Class 2

Introduction to R
and
categorical data

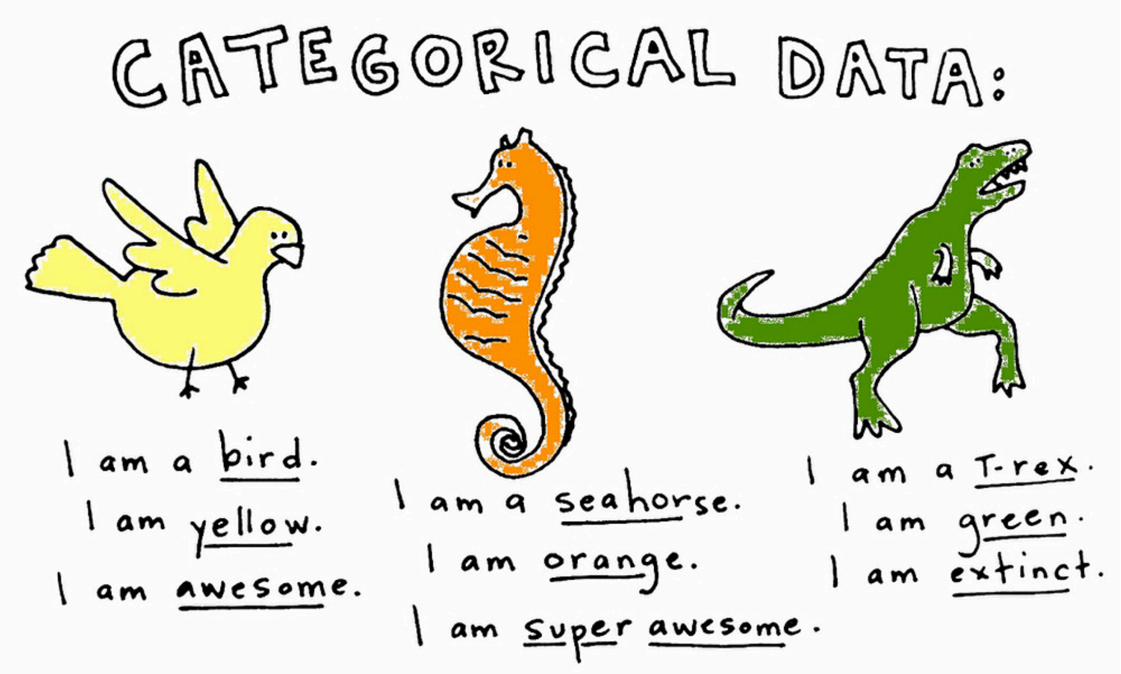
Overview

Review

Introduction to R

Categorical data

- Proportions
- Bar charts and pie plots
- Categorical data in R



Announcement

If you haven't done so yet, please remember to fill out the background survey under the quizzes on Canvas


REVIEW

Quiz time! (not to be turned in)

1. What is a population? All individuals/objects of interest (Truth)

2. What is a sample? A subset of the population (shadows)

3. What is statistical inference?
the sample

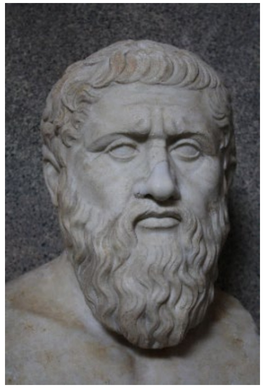
4. What are the rows of a data table called?? Cases/observational units

5. What are the columns of a data table called? Variables

6. What is the difference between categorical and quantitative variables?

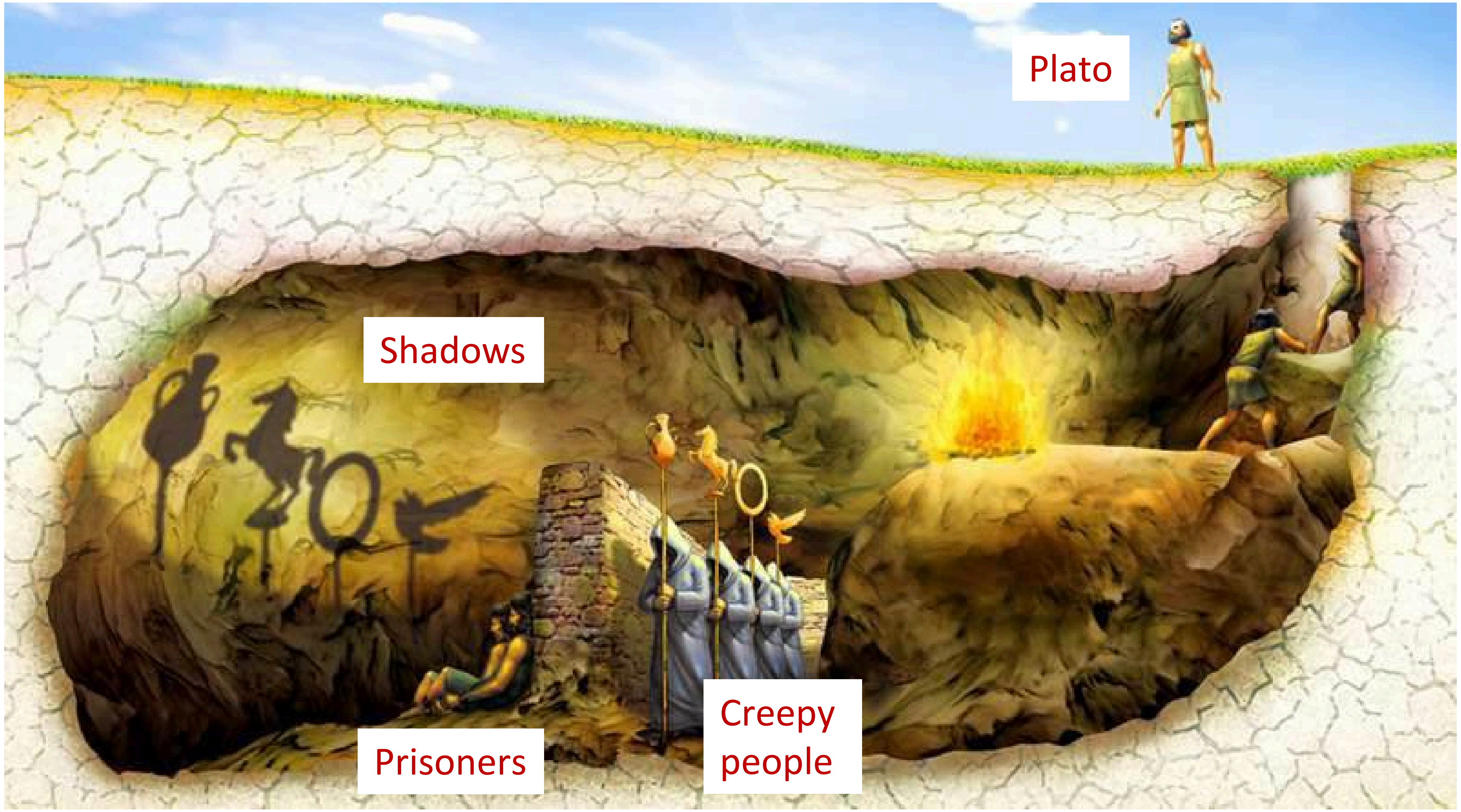
- Categorical variables fall into discrete categories
- Quantitative variables are numbers

