Intro to R for Biologists

IBiS Special Topics, Fall 2021
Class 2: Sept. 27, 2021

Erik Andersen and Shelby Blythe

Outline for today

Packages

Reproducible data organization

Reading in data

Good data stewardship and structure

Markdown

Packages are collections of functions and/or data

- You want to load in a huge data set?

There is a package for that!

- You want to statistically analyze time series data?

There is a package for that!

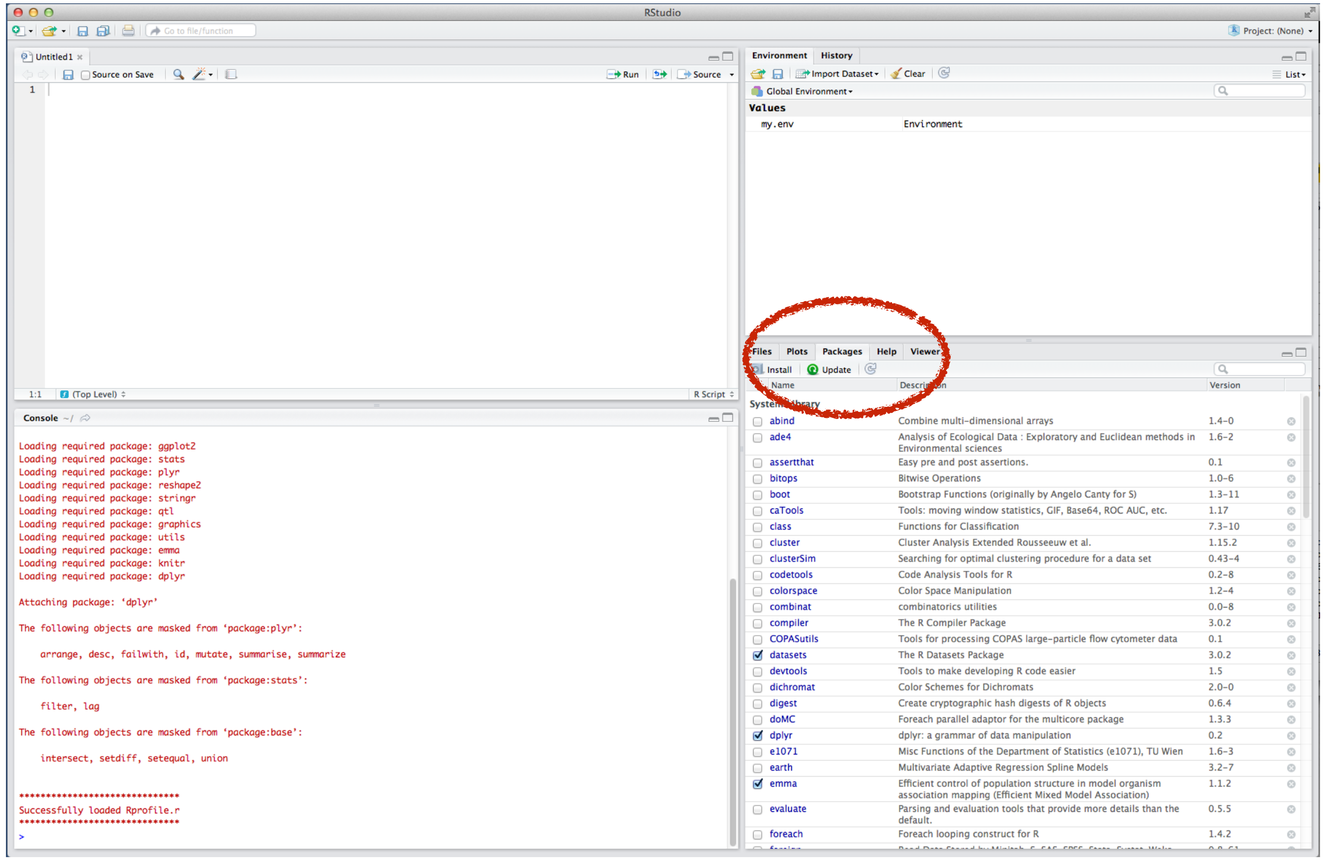
- You want to make a web app for live data analysis?

