Conclusions

INFO 201

Joel Ross

Winter 2017

Project Fair

- **When:** Thur 03/09, 1:30pm
- **Where:** Mary Gates Hall Commons
- **Who**: Open to the public! Informatics faculty and students!
  - (see also INFO 200 Fairs Wed @ 10:30 and 1:30)
- **How**:
  - Set up a laptop (or three!) with your project
  - Present/pitch/demo your project to wandering guests!
  - (One or two group members can wander at a time)
  - Joel will be racing around for 2 minute demos

Today's Objectives

- Reflect on what we've done in this course
- Consider where to go from here

Where we started...

We call putting information in a variable **assigning** that value to the variable. We do this using the *assignment operator* `<-`. For example:

```
# Stores the number 7 into a variable called shoe.size
shoe.size <- 7
```

- *Notice:* variable name goes on the left, value goes on the right!

You can see what value (data) is inside a variable by either typing that variable name as a line of code, or by using R's built-in `print()` function (more on functions later):

```
print(shoe.size)
## [1] 7
```

What we achieved!

Diamond Viewer

Price (in dollars)

2,126 — 18,823

326 2,326 4,326 8,326 12,326 16,326 18,823

Carats

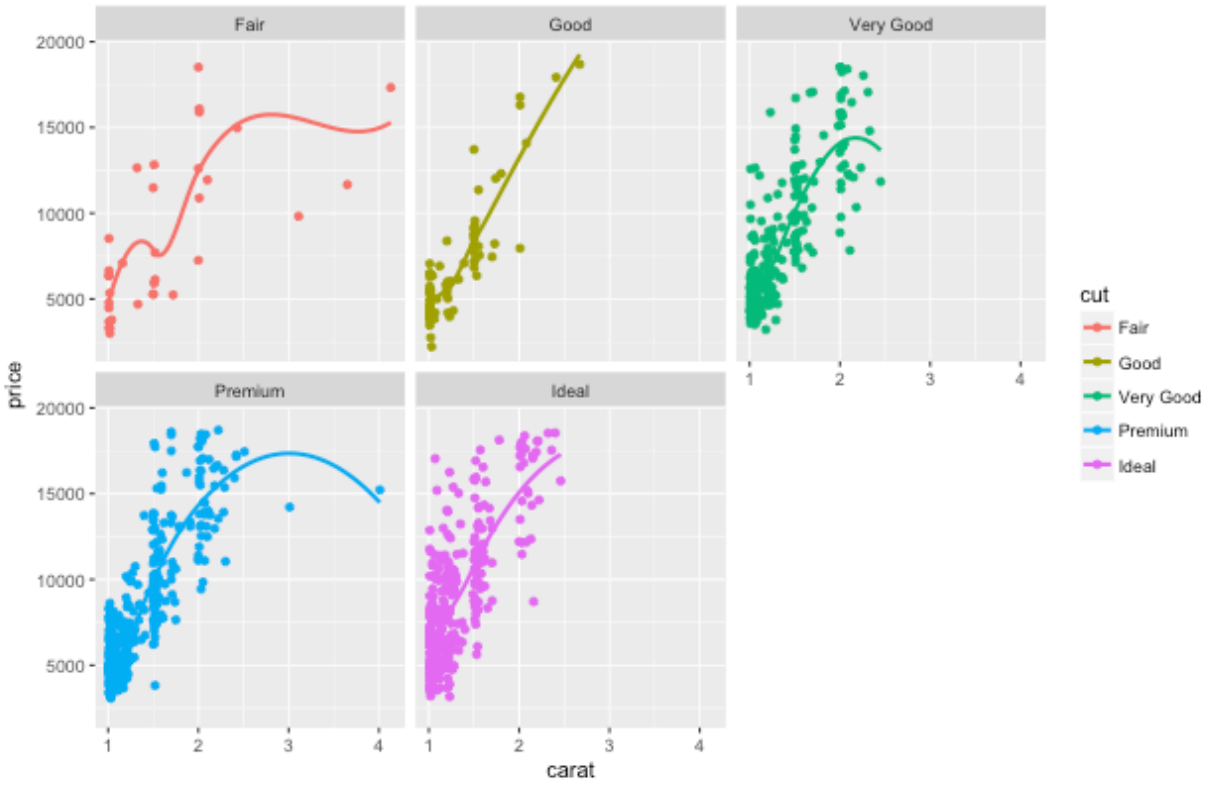
0.2 — 1 — 5.01

0.2 0.7 1.2 1.7 2.2 2.7 3.2 3.7 4.2 4.75.01

☑ **Show Trendline**

Facet By

cut



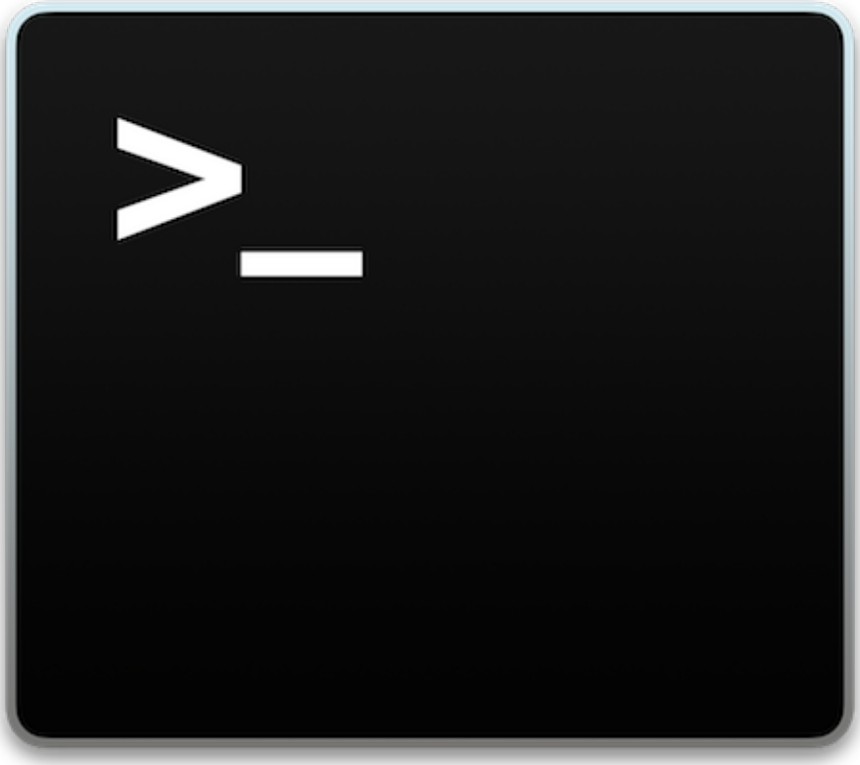
Show 25 entries Search:

carat	cut	color	clarity	depth	table	price	x	y	z
1.33	Ideal	J	VS1	62.5	58	6449	6.99	7.03	4.38
1.25	Ideal	G	VVS2	62.5	54	10636	6.88	6.93	4.31
2.01	Very	H	VS2	63.5	59	16677	7.92	7.86	5.01

Things We Learned

https://github.com/info201-w17?q=module

Command-Line and Version Control



You can use these everywhere!

