

431 Class 03

`thomaseLove.github.io/431`

2020-09-01

PUBLIC SERVICE ANNOUNCEMENT:

OUR DIFFERENT WAYS OF WRITING DATES AS NUMBERS CAN LEAD TO ONLINE CONFUSION. THAT'S WHY IN 1988 ISO SET A GLOBAL STANDARD NUMERIC DATE FORMAT.

THIS IS *THE* CORRECT WAY TO WRITE NUMERIC DATES:

2013-02-27


THE FOLLOWING FORMATS ARE THEREFORE DISCOURAGED:

02/27/2013 02/27/13 27/02/2013 27/02/13

20130227 2013.02.27 27.02.13 27-02-13

27.2.13 2013. II. 27. $27\frac{1}{2}$ -13 2013.158904109

MMXIII-II-XXVII MMXIII $\frac{LVII}{CCCLXV}$ 1330300800

$((3+3) \times ((111+1) - 1) \times 3 / 3 - 1 / 3^3)$ ~~2013~~  hiss

10/11011/1101 02/27/20/13 $\begin{array}{cccc} 2 & 3 & 1 & 4 \\ 0 & 1 & 2 & 3 & 7 \\ 5 & 6 & 7 & 8 \end{array}$



Michael Donohoe ✓

@donohoe

Follow



Comprehensive map of all countries in the world that use the MMDDYYYY format



5:29 PM - 11 May 2015

Today's Agenda

- 1 R, RStudio, R Packages and R Markdown
- 2 “Live” Demo: The “Short” Survey

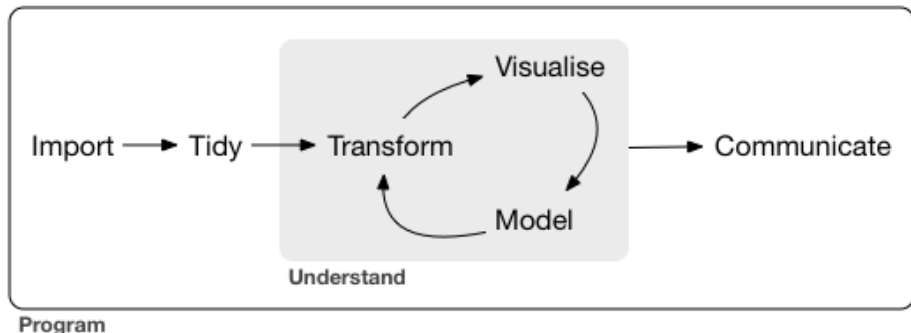
Everything R

- ➊ **R** is a computer language designed primarily for statistical computing and graphics. We use R to make sense of data.
- ➋ **RStudio** is an integrated development environment (IDE) for R. It includes a console, editor and tools for plotting, history, debugging and workspace management. We use RStudio to control our R experience.
- ➌ **R Packages** are collections of functions and code to help expand what base R can do. A key set of packages for doing data science in a coherent and enjoyable way are collectively known as the tidyverse.
 - We **install** packages (and occasionally update them) on our computer as if they were apps on our phone.
 - We then **load** packages within our R code to use those functions in our work.
- ➍ **R Markdown** is a file format to help us make dynamic documents with R. An R Markdown file ends with .Rmd and is an easy-to-write plain text format containing chunks of embedded R code. Everything we'll build in 431, including our reports, labs, presentation slides, etc., will come from R Markdown.

Chatfield's Six Rules for Data Analysis

- 1 Do not attempt to analyze the data until you understand what is being measured and why.
- 2 Find out how the data were collected.
- 3 Look at the structure of the data.
- 4 Carefully examine the data in an exploratory way, before attempting a more sophisticated analysis.
- 5 Use your common sense at all times.
- 6 Report the results in a clear, self-explanatory way.

Chatfield, Chris (1996) *Problem Solving: A Statistician's Guide*, 2nd ed.



What We'll Do In The Live Demo

- 1 Create a directory called `431-class-03-demo-live` on our computer.
- 2 Download the data and `431-r-template.Rmd` files to that folder from Github.
- 3 Open RStudio (we'll assume a successful installation) and briefly tour the four main windows.
- 4 Start a new R Project to do our work, linked to our chosen directory.
- 5 Use the template to start our R Markdown file.
- 6 Write code to do things in R with the data.
- 7 Write in English to explain the analyses that we're doing.
- 8 "Knit" together the R Markdown file to produce an attractive result in HTML.
- 9 Share the HTML result so that we can all see it.

A Worked “Short” Survey Analysis

(431_class-03-demo-full)

We have updated data on the site in a file called `surveyday1_2020.csv`.

