# Wrap up

INFO 201

## Today’s Outline

Nuts and bolts

Reflect on what we've done

Consider how to move forward

## Nuts and Bolts

## Deliverables

Final projects: due **next Thursday night (12/8)** at midnight

Peer evaluations: due **a week from Monday (12/12)**  at midnight

Course evaluations: **please do this** when you're emailed about them

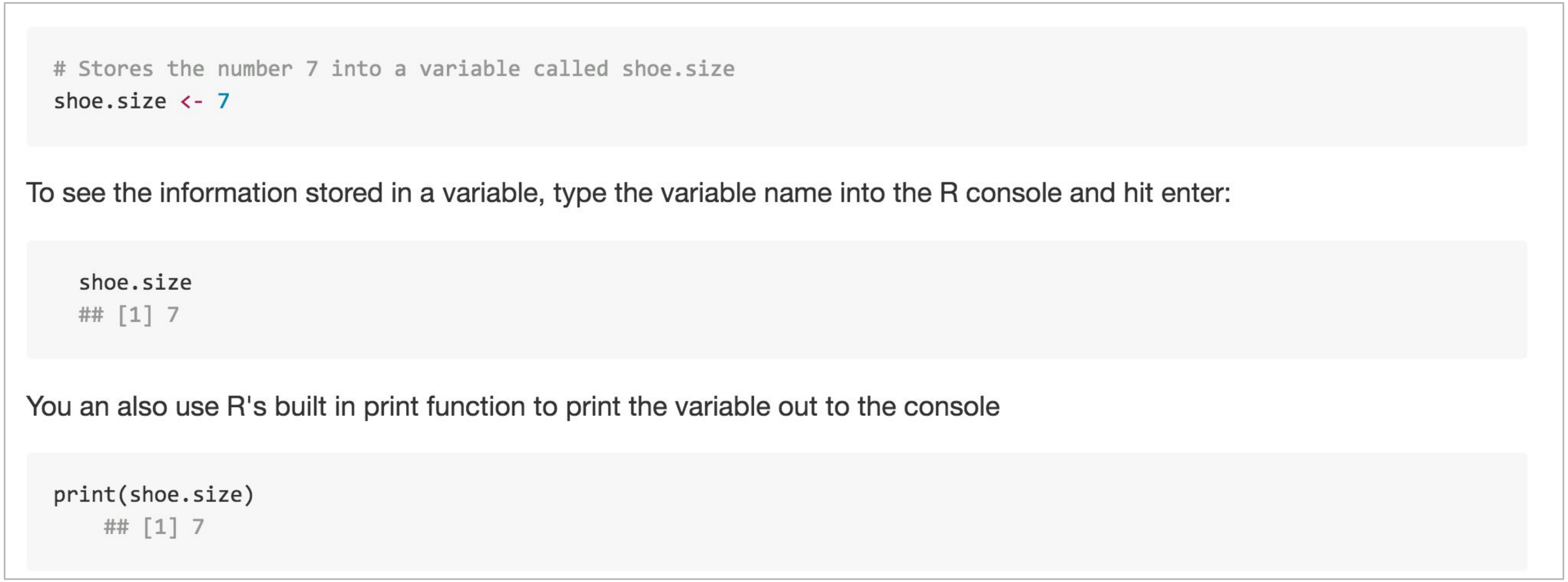
## Class/lab next week

Class time will be an open working session

- Earn a participation point by coming to class once (you can come both times!)

Lab: show the current status of your project (for a participation point)

## Looking back…



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

To see the information stored in a variable, type the variable name into the R console and hit enter:

