
Welcome to DATA 151

I'm so glad you're here!



DATA 151: CLASS 3A

INTRODUCTION TO DATA SCIENCE (WITH R)

EXPERIMENTAL DESIGN AND INTRODUCTION TO R



ANNOUNCEMENTS



RELEVANT READING

INTRODUCTION TO DATA SCIENCE



DATA ANALYSIS AND PREDICTION ALGORITHMS WITH R

Rafael A Irizarry

Introduction to Data Science:

- Tuesday:
 - Ch 1: Getting Started with R and R Studio
 - Ch 2: R Basics
- Thursday:
 - Ch 3: Programming basics

HOMWORK REMINDER

Due this week:

- *HW #2: Practice Problems (due on WISE 9/15)*
- *Project Milestone #0: Communication Plan*
 - ***Due on WISE 9/15***
 - *One submission per group*



EXPERIMENTAL DESIGN

RELATIONSHIPS BETWEEN VARIABLES

Many analyses are motivated by a researcher looking for a relationship between two variables.

Definitions:

- **Response/Dependent variable (Y):** the variable one suspects is affected by the explanatory variable(s).
 - Variable that is of interest to study
- **Explanatory/Independent variable (X):** the variable whose effect one wants to study
 - Is thought to explain or influence the response variable

PRINCIPLES OF EXPERIMENTAL DESIGN

Randomized experiments are build on four principles:

- **1) Control**
 - (verb) Control for lurking variables that might affect the response, most simply by comparing two or more treatments
 - (noun) May also be referred to as the “non-treatment”
- **2) Randomization**
 - Use chance to assign experimental units to treatments

PRINCIPLES OF EXPERIMENTAL DESIGN

- **3) Replication**
 - Use enough experimental units in each group to reduce chance of variation in the results
- **4) Blocking**
 - The arranging of experimental units in groups (blocks) that are known to be similar to one another
 - Blocking factors is typically a source of variability but not the primary interest
 - Common Examples: Space and time...

TYPES OF EXPERIMENTAL DESIGNS

1. Completely Randomized Design
2. Randomized Block Design
3. Matched Pairs Design

COMPLETELY RANDOMIZED DESIGN

- Also known as **CRD**
- The simplest experimental design, in terms of analysis and convenience
- Subjects are randomly assigned to treatments
- Typically done by listing treatment levels and randomly assigning random numbers to each

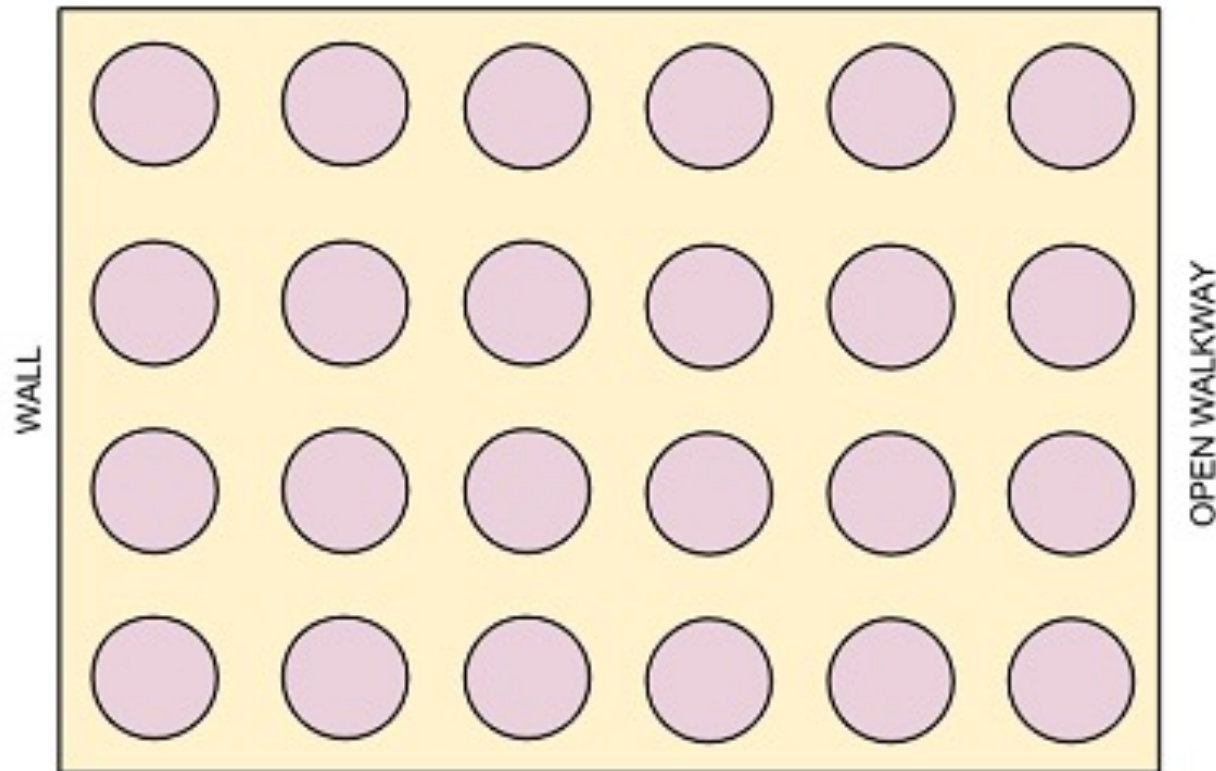
COMPLETELY RANDOMIZED DESIGN

Consider the set up:

- In a greenhouse experiment we want to study a single factors (fertilizer) with 4 levels
- We have enough space for 24 experimental units (a potted plant)
- To maintain balance in the experiment, we will have 6 replications of each treatment