7. Who is this?

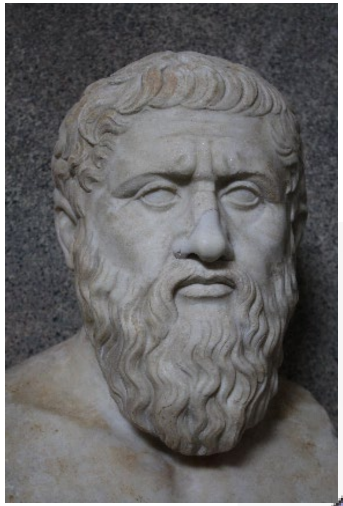


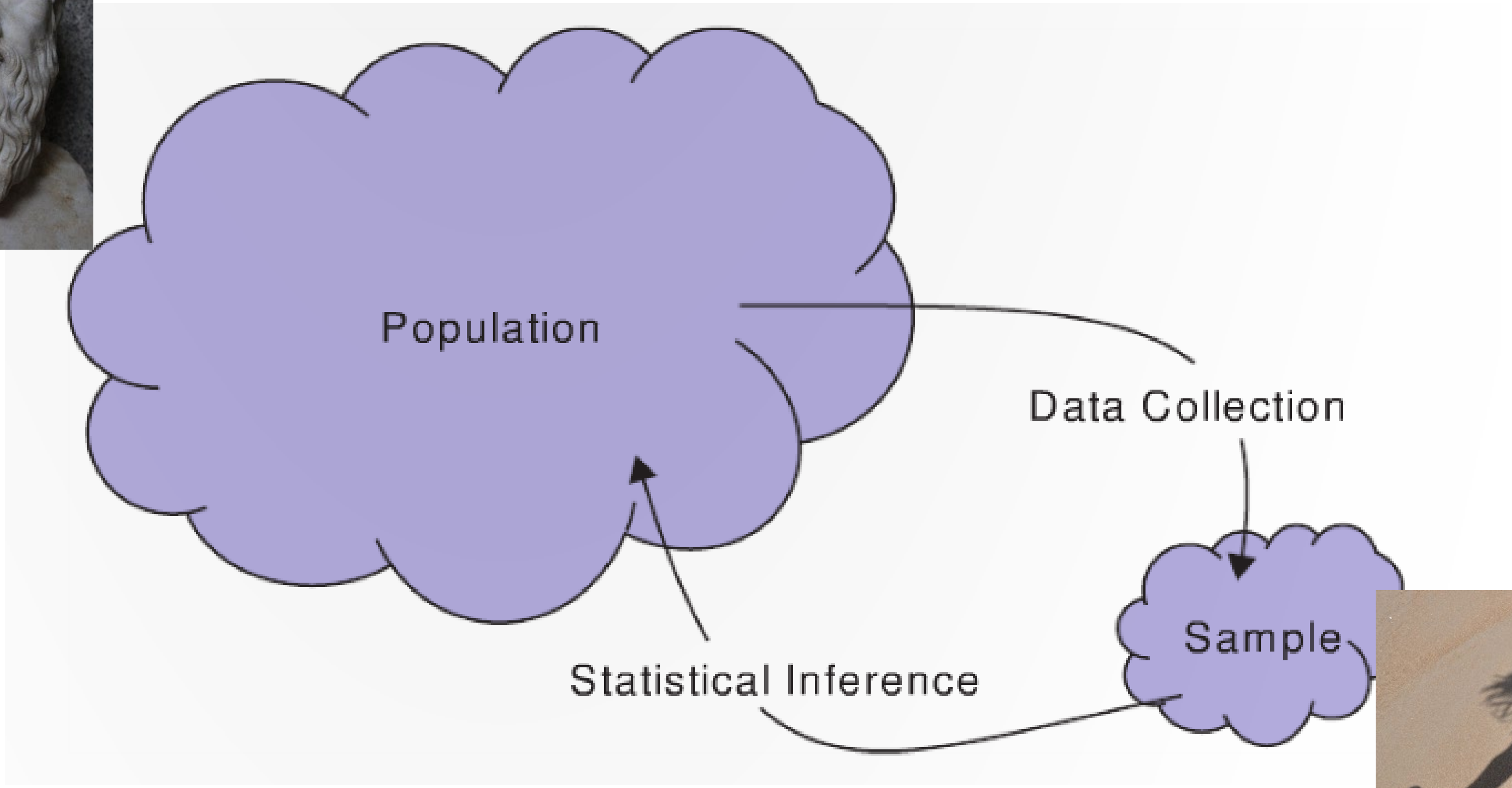
Plato

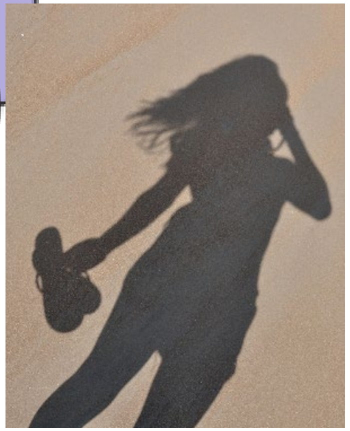
Plato’s cave



From The Republic (~ 380 BCE)







