

# A Walk on the Side

an introduction to R for data analysis

*Online  
edition!*

...

GW Libraries Workshop  
January 22, 2021

[go.gwu.edu/rworkshop](https://go.gwu.edu/rworkshop)

# Logistics

- Schedule (approximate)
  - 10-12 R, with 1 ☕ break
  - 12-1 Lunch break
  - 1-3 R, with 1 ☕ break
- Kiri can provide individual help
  - We can use Zoom breakout rooms if needed
- Collaborative Notes document: [bit.ly/r\\_oct1](https://bit.ly/r_oct1)





# Goals



# Learning Objectives



[Hopefully] You will learn how to do some of the following:

- Set up your laptop with R & RStudio (done!)
- Write and run an R program in RStudio
- Use variables of different types in R
- Use vectors and data frames in R to represent data
- Import & export data files
- "Wrangle" data in R
- Explore data in R with basic statistics and data visualizations
- Learn how to look for help to overcome obstacles

# Agenda

- About R and RStudio
- Along the way: How to get help
- Hands-on:
  - variables
  - logical expressions
  - values, vectors, and data frames
  - R Studio projects
  - reading in data
  - exploring data
  - data wrangling:  
cleaning and reshaping
  - data visualization
  - data analysis
  - functions
  - R Markdown / reports
- Resources for further learning



# Acknowledgments



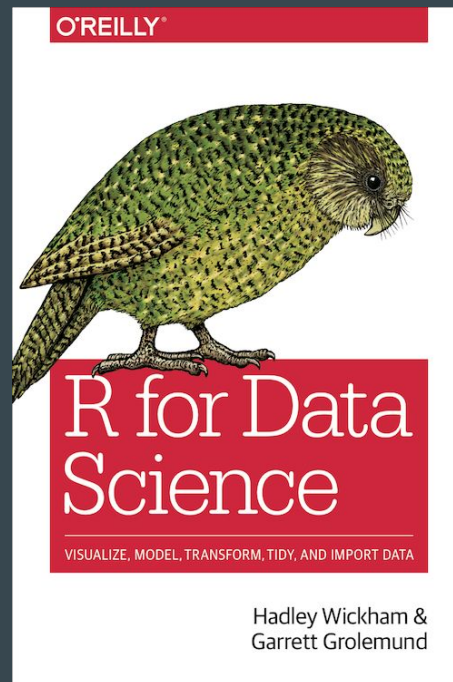
Teaching basic lab skills  
for research computing

**DATA CARPENTRY**

BUILDING COMMUNITIES TEACHING UNIVERSAL DATA LITERACY

**R Tutorial**

An R Introduction to Statistics



# Workshop Housekeeping



Ask questions! Either via voice or chat

Use chat to help each other out

If something is confusing in the workshop, let us know.

# About R

- Free/Open source
- Cross-platform (Mac, Windows, Linux)
- For statistical computing (and data visualization)
- CRAN - [r-project.org](https://r-project.org)
  - [R packages](#)
  - [R journal](#)