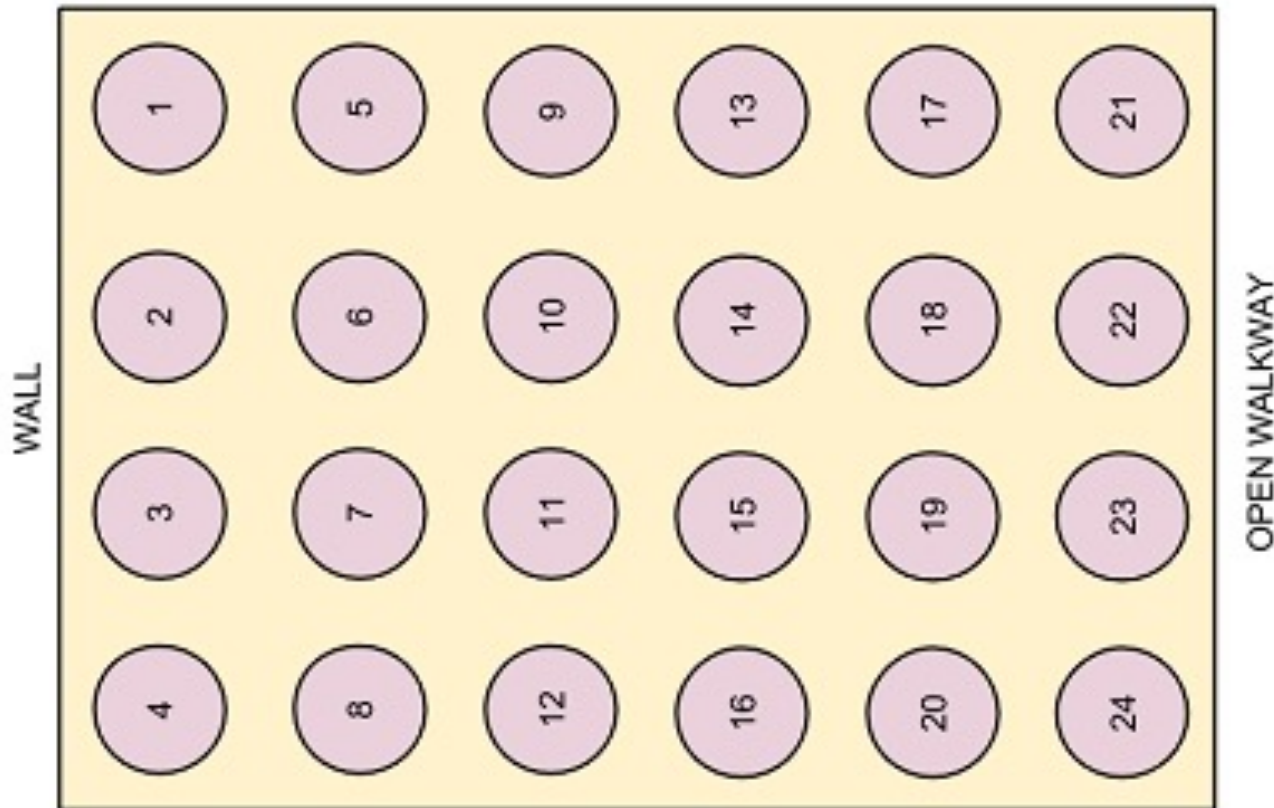
COMPLETELY RANDOMIZED DESIGN

Greenhouse Diagram and bench used for the experiment (viewed from above):



COMPLETELY RANDOMIZED DESIGN

Step 1: Assign it experimental unit a unique id



Source: <https://newonlinecourses.science.psu.edu/stat502/node/175/>

COMPLETELY RANDOMIZED DESIGN

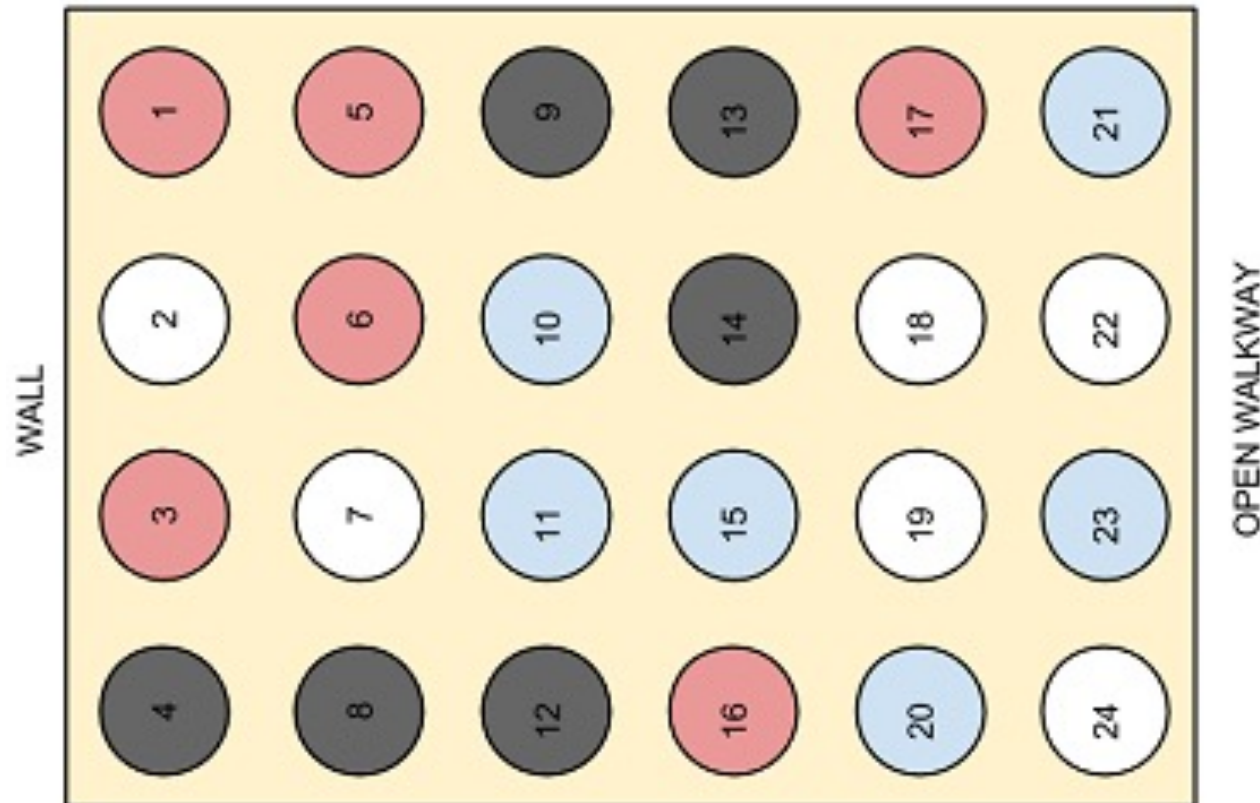
Step 2: Randomly assign each experimental units to treatments

Fertilizer 1 - Blue

Fertilizer 2 - Red

Fertilizer 3 - Black

No Fertilizer -
White (control)



COMPLETELY RANDOMIZED DESIGN

... but what if there are known nutrient gradients across the bench?

A vanilla CRD will not control for this!

RANDOMIZED (COMPLETE) BLOCK DESIGN

- Also known as **RCBD**
- Variation between blocks is accounted for assigning at least one of each treatment to each block
- Effects of blocks not of interest
- Standard design for agricultural experiments

RANDOMIZED (COMPLETE) BLOCK DESIGN

In a **block design**, the random assignment of experimental units to treatments is carried out within each block

What are the steps in performing a blocked experiment?

1. Form groups (blocks)
 - All individuals within each block should be similar in regard to the lurking variable
2. Within each block, randomly assign experimental units to each treatment.

