

# Session 1: Introduction to R

# Overview

Introductions and introduction to R

Basic operations and data types

Packages

R Markdown

Data Frames

# Introductions

## My background:

- Visiting associate professor at Yale
- Associate professor of Statistics, Hampshire College
- Research Affiliate, Center for Brains, Minds and Machines at MIT

## How I use R:

- Teaching: Statistics and Data Science courses using R
- Research: Develop/use machine learning to analyze neural data
  - [NeuroDecodeR package](#)

# Introductions

Let's do some quick introductions

- Your name
- Your academic department affiliation
- How are you interested in using R in your teaching/research?
- What you are most interested in learning in this workshop?
- Anything else you would like to share

# Workshop plan: today



**Session 1:** Basic programming in R

**Session 2:** Statistics and plots, for loops, writing functions

**Session 3:** Basic statistical inference and simple linear regression

**Session 4:** Multiple regression, logistic regression, ANOVA

# Workshop plan: tomorrow

**Session 1:** Data wrangling with dplyr

**Session 2:** Data visualization with ggplot

**Session 3:** Joining and mapping data

**Session 4:** String manipulation and/or interactive graphics

Even if you don't like today, come back tomorrow b/c that's when we're going to be doing the really fun stuff!



# Why use R?

**Why ?**

**R**

1. Free Installation
2. Hottest Trend
3. Independent Platform
4. Latest cutting edge technology
5. Has Large community of users

**Data Flair**

# Data analysis software vs. programming languages

Advantages of using data analysis software (STATA, SAS, SPSS, Minitab, etc.):

- Can be easier/quicker to do the basics: you don't need to know how to program!

Advantages of using data analysis programming languages

- Reproducibility! Can see exactly which steps were taken in an analysis.
  - Can cut down on mistakes
- Ability to extend beyond functions built into software
  - Particularly useful for computational methods that work in more situations





## AS SEEN BY USERS OF ...

STATA



R



sas



python



SPSS



STATA

R



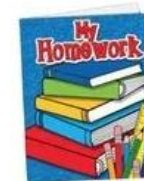
sas



python



SPSS



# Data analysis programming languages



## MATLAB

- Disadvantage: Expensive
  - Makes it harder to distribute results because others might not own the software



## Julia

- Disadvantage: Newer so less of a community, less functionality implemented



## Python

- Advantages: general purpose programming language so useful if one already knows Python or if one wants to integrate into larger software ecosystem
- Disadvantage: Not quite as good as R for data analysis



## R

- Advantages: Rich ecosystem designed specifically for reproducible data analysis
  - Free RStudio IDE, reproducibility, powerful packages, etc.
- Disadvantages: Not good for larger software development

# What are the strengths of R?

Statistical analyses: basic and advanced

Reproducibility: R Markdown

Tidyverse: Powerful packages for graphics, data manipulations, etc.

Interactive graphics: plotly and Shiny

Demos!