We will come back to this soon, but first a question…

Question

Q: What programming language do pirates use?
A: Arrrr

Q: Worst joke of the semester?
Please answer below…

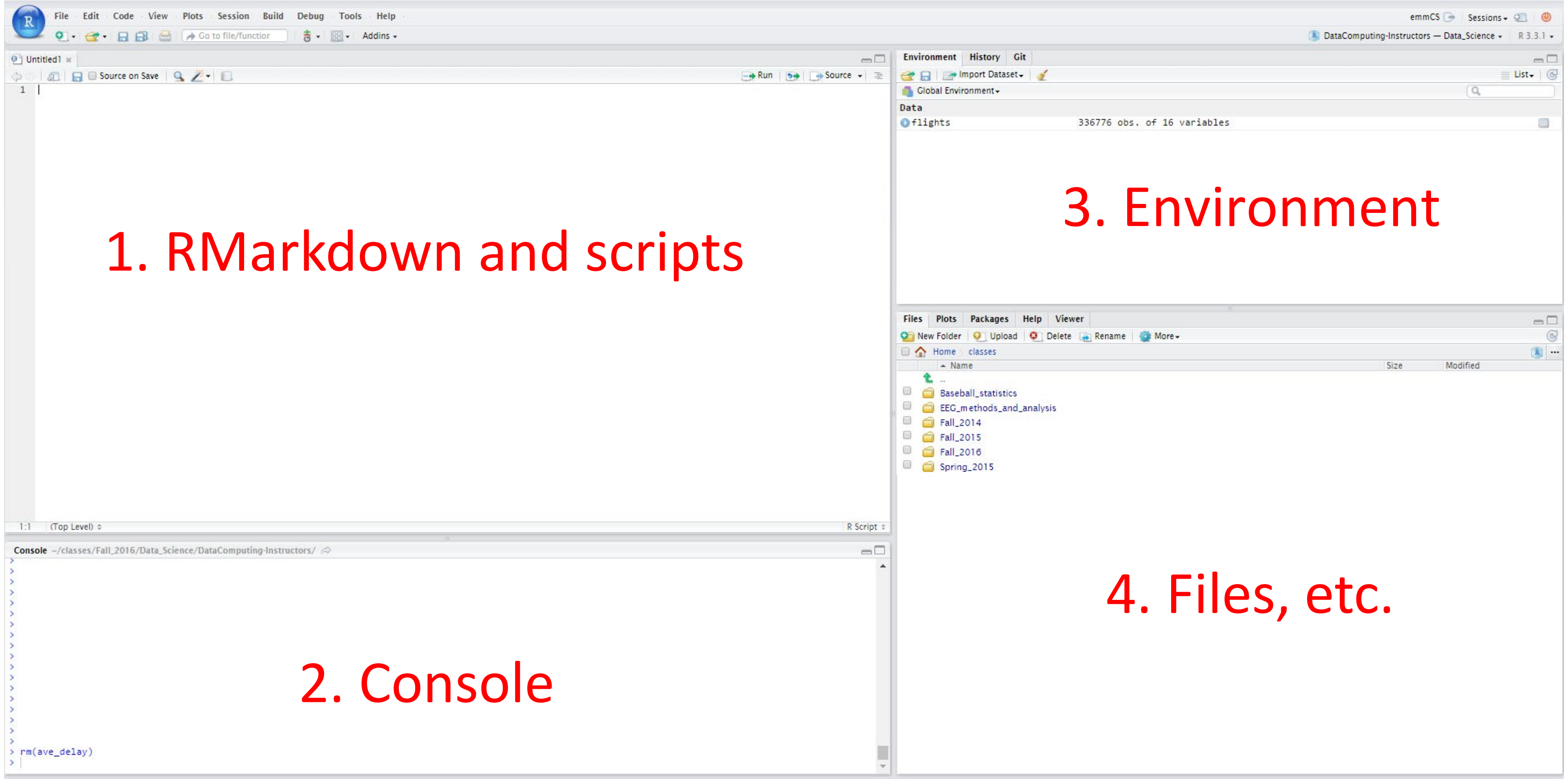
Introduction to R

R and R Studio

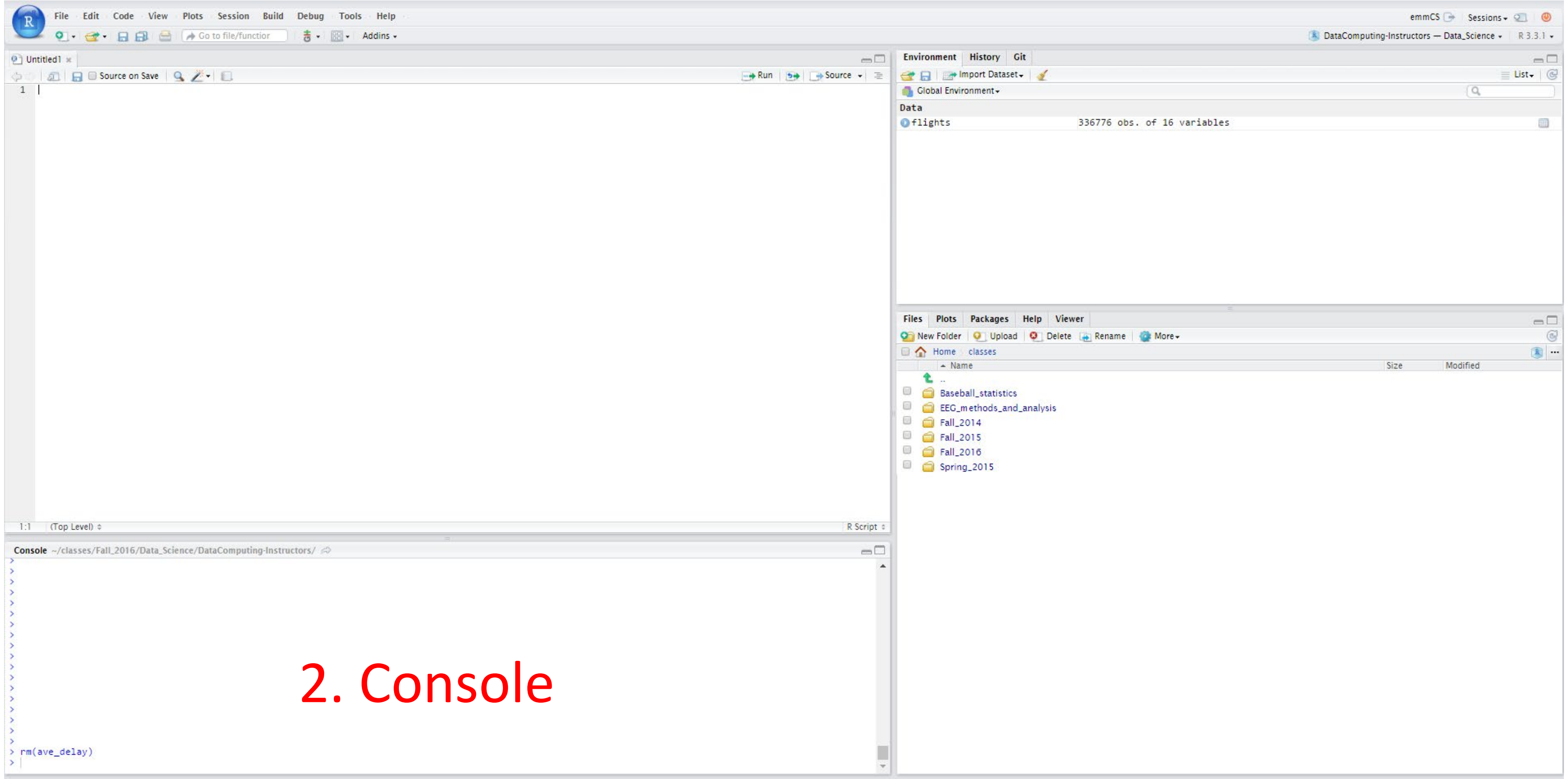
R: Engine

RStudio: Dashboard

RStudio layout



RStudio layout

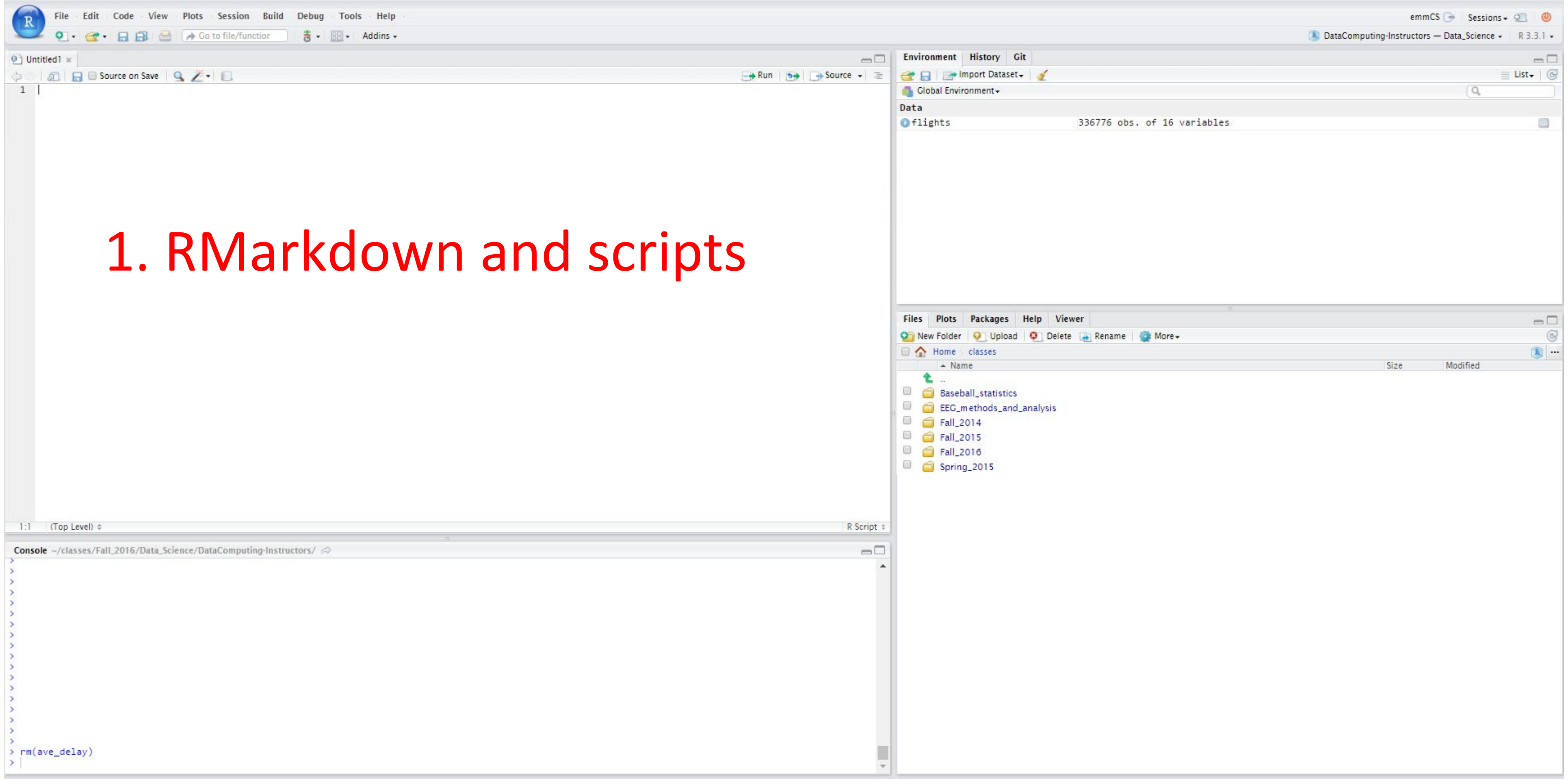


<u>R as a calculator</u>

> 2 + 2

> 7 * 5

RStudio layout



R Basics

Please open R Studio and follow along!