Variables and Functions

```
SimulateGroups <- function(mean, sd, num_cars) {
  # Simulate 100 cars w/mean speed 50
  cars <- rnorm(n = num_cars, mean = mean, sd = sd)

  # A function to determine if a car is slower than all of the cars
  # in front of it (which createa a new group of cars **behind** it)
  SlowerThan <- function(index) {
    return(cars[index] < min(cars[1:index - 1]))
  }

  # Apply the slower_than function to all of the cars
  new.groups <- lapply(2:length(cars), SlowerThan)

  # Determine number of groups created
  groups <- length(new.groups[new.groups == TRUE]) + 1
  return(groups)
}
```

http://fivethirtyeight.com/features/how-many-cars-will-get-stuck-in-traffic/

Working with formal language and abstractions

Data Structures

Vectors

```
dogs <- c("Fido", "Spot", "Sparky")
numbers <- c(1,2,2,3,5,8,13,21,34)  # Fibonacci!
nineties <- 90:99  # 90  91  92  ...  99
```

Lists

```
person <- list(first.name = "Ada", job = "Programmer",
               salary = 78000, in.union = TRUE)
person$first.name  # person's first.name ("Ada")
person$salary  # person's salary (78000)
```

Data Frames

```
name <- c('Ada','Bob','Chris','Diya','Emma')
height <- 58:62
weight <- c(115, 117, 120, 123, 126)
my.data <- data.frame(name, height, weight)
```

Organizing and manipulating data

Debugging

1. What are you **trying** to achieve?

When I (as a user) do **Foo**, the program should do **Bar**

2. What is **actually** happening?

When I (as a user) do **Foo**, the program does **Baz** instead!

Systematically solving problems!

Data Wrangling

Grammar for Data Manipulation

Words (***verbs***) used to describe ways to manipulate data:

- **Select** the columns of interest
- **Filter** out irrelevant data to keep rows of interest
- **Mutate** a data set by adding more columns
- **Arrange** the rows in a data set
- **Summarize** the data (e.g., calculate the *mean*, *median*, *maximum*, etc).