# R and R Studio

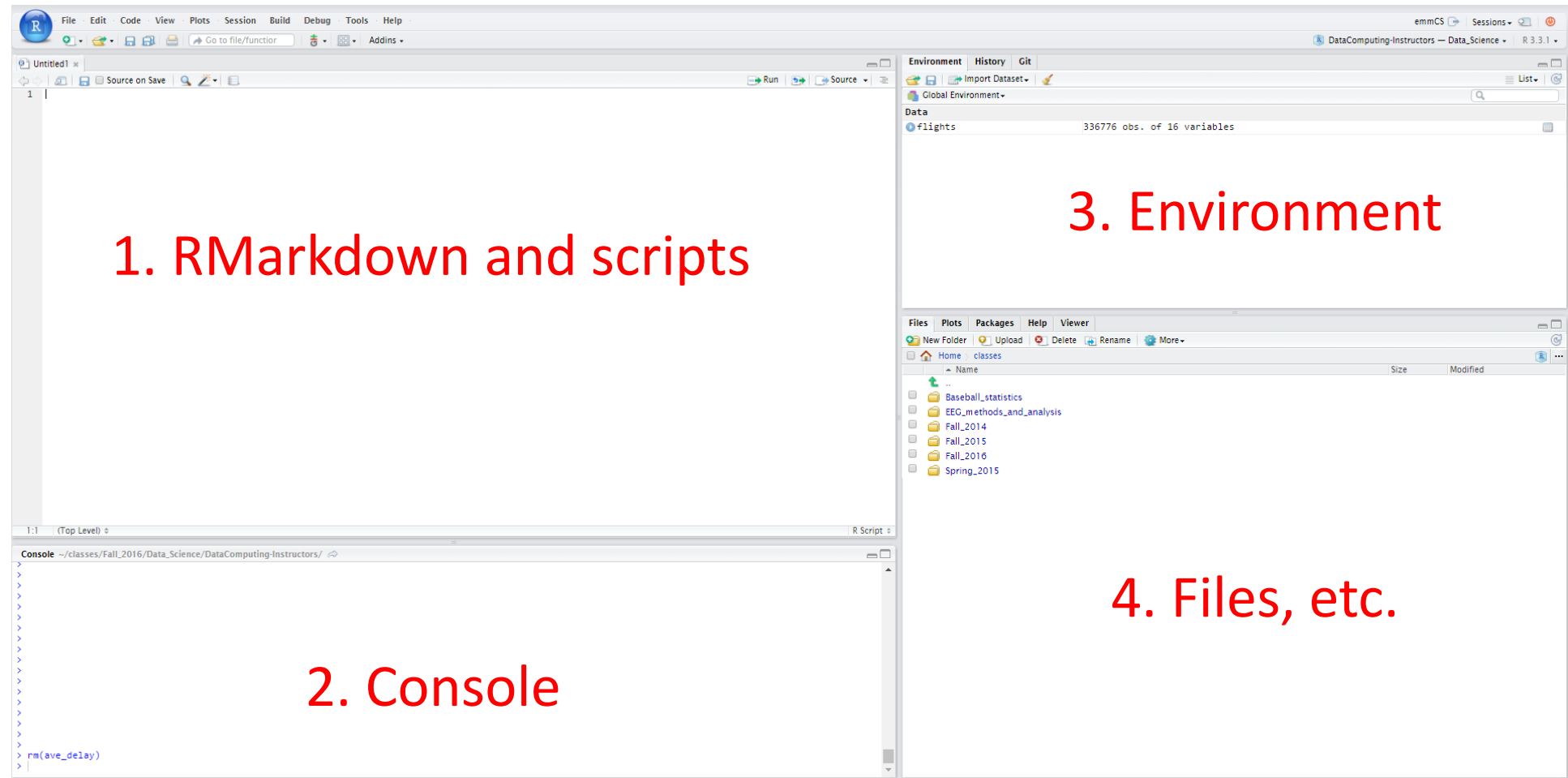
**R: Engine**



**RStudio: Dashboard**

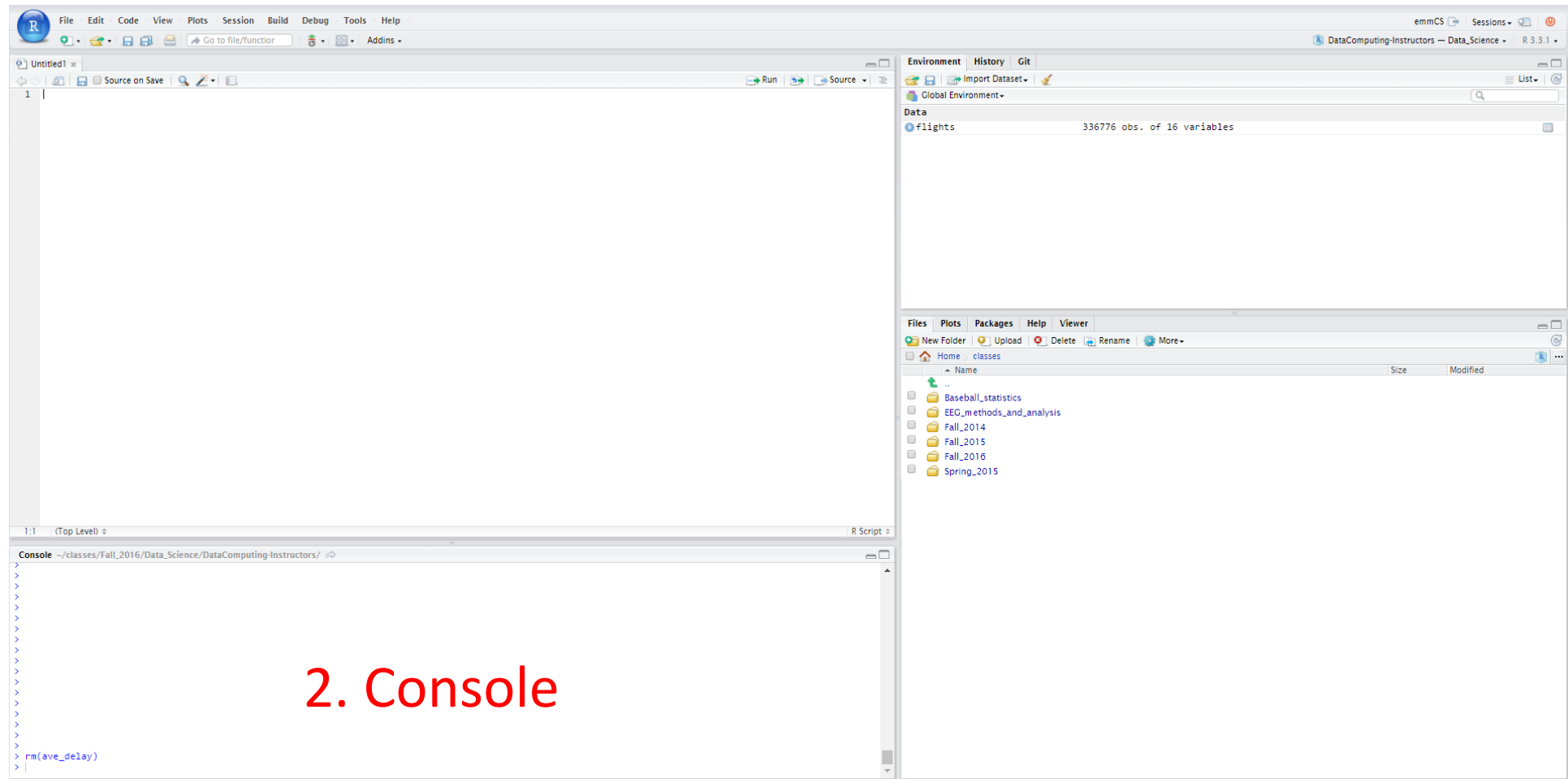


# RStudio layout





# RStudio layout

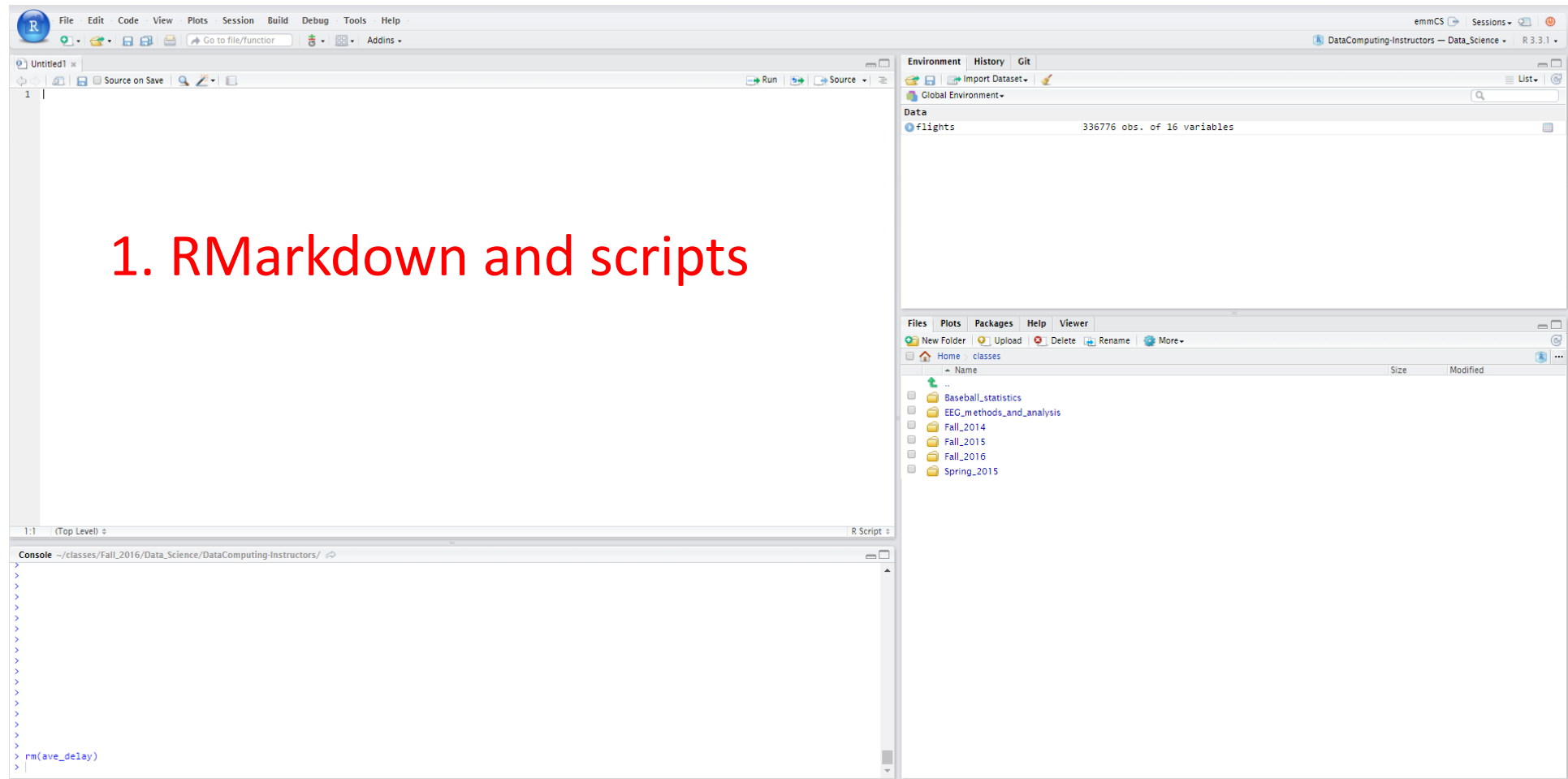


## R as a calculator

> 2 + 2

> 7 \* 5

# RStudio layout



## Create a new script

File -> New File -> R Script

Save the script with a reasonable name, e.g., my\_notes.R



# R Basics

Arithmetic:

```
> 2 + 2
```

```
> 7 * 5
```

Assignment of values to ***objects***:

```
> a <- 4
```

```
> b <- 7
```

```
> z <- a + b
```

```
> z
```

```
[1] 11
```

Number journey...