`431_class-03-demo-full` R Markdown file, used to build HTML (and PDF) results.

- Key verbs in the tidyverse for data wrangling
 - `select`, `filter`, `count`, `arrange`, `mutate`, `group_by`, `summarize`
- Visualizing a single quantitative variable
- Comparing a distribution of a quantity within groups
 - Faceted histogram
 - Comparison boxplot
- Obtaining numerical summaries
- Scatterplots to describe associations

R Studio Primers

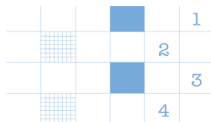
Learn data science basics with the interactive tutorials below.

The Basics



Start here to learn the skills that you will rely on in every analysis (and every primer that follows): how to inspect, visualize, subset, and transform your data, as well as how to run code.

Work with Data



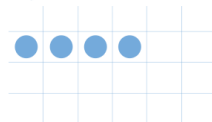
Learn the most important data handling skills in R: how to extract values from a table, subset tables, calculate summary statistics, and derive new variables.

Visualize Data



Learn how to use ggplot2 to make any type of plot with your data. Then learn the best ways to visualize patterns within values and relationships between variables.

Tidy Your Data



Unlock the tidyverse by learning how to make and use tidy data, the data format designed for R.

To The Live Demo!



Numerical quantities focus on expected values, graphical summaries on unexpected values.

-- John **Tukey**

Suppose we start from here...

```
library(magrittr); library(tidyverse)
```

```
-- Attaching packages -----
```

```
v ggplot2 3.3.2      v purrr  0.3.4
v tibble  3.0.3      v dplyr   1.0.2
v tidyr   1.1.2      v stringr 1.4.0
v readr   1.3.1      v forcats 0.5.0
```

```
-- Conflicts -----
```

```
x tidyr::extract()    masks magrittr::extract()
x dplyr::filter()     masks stats::filter()
x dplyr::lag()         masks stats::lag()
x purrr::set_names()  masks magrittr::set_names()
```

and here...

```
day1 <- read_csv("surveyday1_2020.csv")
```

Parsed with column specification:

```
cols(  
  .default = col_double(),  
  sex = col_character(),  
  glasses = col_character(),  
  english = col_character(),  
  favcolor = col_character()  
)
```

See `spec(...)` for full column specifications.

Analyzing the Survey Data

```
mosaic::favstats(~ height.in, data = day1)
```

Registered S3 method overwritten by 'mosaic':

```
method                                from  
fortify.SpatialPolygonsDataFrame ggplot2
```

min	Q1	median	Q3	max	mean	sd	n	missing
57	64	67	70	77.5	67.11905	3.7355	378	4

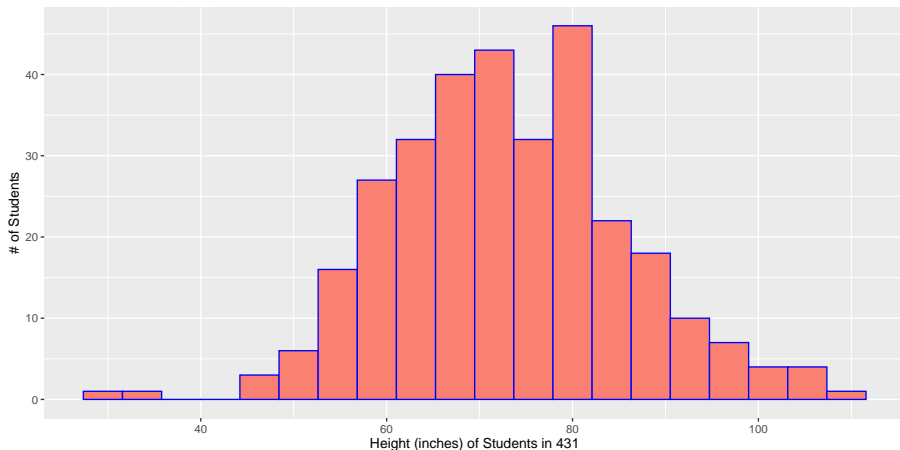
Analyzing the Survey Data - A little challenge

Can you reproduce the following. . .