There is a package for that!

- You want to make beautiful 3D interactive plots?

There is a package for that!

Let’s install some packages



1. Click on Packages in bottom right pane
2. Click on Install
3. Make sure the Install from drop down is CRAN
4. Enter **`devtools`**
5. Enter **`rio, readr, data.table`**

Where to find packages?

stackoverflow

Outline for today

Packages

Reproducible data organization

Reading in data

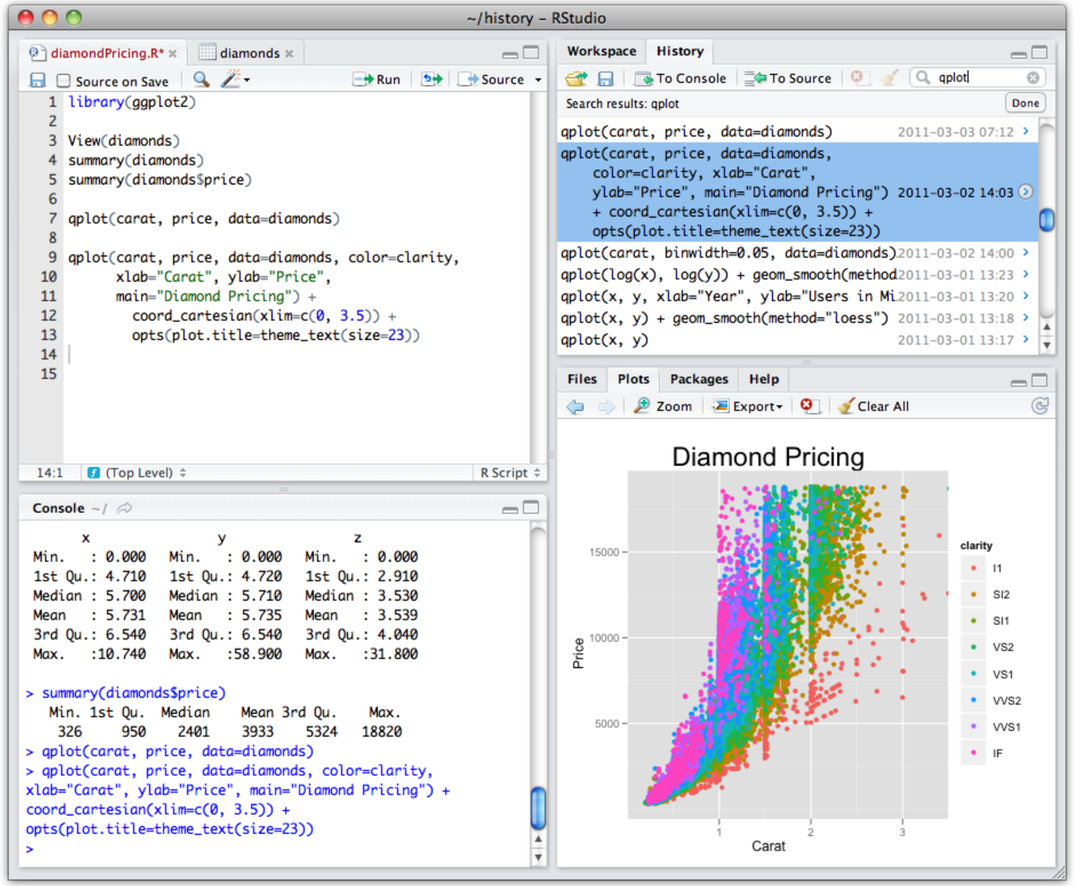
Good data stewardship and structure

Markdown

R helps you set up reproducible research practices

… but keeping track of all your work can be a challenge

- Part of the goal of this class is to teach you strategies for developing an analysis, and keeping track of a project as you would with a lab notebook.
- This starts with how you organize your file system, and how you name files.
- It extends to how you input commands into the editor, and learning to code in a predictable and reproducible way.



The standard should ultimately be: whatever code you write, that you could hand it to a stranger and it would run (and give the same results) on a completely different computer.