# Number journey

```
> a <- 7
```

```
> b <- 52
```

```
> d <- a * b
```

```
> d
```

```
[1] 364
```

# Character strings and Booleans

```
> a <- 7
```

```
> s <- "s is a terrible name for an object"
```

```
> b <- TRUE
```

```
> class(a)
```

```
[1] numeric
```

```
> class(s)
```

```
[1] character
```

# Functions

Functions use parenthesis: functionName(x)

```
> sqrt(49)
```

```
> tolower("DATA is AWESOME!")
```

To get help

```
> ? sqrt
```

One can add comments to your code

```
> sqrt(49)  # this takes the square root of 49
```

# Vectors

Vectors are ordered sequences of numbers or letters

The `c()` function is used to create vectors

```
> v <- c(5, 232, 5, 543)
```

```
> s <- c("statistics", "data", "science", "fun")
```

One can access elements of a vector using square brackets `[]`

```
> s[4]      # what will the answer be?
```

We can get multiple elements from a vector too

```
> s[c(1, 2)]
```

# Vectors continued

One can assign a sequence of numbers to a vector

```
> z <- 2:10
```

```
> z[3]
```

One can test which elements are greater than a value

```
> z > 3
```

Can add names to vector elements

```
> names(v) <- c("first", "second", "third", "fourth")
```

# Vectors continued

One can also apply functions to vectors

```
> z <- 2:10
```

```
> sqrt(z)
```

```
> mean(z)
```

# Questions?





# R packages

Packages add additional functionality to R

We will use many additional packages in this class

- gplyr, ggplot2, tidyr, etc.



# R workshop package

I have written a package to allow you to more easily download material for this workshop

```
> install.packages("devtools")  
> devtools::install_github("emeyers/rworkshop")
```

Once you have installed this package, please run the following command to download an R Markdown document that has code for this morning session

- > `rworkshop::download_class_code(1)`

# R Markdown

R Markdown (.Rmd files) allow you to embed written descriptions, R code and the output of that code into a nice looking document

Creates a way to do reproducible research!



# R Markdown

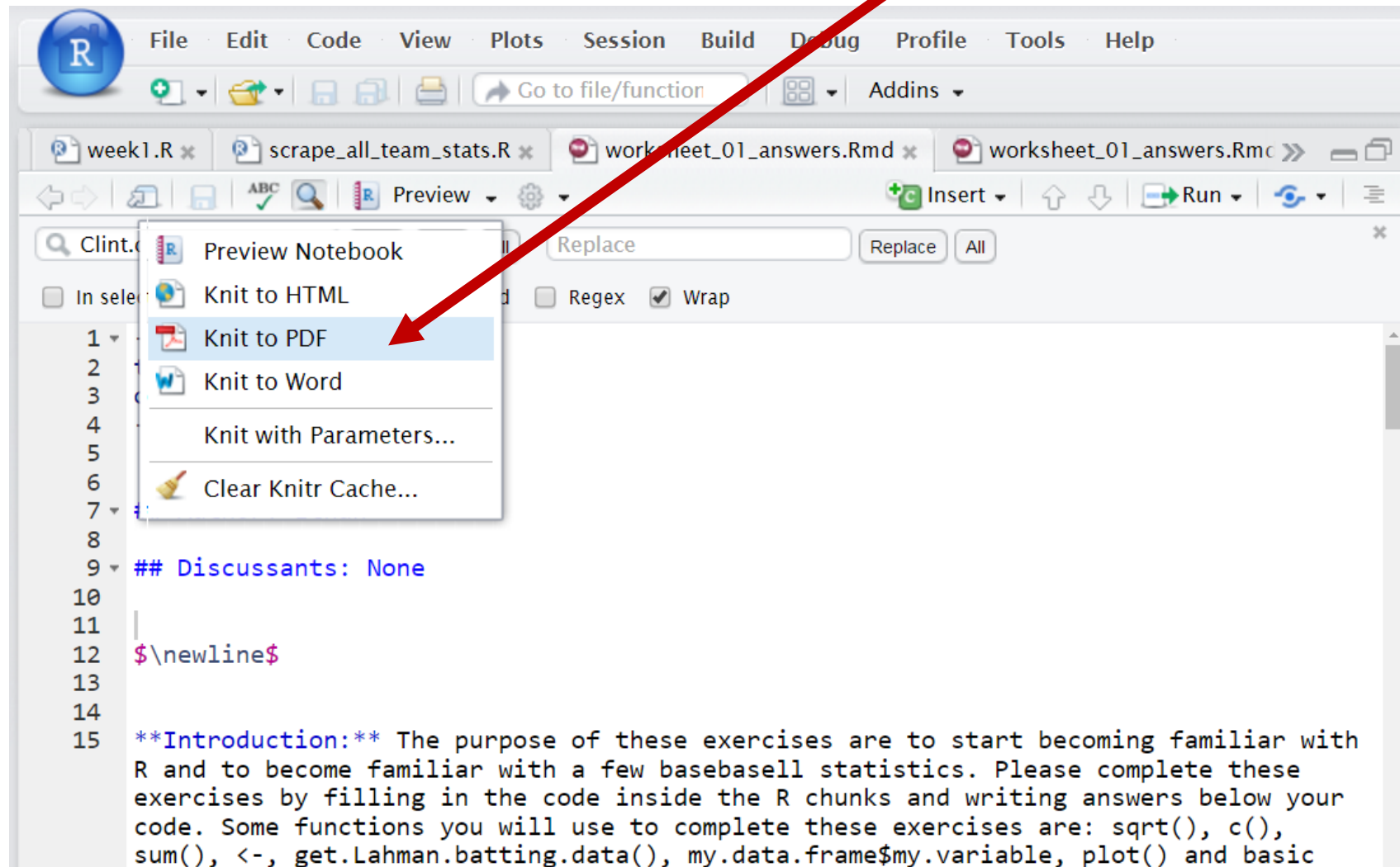
Everything in R chunks is executed as code:

```
```${r}  
  # this is a comment  
  # the following code will be executed  
  2 + 3  
```
```

