Introduction

Welcome to class!

1. Introductions

2. Class overview

3. Getting R up and running

[Photo by Belinda Fewings on Unsplash]

Before we start ..

Poll: How are you feeling right now?

About Us

Carrie Wright (she/her)

Senior Staff Scientist and Training Lead, Fred Hutchinson Cancer Center

Associate, Department of Biostatistics, JHSPH

PhD in Biomedical Sciences

Email: cwright2@fredhutch.org Web: https://carriewright11.github.io

About Us

Ava Hoffman (she/her)

Senior Staff Scientist, Fred Hutchinson Cancer Center

Associate, Department of Biostatistics, JHSPH

PhD in Ecology

Email: ahoffma2@fredhutch.org Web: https://avahoffman.com

About Us: TA

Elizabeth Humphries (she/her)

Staff Scientist, Fred Hutchinson Cancer Center

PhD in Molecular Epidemiology

Email: ehumphri@fredhutch.org

NOTE this is not her dog

About you!

Please introduce yourself on Slack!

[Slack Workspace](#)

The Learning Curve

Learning a programming language can be very intense and sometimes overwhelming.

We recommend fully diving in and minimizing other commitments to get the most out of this course.

Like learning a spoken language, programming takes **practice**.

The Learning Curve

Learning R has been career changing for all of us, and we want to share that!

We want you to succeed – We will get through this together!

What is R?

- R is a language and environment for statistical computing and graphics developed in 1991

- R is the open source implementation of the S language, which was developed by Bell laboratories in the 70s.

- The aim of the S language, as expressed by John Chambers, is "to turn ideas into software, quickly and faithfully"

[source: http://www.r-project.org/, https://en.wikipedia.org/wiki/S_(programming_language), https://en.wikipedia.org/wiki/Bell_Labs)]

What is R?

- **R**oss Ihaka and **R**obert Gentleman at the University of Auckland, New Zealand developed R

- R is both open source and open development

[source: http://www.r-project.org/, https://en.wikipedia.org/wiki/R_(programming_language)]

Why R?

- Free (open source)

- High level language designed for statistical computing

- Powerful and flexible - especially for data wrangling and visualization

- Extensive add-on software (packages)

- Strong community

[source: https://github.com/rladies/meetup-presentations_baltimore]

Why not R?

- Little centralized support, relies on online community and package developers

- Annoying to update

- Slower, and more memory intensive, than the more traditional programming languages (C, Perl, Python)
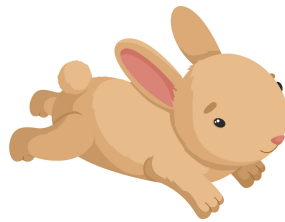
FAST   SLOW

[source -School vector created by nizovatina - www.freepik.com]

Introductions

What do you hope to get out of the class?

Why do you want to use R?

[Photo by Nick Fewings on Unsplash]

Logistics

Course Website

https://daseh.org/

Materials will be uploaded the night before class. We are constantly trying to improve content! Please refresh/download materials before class.

Learning Objectives

- Understanding basic programming syntax

- Reading data into R

- Recoding and manipulating data

- Using add-on packages (more on what this is soon!)

- Making exploratory plots

- Performing basic statistical tests

- Writing R functions

- **Building intuition**

Course Format

ONLINE VIRTUAL COURSE

- Lecture with slides, interactive
- Lab/Practical experience
- Two 10 min breaks each day - timing may vary
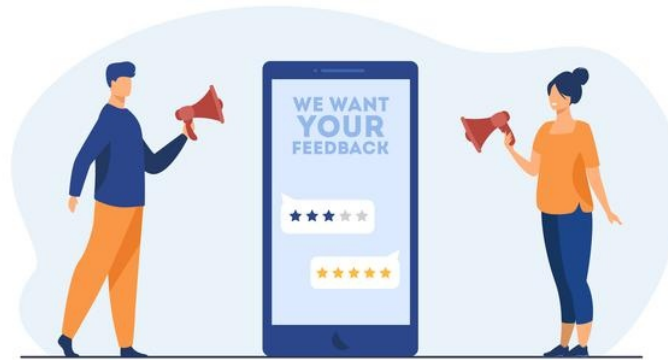- July 8-18, 10:30am - 2:00pm PST on Zoom

IN-PERSON CODE-A-THON

- Mostly independent group work
- Frequent check-ins with instructors and other groups
- Some lectures about the practical aspects of coding
- July 29-31 (in person in Seattle)

Pulse Check Survey

https://forms.gle/hQ7jdHZSZoDpHAdE6

Let us know anonymously how you're doing with the material.