All research data collected in projects funded by NIH and NSF should be FAIR (Findable, Accessible, Interoperable, Resuable).

Those requirements can be summarized as:

- All raw and processed data are published with manuscript as supplemental data
- All supplemental data should be flat files (csv, tsv)
- All code to take raw data to processed and to make plots should be published with the manuscript.
- All code should have all dependencies described (if not containerized).

It’s time to get organized…

```
MyProject/
  data/
    raw/
    processed/
  scripts/
  results/
  readme.md
```

```
MyProject/data/raw/
```

It’s time to get organized…

Using that file structure…

- Transfer to a Github repository, track your code, data, etc.
- Create a RStudio Project and track your package versions, code, data, etc.
- Connect your RStudio Project to Github

Outline for today

Packages

Reproducible data organization

Reading in data

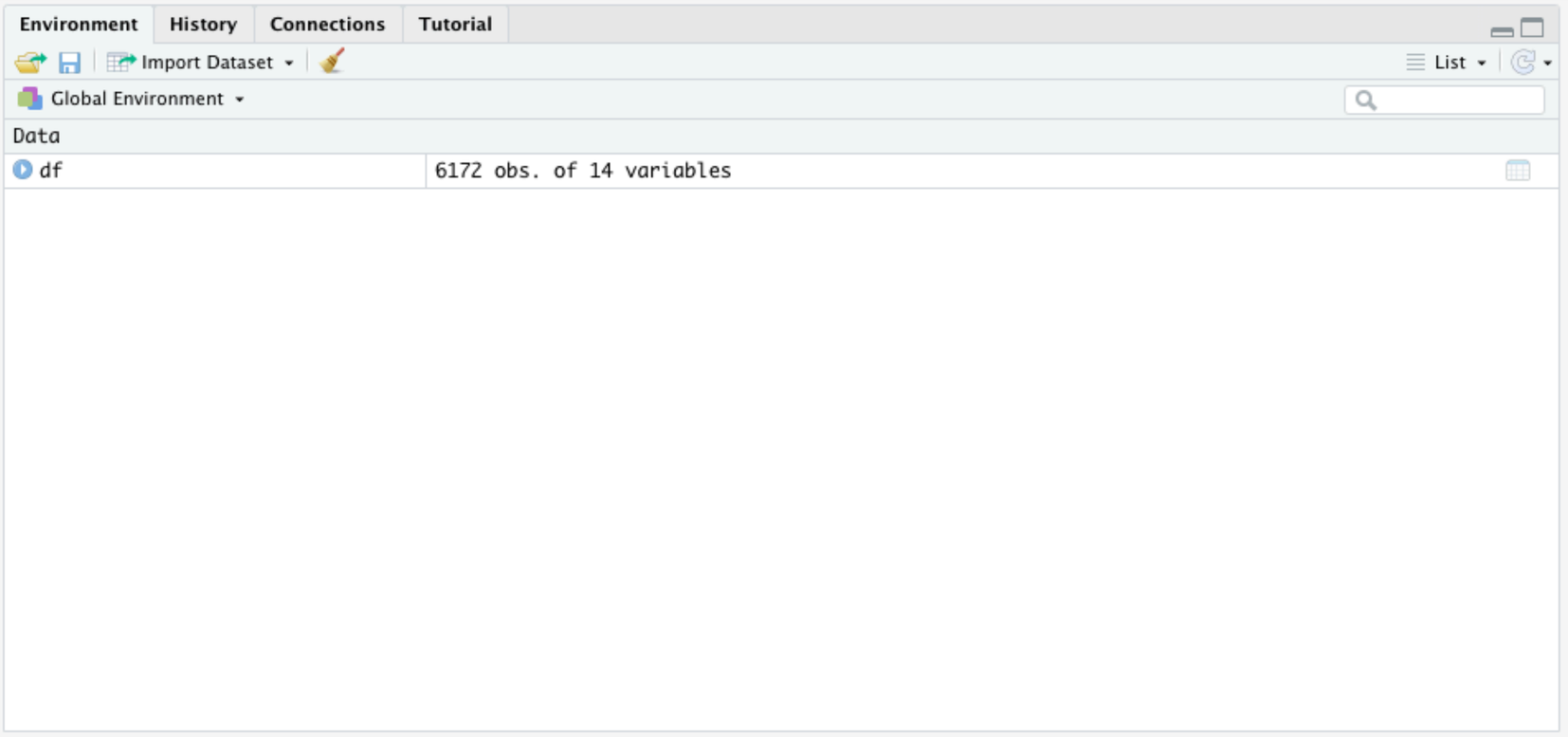
Good data stewardship and structure

Markdown

Reading in data

`read.delim()` is a generic function to read in flat files, like csv or tsv files.

```