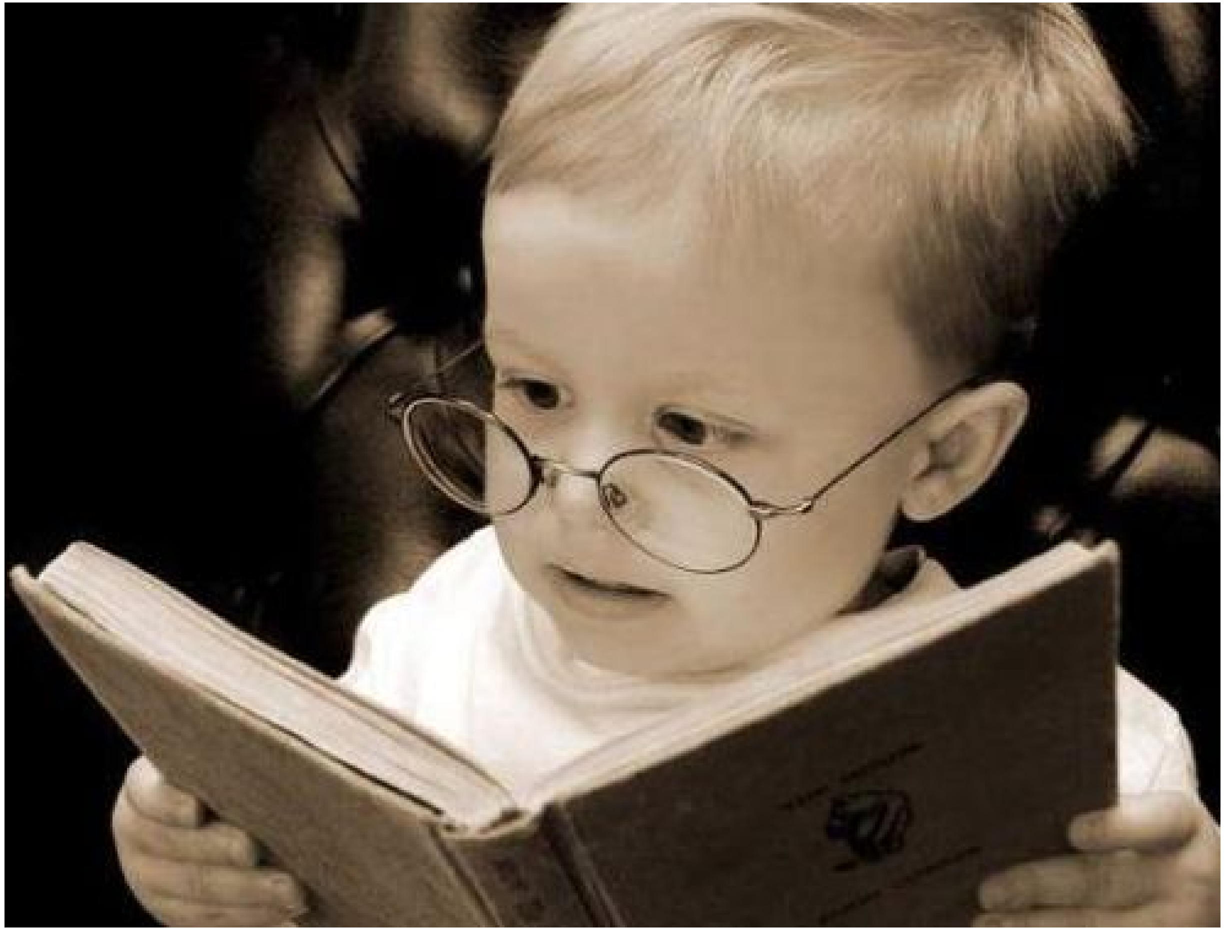
```
  shoe.size
  ## [1] 7
```

You an also use R's built in print function to print the variable out to the console

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

You've come a long way...

You'll use version control in every project moving forward (and if not...)



You've learned how to teach yourself

## Looking forward…

There is a very long, straight highway with some number of cars (N) placed somewhere along it, randomly. The highway is only one lane, so the cars can't pass each other. Each car is going in the same direction, and each driver has a distinct positive speed at which she prefers to travel. Each preferred speed is chosen at random. Each driver travels at her preferred speed unless she gets stuck behind a slower car, in which case she remains stuck behind the slower car. On average, how many groups of cars will eventually form? (A group is one or more cars traveling at the same speed.)