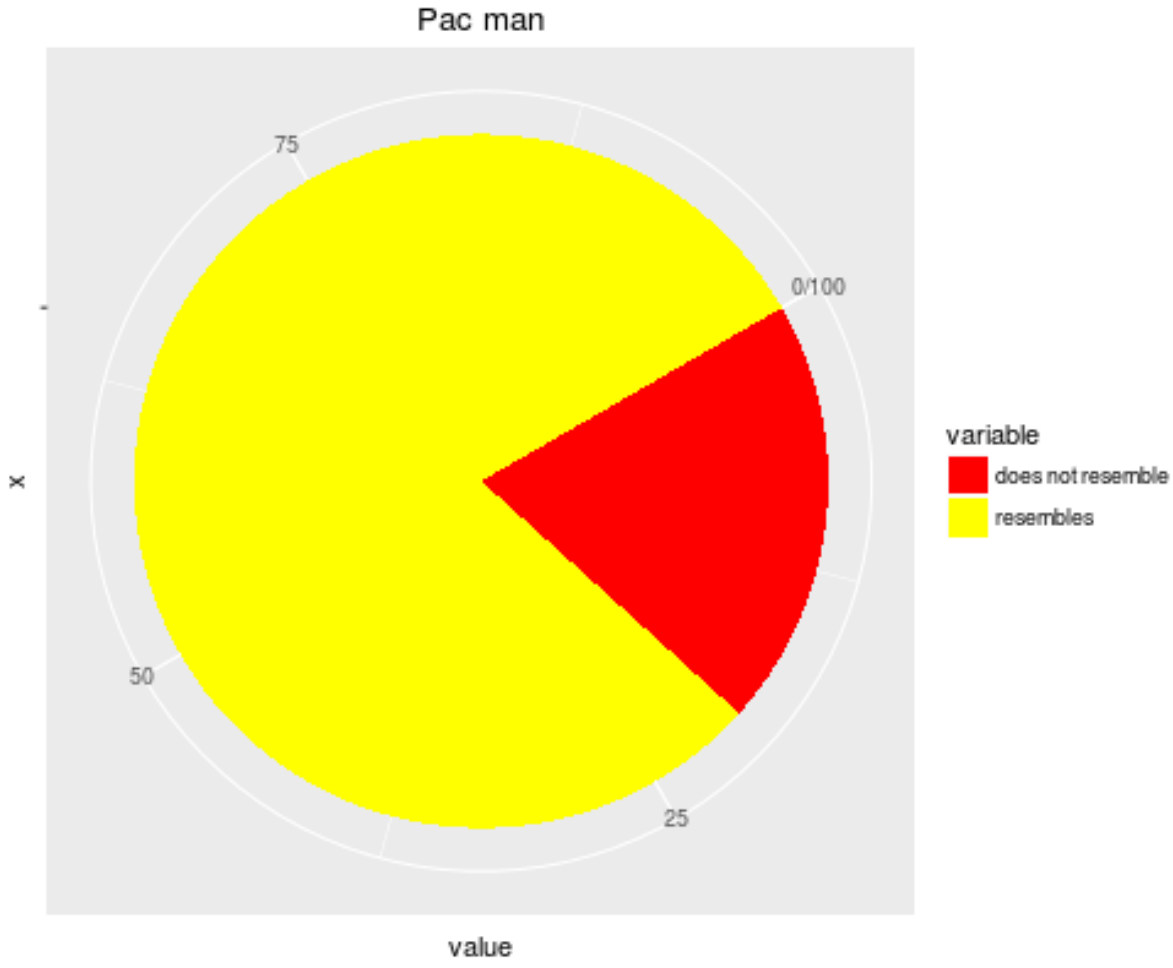
Converting data into information

Accessing Data

```
# a GitHub search for `dplyr`
uri <- "https://api.github.com/search/repositories?q=dplyr"
response <- GET(uri)
body.data <- fromJSON(content(response, "text"))  # extract and parse

# is it a data frame already?
is.data.frame(body.data)  # FALSE

# inspect the data!
str(body.data)  # view as a formatted string
names(body.data)  # view the tag names
  # looking at the JSON data itself (e.g., in the browser),
  # `items` is the key that contains the value we want

# extract the (useful) data
items <- body.data$items  # extract from the list
is.data.frame(items)  # TRUE; we can work with that!
```