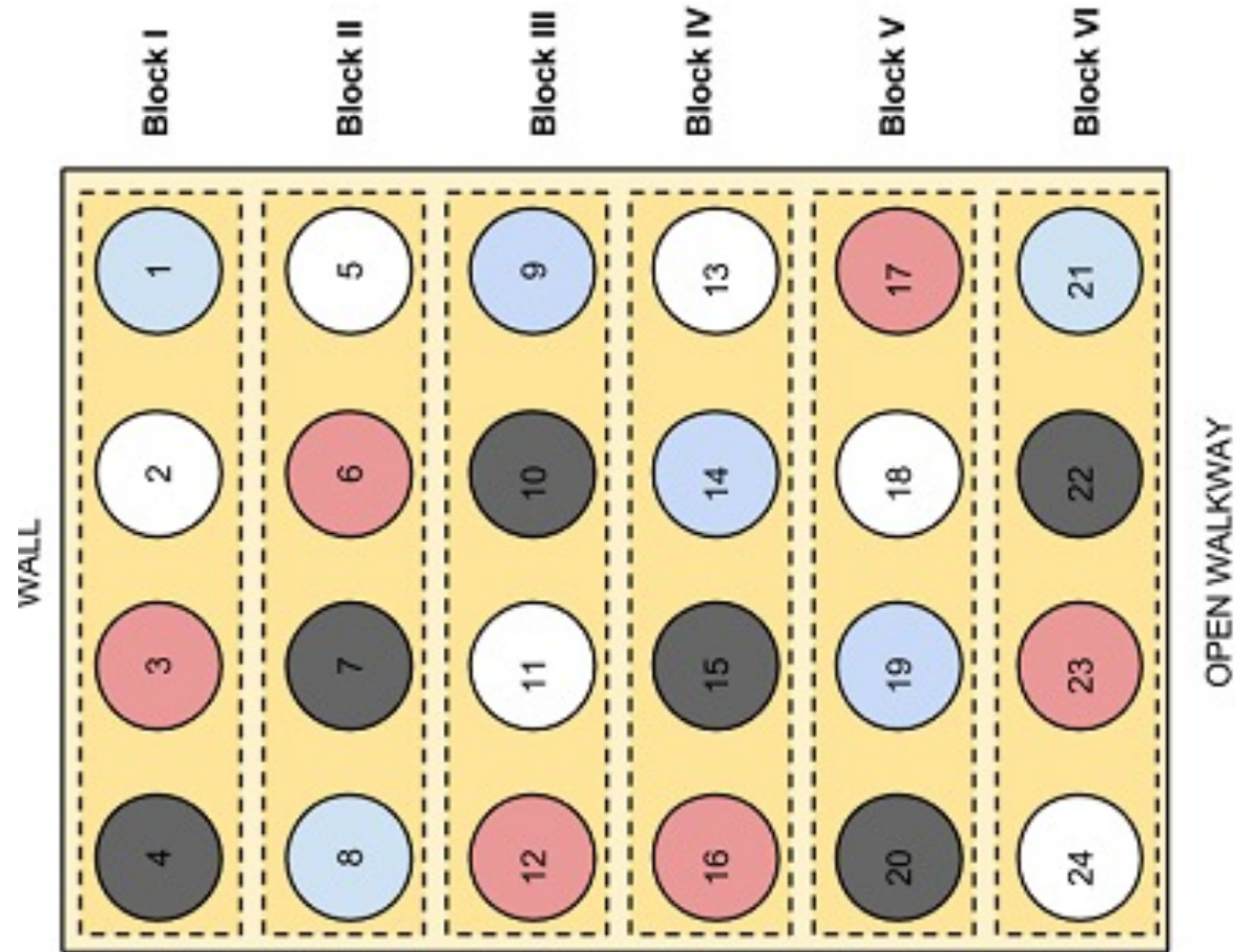
RANDOMIZED (COMPLETE) BLOCK DESIGN

Fertilizer 1 - Blue

Fertilizer 2 - Red

Fertilizer 3 - Black

No Fertilizer -
White (control)



MATCHED PAIRS (AN EXTENSION OF BLOCKING)

- Big Idea: Create blocks by matching pairs of similar experimental units
- Chance is used to determine which unit in each pair gets each treatment

Ex: **Pre-Post (Before After) Studies**

Data from the same individual is related (treat like a block)

1. Assess baseline
2. Assign treatment
3. Find difference after



WORKSHEET EXAMPLES



WORKSHEET EXAMPLES

Example of a blocked design:

An experiment that showed that high doses of omega-3 fats might be a benefit to people with bipolar disorder involved a control group of subjects who received a placebo. Researchers hoped to design a study with two treatment groups, one taking a high dose of omega-3 fatty acids and the other a placebo. Suppose researchers recognized that some of the participants in the study were very active people who walked a lot or got vigorous exercise several times a week, while others tended to be more sedentary. Design a Blocked Experiment, blocking on activity level.

WORKSHEET EXAMPLES

Example: The Blood Lactate Example - A Matched-Pairs (Before and After) design

The effect of exercise on the amount of lactic acid in the blood was examined by researchers. In a particular study, eight men who were attending a week-long training camp were randomly selected to participate in the study. The blood lactate levels (in mmol/L (millimoles per liter of blood)) were measured before and after playing three games of racquetball for each of the 8 men. Researchers wanted to determine if exercise increased blood lactate levels. Explain why this is an example of a matched-pairs design.



INTRODUCTION TO R

WHAT IS R?

Free and open source programming language created by statisticians
as an interactive environment for data analysis

Benefits:

- Relatively quick to learn
- Large community of R users (and online support)
- You can edit and save scripts (rather than point and click)

WHAT IS R STUDIO?

R Studio is an **IDE (Integrated Development Environment)**

R Studio runs on top of R (the programming language and compiler) to provide a more aesthetic and organized experience for programming in R

R STUDIO CLOUD

The screenshot displays the RStudio Cloud web interface. At the top, a browser tab is labeled "RStudio Cloud" and the address bar shows the URL "rstudio.cloud/spaces/279080/content/all?sort=name_asc". The interface includes a left sidebar with a "Spaces" list containing "Your Workspace" and several university-related spaces, with "FA22//DATA151" selected. The main content area is titled "FA22//DATA151" and "Willamette University Cloud". It features a "Projects" tab and a list of project categories: "All Projects", "Your Projects", "Archive", and "Trash". The "All Projects" section shows two projects: "classExamples" and "professorSmalley". Each project entry includes an RStudio icon, the project name, the user "Heather Kitada Smalley", a "Space members" link, and a creation timestamp. The footer contains the RStudio Cloud logo, social media links for LinkedIn, Facebook, Twitter, and GitHub, and the copyright notice "© 2020 RStudio, PBC".

RStudio Cloud

FA22//DATA151
Willamette University Cloud

Projects Members Usage About

Spaces

- Your Workspace
- FA22//DATA151
Willamette University Cloud
- FA22//DATA502
Willamette University Cloud
- HeatherClassPrep
Willamette University Cloud
- HeatherResearch
Willamette University Cloud
- python test
Heather Kitada Smalley
- SP22//MATH239
Willamette University Cloud
- SP22//MATH266
Willamette University Cloud
- + New Space

All Projects (2)

TYPE * ACCESS * SORT A Z

New Project

classExamples

RStudio Project Heather Kitada Smalley Space members Created Sep 9, 2022 10:03 PM

professorSmalley

RStudio Project Heather Kitada Smalley Private Created Sep 9, 2022 10:02 PM

RStudio Cloud

Terms Status

© 2020 RStudio, PBC

MAKING A PROJECT IN OUR CLASS "SPACE"

All Projects (2)

New Project ▾

TYPE



ACCESS



SORT

classExamples



RStudio Project



Heather Kitada Smalley



Space m



New RStudio Project



New Jupyter Project



New Project from Git Repository

MAKING A PROJECT IN OUR CLASS "SPACE"

FA22//DATA151 /

hsmalley

Click to name your project

RAM

HS Heather Kitada Smalley

Deploying Project

MAKING YOUR FIRST R STUDIO SCRIPT

FA22//DATA151 / hsmalley

RAM ⚙️ ⋮ HS Heather Kitada Smalley ^

File Edit Code View Plots Session Build Debug Profile Tools Help

+ ↵ ↻ ↺ ↻ Go to file/function Addins R 4.2.1

Console Terminal x Background Jobs x

R 4.2.1 · /cloud/project/

```
R version 4.2.1 (2022-06-23) -- "Funny-Looking Kid"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (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.

> |
```

Environment History Connections Tutorial

Import Dataset 125 MiB List ↻

R Global Environment

Environment is empty

Files Plots Packages Help Viewer Presentation

+ Folder + Blank File + Upload × Delete + Rename ⚙️ ↻

Cloud > project

	▲ Name	Size	Modified
↑	..		
<input type="checkbox"/>	.Rhistory	0 B	Sep 11, 2022, 11:51
<input type="checkbox"/>	project.Rproj	205 B	Sep 11, 2022, 11:53

MAKING YOUR FIRST R STUDIO SCRIPT

FA22//DATA151 / hsmalley

RAM ⚙️ ⋮ HS Heather Kitada Smalley ^

R 4.2.1

File Edit Code View Plots Session Build Debug Profile Tools Help

+ Go to file/function Addins

R Script ⌘⇧N Round Jobs x

Quarto Document Create a new R script

Quarto Presentation...