df <- read.delim(file="~/R_for_Biologists/data/lego.csv", header=TRUE, sep=",")
```



Reading in data

Other packages have much more powerful data reading capabilities.

readr, `read_csv()` or `read_tsv()`

rio, `import()`

data.table, `read_csv()`

Data structures in R

Data.frames are data organized into rows and columns.

Data frames are organized by [row,column]. You must use the comma!

You can look at a vector of the data frame using $, *i.e.*, **`df$column`**

Or you can specify the row or column to output, *i.e.*, **`df[,9]`** outputs the ninth column

A new data object is in your workspace. Click on it. Type **`df`** to display all of it.
Try **`head(df)`** and **`tail(df)`**.
Try **`summary(df)`**.
Try **`View(df)`.**

Take some time to move through your data

1. Are there missing values? `is.na()`
2. What are the column names? `colnames()`
3. What are the dimensions? `dim()`
4. What are the row values? `rownames()`
5. Are the data organized long or wide?

Outline for today

Packages

Reproducible data organization

Reading in data

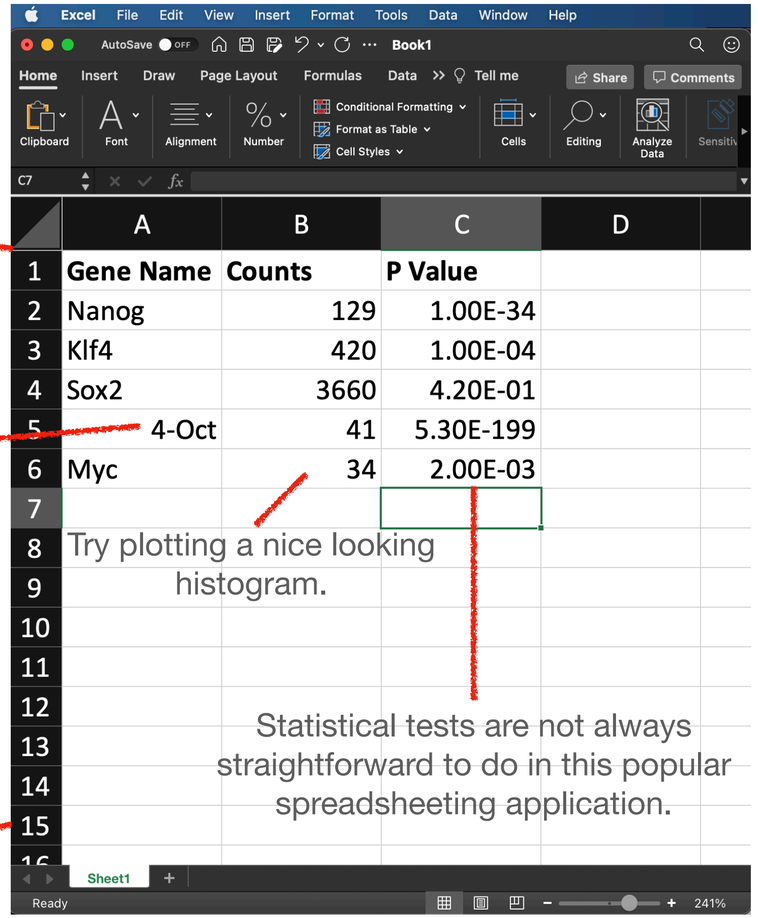
Good data stewardship and structure

Markdown

But most data are not single vectors.