A. That fill color is called *salmon*, I used 20 bins.

Heights of 378 students in 431

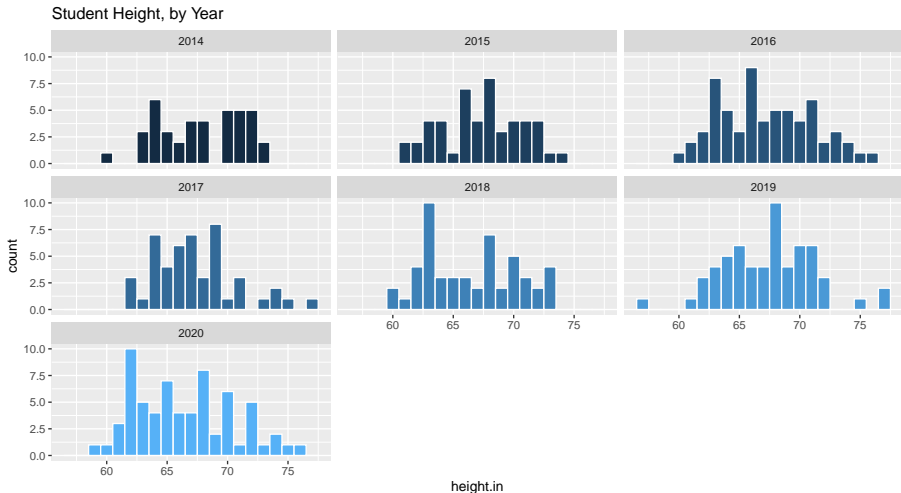
4 students had missing heights



Code for Part A.

```
day1 %>% filter(complete.cases(pulse)) %>%  
  ggplot(data = ., aes(x = pulse)) +  
  geom_histogram(bins = 20, col = "blue", fill = "salmon") +  
  labs(x = "Height (inches) of Students in 431",  
       y = "# of Students",  
       title = "Heights of 378 students in 431",  
       subtitle = "4 students had missing heights")
```

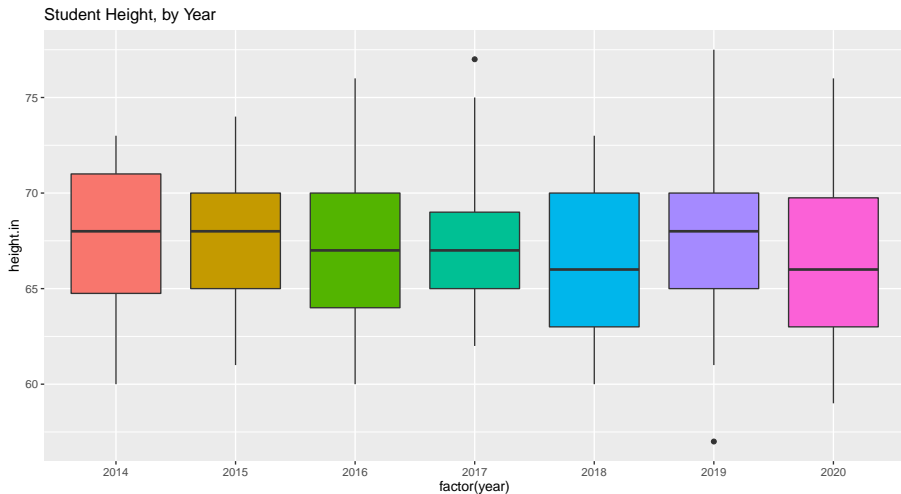
B. Histograms of Heights, Faceted by Year (binwidth = 1 inch)



Code for Plot B.

```
day1 %>% filter(complete.cases(height.in)) %>%  
  ggplot(data = ., aes(x = height.in, fill = year)) +  
  geom_histogram(binwidth = 1, col = "white") +  
  facet_wrap(~ year) +  
  guides(fill = FALSE) +  
  labs(title = "Student Height, by Year")
```

C. Boxplots of Age Guesses, by Year



Code for Plot C

```
day1 %>% filter(complete.cases(height.in)) %>%  
  ggplot(data = ., aes(x = factor(year), y = height.in,  
                        fill = factor(year))) +  
  geom_boxplot() +  
  guides(fill = FALSE) +  
  labs(title = "Student Height, by Year")
```

Table summarizing Student Heights, by Year

```
library(knitr)
mosaic::favstats(height.in ~ year, data = day1) %>%
  kable(digits = 1)
```

year	min	Q1	median	Q3	max	mean	sd	n	missing
2014	60	64.8	68	71.0	73.0	67.8	3.5	40	2
2015	61	65.0	68	70.0	74.0	67.3	3.3	49	0
2016	60	64.0	67	70.0	76.0	67.2	3.9	64	0
2017	62	65.0	67	69.0	77.0	67.4	3.5	48	0
2018	60	63.0	66	70.0	73.0	66.5	3.8	51	0
2019	57	65.0	68	70.0	77.5	67.4	3.8	60	1
2020	59	63.0	66	69.8	76.0	66.4	4.1	66	1