Arithmetic:

```
>  2 + 3
>  7 * 5
```

Assignment:

```
>  a <- 4
>  b <- 7
>  z  <- a + b
>  z
[1]  11
```

Review: Character strings and booleans

```
> a <- 7
> s <- "Statistics is great!"
> 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
```

Question

Q: What kind of grades did the pirate get in Introductory Statistics?

A: High Seas

Q: Worst joke of the semester?

A: Not likely

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("these", "are", "strings")
```

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

```
> s[3]        # what will the answer be?
```

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
```

Question

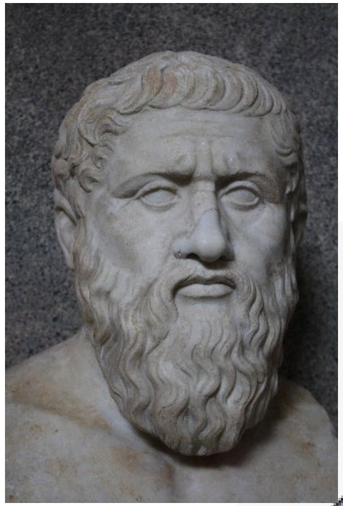
Q: What was the movie, ‘Pirates of the Caribbean’ rated?

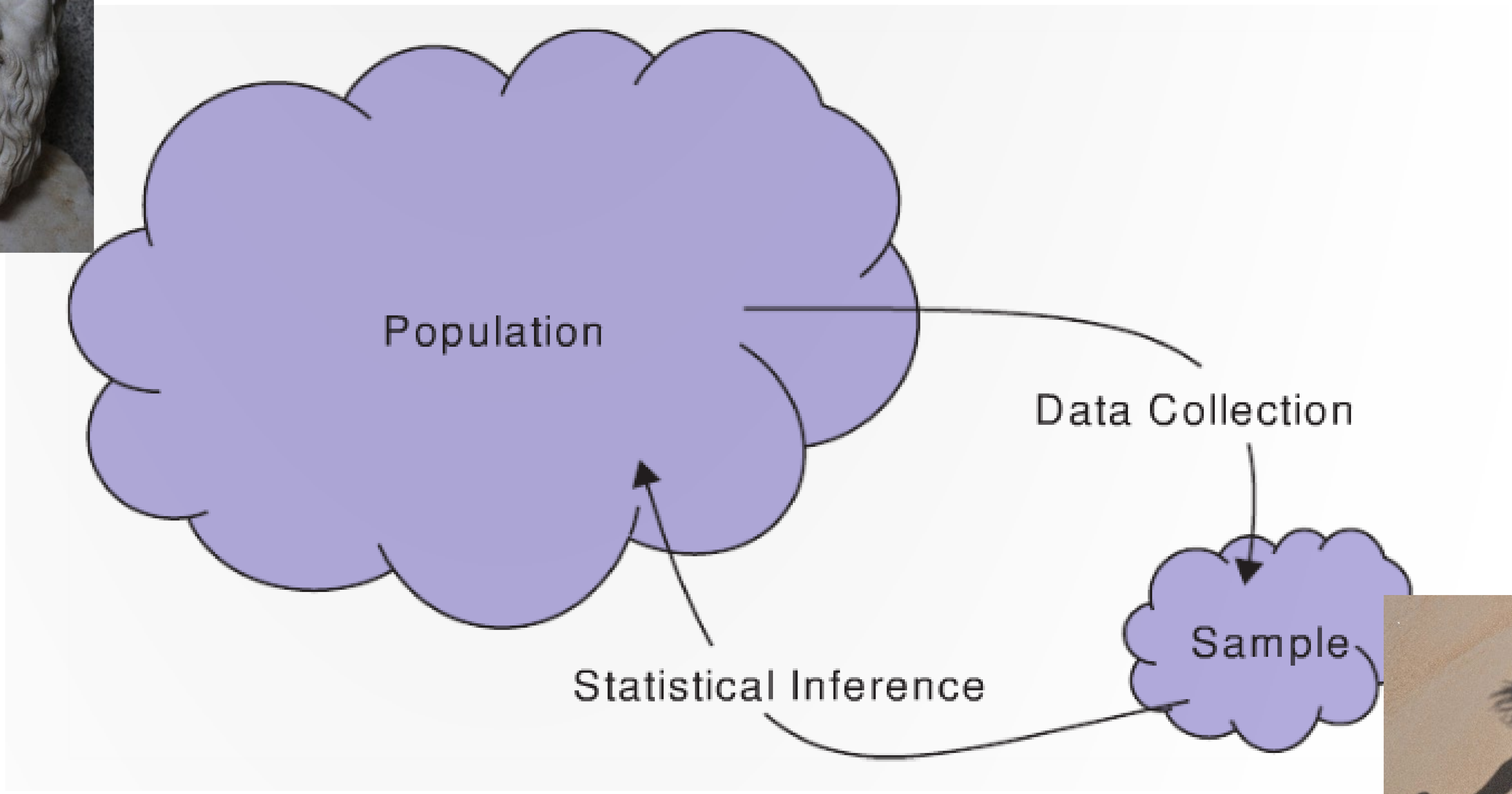
A: PG-13

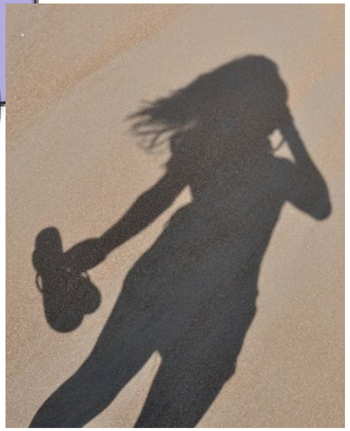
Q: Worst joke of the semester?

A: We are just getting started!

Now back to fundamental concepts in Statistics…




Population
Data Collection
Sample
Statistical Inference



Categorical variables

The sprinkle business (fictional)

ACME CORPORATION

PERFECT Corporation

ACME corporation believes that if they had the correct ratio (proportion) of red sprinkles that PERFECT corporation uses, their sales will increase

Where do samples/data come from?

To assess the proportion of sprinkles that PERFECT corporation uses, AMCE sampled 100 of PERFECT corporation's sprinkles

- The ***sample size*** is 100 (n = 100)

1	orange
2	red
3	green
4	white
5	white
6	white
7	white
8	white
9	red

Sampling example

<u>Questions:</u>

1) What are the observational units (cases)?

2) What is the variable?

3) Is the variable categorical or quantitative?

4) What is the population?

5) Do you think the samples we are getting are representative of the population?