R Notebook

R Markdown...

Shiny Web App...

Plumber API...

Text File

C++ File

Python Script

SQL Script

Stan File

D3 Script

R Sweave

R HTML

R Documentation...

Environment History Connections Tutorial

Import Dataset 125 MiB

R Global Environment

Environment is empty

Files Plots Packages Help Viewer Presentation

Folder Blank File Upload Delete Rename

Cloud > project

	▲ Name	Size	Modified
	..		
<input type="checkbox"/>	.Rhistory	0 B	Sep 11, 2022, 11:51
<input type="checkbox"/>	project.Rproj	205 B	Sep 11, 2022, 11:53

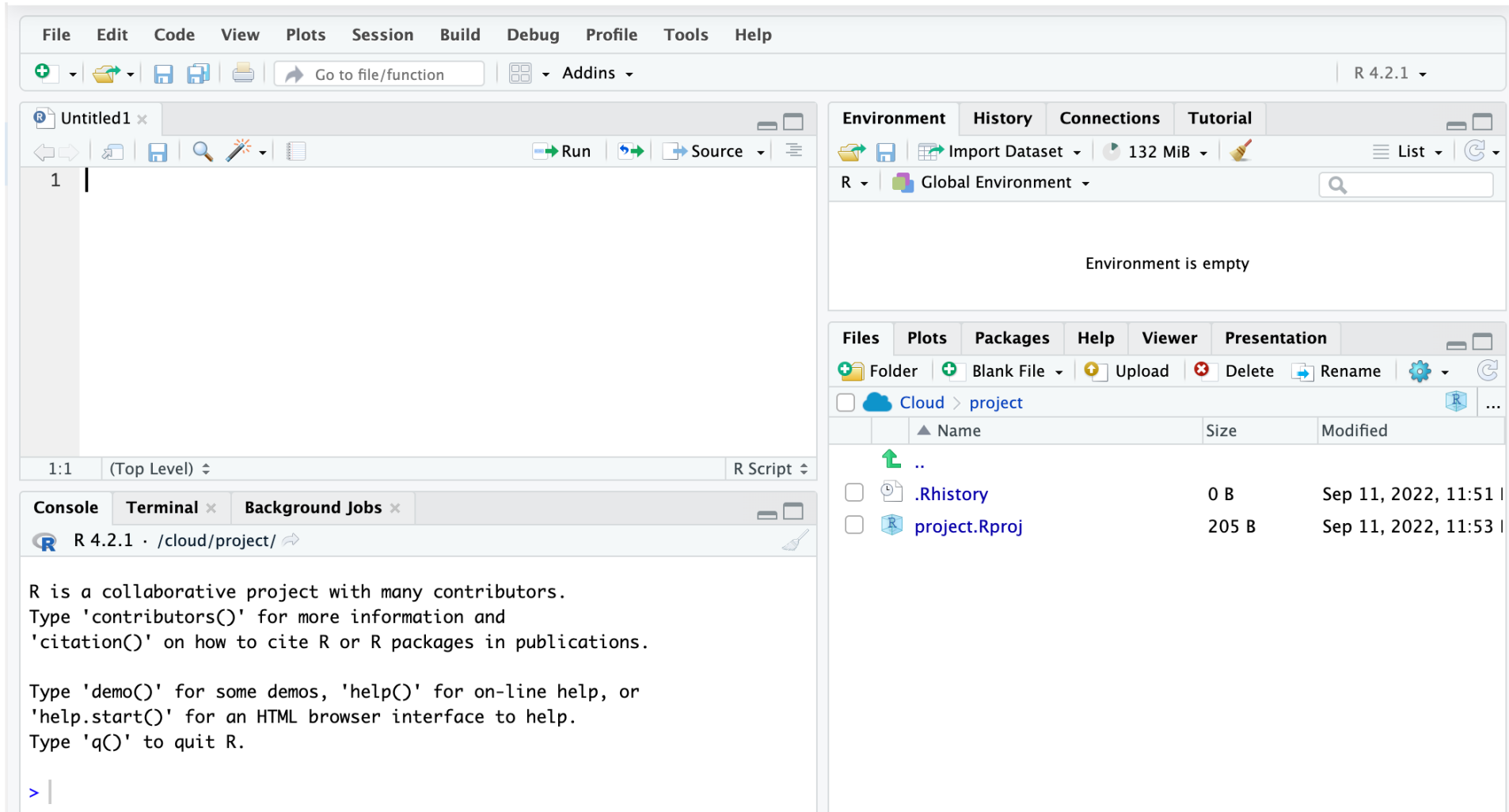
```
23) -- "Funny-Looking Kid"
Foundation for Statistical Computing
gnu (64-bit)

comes with ABSOLUTELY NO WARRANTY.
tribute it under certain conditions.
nce()' for distribution details.

ect with many contributors.
more information and
ce R or R packages in publications.

mos, 'help()' for on-line help, or
browser interface to help.
```

MAKING YOUR FIRST R STUDIO SCRIPT



SAVING AN R SCRIPT

The screenshot displays the RStudio IDE interface. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. The toolbar contains icons for creating a new file, opening a file, saving, and running code. The main editor window shows a file named 'Untitled1' with a single line of code '1'. A tooltip 'Save current document (⌘S)' is visible over the editor. The bottom-left pane shows the console output for R 4.2.1, displaying instructions on how to use R, including commands like 'contributors()', 'citation()', 'demo()', 'help()', 'help.start()', and 'q()'. The bottom-right pane shows the Environment, History, Connections, and Tutorial tabs. The Environment tab is active, showing 'Global Environment' with a memory usage of 132 MiB. The Files pane shows a directory structure with a 'project' folder containing files like '.Rhistory' and 'project.Rproj'.

File Edit Code View Plots Session Build Debug Profile Tools Help

+ - - - - - Go to file/function - - - - - Addins - R 4.2.1

Untitled1 x

1

Save current document (⌘S)

1:1 (Top Level) R Script

Console Terminal Background Jobs

R 4.2.1 · /cloud/project/

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.