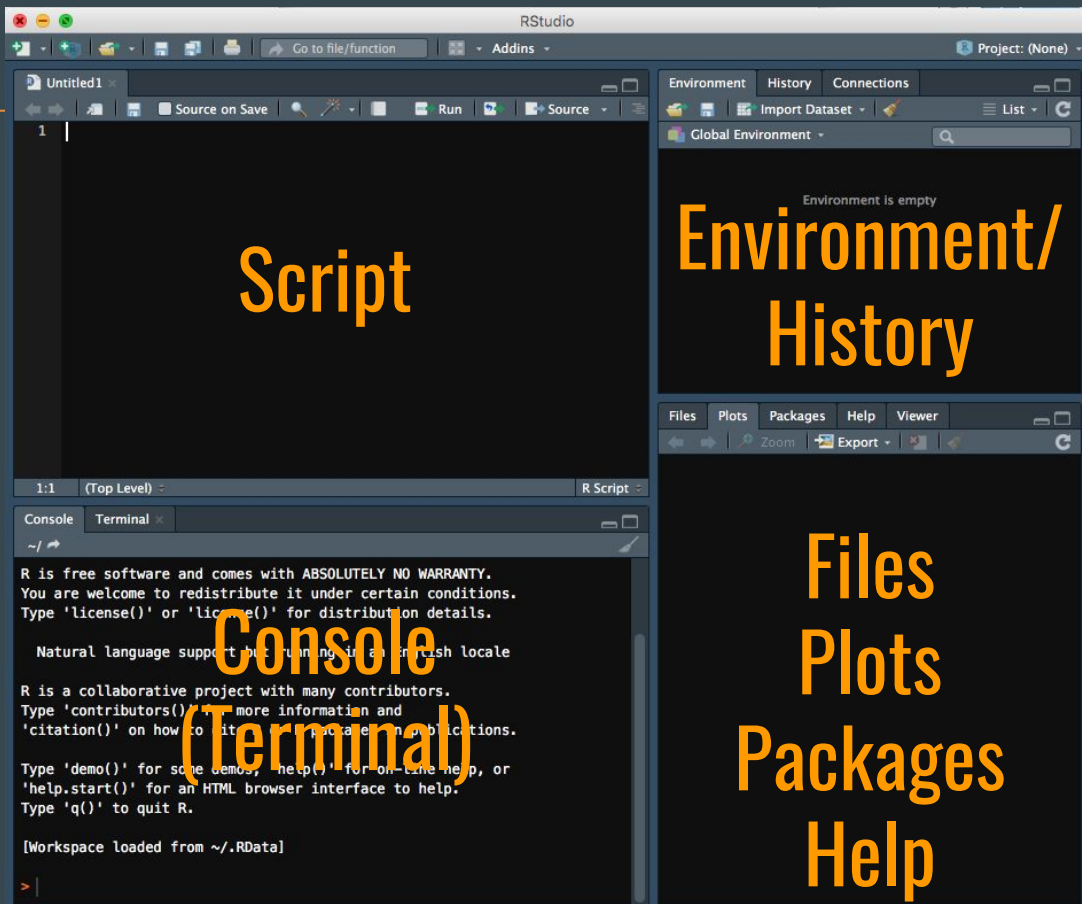




# Reasons ReseaRcheRs pRefeR R

- Scripted language (vs. point/click)
- Features built around working with data
- Reproducibility
- Interdisciplinary
- Extensible
- Beautiful data visualization
- Community - RStudio Community, Stack Overflow

# R Studio



A WALK ON THE R SIDE



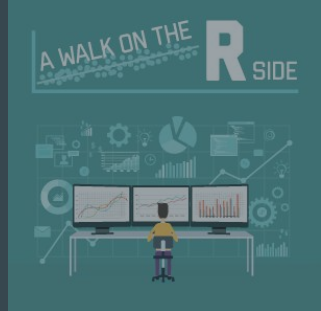
# Variables/Objects

"Binding" data to a named object/variable allows you to store data in memory and access it later.

```
x <- 5
```

```
y <- c("Washington", "Chicago", "Washington", "Boston")
```

```
z <- data.frame(pt_id = c("A001", "B204"), bpm = c(60, 72))
```





# Variables

- Try using R as a "calculator" in the Console
  - Try some mathematical functions, too
- Create some variables
  - variable naming
  - `<-` for assigning values to variables (Option - on Mac, Alt - on Win)
  - numeric, character, logical
  - Watch the Environment pane!
  - `typeof()`
  - Coercion w/ `as.integer`, `as.character`, `as.logical`, `as...`

# Logical Expressions

- Operators include:  
==, <, >, ! (not), & (and), | (or), etc.





# Basic Data Structures

## Atomic Vector

10.2
------

## Vector

1	10.2
2	11.3
3	11.5
4	12.0

## Data Frame

	time	temp	boiling
1	51	10.2	FALSE
2	58	11.3	FALSE
3	63	11.5	FALSE
4	70	12.0	TRUE



# Vectors

# Vectors

- A vector is
  - A sequence of data elements (components) all of the same type.
- Create vectors with `c()` (short for "combine")