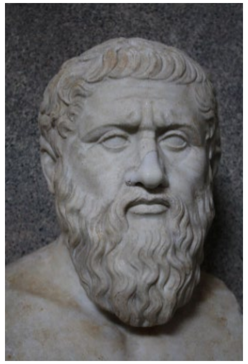
1	orange
2	red
3	green
4	white
5	white
6	white
7	white
8	white
9	red

Population parameters vs. sample statistics

A **statistic** is a number that is computed from ***data in a sample***

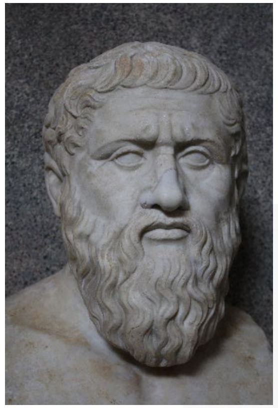
- Not to be confused with Statistics, which is a field of study

A **parameter** is a number that describes some aspect of a ***population***

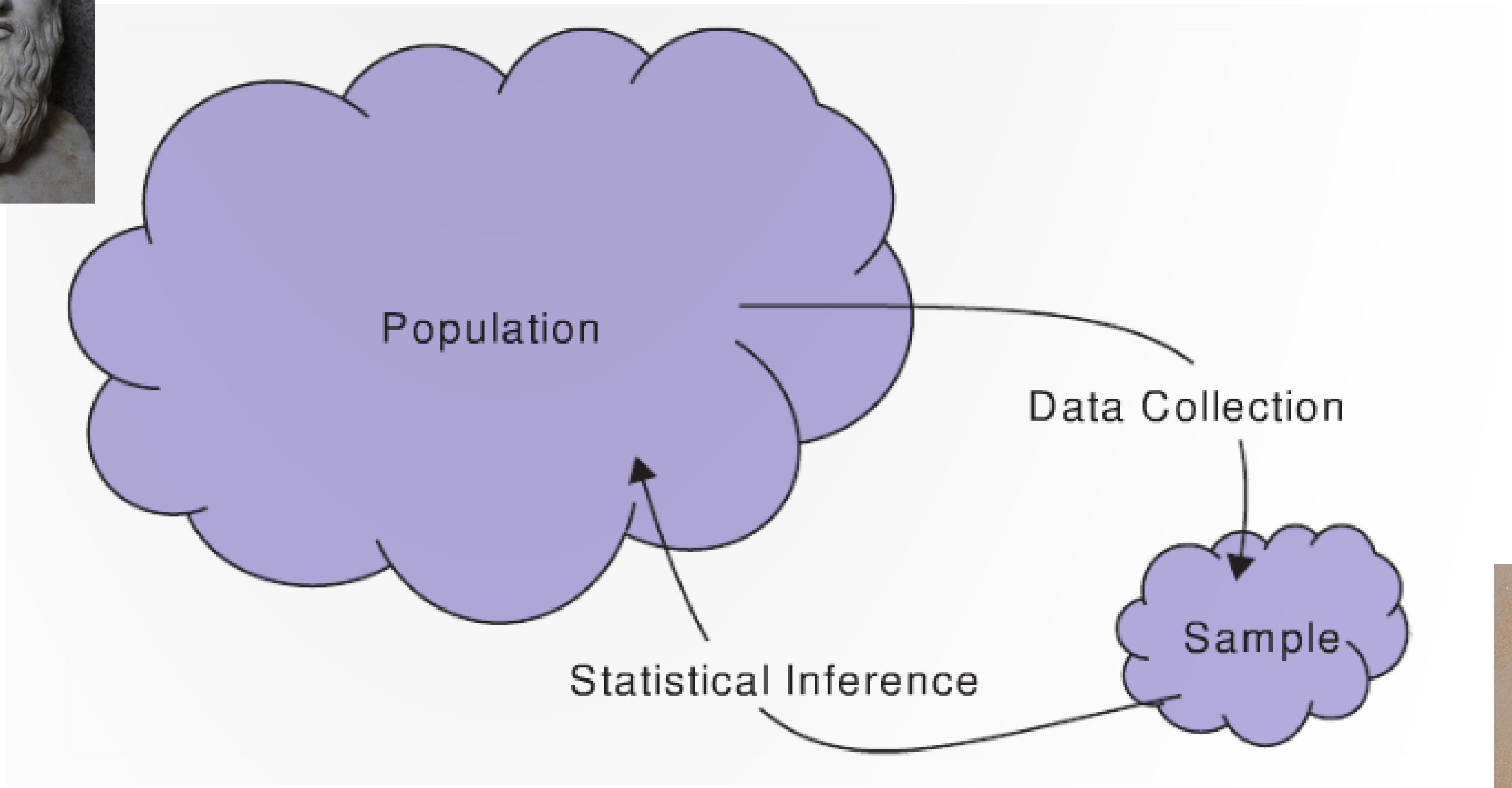


?

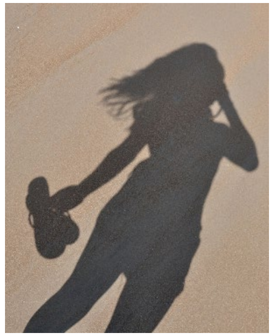
Parameters and statistics



Parameters



statistics



Proportions

For a *single* ***categorical variable***, the main ***statistic*** of interest is the *proportion* in each category

- E.g., the proportion of red sprinkles

$$\text{Proportion in a category} = \frac{\text{number in that category}}{\text{total number}}$$

Example proportion of red sprinkles

The sample

- orange, red, green, white, white, white, …, pink

The proportion for a ***sample*** is denoted $\mathbf{\hat{p}}$ (pronounced "p-hat")

- $\hat{p}_{red}$ = 13/100 = 0.13

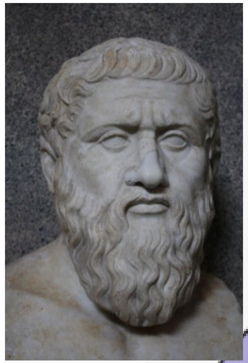
The proportion for a ***population*** is denoted $\boldsymbol{\pi}$ (the book uses p)