> |

Environment History Connections Tutorial

Import Dataset 132 MiB

R Global Environment

Environment is empty

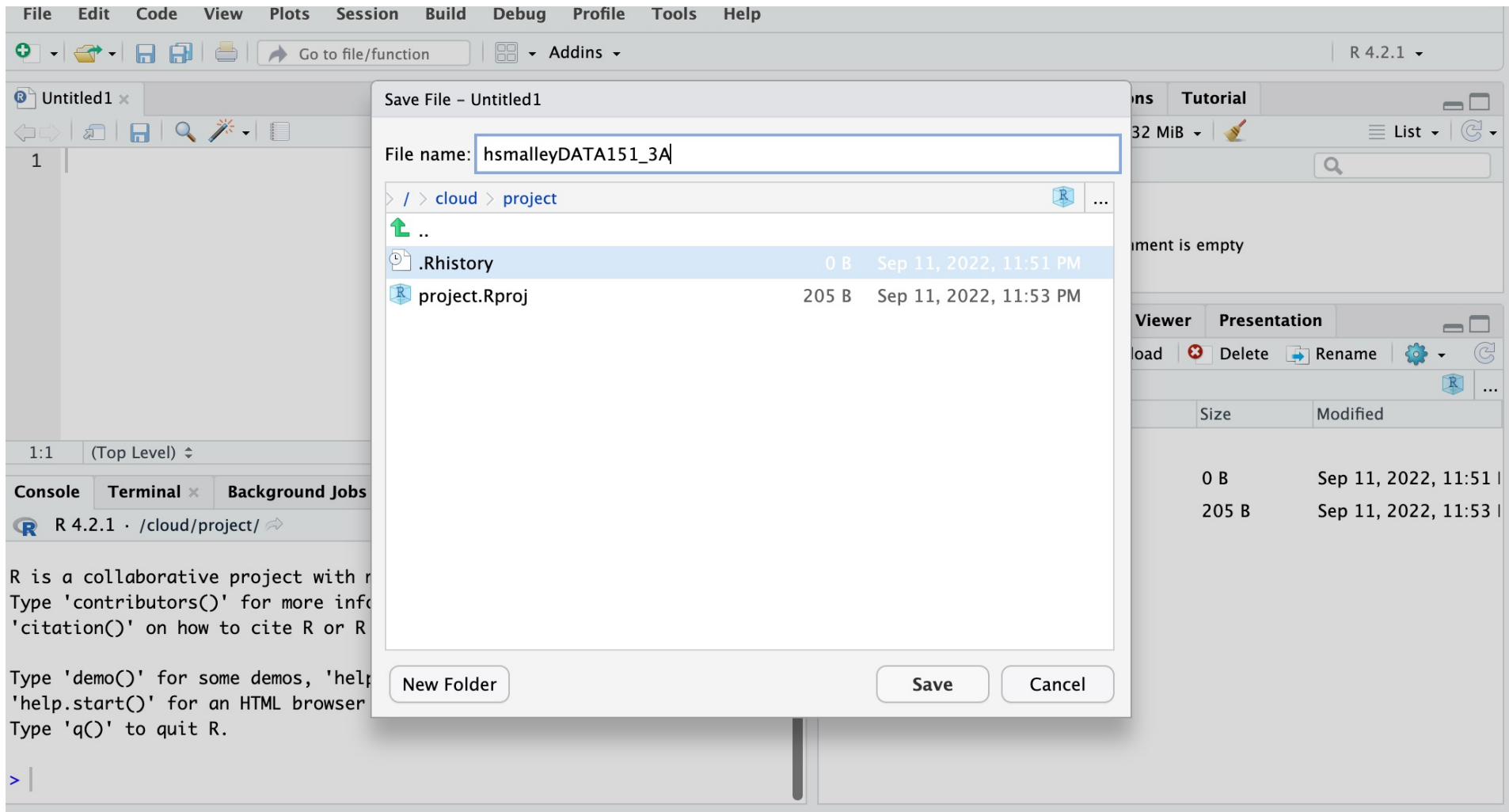
Files Plots Packages Help Viewer Presentation

Folder Blank File Upload Delete Rename

Cloud > project

	Name	Size	Modified
..			
.Rhistory		0 B	Sep 11, 2022, 11:51
project.Rproj		205 B	Sep 11, 2022, 11:53