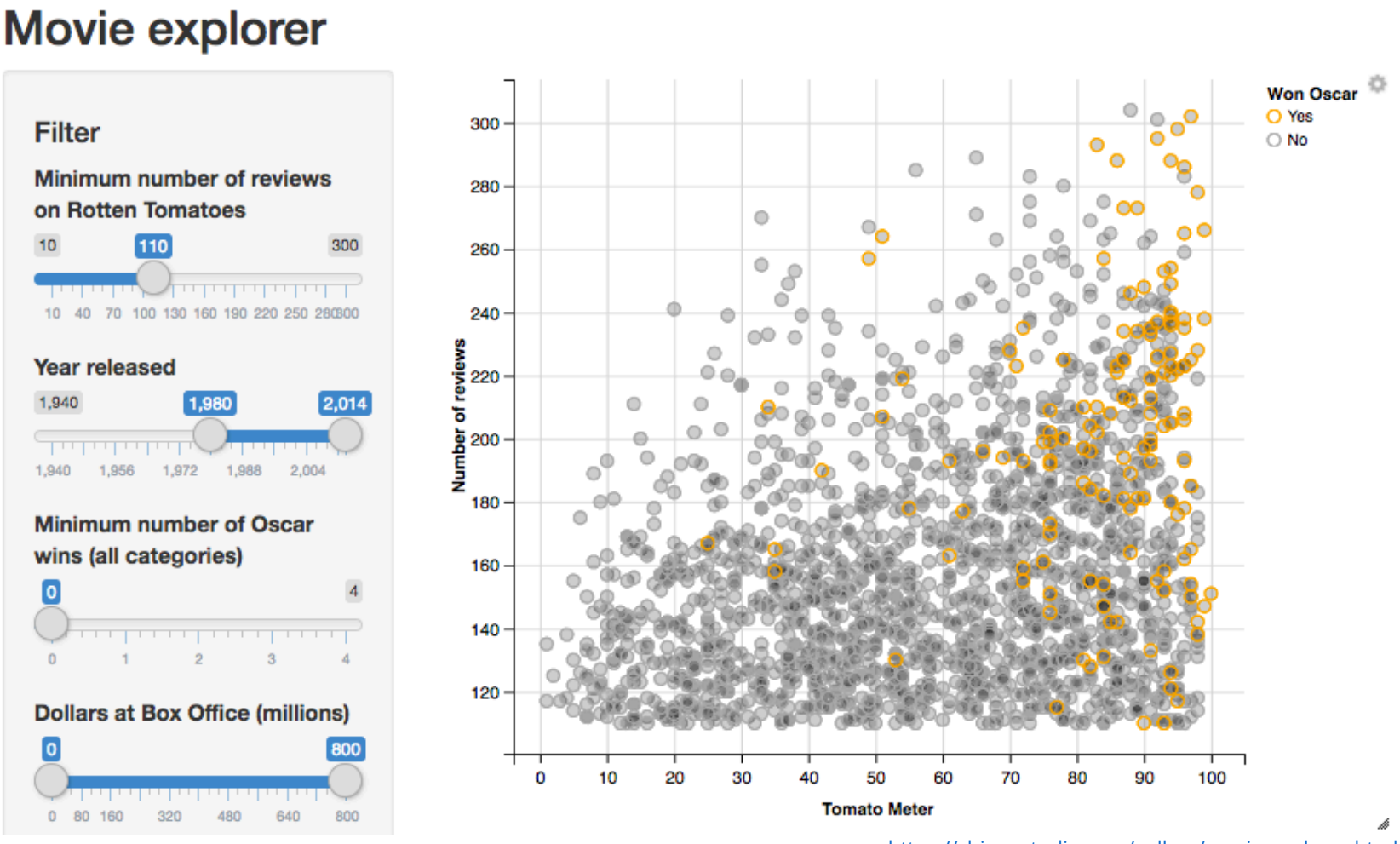
Acquiring and cleaning data to process

Data Visualization



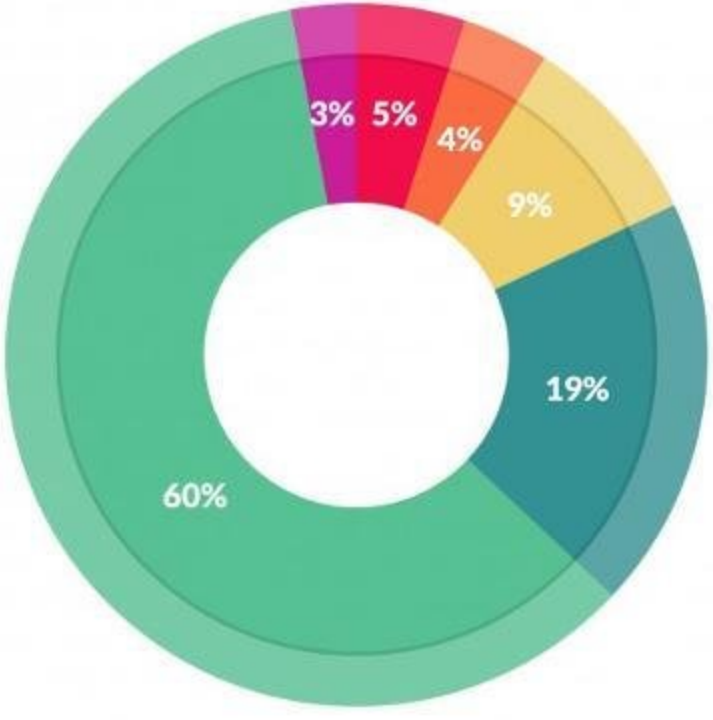
Presenting information effectively

Data Interaction



https://shiny.rstudio.com/gallery/movie-explorer.html

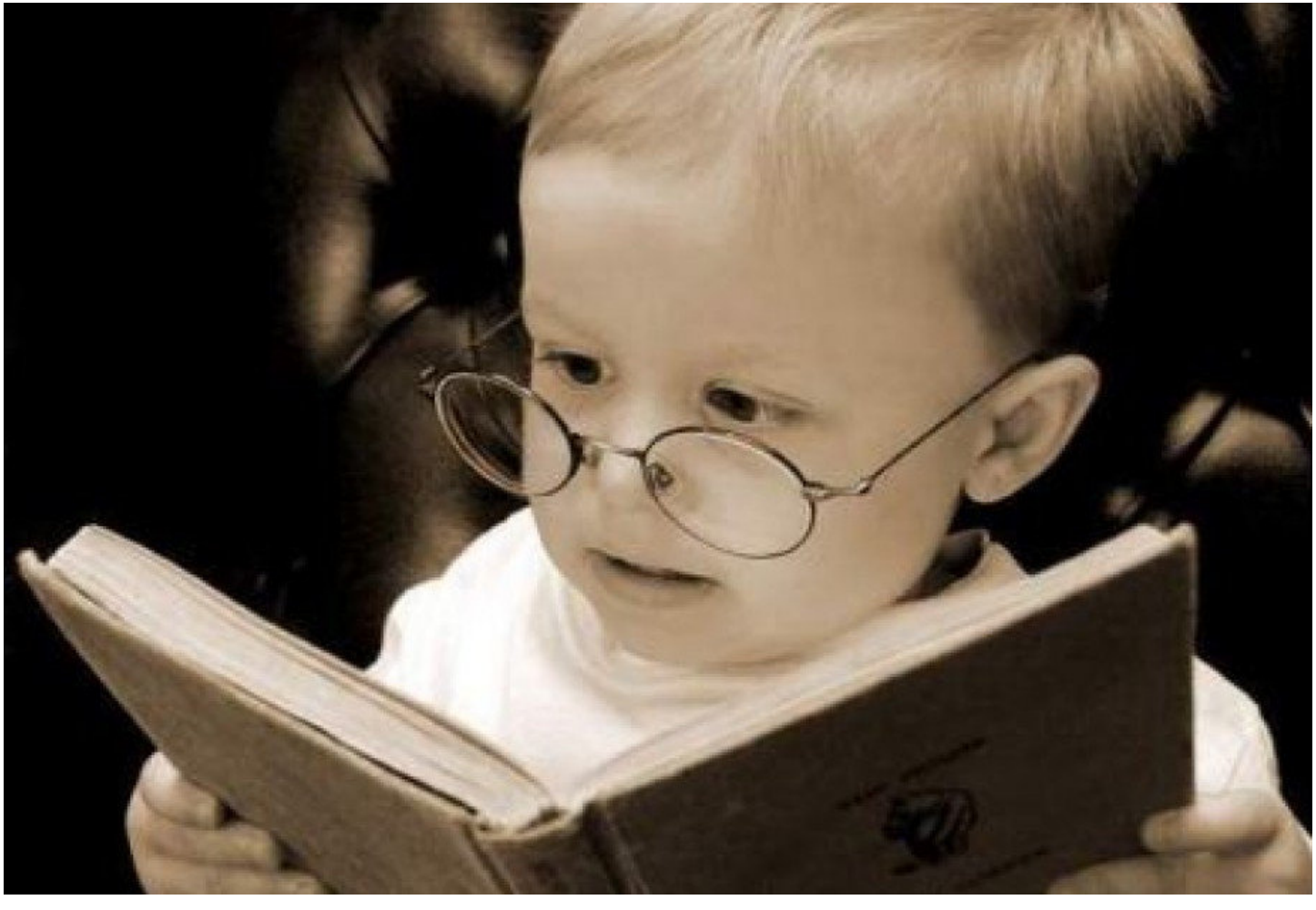
Creating interactive platforms



What data scientists spend the most time doing

- *Building training sets: 3%*
- *Cleaning and organizing data: 60%*
- *Collecting data sets; 19%*
- *Mining data for patterns: 9%*
- *Refining algorithms: 4%*
- *Other: 5%*

Learning On Your Own!