- π_{red} proportion if we had measured all sprinkles in the population

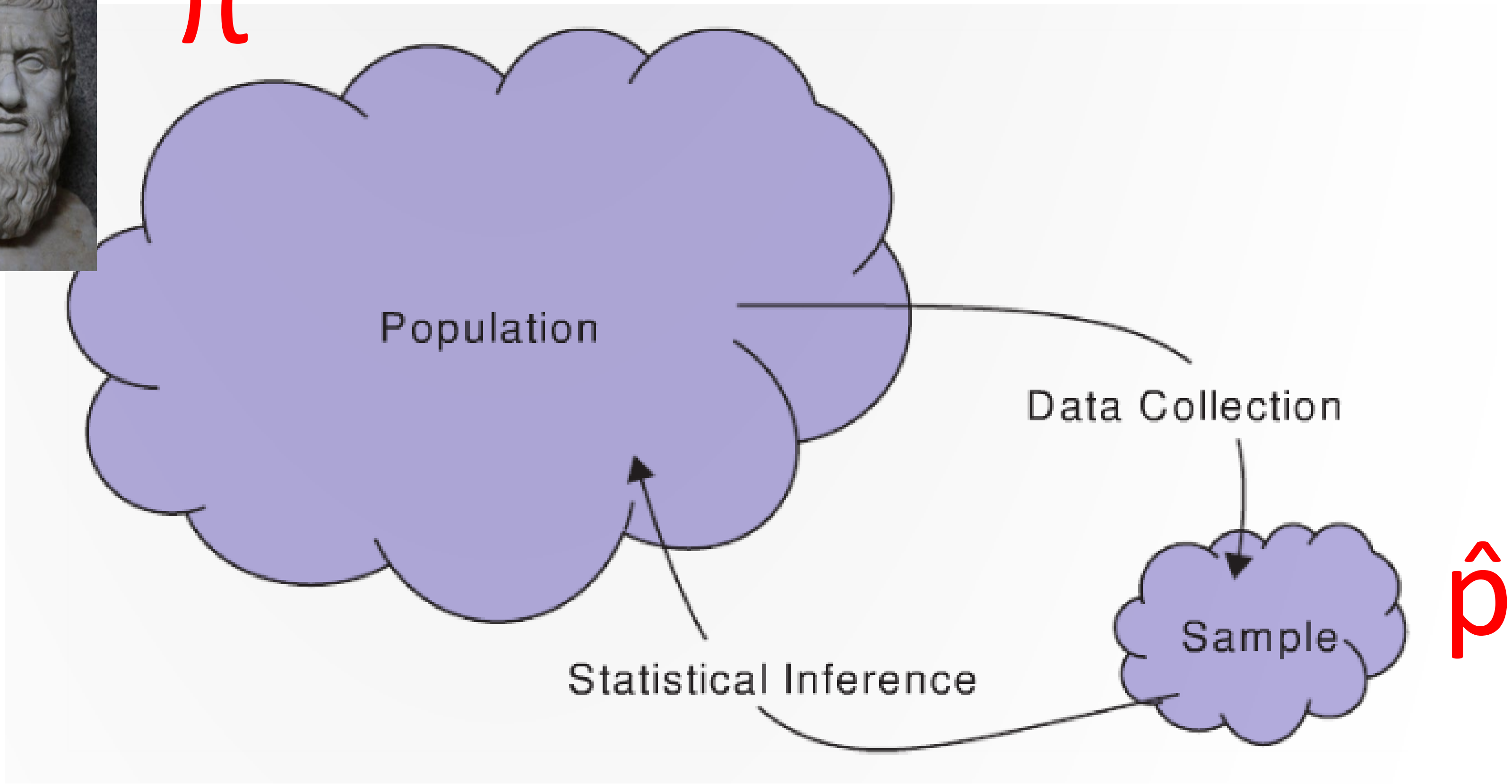
$\hat{p}$ is a ***point estimate*** of π

- i.e., $\hat{p}$ our best guess of what π is

Sample vs. Population proportion



π



$\hat{p}$

Different samples yield different values for the statistic

$\hat{p}_{s1\_red} = 0.13$

$\hat{p}_{s2\text{-}red} = 0.11$

$\hat{p}_{s3\text{-}red} = 0.15$

Calculating counts on a categorical variable

The count of how many items are in each category can be summarized in a ***frequency table***

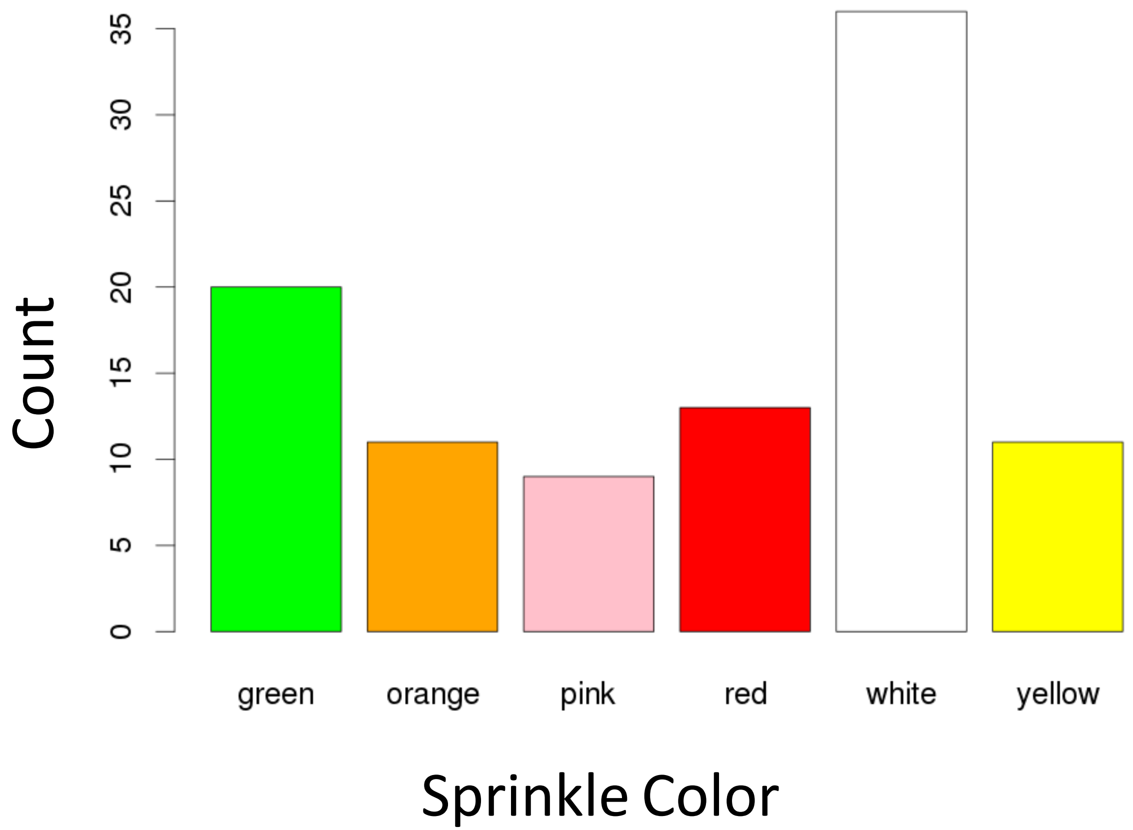
Color	**green**	**orange**	**pink**	**red**	**white**	**yellow**		**Total**
Count	20	11	9	13	36	11		100

Calculating proportions (relative frequencies)

We can convert a frequency table into a ***relative frequency table*** by dividing each cell by the total number of items

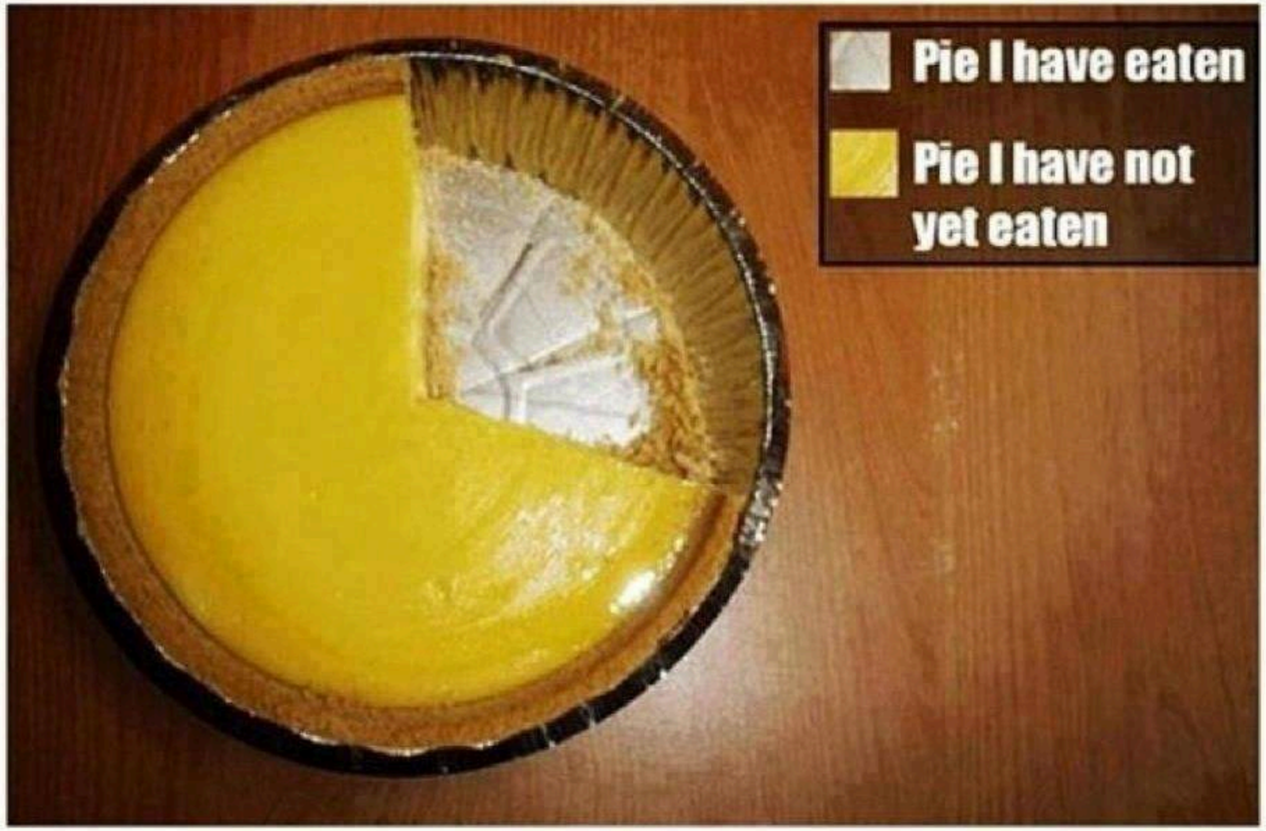
Color	**green**	**orange**	**pink**	**red**	**white**	**yellow**		**Total**
Count	.20	.11	.09	.13	.36	.11		1