We deal with tables. But for most folks, tables = spreadsheets

- Spreadsheets are mostly fine for small data tables.
  - What happens when they get big?
  - What happens when you need to extract subsets of data?
  - What happens when you have to do a repetitive task?
- **This is where R can help.**



You are restricted to this horrible rectangular space

This is a date now for some reason.

Try plotting a nice looking histogram.

Statistical tests are not always straightforward to do in this popular spreadsheeting application.

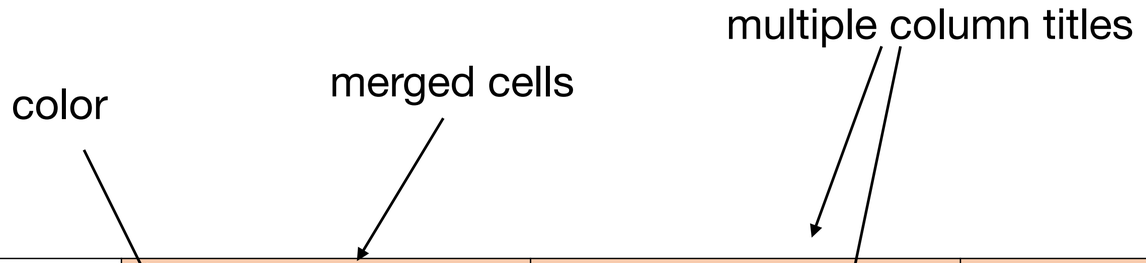
Big data sets can have millions of rows (limited here to ~$1x10^6$)

Messy vs. clean data

“Happy families are all alike; every unhappy family is unhappy in its own way.” — Leo Tolstoy

“Tidy datasets are all alike, but every messy dataset is messy in its own way.” — Hadley Wickham

Messy vs. clean data



	Replicate1			Replicate2			Replicate3			Replicate4		
PLATE1	day0	day2	day2/day0	day0	day2	day2/day0	day0	day2	day2/day0	day0	day2	day2/day0
BRC20067	229	2	0.8733624454148	138	0	0	234	0	0	136	0	0
DL238	285	183	64.210526315789	334	161	48.2035928143713	179	71	39.6648044692737	224	110	49.1071428571429
A3	166	0	0	104	0	0	231	0	0	251	0	0
A6_R1	57	0	0	62	0	0	60	0	0	75	0	0
A6_R2	150	0	0	256	6	2.34375	265	4	1.50943396226415	236	0	0
B2	187	2	1.0695187165775	165	1	0.606060606060606	183	0	0	171	4	2.33918128654971
B3	179	12	6.7039106145251	202	26	12.8712871287129	251	22	8.76494023904382	243	26	10.6995884773663
B6	128	15	11.71875	113	9	7.9646017699115	220	10	4.54545454545455	214	14	6.54205607476636
B8	268	0	0	155	0	0	164	1	0.60975609756097	175	0	0
B9	93	2	2.1505376344086	118	2	1.69491525423729	132	0	0	111	2	1.8018018018018
B10	188	2	1.0638297872340	379	2	0.527704485488127	225	0	0			
C6	169	0	0	129	1	0.775193798449611	130	0	0	115	1	0.869565217391304

Q: Clean or messy?

Messy vs. clean data

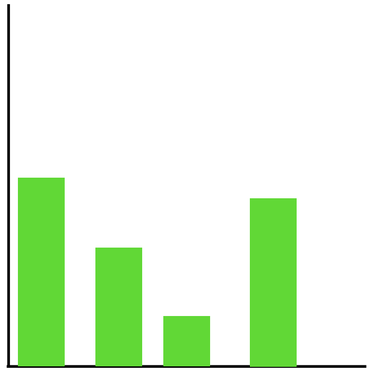
strain	survival	rep	day
BRC20067	0.873362445	A	1
DL238	64.210526315	A	1
A3		A	1
A6_R1		A	1
A6_R2		A	1
B2	1.069518716	A	1
B3	6.703910614	A	1
B6	11.7187	A	1
B8		A	1
B9	2.150537634	A	1
B10	1.063829787	A	1
C6		A	1
BRC20067		B	1
DL238	48.203592814	B	1
A3		B	1
A6_R1		B	1
A6_R2	2.3437	B	1
B2	0.606060606	B	1
B3	12.871287128	B	1
B6	7.964601769	B	1
B8		B	1