For example, if the car in the very front happens to be slowest, there will be exactly one group — everybody will eventually pile up behind the slowpoke. If the cars happen to end up in order, fastest to slowest, there will be N groups — no car ever gets stuck behind a slower car.

```r
# Simulate 100 cars w/mean speed 50
cars <- rnorm(n = 100, mean = 50, sd = 5)

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

# Apply the SlowerThan 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
```

## An approach in R

```r
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)
}
```

## Wrap it in a function

```r
RepeatSimulation <- function(num_sims = 10, mean = 50, sd = 5, num_cars = 100) {
  # Create an empty vector to store your results
  results <- vector()

  # Run your simulation 100 times, and track your results
  for(i in 1:num_sims) {
    results <- c(results, SimulateGroups(mean, sd, num_cars))
  }
  # Work with your results
  return(hist(results))
}
```

Run it a bunch of times...

```
# Create UI
shinyUI(fluidPage(
  # UI for the traffic simulation
  titlePanel('Traffic Simulation'),

  # Controls
  sidebarLayout(
    sidebarPanel(
      sliderInput("num_cars", "Number of Cars:",
                  min = 10, max = 1000, value = 100, step = 10),
      sliderInput("num_sims", "Iterations of Simulation",
                  min = 10, max = 1000, value = 100, step= 10),
      sliderInput("speed", "Average Speed",
                  min = 10, max = 150, value = 40, step= 5),
    ),
    # Render plot
    mainPanel(
      plotOutput("histogram")
    )
  )
))
```

## Make it Shiny (ui.R)

```r
# Run the traffic simulation
source('traffic_sim.R')
shinyServer(function(input, output) {
  output$histogram <- renderPlot({
    repeat_simulation(
        num_sims = input$num_sims,
        mean = input$speed,
        sd = input$deviation,
        num_cars = input$num_cars
    )
  })
})
```