# Let's pause to explore some useful tabs in RStudio

~/R Projects/rstudio-testproject - master - RStudio

Workshop.R x gapminder x

Source on Save Run Addins

```
1 library('tidyverse')
2 gapminder <- read_csv('data/gapminder.csv')
3
4 by_year <- gapminder %>%
5   group_by(year) %>%
6   summarize(weighted_avg_lifeExp = sum(pop*lifeExp)/sum(pop))
7
8 # Plot the data (scatterplot)
9 plot(y = by_year$weighted_avg_lifeExp, x = by_year$year, col='blue')
10 # Build a linear regression model
11 mod = lm(data = by_year, weighted_avg_lifeExp ~ year)
12 # Plot the line
13 abline(mod)
14
15 # or using ggplot2:
16 ggplot(data = gapminder, aes(x=year, y=lifeExp, base_indent=1, color=continent)) +
17   geom_point() +
18   # ...
19
20 5:1 (Top Level) R Script

Environment History Connections Git



Global Environment



|           |                                |
|-----------|--------------------------------|
| df        | 3 obs. of 2 variables          |
| gapminder | 1704 obs. of 6 variables       |
| housedata | 1460 obs. of 81 variables      |
| lemod     | List of 12                     |
| mod       | List of 12                     |
| mx        | logi [1:3, 1:2] NA NA NA NA NA |
| mx2       | List of 6                      |



Values



|         |                         |
|---------|-------------------------|
| primes  | num [1:6] 2 3 5 7 11 13 |
| testnum | 5                       |



Files Packages Help



R: Reduces multiple values down to a single value



summarise (dplyr)



R Documentation



Reduces multiple values down to a single value



Description



summarise() is typically used on grouped data created by group_by(). The output will have one row for each group.



Usage



```
summarise(.data, ...)
```



```
summarize(.data, ...)
```



Arguments



.data A tbl. All main verbs are S3 generics and provide methods for tbl_df(), dtplyr::tbl_dt() and dbplyr::tbl_dbi().



... Name-value pairs of summary functions. The name will be the name of the variable in the result. The value should be an expression that returns a single value like min(x), n(), or sum(is.na(y)).



Console



```
~/R Projects/rstudio-testproject - master - RStudio
```



```
[1,]
[1,] 1
[2,] 2
[3,] "A"
[4,] "b"
[5,] 2
[6,] 2
> mx2 = matrix(list(1, 2, "A", "b"), nrow=2, ncol=2)
> mx2
      [,1] [,2]
[1,] 1    "A"
[2,] 2    "b"
> mx2 = matrix(list(1, 2, "A", 3, "b", 5), nrow=3, ncol=2)
> mx2
      [,1] [,2]
[1,] 1    3
[2,] 2    "b"
[3,] "A"  5
>
```


```



# Data Frames



# Data Frames

- A **data.frame** stores a data table
- Comprised of **vectors** of equal length. Vectors become columns.
- Columns and rows can have names.
- **tibble** (from the tibble package) has some advantages over **data.frame**



# A brief word on **list** and **matrix**



# Projects in RStudio

# Projects in RStudio

## Recommendations:

- Use [Github for] **version control!**
- Create **folders** to keep things organized





It's time to **import** some data!



# Data Importing

- Prepare data as "tidy"
  - rectangular
  - one table per file
  - rows are observations, columns are variables
- Formats: CSV, TSV, Excel, Fixed-Width, JSON... and with the right packages: Stata, SPSS, SAS... (using **rio** or **haven**)
- A word about "big data" (consider **data.table**)





# R Packages

# Installing and loading R packages

- `install.packages('mypackage')`
- `library(mypackage)`

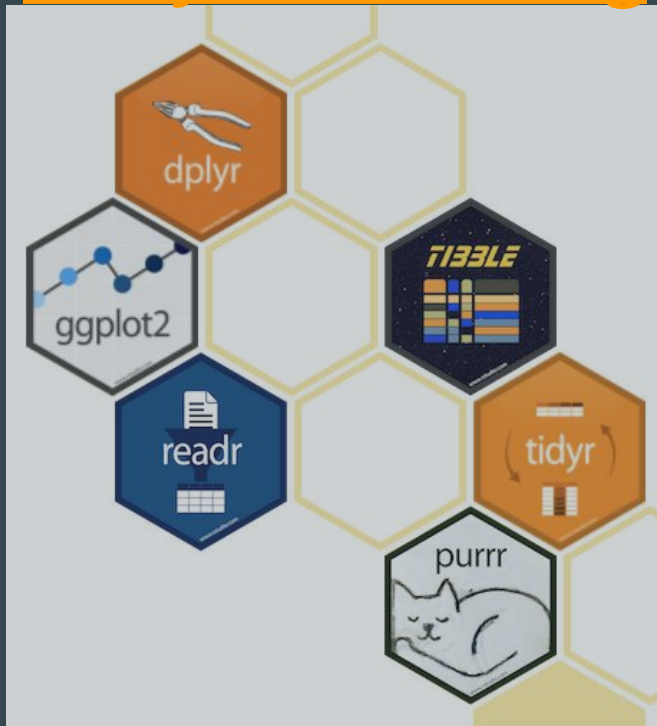




# Tidyverse Core Packages

[tidyverse.org](https://tidyverse.org)

