

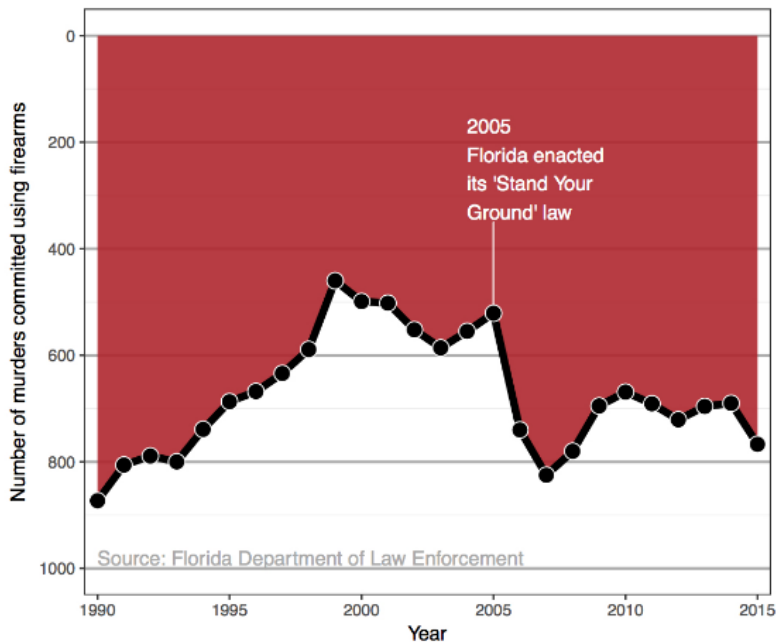
## Data Ethics, P-hacking, Reproducibility, and Data Journalism

- [Unethical misrepresentation of data and correcting for inflation](#)
- [P-hacking](#)
- [Data Journalism](#)
- [Filter and sort the data using dplyr package](#)
- [Write the data to a csv file](#)
- [Group\\_by and summarize the data](#)
- [Join two data frames \(different types of joins\)](#)
- [More info on merging datasets](#)
- [Introduction to Tableau – MoCo High School Districts Visualization](#)
- [Week 4 Homework Assignment](#)

“A good chart isn’t an illustration but a visual argument,” Alberto Cairo from *How Charts Lie*.

## Unethical Misrepresentation of Data

1. Take a minute to discover what is astonishingly wrong with the following graph from Florida Department of Law Enforcement:



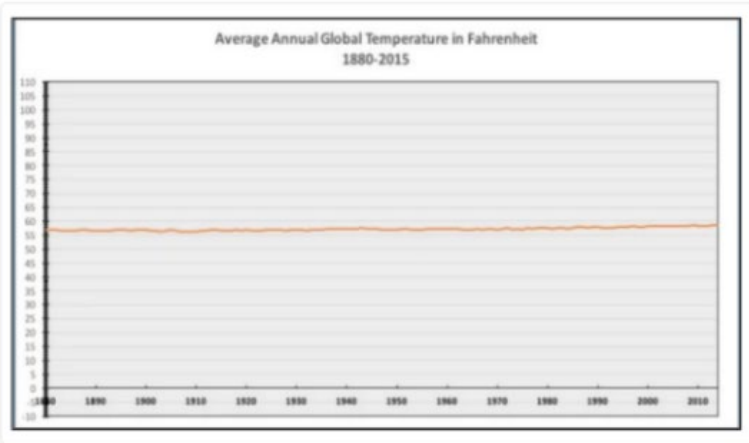
(Here is a clue: focus on the y-axis)

2. Here is another example. Again, look at the y-axis:

**NR** National Review  

The only [#climatechange](#) chart you need to see.  
[natl.re/wPKpro](http://natl.re/wPKpro)

(h/t [@powerlineUS](#))



RETWEETS 413 LIKES 318

1:36 PM - 14 Dec 2015

3. The following bar graph was presented by Georgia's Department of Health when reporting COVID19 data:

**Top 5 Counties with the Greatest Number of Confirmed COVID-19 (Reproduction of Figure)**

The chart below represents the most impacted counties over the past 15 days and the number of cases over time. The table below also represents the number of deaths and hospitalizations in each of those impacted counties.