NAMING AN R SCRIPT

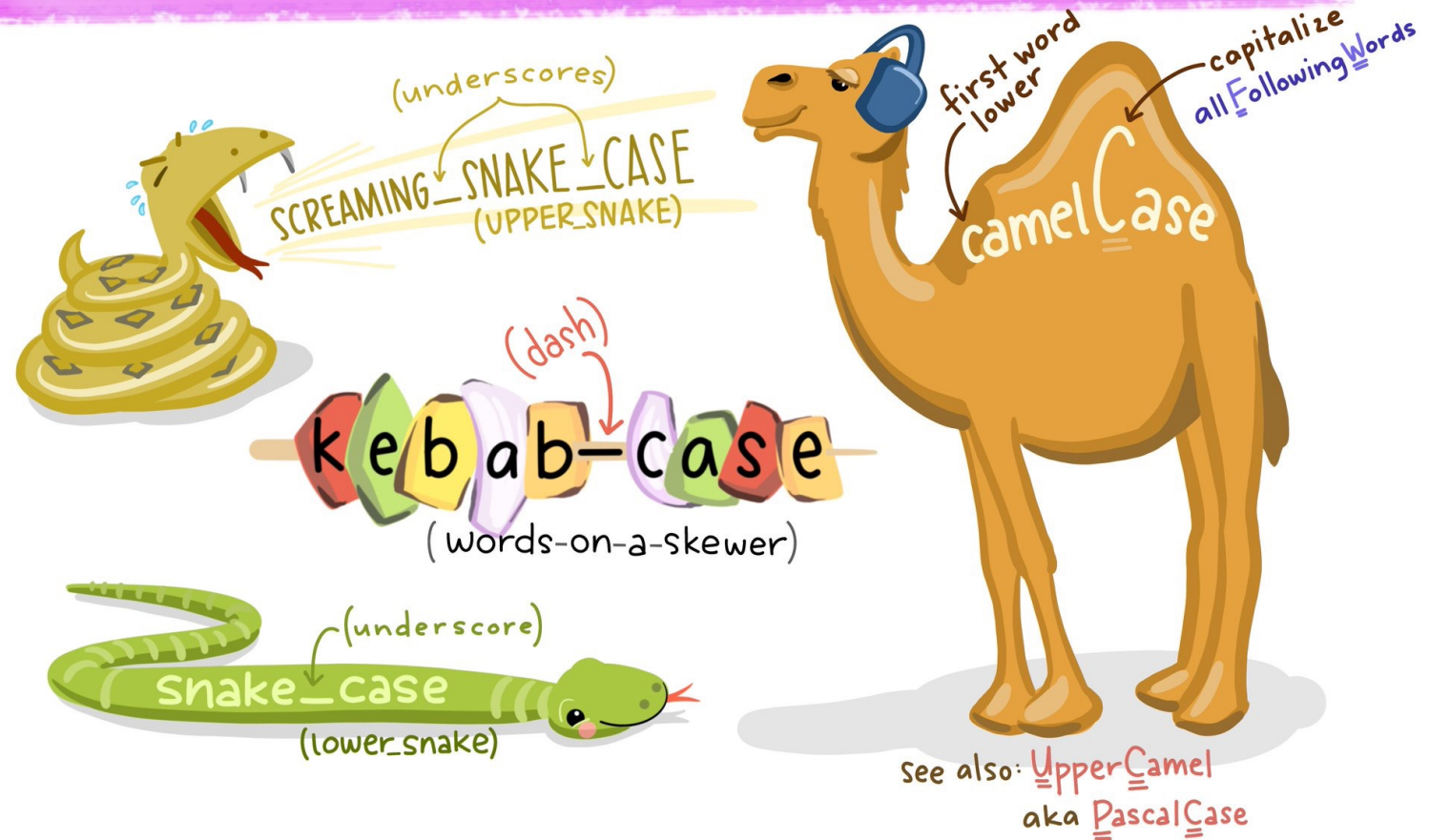


NAMING CONVENTIONS

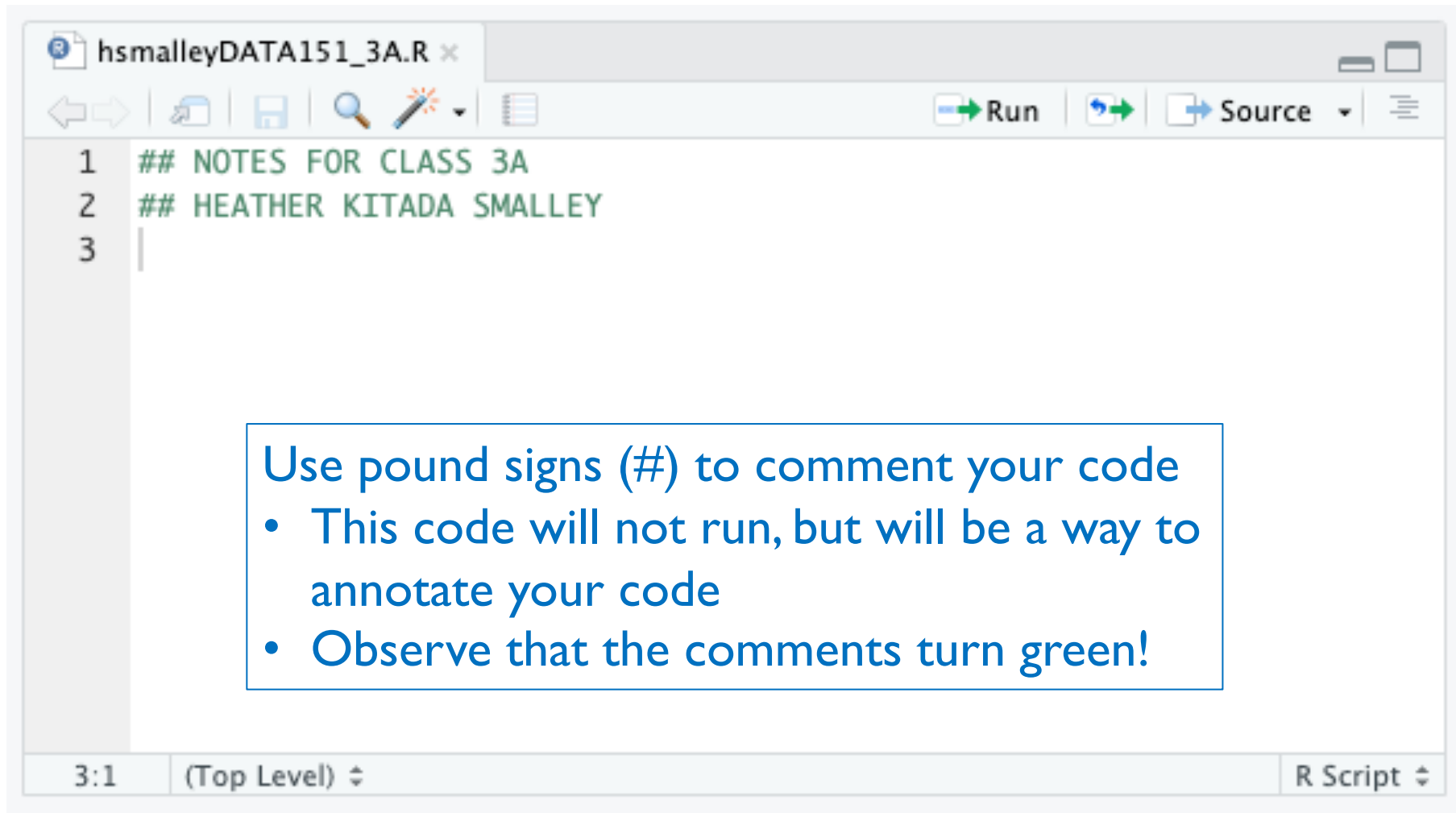


Allison Horst
@allison_horst

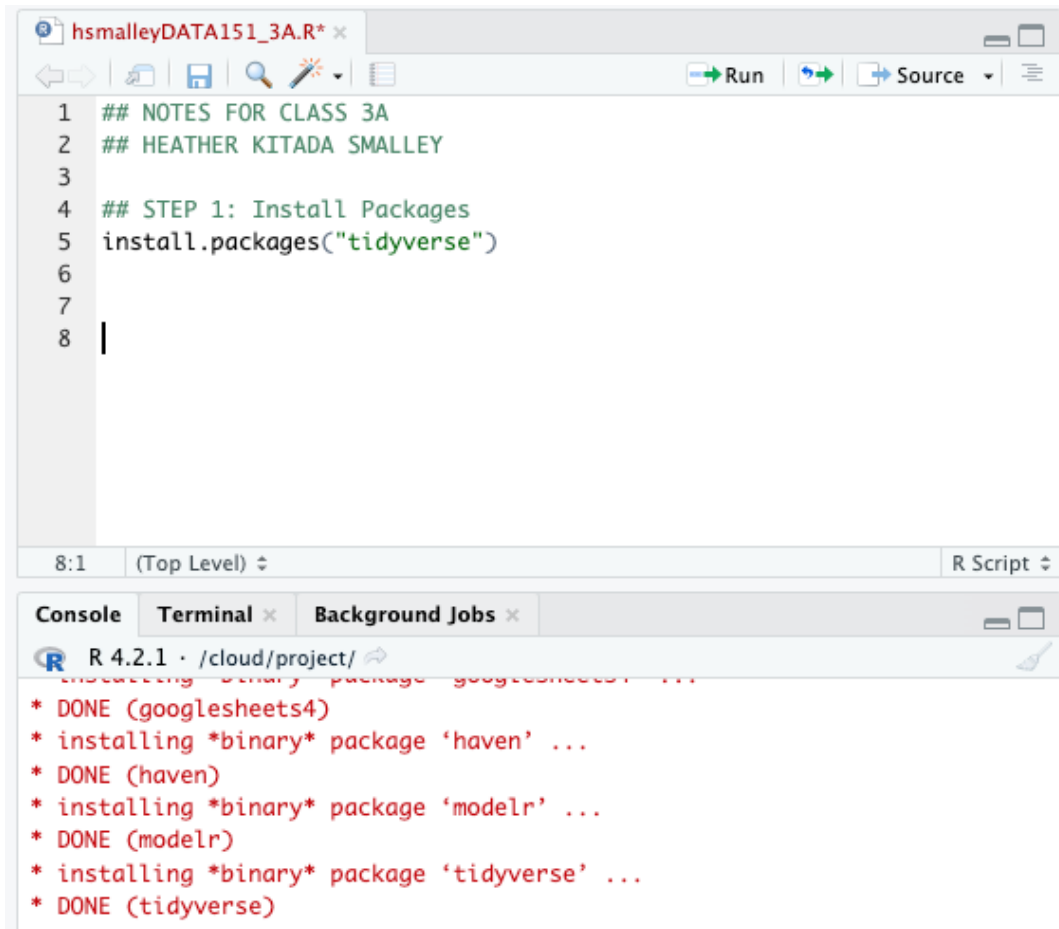
in that case...