## Make it Shiny (server.R)

Module-15 demo-2

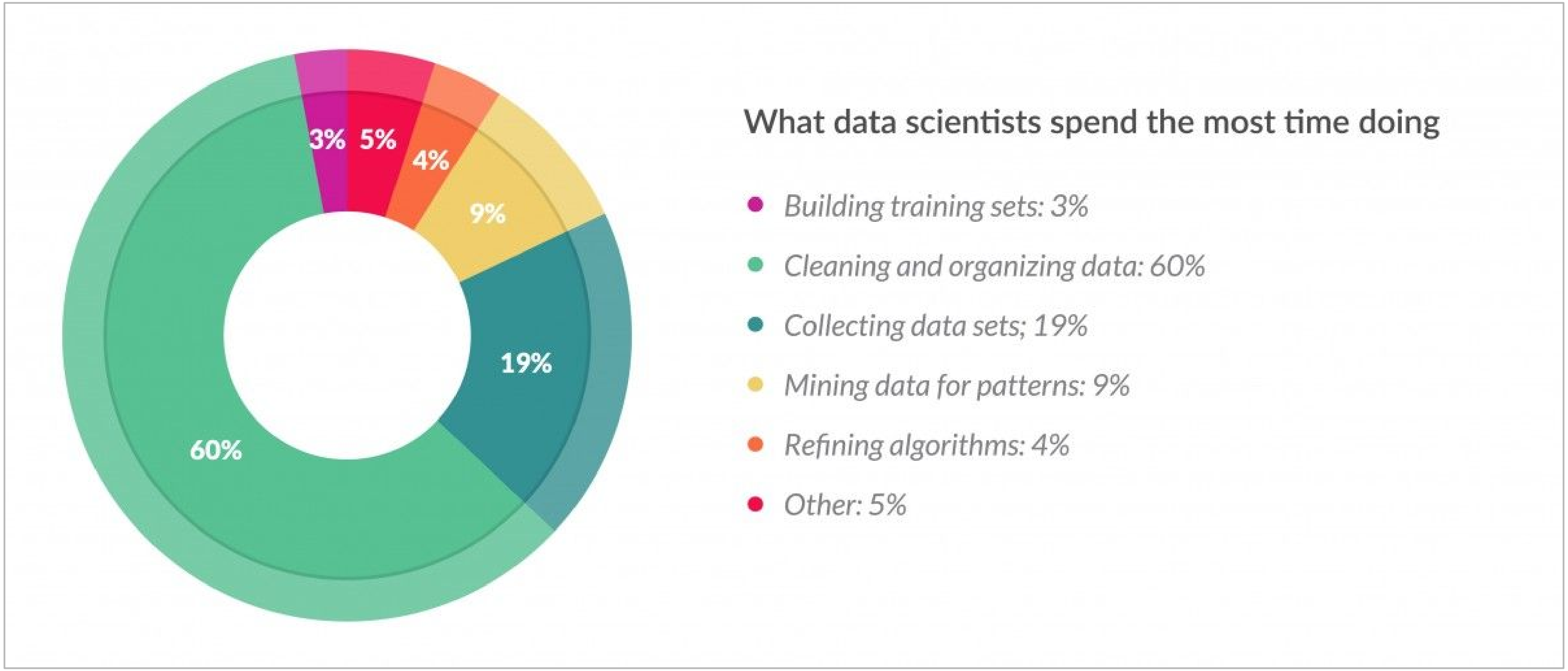
## Riddle links

Riddle

Answer (scroll down)

More involved response (app, code)

*I took this class to solve riddles…?*



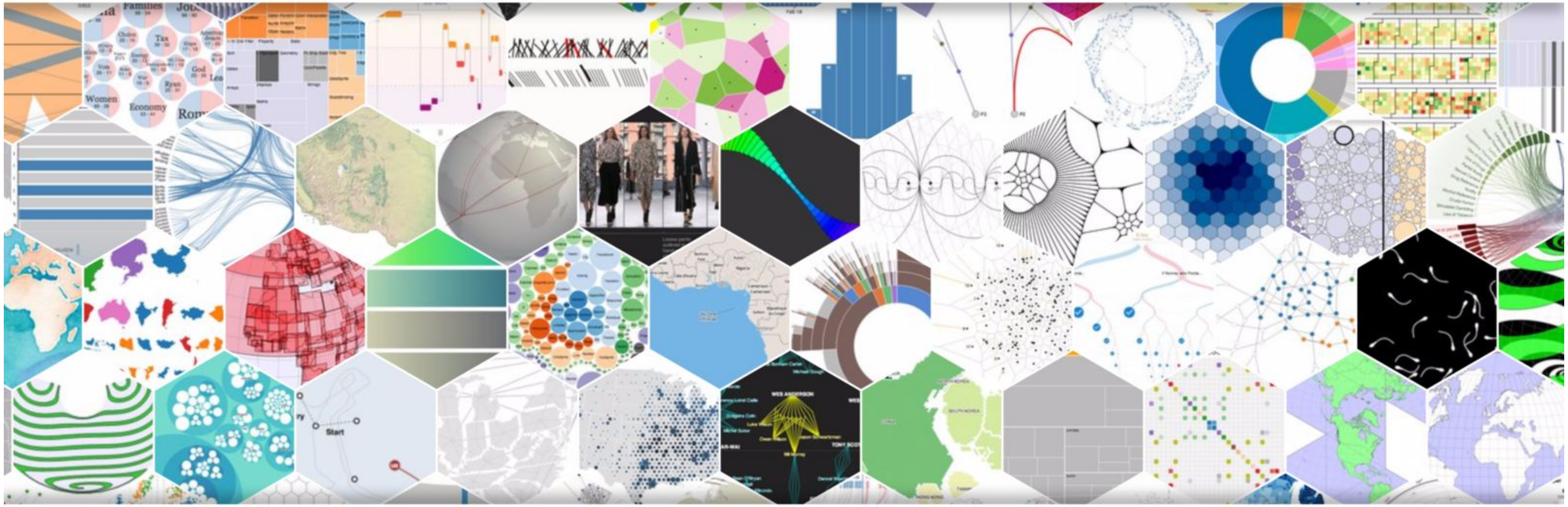
## Need to know foundational skills

## How Computer Scientists Are Using Twitter to Predict Gentrification

Cambridge researchers have created a way to predict a neighborhood's fortunes in coming years by analyzing social media data

