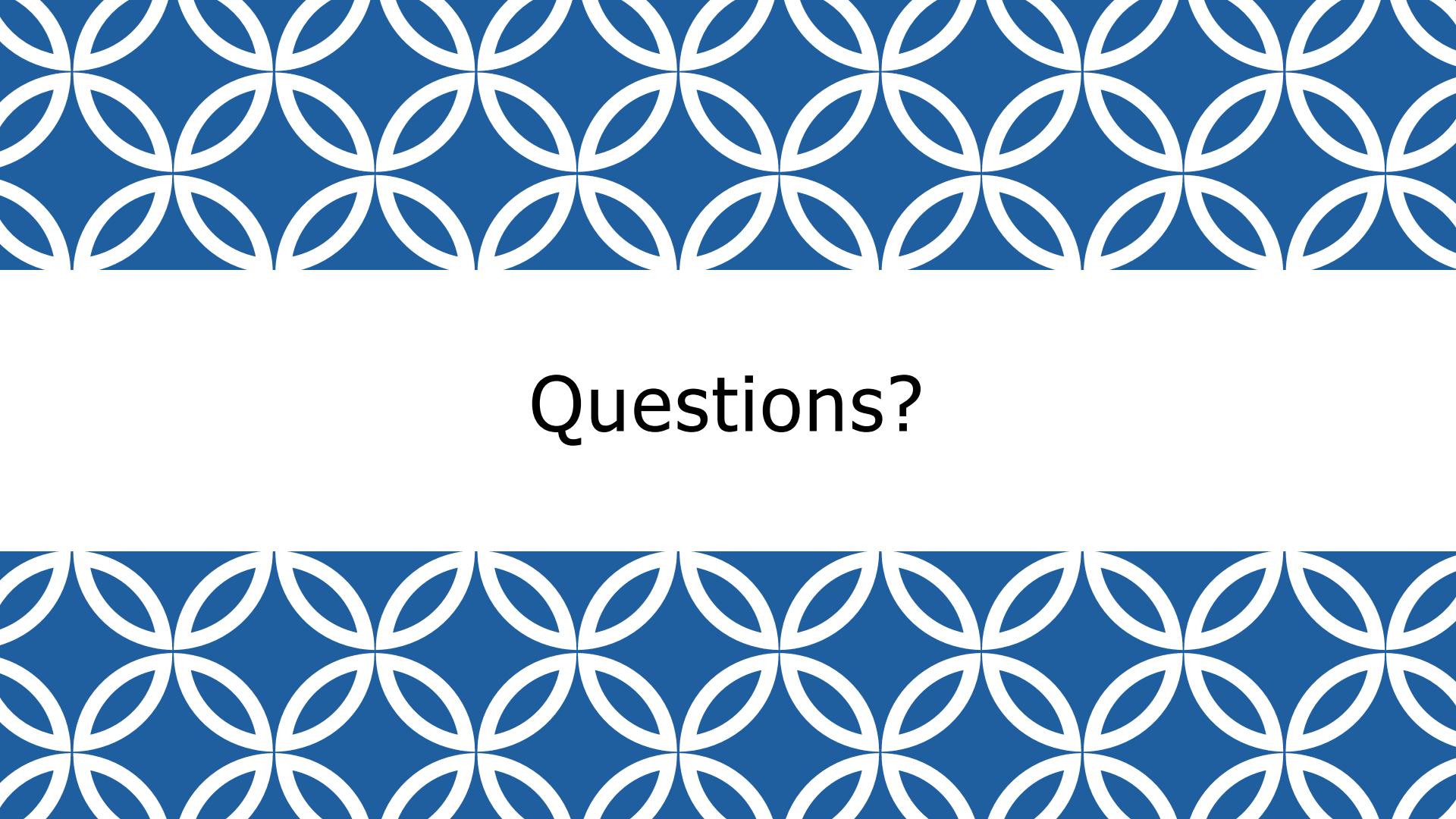
Briefly, reads were mapped to the reference genome using `bwa` and peaks were called for each sample independently using MACS2. Sample-specific peaks were combined into a consensus peak-set with `bedtools`. `featureCounts` was used to count the number of reads relative to the consensus peak-set across all samples. The resulting raw count matrix served as input to `DESeq2` for differential accessibility analysis. Downstream analyses, including plotting and GSEA, were performed in `R`.

A detailed results description can be found under `pipeline_info/results_description.html`. A full list of software versions can be found at `pipeline_info/software_versions.csv`

### Output files

The workflow generates a directory of files: `atacseq-results`.

```
atacseq-results
├── peakannotatedresults.xlsx
├── bigwig
│   ├── ...
│   └── <sample>.mLb.cLN.bigWig
└── DiffAnalysis
    └── consensus_peaks.mRp.cLN.annotatePeaks.txt
        └── consensus_peaks.mRp.cLN.log
```



# Questions?