Visualizing categorical data: The Bar Chart



Visualizing categorical data: The Pie Chart

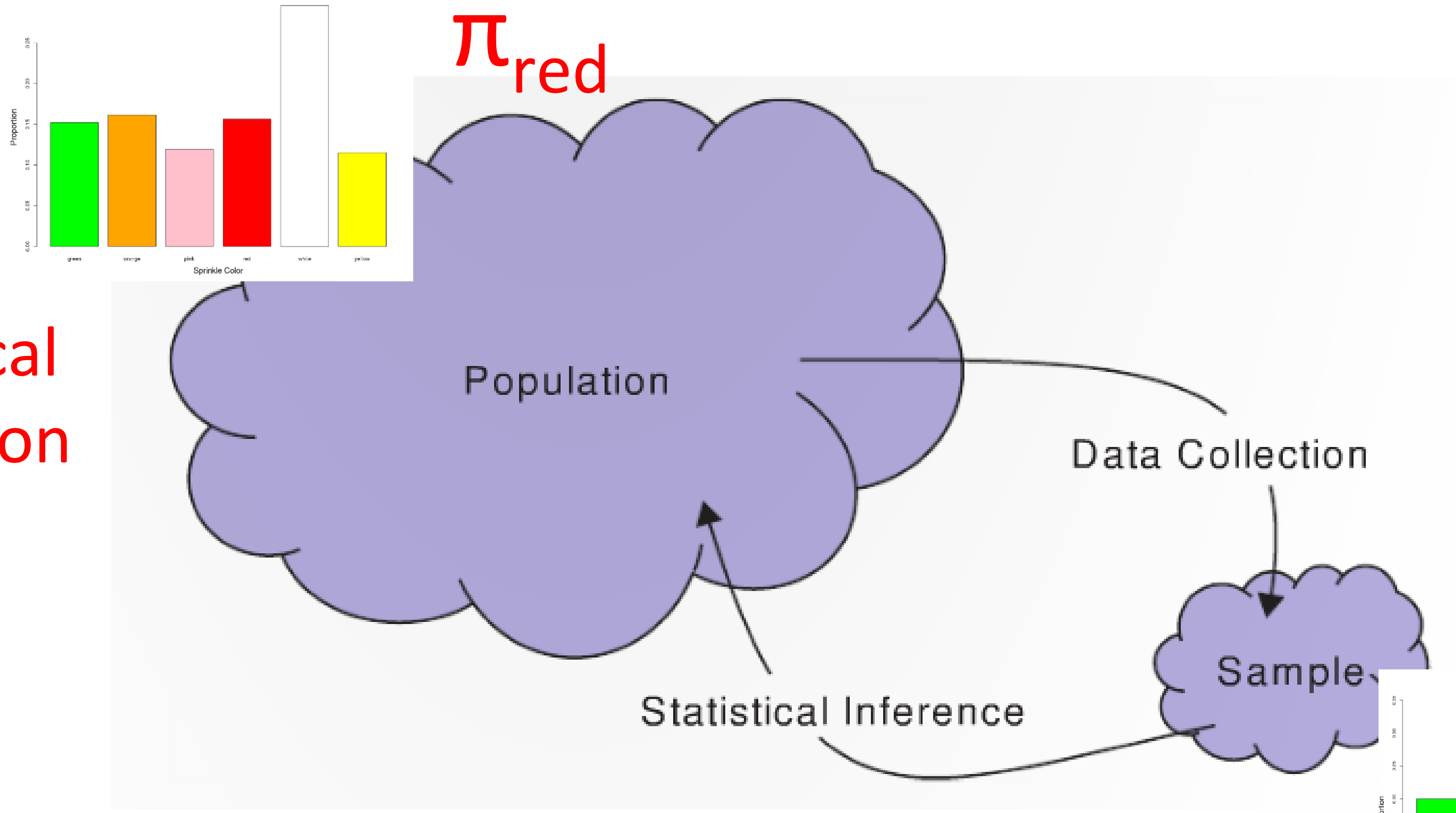
World's Most Accurate Pie Chart



Summary: Sample and Population proportion

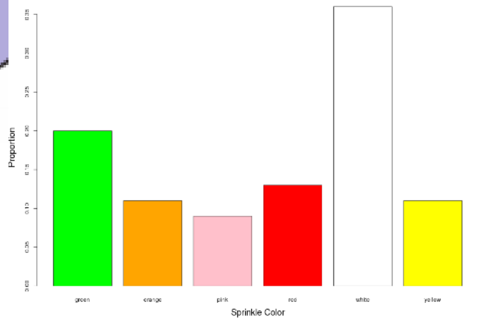
π_{red}

Categorical distribution



Bar chart

$\hat{p}_{red}$



Let’s sample virtual sprinkles…

Sampling virtual sprinkles

```
library(SDS100)

sprinkle_sample <- get_sprinkle_sample(100)

sprinkle_count_table <- table(sprinkle_sample)
sprinkle_prop_table <- prop.table(sprinkle_count_table)

barplot(sprinkle_count_table)
pie(sprinkle_count_table)
```

Summary of concepts

1. A **statistic** is a number that is computed from ***data in a sample***

- The number of items in a sample is called the ***sample size*** and is usually denoted with the symbol n

2. A **parameter** is a number that describes some aspect of a ***population***

3. A point estimate is using a value of a statistic as a guess for the value of a parameter

4. When calculating proportions:

- The proportion statistic is denoted $\hat{p}$
- The population proportion is denoted π
- Thus $\hat{p}$ is a ***point estimate*** of π

5. Proportions can be summarized in a **relative frequency table** and can be visualized using **bar plots** and **pie charts**

Summary of R

```
# a vector of character strings (or factors)
my_sample <- c("orange", "red", "green", "white", " white", ... )

# creating a table using the table() function
my_table <- table(my_sample)

# creating a frequency table using the prop.table() function
prop.table(my_table)

# creating bar and pie charts
barplot(my_table)
pie(my_table)
```

rmarkdown

www.rstudio.com

R Markdown

R Markdown (.Rmd files) documents allow you to combine written descriptions with R analysis code.

You can then ‘knit’ these documents to create nice looking report.

All homework in this class will be done using R Markdown.

R Markdown document structure

R Markdown documents have written sections and code sections.