Rules for tidy data

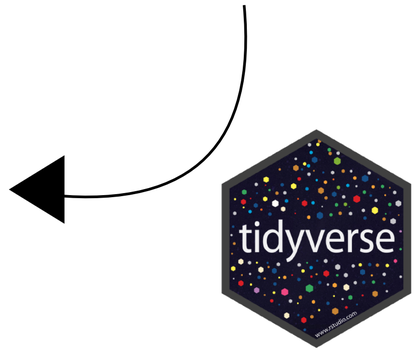
- Each **variable** must have its own **column**
- Each **observation** must have its own **row**
- Each **value** must have its own **cell**

Messy vs. clean data

	Replicate1			Replicate2			Replicate3			Replicate4		
PLATE1	day0	day2	day2/day0	day0	day2	day2/day0	day0	day2	day2/day0	day0	day2	day2/day0
BRC20067	229	2	0.8733624454148	138	0	0	234	0	0	136	0	0
DL238	285	183	64.210526315789	334	161	48.2035928143713	179	71	39.6648044692737	224	110	49.1071428571429
A3	166	0	0	104	0	0	231	0	0	251	0	0
A6_R1	57	0	0	62	0	0	60	0	0	75	0	0
A6_R2	150	0	0	256	6	2.34375	265	4	1.50943396226415	236	0	0
B2	187	2	1.0695187165775	165	1	0.606060606060606	183	0	0	171	4	2.33918128654971
B3	179	12	6.703910614525	202	26	12.8712871287129	251	22	8.76494023904382	243	26	10.6995884773663
B6	128	15	11.71875	113	9	7.9646017699115	220	10	4.54545454545455	214	14	6.54205607476636
B8	268	0	0	155	0	0	164	1	0.60975609756097	175	0	0
B9	93	2	2.150537634408	118	2	1.69491525423729	132	0	0	111	2	1.8018018018018
B10	188	2	1.0638297872340	379	2	0.527704485488121	225	0	0			
C6	169	0	0	129	1	0.775193798449611	130	0	0	115	1	0.869565217391304



strain	survival	rep	day
BRC20067	0.873362445414	A	1
DL238	64.21052631578	A	1
A3	0	A	1
A6_R1	0	A	1
A6_R2	0	A	1
B2	1.069518716577	A	1
B3	6.703910614525	A	1
B6	11.71875	A	1
B8	0	A	1
B9	2.150537634408	A	1
B10	1.063829787234	A	1
C6	0	A	1



Some thoughts about organizing raw data

- Be consistent.
- Write dates as `YYYY-MM-DD` or YYYYMMDD.
- Fill in all of the cells. Use NA. No spaces
- Put just one thing in a cell.
- Make it a rectangle.
- Create a data dictionary.
- No calculations in the raw data files.
- Don’t use font color or highlighting as data.
- Choose good names for things.
- Make backups.
- Use data validation to avoid data entry mistakes. Unit tests!
- Save the data in plain text files.

from Karl Broman (UW Madison), check out his fantastic blog
Other great rules here: https://github.com/jtleek/datasharing

Outline for today

Packages

Reproducible data organization

Reading in data

Good data stewardship and structure

Markdown

What about that readme.md file?