Everything outside R chunks appears as text

# Knitting to a pdf

Turn in a pdf or html document  
with your solutions to Canvas



# R Markdown

Note: When you knit, RMarkdown files **do not have access to variables in the global environment**, but instead have their own environment.

Why is this a good thing???

# Formatting in R Markdown

We can add formatting to text outside the code chunks

Examples:

`## Level 2 header`

`**bold**`

``

# LaTeX in R Markdown

We can also add LaTeX symbols to documents using  $\text{\symbol{}} syntax$

For example, try these:

$\theta$

$\hat{p}$

$\hat{\theta}$

Knit early and knit often to avoid errors!!!



# LaTeX in R Markdown

I have added a link on Canvas in the resources section to help [find LaTeX symbols](#)

How else could you get help to learn more about LaTeX symbols?

# Avoid hard to debug code!

Only change a few lines at a time and then knit your document to make sure everything is working!

If your document isn't knitting:


- **For code chunks:** use the `# symbol` to comment out code until you can find the line of code that is giving the error message
- **Outside of code chunk:** cut out part of the document until it knits and then paste it back

# Questions?



# Data frames

Data frames contain structured data

|  | age | body_type      | diet              | drinks   | drugs     | education                         |
|---|-----|----------------|-------------------|----------|-----------|-----------------------------------|
| 1   | 22  | a little extra | strictly anything | socially | never     | working on college/university     |
| 2   | 35  | average        | mostly other      | often    | sometimes | working on space camp             |
| 3   | 38  | thin           | anything          | socially | NA        | graduated from masters program    |
| 4   | 23  | thin           | vegetarian        | socially | NA        | working on college/university     |
| 5   | 29  | athletic       | NA                | socially | never     | graduated from college/university |
| 6   | 29  | average        | mostly anything   | socially | NA        | graduated from college/university |


# OK Cupid data




49,638 online now

[View my profile](#)  
[My photos](#)  
[Settings](#)


You might like...