Everything in R chunks is executed as code:

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

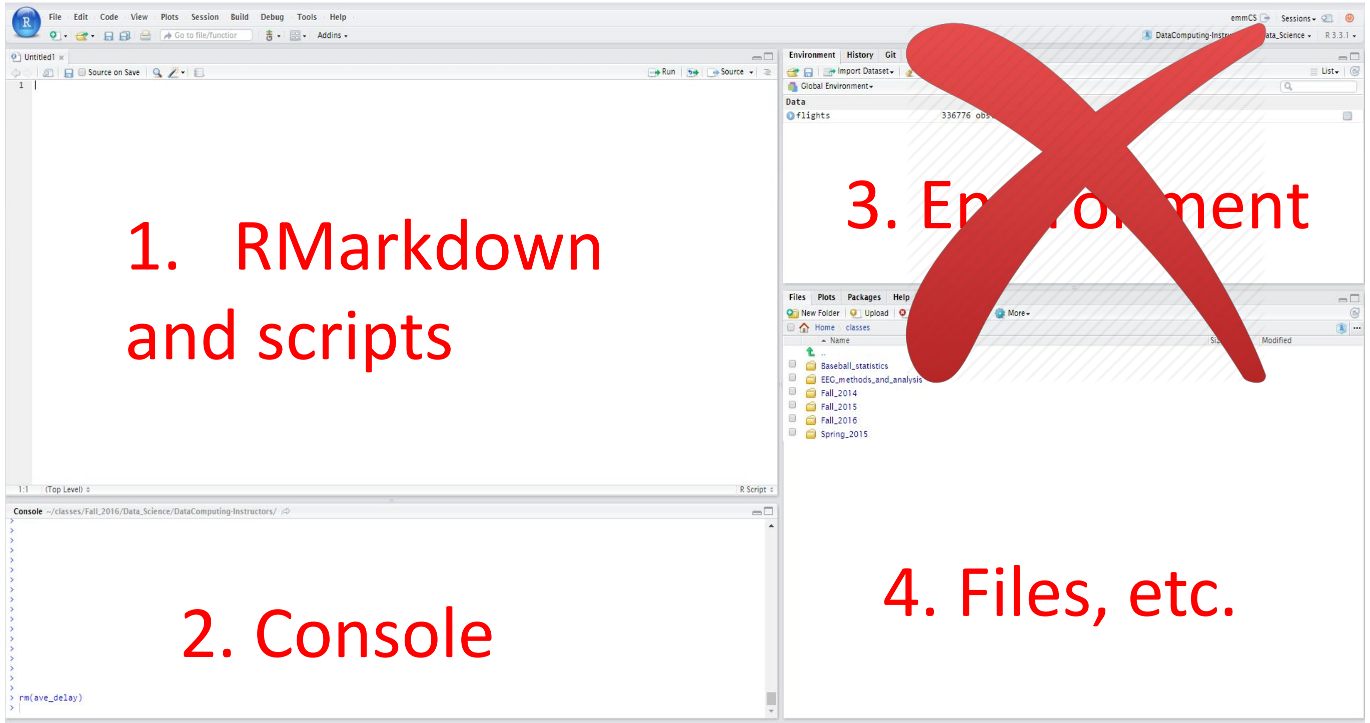
Everything outside R chunks appears as text.

R Markdown

Note: R Markdown documents **do not have access to variables in the global environment!**

Instead have their own environment.

Why is this a good thing???



R Markdown

Special LaTeX characters can be embedding in the text regions outside of the code chunks

Examples:

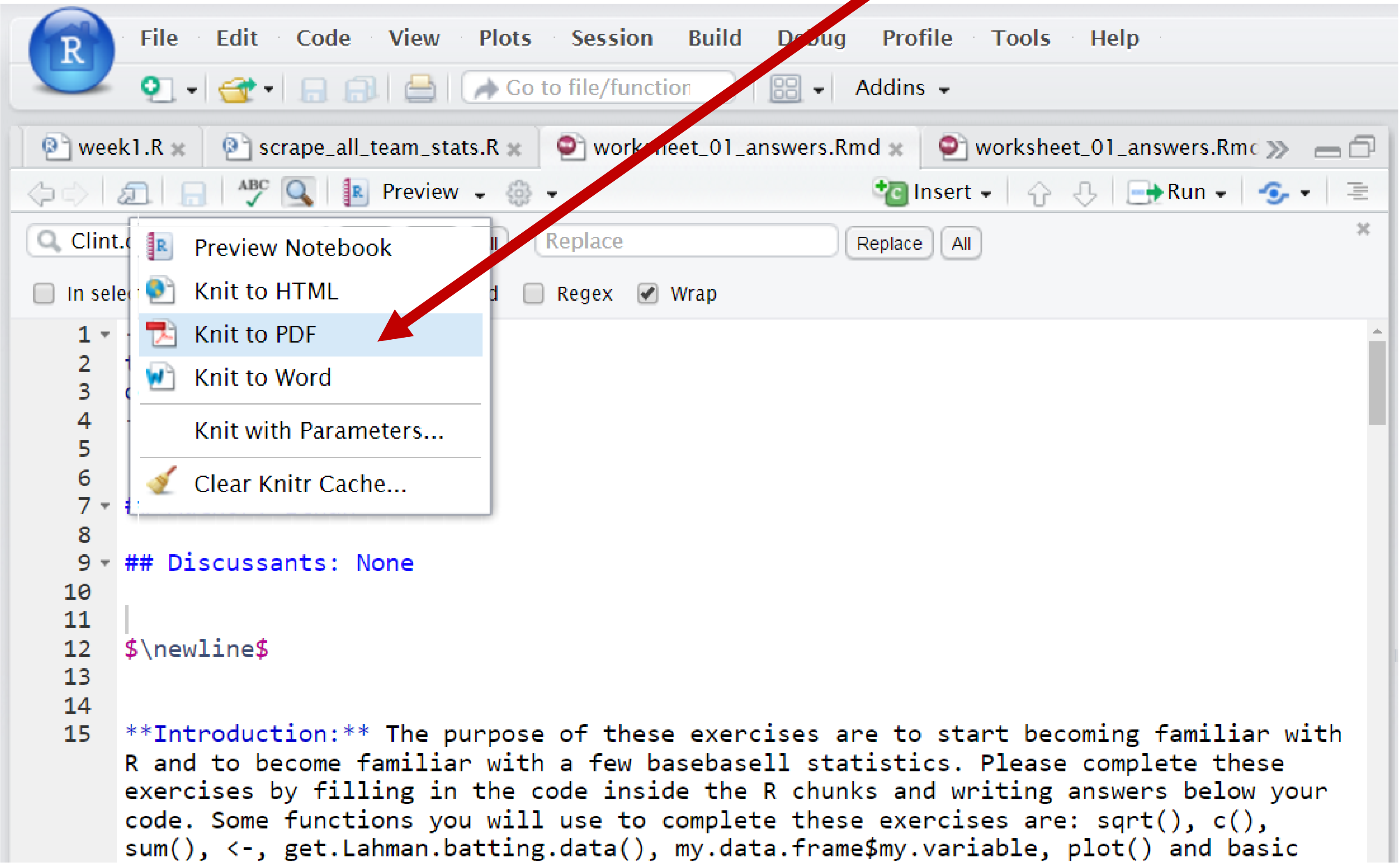
`$\pi$`

`$\hat{p}$`

`$\hat{p}_{red}$`

Knitting to a pdf

Turn in a pdf of your solutions to Gradescope



Avoid hard to debug code!

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

Comment out parts of the code that isn’t working (using the # symbol) until you can find the line of code that is giving the error message

Homework 0

To practice the material from the first week of class I have created an R Markdown document called ‘homework 0’

You will not turn in this homework, it is purely for practice!

Do not worry if you run into any technical difficulties with the homework, the purpose of this homework is to work out any issues so you will be all set for the first real homework.