[source - Banner vector created by pch.vector - www.freepik.com]

Homework

While we do have homework assignments on the course schedule, these are **strictly optional!!!**

We encourage you to try the assignments, as the best way to get comfortable with any programming language is through practice.

Your Setup

If you can, we suggest working virtually with a **large monitor or two screens**. This setup allows you to follow along on Zoom while also doing the hands-on coding.
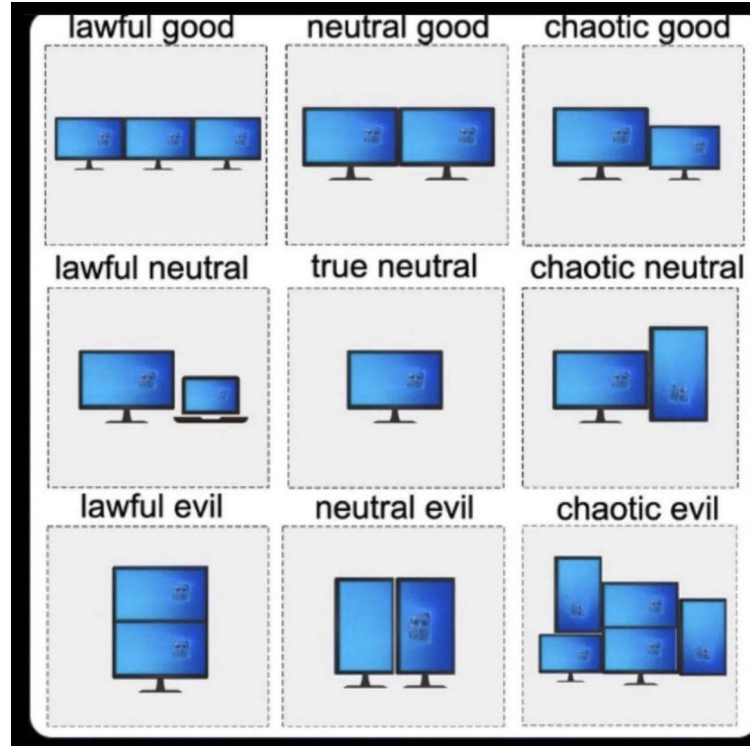
[source - reddit.com]

Research Survey

Research Survey

We are collecting data about user experience with our course to learn more about how to improve the data science education experience. This data may ultimately be used for a research publication and reporting to the NIH.

https://forms.gle/e2CQFDJsgyZwLV3S9

Getting Started

Installing R

- Install the latest R version (4.3.2 (called 'Eye Holes') as of October 31, 2023)

- Install RStudio

More detailed instructions on the website.

RStudio is an **integrated development environment** (IDE) that makes it easier to work with R.

More on that soon!

Getting files from downloads

This course will involve moving files around on your computer and downloading files.

If you are new to this - check out these videos.

If you have a PC: https://youtu.be/we6vwB7DsNU

If you have a Mac: https://www.youtube.com/watch?v=Ao9e0cDzMrE

You can find these on the resource page of the website.

Basic terms

R jargon: https://link.springer.com/content/pdf/bbm%3A978-1-4419-1318-0%2F1.pdf

Package - a package in R is a bundle or "package" of code (and or possibly data) that can be loaded together for easy repeated use or for **sharing** with others.

Packages are analogous to a software application like Microsoft Word on your computer. Your operating system allows you to use it, just like having R installed (and other required packages) allows you to use packages.

Basic terms

Function - a function is a piece of code that allows you to do something in R. You can write your own, use functions that come directly from installing R, or use functions from additional packages.

You can think of a function as **verb** in R.