**batsignalgalore**  
Chicago



**ursunshine2b**  
Rolling Meadows



**i\_am\_princess86**  
Chicago




Roll the dice!  
Random match

[See more matches](#)

**Favorites**  
You haven't saved anyone

**Profile Completion**  
65%  
Contact 5 new people to get to 70%

[Messages](#)[Matches](#)[Connections](#)[Treasures](#)

Welcome, BigDaddyC\_taco [Sign out](#)



**BigDaddyC\_taco**  
21 / M / Straight / Single  
Chicago, Illinois

Online Now

[About](#)[Photos](#)[Questions](#)[Personality](#)

**My self-summary**

I'm a young, ambitious and outgoing individual. I love traveling, having recently been to South America and through the southern states on a road trip with friends. I'm a very caring/emotional person. I enjoy anything artistic and always up for new activities. Also, I've been told I'm too perfect.

**What I'm doing with my life**

- Working two marketing jobs in downtown and Lincoln Park areas of Chicago.
- Full-time student at DePaul University studying Marketing/Sales.
- Volunteer on South Side of Chicago (Pilsen, Little Village & Englewood).
- Writer for my blog, The Plaid Tie

**My Details**

|             |                  |
|-------------|------------------|
| Last Online | Online now!      |
| Ethnicity   | Hispanic / Latin |
| Height      | 6' 0" (1.83m).   |
| Body Type   | Fit              |
| Diet        | Mostly anything  |
| Smokes      | No               |
| Drinks      | Rarely           |
| Drugs       | Never            |

# Data frames

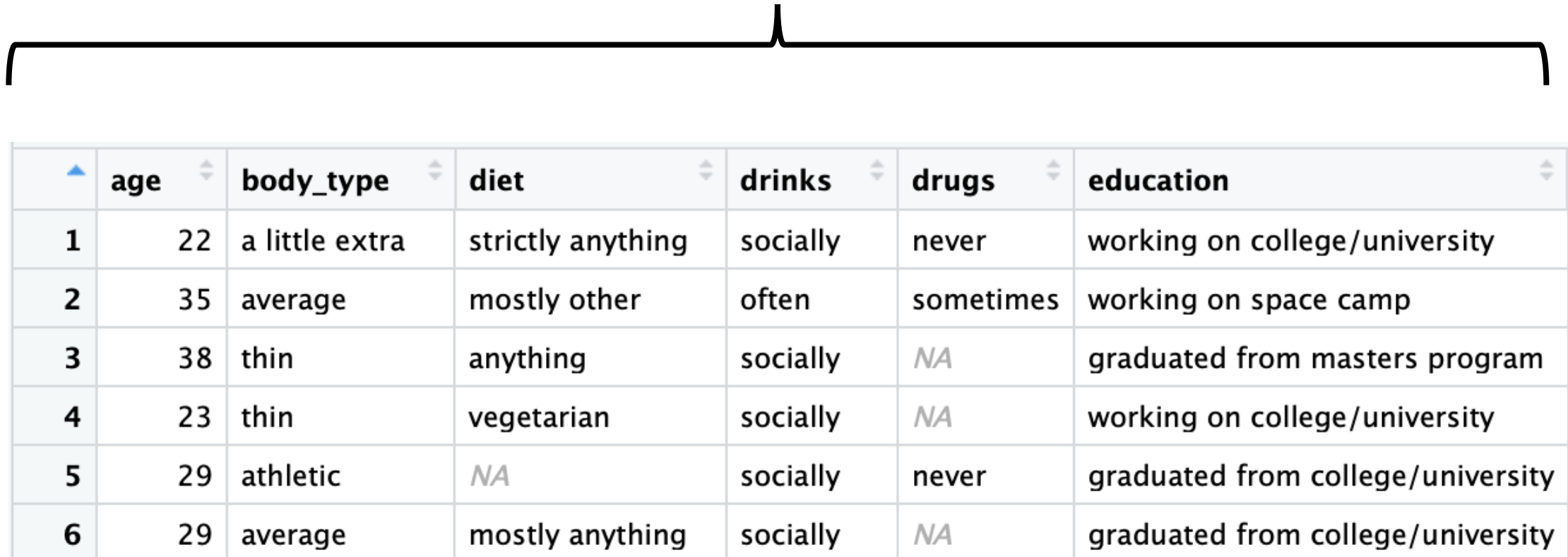
Data frames contain structured data

- > `library(SDS230)`
- > `download_data("profiles_revised.csv")`    # only needs to be run once
- > `profiles <- read.csv("profiles_revised.csv")`
- > `View(profiles)`    # the `View()` function only works in R Studio!