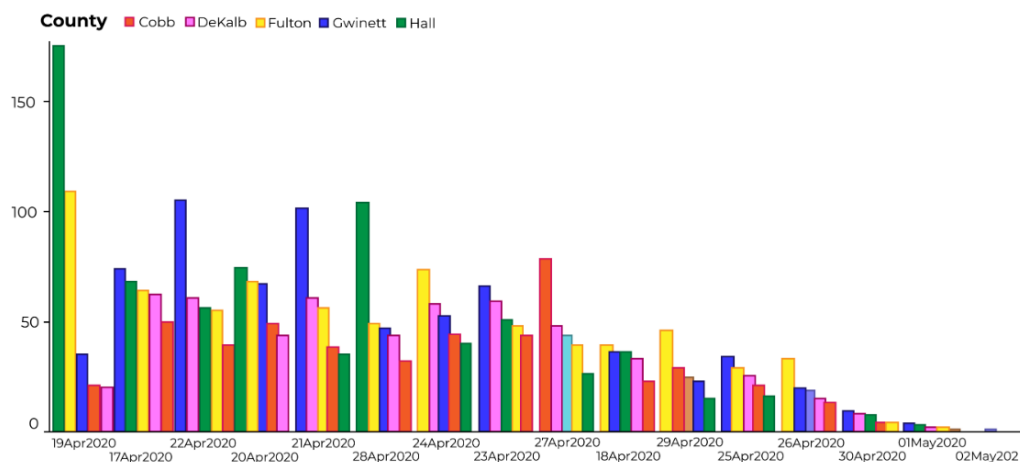
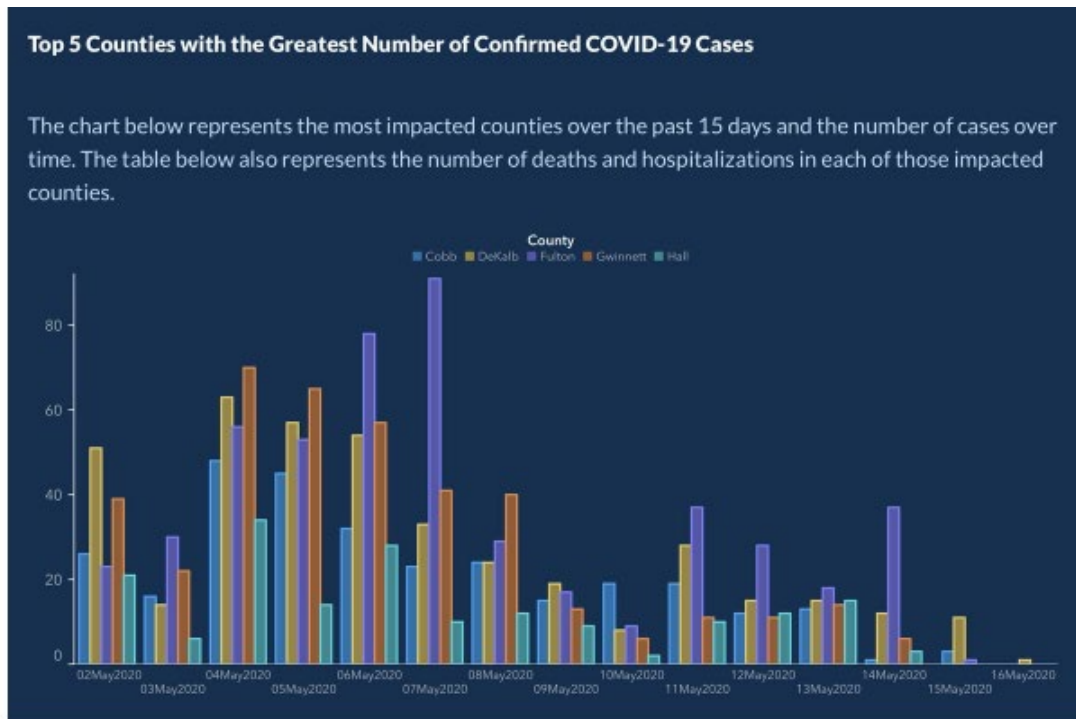


Figure 8.3: A recreation of a misleading display of confirmed COVID-19 cases in Georgia.

<https://dph.georgia.gov/covid-19-daily-status-report>

If you could not see very clearly, the bars are ordered by height, rather than by chronological time. If we reordered the bars by time, the graph would look like this:



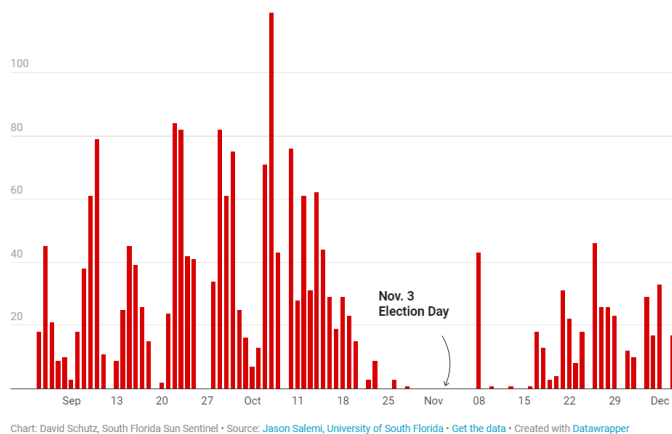
May 17, 2020, Georgia Department of Health, COVID-19 cases for 5 counties across time, sorted by time.