A function might help you add numbers together, create a plot, or organize your data. More on that soon!

```
sum(1, 20234)

[1] 20235
```

Basic terms

Argument - what you pass to a function

- can be data like the number 1 or 20234

```
sum(1, 20234)
```

```
[1] 20235
```

- can be options about how you want the function to work such as `digits`

```
round(0.627, digits = 2)
```

```
[1] 0.63
```

```
round(0.627, digits = 1)
```

```
[1] 0.6
```

Basic terms

Object - an object is something that can be worked with or on in R - can be lots of different things! You can think of objects as **nouns** in R.

- a matrix of numbers
- a plot
- a function
- data

… many more

Variable and Sample

- **Variable**: something measured or counted that is a characteristic about a sample

examples: temperature, length, count, color, category

- **Sample**: individuals that you have data about -

examples: people, houses, viruses etc.

head(iris)

```
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
```

Columns and Rows

[source]

Sample = Row
Variable = Column

Data objects that looks like this is often called a **data frame**.

Fancier versions from the tidyverse are called **tibbles** (more on that soon!).

More on Functions and Packages

- When you download R, it has a "base" set of functions/packages (**base R**)

    - You can install additional packages for your uses from CRAN or GitHub

    - These additional packages are written by RStudio or R users/developers (like us)

    - There are also packages for bioinformatics available at Bioconductor

Using Packages

- Not all packages available on CRAN or GitHub are trustworthy

- Posit makes many useful packages

- How to trust an R package

- Many packages have accompanying academic papers published in peer-reviewed journals

- Widely used packages have better documentation (official and in forums) and are more likely free of errors

Tidyverse and Base R: Two Dialects

We will mostly show you how to use tidyverse packages and functions.

This is a newer set of packages designed for data science that can make your code more **intuitive** as compared to the original older Base R.

Tidyverse advantages:
- **consistent structure** - making it easier to learn how to use different packages
- particularly good for **wrangling** (manipulating, cleaning, joining) data
- more flexible for **visualizing** data

Packages for the tidyverse are managed by a team of respected data scientists at Posit.

See this article for more info.

Package Installation
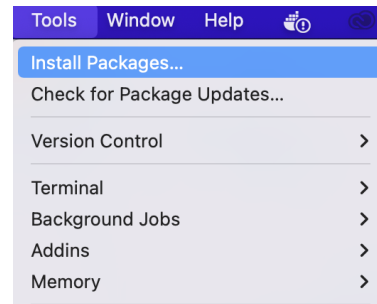
We will practice this in labs :)

Differs depending on the source (CRAN, GitHub, etc)

Must be done **once** for each installation of R (e.g., version 4.2 >> 4.3).
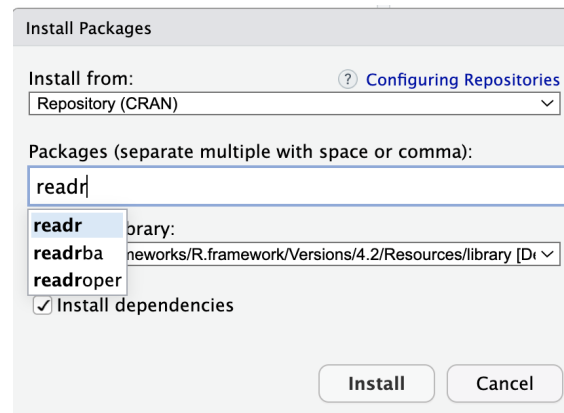
Installing Packages: Dropdown Menu

You can install packages from CRAN using the tool menu in RStudio:

tools > Install Packages

Type in the package name to install.

Installing Packages: Using Code

We use a function called `install.packages()` for CRAN packages.

Here is an example where we "install" the `dplyr` package:

```
install.packages("dplyr")
```

The package name is enclosed in quotation marks.

Loading packages

After installing packages, you will need to "load" them into memory so that you can use them.

This must be done **every time** you start R.

We use a function called `library` to load packages.

Here is an example where we "load" the `dplyr` package:

```
library(dplyr)
```

Quotation marks are optional.