COMMENTING YOUR CODE



INSTALLING PACKAGES



The screenshot shows the RStudio interface. The top pane displays an R script file named 'hsmalleyDATA151_3A.R'. The script contains the following code:

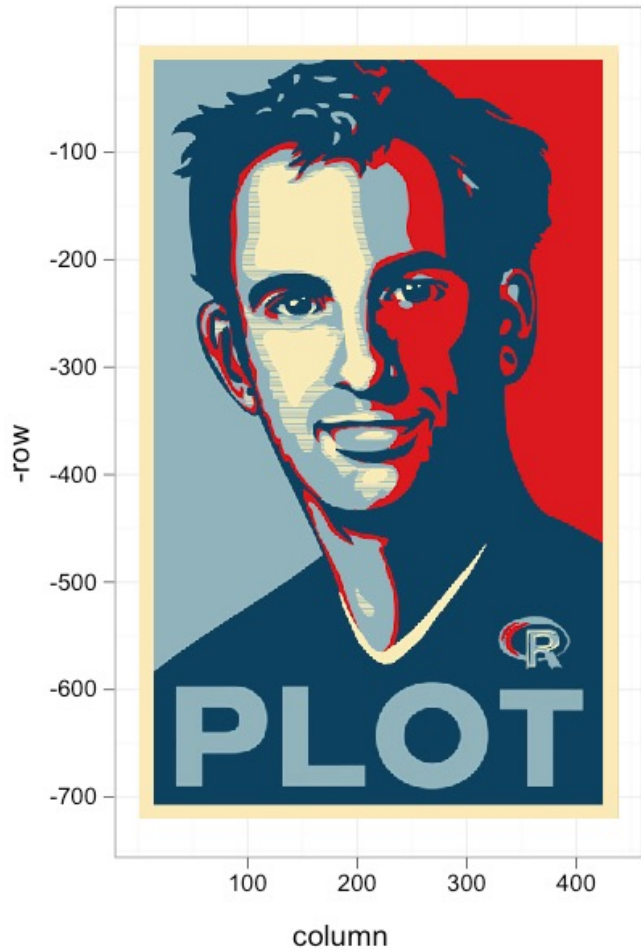
```
1 ## NOTES FOR CLASS 3A
2 ## HEATHER KITADA SMALLEY
3
4 ## STEP 1: Install Packages
5 install.packages("tidyverse")
6
7
8 |
```

The bottom pane shows the console output, indicating the installation of several packages:

```
R 4.2.1 · /cloud/project/
Installing binary package 'googlesheets4' ...
* DONE (googlesheets4)
* installing *binary* package 'haven' ...
* DONE (haven)
* installing *binary* package 'modelr' ...
* DONE (modelr)
* installing *binary* package 'tidyverse' ...
* DONE (tidyverse)
```

- You should only need to install a package once
- Observe the use of quotes in the `install.packages()` command

HADLEY'S TIDYVERSE



CALLING THE LIBRARY

```
> ## STEP 2: Calling the library
> library(tidyverse)
— Attaching packages — tidyverse 1.3.2 —
✓ ggplot2 3.3.6      ✓ purrr 0.3.4
✓ tibble 3.1.8       ✓ dplyr 1.0.10
✓ tidyr 1.2.1        ✓ stringr 1.4.1
✓ readr 2.1.2        ✓ forcats 0.5.2
— Conflicts — tidyverse_conflicts() —
✗ dplyr::filter() masks stats::filter()
✗ dplyr::lag() masks stats::lag()
> |
```

Note: Conflicts can occur when the library contains functions with the same name as another function

CALLING R BUILT IN DATASETS TO THE ENVIRONMENT

```
1 ## NOTES FOR CLASS 3A
2 ## HEATHER KITADA SMALLEY
3
4 ## STEP 1: Install Packages
5 #install.packages("tidyverse")
6
7 ## STEP 2: Calling the library
8 library(tidyverse)
9
10 ## STEP 3: Data sets in the Environment
11 data()
12
13
```

Environment

Global Environment

Environment is empty

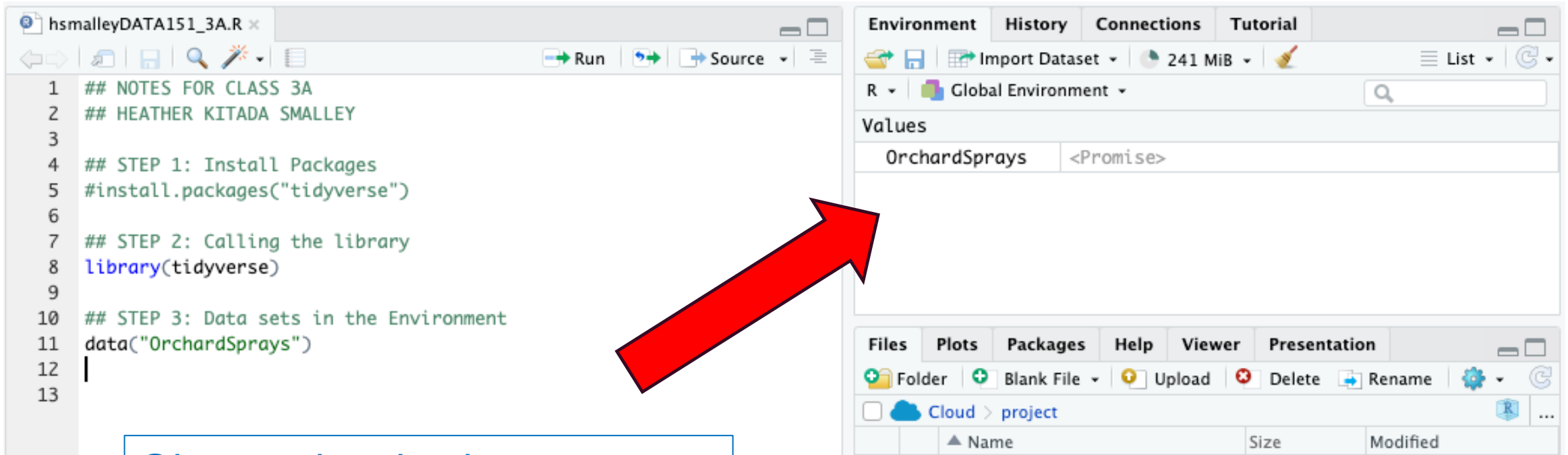
Files Plots Packages Help Viewer Presentation

Upload Delete Rename

	Size	Modified
OrchardSprays	0 B	Sep 11, 2022, 11:51
occupationalStatus	161 B	Sep 12, 2022, 1:06 P

OrchardSprays
Potency of Orchard Sprays
An experiment was conducted to assess the potency of various constituents of orchard sprays in repelling honeybees, using a Latin square design.
Press F1 for additional help

CALLING R BUILT IN DATASETS TO THE ENVIRONMENT



The screenshot displays the RStudio IDE. The script editor on the left contains the following code:

```
1 ## NOTES FOR CLASS 3A
2 ## HEATHER KITADA SMALLEY
3
4 ## STEP 1: Install Packages
5 #install.packages("tidyverse")
6
7 ## STEP 2: Calling the library
8 library(tidyverse)
9
10 ## STEP 3: Data sets in the Environment
11 data("OrchardSprays")
12 |
13
```

The Environment pane on the right shows the Global Environment with the following values:

Values
OrchardSprays

A red arrow points from the `data("OrchardSprays")` line in the script to the `OrchardSprays` entry in the Environment pane.