What's Next?

Evan Frawley

One more inspiring idea...

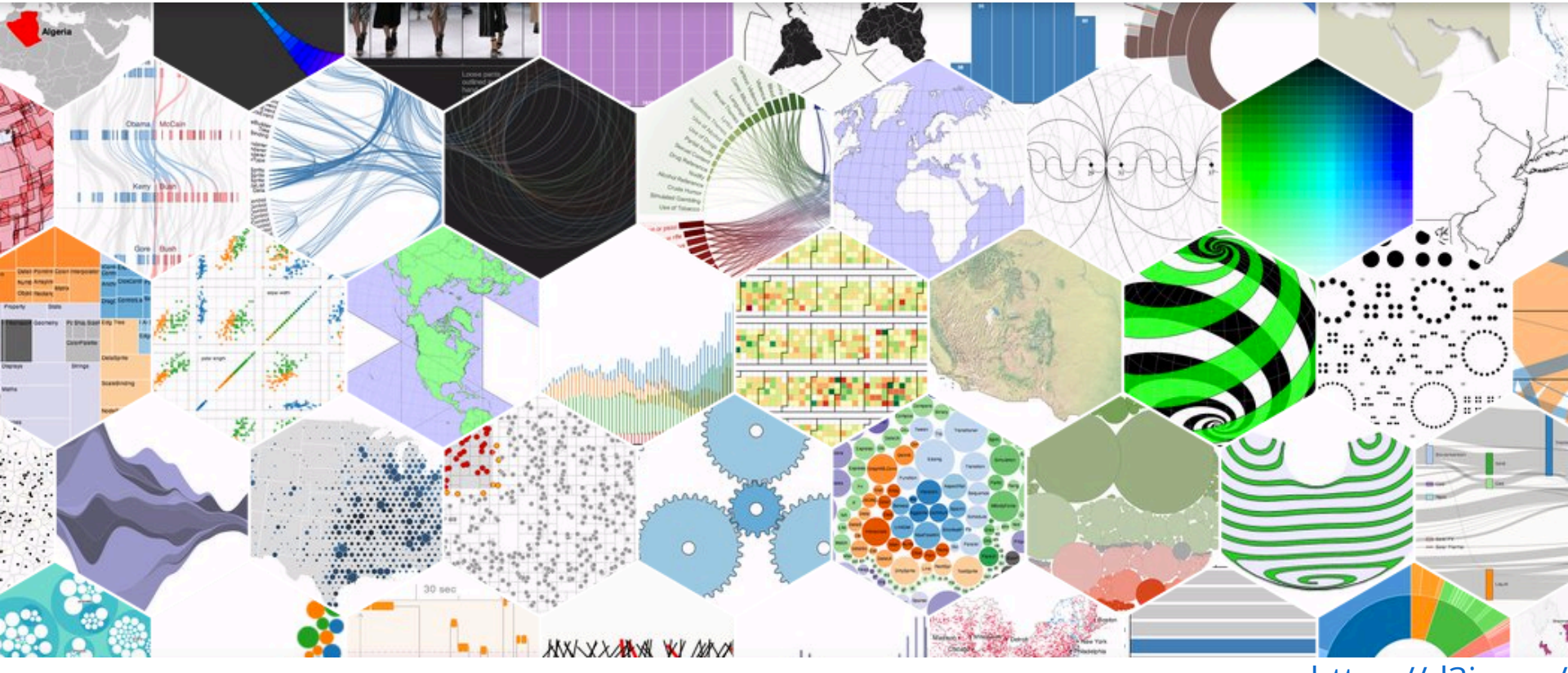

nEs
2016
data sketch|es
a Nadieh & Shirley collaboration


To Do List:
1.
2.
3.
4.
5.
6.
7.