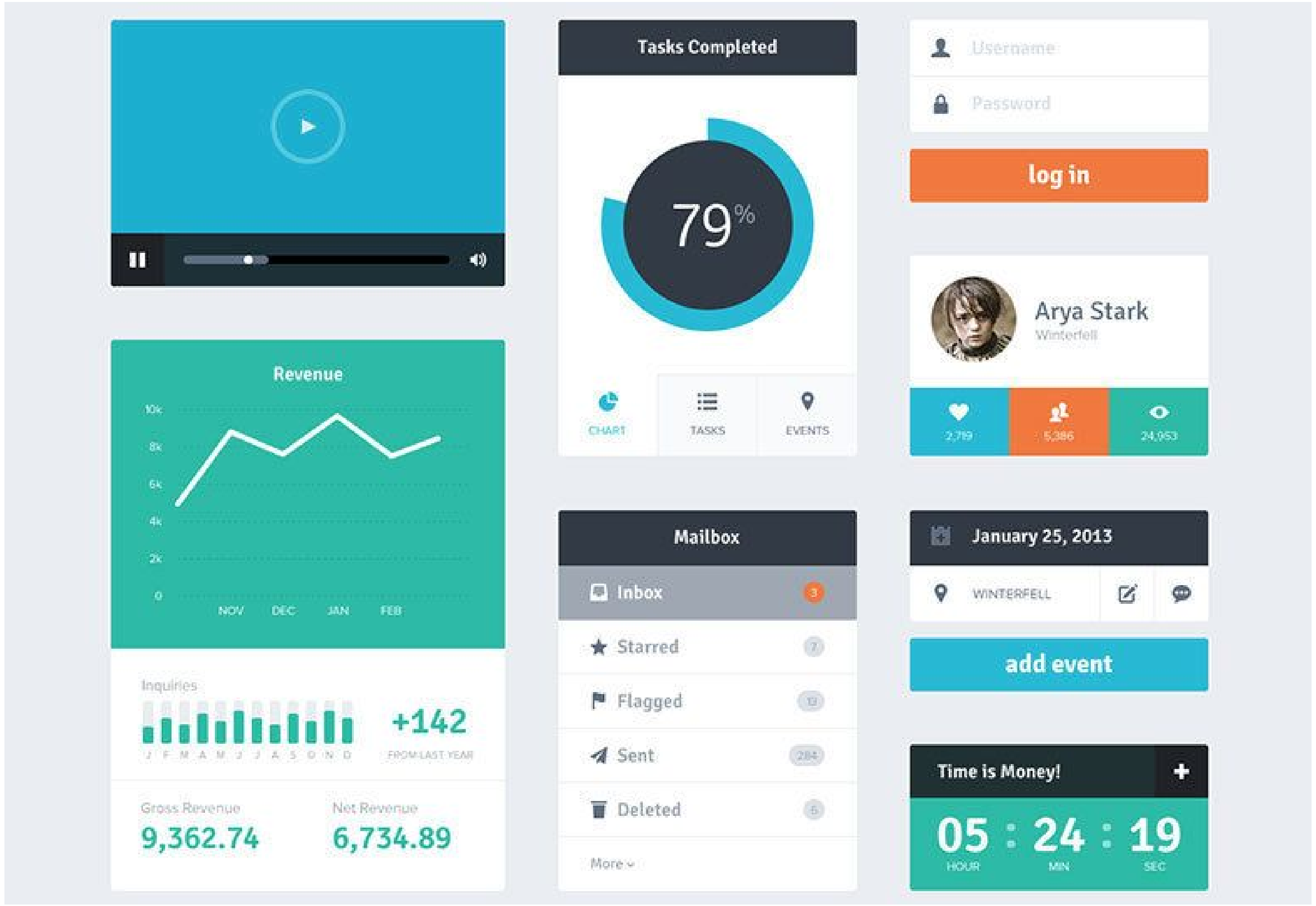
Many pertinent / interesting questions to be asked

*We are building a **program in R** that will be used to accurately perform statistical analysis on brain network data. The end goal here is to **detect Alzheimer's**. Skills from the course that I use include **collaborative coding** techniques using **Git** and BitBucket, reading **R documentation**, and **structuring code**. The project will eventually require me to build an **interactive interface using R and Shiny.***



Expand your visualization programming skills

Expand your analytical skills



Study design and design thinking

Bring these skills to a field of your interest!

Be a
NICE
human

## Thank you to the TAs!

## Upcoming

Final projects: due **next Thursday night (12/8)** at midnight

Peer evaluations: due **a week from Monday (12/12)**  at midnight

Course evaluations: **please do this** when you're emailed about them

Thank you.