Observe that the data set is now available in the R environment

LEARNING ABOUT THE DATA

The screenshot displays the RStudio interface with three main panels:

- Script Editor:** Contains an R script file named `hsmalleyDATA151_3A.R` with the following content:

```
1 ## NOTES FOR CLASS 3A
2 ## HEATHER KITADA SMALLEY
3
4 ## STEP 1: Install Packages
5 #install.packages("tidyverse")
6
7 ## STEP 2: Calling the library
8 library(tidyverse)
9
10 ## STEP 3: Data sets in the Environment
11 data("OrchardSprays")
12
13 ## STEP 4: Learning about the data
14 help("OrchardSprays")
15 ?OrchardSprays
16
```
- Environment Panel:** Shows the `Global Environment` with a search bar and a table of values. The table has one entry: `OrchardSprays` with a value of `<Promise>`.
- Console Panel:** Shows the output of the R script execution. It includes the message `Attaching packages: tidyverse 1.3.2`, a list of installed packages (e.g., `ggplot2 3.3.6`, `dplyr 1.0.10`), and conflict warnings for `dplyr::filter()` and `dplyr::lag()` masking functions from the `stats` package. The final output shows the execution of `data("OrchardSprays")` and `help("OrchardSprays")`.
- Help Panel:** Displays the R documentation for `OrchardSprays`. It includes the title `Potency of Orchard Sprays`, a description of the experiment, usage instructions, and format details. The format section states: "A data frame with 64 observations on 4 variables." and lists the variables: `[,1] rowpos` (numeric Row of the design), `[,2] colpos` (numeric Column of the design), and `[,3] treatment factor` (Treatment level).

LEARNING ABOUT THE DATA

Details

Individual cells of dry comb were filled with measured amounts of lime sulphur emulsion in sucrose solution. Seven different concentrations of lime sulphur ranging from a concentration of 1/100 to 1/1,562,500 in successive factors of 1/5 were used as well as a solution containing no lime sulphur.

The responses for the different solutions were obtained by releasing 100 bees into the chamber for two hours, and then measuring the decrease in volume of the solutions in the various cells.

An 8×8 Latin square design was used and the treatments were coded as follows:

A highest level of lime sulphur

B next highest level of lime sulphur

.

.

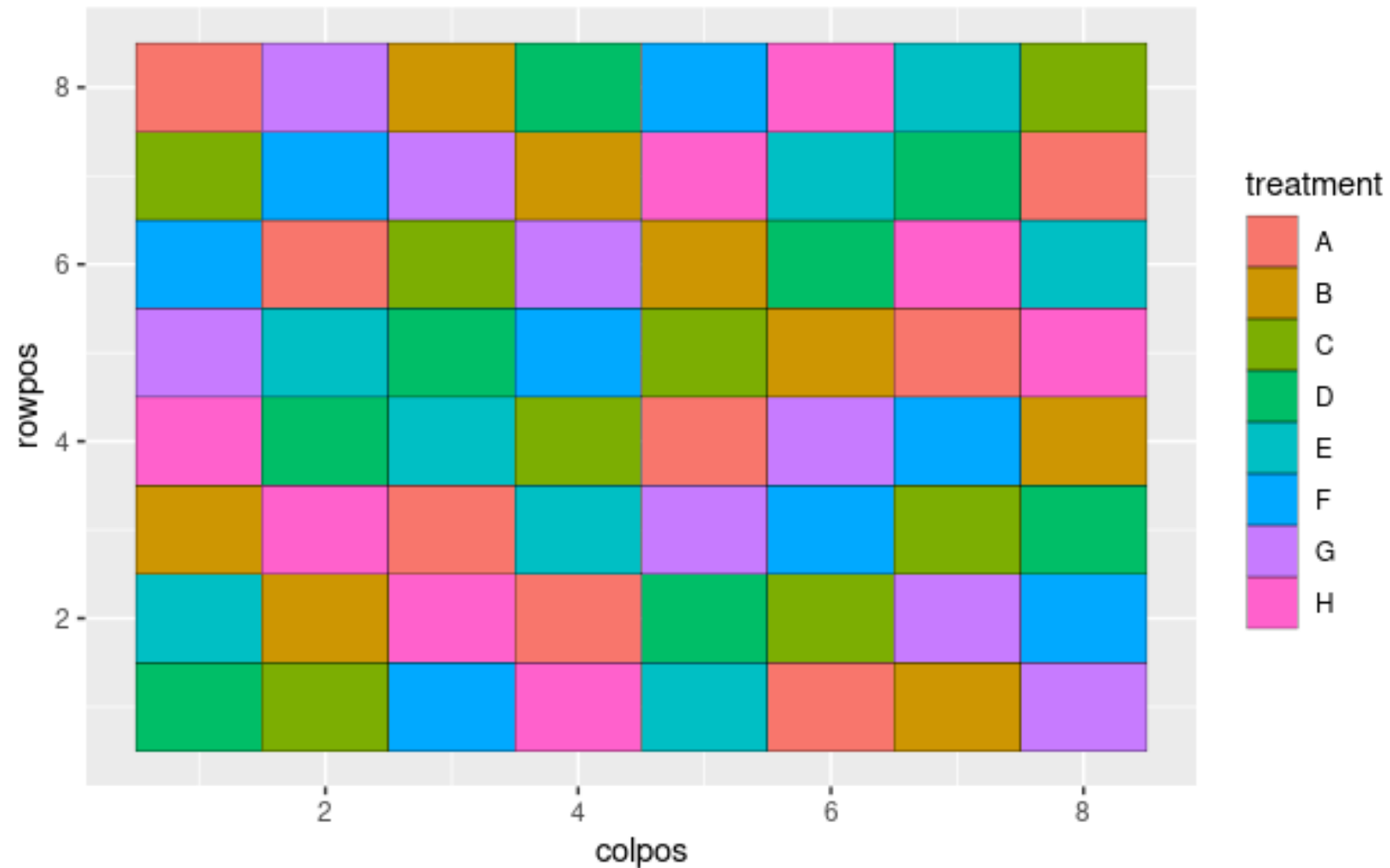
.

G lowest level of lime sulphur

H no lime sulphur



WHAT DOES THE EXPERIMENT LOOK LIKE?



WORKSHEET TIME!

Part I: Experimental Design:

1) What are the response and explanatory variables in this study?

2) Are the four principles of a randomized experiment present? Verify each and explain.

- 1.
- 2.
- 3.
- 4.

If the four principles are met, do you feel comfortable making a cause and effect conclusion?