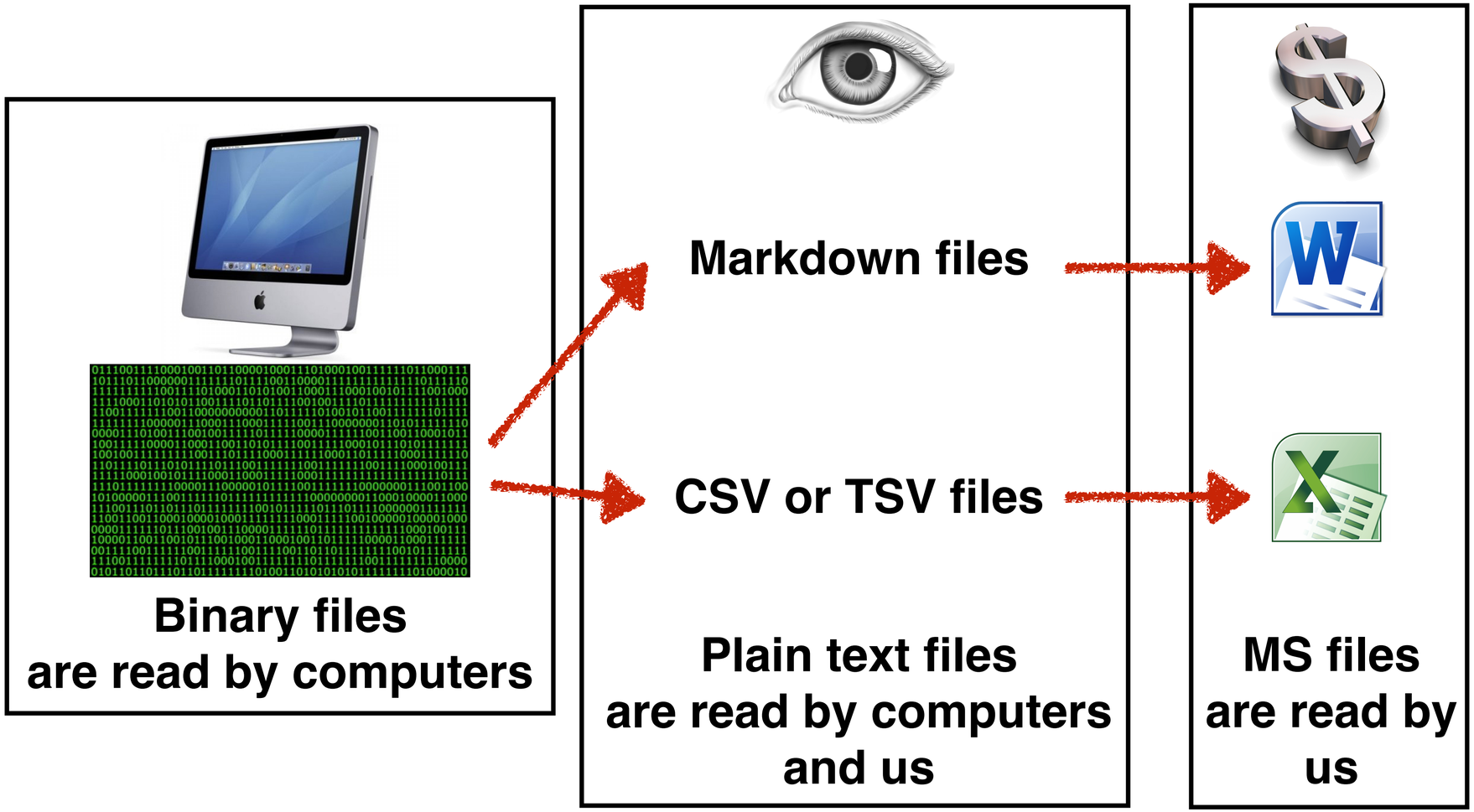
md = markdown

Markdown is the standard for software documentation.

Easily written, read, and translated.

Portable to HTML, PDF, etc.

We need files that the computer and we can read.



What about that readme.md file?

md = markdown

what you type

```
Heading
=======

Sub-heading
-----------

h3. Traditional html title

Paragraphs are separated
by a blank line.

Let 2 spaces at the end of a line to do a
line break

Text attributes *italic*,
**bold**, `monospace`.

A [link](http://example.com).
<<<   No space between ] and (  >>>

Shopping list:

  * apples
  * oranges
  * pears

Numbered list:

  1. apples
  2. oranges
  3. pears

The rain---not the reign---in
Spain.
```

what is rendered

Heading

Sub-heading

Traditional html title

Paragraphs are separated by a blank line.

Let 2 spaces at the end of a line to do a line break

Text attributes *italic*, **bold**, `monospace`.

A link.