|   | age | body_type      | diet              | drinks   | drugs     | education                         |
|---|-----|----------------|-------------------|----------|-----------|-----------------------------------|
| 1 | 22  | a little extra | strictly anything | socially | never     | working on college/university     |
| 2 | 35  | average        | mostly other      | often    | sometimes | working on space camp             |
| 3 | 38  | thin           | anything          | socially | NA        | graduated from masters program    |
| 4 | 23  | thin           | vegetarian        | socially | NA        | working on college/university     |
| 5 | 29  | athletic       | NA                | socially | never     | graduated from college/university |
| 6 | 29  | average        | mostly anything   | socially | NA        | graduated from college/university |

# Data Frames

## Variables



|   | age | body_type      | diet              | drinks   | drugs     | education                         |
|---|-----|----------------|-------------------|----------|-----------|-----------------------------------|
| 1 | 22  | a little extra | strictly anything | socially | never     | working on college/university     |
| 2 | 35  | average        | mostly other      | often    | sometimes | working on space camp             |
| 3 | 38  | thin           | anything          | socially | NA        | graduated from masters program    |
| 4 | 23  | thin           | vegetarian        | socially | NA        | working on college/university     |
| 5 | 29  | athletic       | NA                | socially | never     | graduated from college/university |
| 6 | 29  | average        | mostly anything   | socially | NA        | graduated from college/university |

Cases

# An Example Dataset

Quantitative Variable

Categorical Variable

Cases  
(observational units)

|   | age | body_type      | diet              | drinks   | drugs     | education                         |
|---|-----|----------------|-------------------|----------|-----------|-----------------------------------|
| 1 | 22  | a little extra | strictly anything | socially | never     | working on college/university     |
| 2 | 35  | average        | mostly other      | often    | sometimes | working on space camp             |
| 3 | 38  | thin           | anything          | socially | NA        | graduated from masters program    |
| 4 | 23  | thin           | vegetarian        | socially | NA        | working on college/university     |
| 5 | 29  | athletic       | NA                | socially | never     | graduated from college/university |
| 6 | 29  | average        | mostly anything   | socially | NA        | graduated from college/university |



# Data frames

We can extract the columns of a data frame as vector objects using the \$ symbol

```
> the_ages <- profiles$age
```

Can you get the `mean()` age of users in this data set?

```
> mean(the_ages)
```

# Extracting rows from a data frame

We can extract rows from a data frame in a similar way as extracting values from a vector by using the square brackets

```
> profiles[1, ] # returns the first row of the data frame
```

```
> profiles[, 1] # returns the first column of the data
```

Note, the first column of the profiles data frame is the variable *age*, so we can also get the first column using:

```
> profiles$age # this is the same as profiles[, 1]
```

# Extracting rows from a data frame

We can also create vectors of numbers or Booleans specifying which rows we want to extract from a data frame

```
# create a vector with the numbers 1, 10, 20
```

```
> my_vec <- c(1, 10, 20)
```

```
# use my_vec to get the 1st, 10th, and 20th row in profiles
```

```
> small_profiles <- profiles[my_vec, ]
```

```
> dim(small_profiles) # number of rows and columns in the data frame
```

# Extracting rows from a data frame

Finally, we can also extract rows by creating a Boolean vector that is of the same length as the number of rows in the data frame

**TRUE** values will be extracted from the data frame while **FALSE** values will not

```
# create a vector of booleans
```

```
> my_bools <- c(TRUE, FALSE, TRUE)
```

```
# use the Boolean vector to get the 1st and 3rd row
```

```
> small_profiles[my_bools, ]
```

# Questions?



Break time?