Practice Programming

Study Design

Explore Visualizations

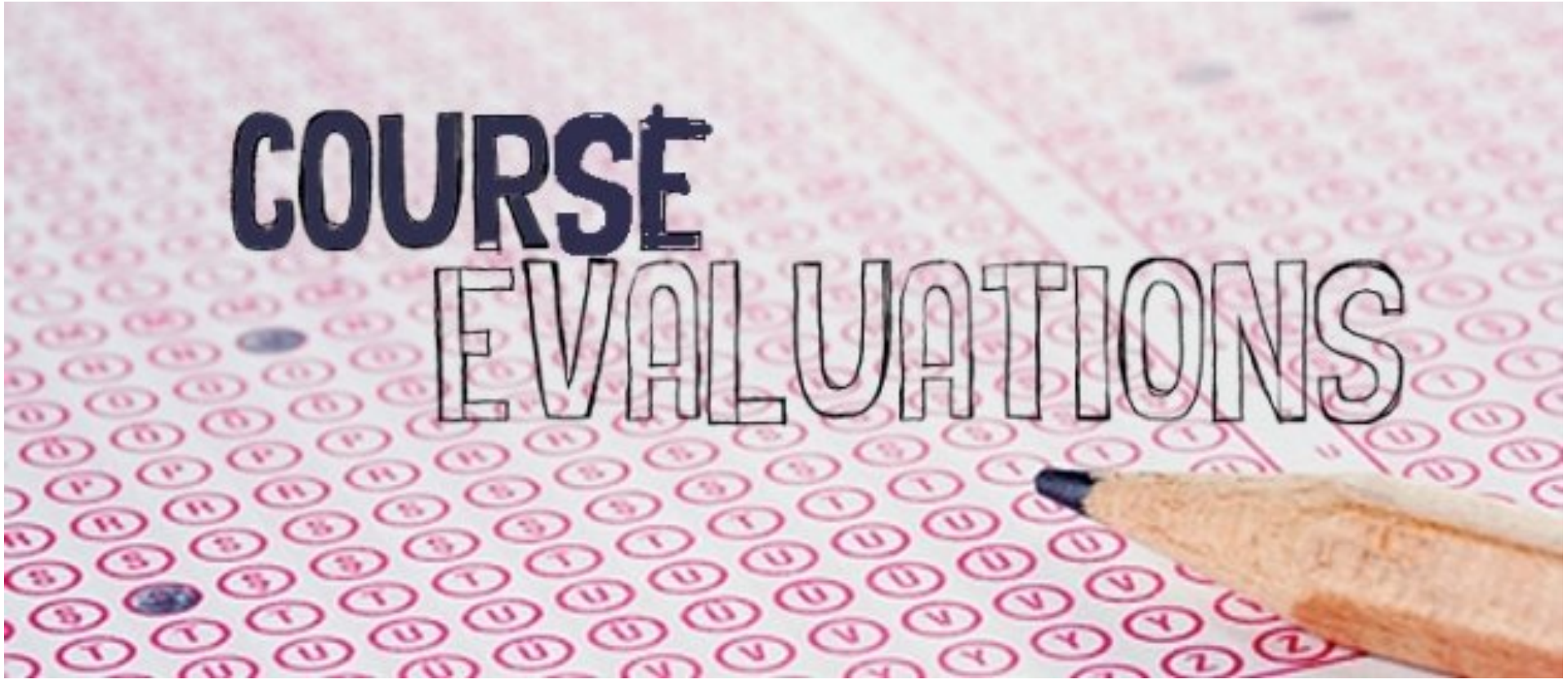


https://d3js.org/

Bring these skills to a field of your interest!


KEEP
CALM
AND MAKE THE WORLD
A BETTER
PLACE

Thank you to the TAs!



Check your email

DO THIS ASAP!!

Action Items!

- **Course Evaluation**
- Project Fair **Thursday** (show up early!)
  - Submit Project on Canvas
  - Fill out Peer Evaluation