Shopping list:

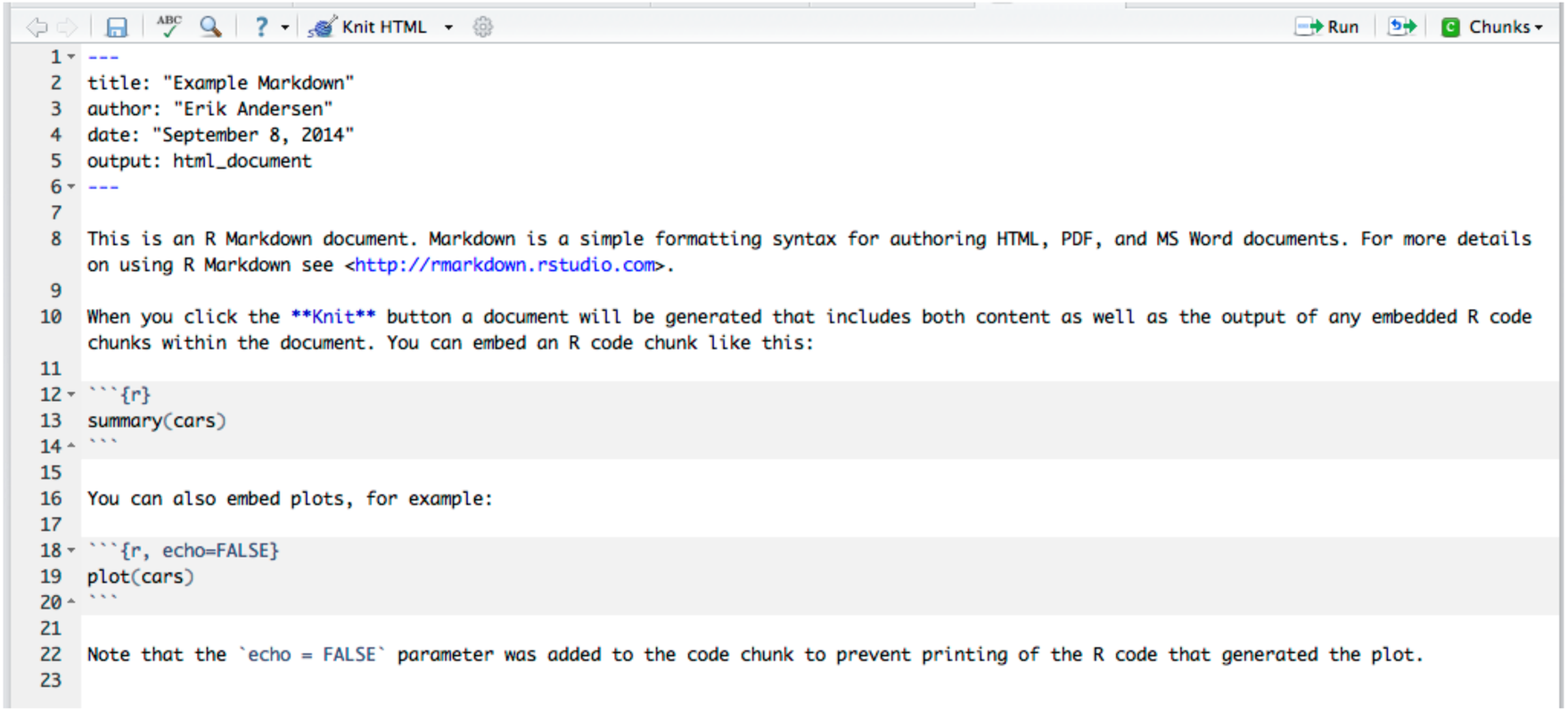
- apples
- oranges
- pears

Numbered list:

1. apples
2. oranges
3. pears

The rain—not the reign—in Spain.

RStudio can generate markdown reports



1. **Go to File, New File, R Markdown**
2. **Install the necessary packages**
4. **Knit an HTML report**

RStudio can generate markdown reports

1. Go to the class repo
2. Open the Class1_Activities.Rmd

In-class activity time

The most important part of data analysis is to *think* about your data

What do you want to test?

How will you show that conclusion?

Put the goal at the top of your markdown report