Installing + Loading packages

Install package **VS** Load package

Once per R version	Every R session
Can go in the console	At the top of your code
`install.packages("dplyr")`	`library(dplyr)`

Installing + Loading packages

install.packages("light")   library("light")

Install once

Use many times

Images sourced from https://www.wikihow.com/Change-a-Light-Bulb
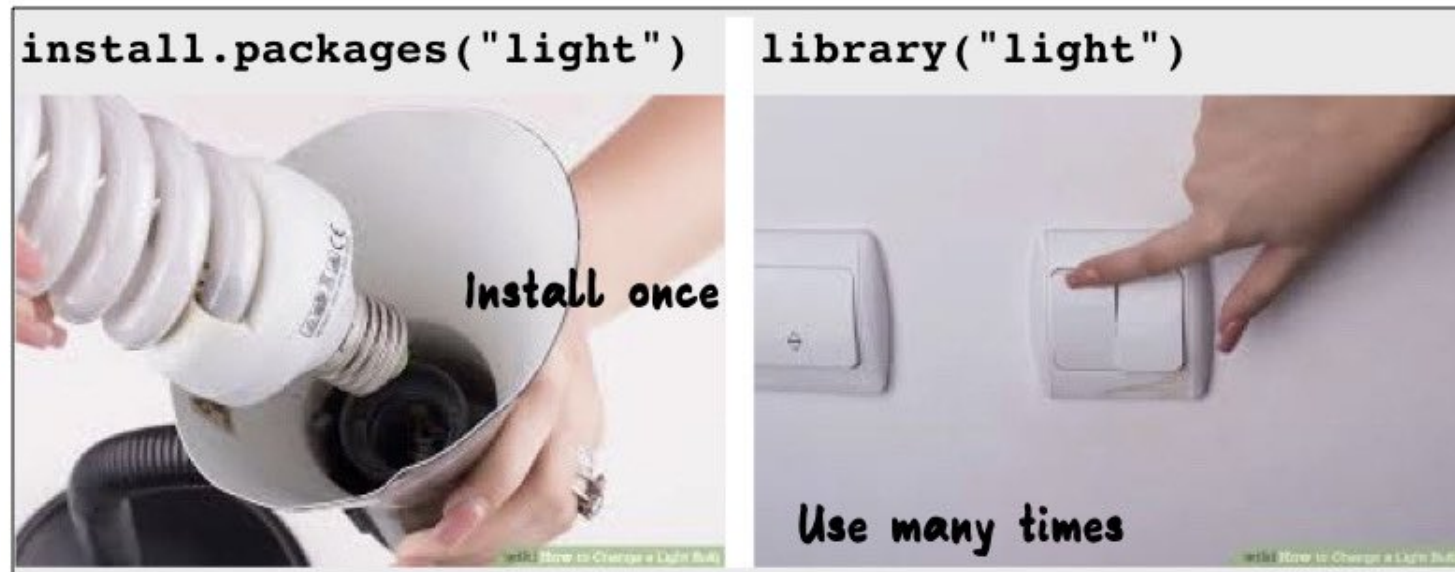
Let's practice!

Installing **remotes** and **dasehr**

Install the `remotes` package.

```r
install.packages("remotes")
```

Then load the package.

```r
library(remotes)
```

Installing **remotes** and **dasehr**

Next, run the following.

It will install our custom package, `dasehr` from GitHub.

```
install_github("fhdsl/dasehr")
```

Where to find help

Useful (+ mostly Free) Resources

Found on our website under the `Resources` tab:
https://daseh.org/resources.html

- videos from previous offerings of the class
- cheatsheets for each class

Help!!!

Error messages can be scary!

- Check out the FAQ/Help page on the website: https://daseh.org/help.html
- Ask questions in Slack! Copy+pasting your error messages is really helpful!

We will also dedicate time today to debug any installation issues

ME RUNNING TO GET HELP

Summary

- R is a powerful data visualization and analysis software language.

- Add-on **packages** like the `tidyverse` can help make R more intuitive.

- **Functions** (like verbs) perform specific tasks in R and are found within packages.

- **Arguments** within functions specify how to perform a function.

- **Objects** (like nouns) are data or variables.

- We will be both installing and loading packages.

- Materials will be updated frequently as we improve it. Please use the **Google Form survey** so you can provide feedback throughout the class!

- Lots of **resources** can be found on the website. *You will have access to the website after the class is over.*

⬚ [Class Website](#)

Website tour!

 [Class Website](#)

Image by [Gerd Altmann](#) from [Pixabay](#)