- ggplot2 - graphics
- dplyr - data manipulation
- tidyr - tidying data
- readr - reading in data
- tibble - modern data frame
- purrr - functional programming



## Other often-used R packages

Loading in various data file types ♦ haven, readxl

Mapping ♦ rgdal, tmap, leaflet

Analyzing 2D and 3D shapes ♦ geomorph

Genomic data ♦ bioconductor

Cluster analyses ♦ cluster

Time series data ♦ forecast

Text mining ♦ qdap, sentimentr, tidytext

graph/network analysis ♦ igraph, sna

Interactive web visualizations ♦ shiny

Web scraping ♦ rvest



# Exploring Data

- head, tail
- subsetting
- slicing and dicing





# Data Wrangling

[flickr.com/photos/thewomensmuseum/3687975017/](https://www.flickr.com/photos/thewomensmuseum/3687975017/)



# Data Transformation using the dplyr package

- filter()
- arrange()
- select()
- mutate()
- summarize()
- group\_by()
- ...

You will want to use a "pipe": `%>%`  
(shortcut: **control-shift-M**)



# Data Tidying with dplyr

- `gather()`
- `spread()`
- `separate()`
- `unite()`





# Joining with dplyr

"Merges" tables together

- `left_join()`
- `right_join()`
- ...





# Data Visualization with "base R" and ggplot



# Data Analysis



# Functions



# R Markdown



# R Markdown

- A format for writing reproducible, dynamic reports with R (as HTML, PDF, MS Word, and more)
- [rmarkdown.rstudio.com](https://rmarkdown.rstudio.com)
- # Header 1  
## Header 2  
*\*Italic\** **\*\*bold\*\***
- Insert R code directly into your document

```
```{r setup}
# your R code goes here
```
```
- Include LaTeX code with \$ or \$\$



# R Shiny



## Some Handy R Links



# Tutorials

- RStudio R paths: [education.rstudio.com/learn/](https://education.rstudio.com/learn/)
- Data Carpentry & Software Carpentry:
  - [datacarpentry.org/R-ecology-lesson/](https://datacarpentry.org/R-ecology-lesson/)
  - [datacarpentry.org/r-socialsci/](https://datacarpentry.org/r-socialsci/)
  - [swcarpentry.github.io/r-novice-inflammation](https://swcarpentry.github.io/r-novice-inflammation)
  - [swcarpentry.github.io/r-novice-gapminder](https://swcarpentry.github.io/r-novice-gapminder)
- LinkedIn Learning @ GW: [go.gwu.edu/linkedinlearning](https://go.gwu.edu/linkedinlearning)
- [r-tutor.com/r-introduction](https://r-tutor.com/r-introduction) & [r-tutor.com/elementary-statistics](https://r-tutor.com/elementary-statistics)
- R Graph Gallery (w/code): [r-graph-gallery.com](https://r-graph-gallery.com)



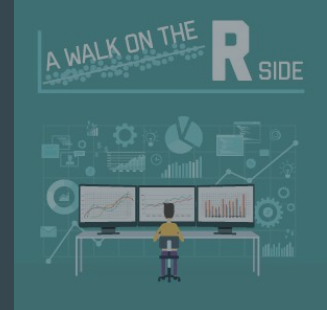
# Books you can access for free

- Free books online - Hadley Wickham:
  - R for Data Science [r4ds.had.co.nz](https://r4ds.had.co.nz)
  - Advanced R [adv-r.hadley.nz/](https://adv-r.hadley.nz/)
- Through your GW library privileges:

ADVANCED SEARCH

Search for: ☐ Catalog + Articles ☒ Catalog ☐ Articles

Subject ▼ contains ▼ R (Computer programming language)



# Reference Links

- R language (CRAN): [r-project.org](https://r-project.org)
- R search engine: [rseek.org](https://rseek.org)
- [rstudio.com](https://rstudio.com)
  - Cheat Sheets! [rstudio.com/resources/cheatsheets](https://rstudio.com/resources/cheatsheets)
- [stackoverflow.com](https://stackoverflow.com)



# Thanks!

Dan Kerchner

[kerchner@gwu.edu](mailto:kerchner@gwu.edu)

These slides: [go.gwu.edu/rworkshop](https://go.gwu.edu/rworkshop)

R or Statistics Appointments: [calendly.com/statistical-consulting-gw](https://calendly.com/statistical-consulting-gw)

Appointments with me: [calendly.com/kerchner](https://calendly.com/kerchner)

Coding consultations (Python, git, etc.): [calendly.com/gwul-coding/](https://calendly.com/gwul-coding/)