All examples were taken from **Modern Data Science with R** by Baumer, Kaplan, and Horton

- Then there was FLORIDA.... Here is the story about the gap in data just before the election in Nov. 2020: <https://www.sun-sentinel.com/coronavirus/fl-ne-ss-prem-covid-deaths-florida-election-20201216-f4kgezjf4rf75ppumt4omxfsxxy-story.html>

#### Gap in reporting backlog of COVID-19 deaths

It takes days or weeks for Florida to report a COVID-19 death. Ten days before the election, the state stopped reporting older deaths. The bars show daily reported deaths that were older than one month.



*With minor exceptions, Florida stopped including long-backlogged deaths in its daily counts on Oct. 24, 10 days before the Nov. 3 election, and resumed consistently including them on Nov. 17, two weeks after the election.*

## Fired Florida Data Scientist Launches A Coronavirus Dashboard of Her Own (NPR)

<https://www.npr.org/2020/06/14/876584284/fired-florida-data-scientist-launches-a-coronavirus-dashboard-of-her-own>



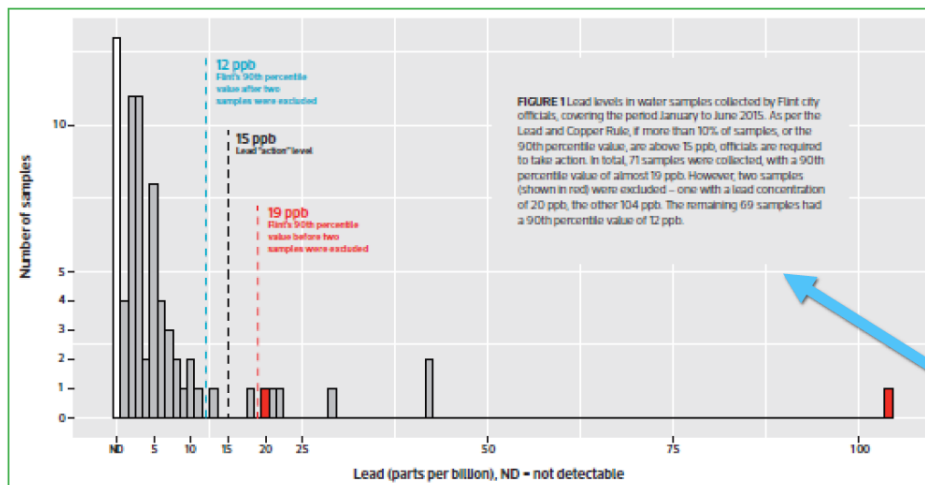
Rebekah Jones says she was fired after she refused to manipulate coronavirus data at the Florida Health Department. Now she has launched her own COVID-19 data portal for the state.

Screenshot by NPR/ Florida's Community Coronavirus Dashboard

### 5. Flint, MI – Lead in Water 2017

## Statistical Tricks to Manipulate Findings

A recent example of this occurred in Flint, MI



## The murky tale of Flint's deceptive water data

When children in Flint, Michigan showed signs of lead poisoning, residents rightly suspected their tap water was to blame. Activists during the fact for months, but the official water test data was misleading – so citizens fought back with statistics of their own. By Robert Langlois/Reuters



Lead levels in water samples collected by Flint city officials, covering the period January to June 2015. As per the Lead and Copper Rule, if more than 10% of samples, or the 90th percentile value, are above 15 ppb, officials are required to take action. In total, 71 samples were collected with a 90th percentile value of almost 19 ppb. However, 2 samples (in red) were excluded – one with a lead concentration of 20 ppb, the other 104 ppb. The remaining 69 samples had 90th percentile value of 12 ppb.

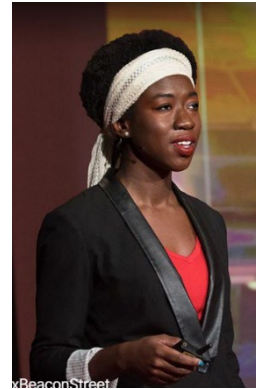
April 2017 SignificanceMagazine.com

## P-Hacking

P-hacking refers to a practice where researchers select the analysis that yields a pleasing result. The  $p$  refers to the p-value, which is essentially a measure of how surprising the results of a study would be if the effect you are looking for is not there (Wired, “We’re All ‘P-Hacking’ Now”, 11/26/19) Read more of this article as a part of your homework assignment.

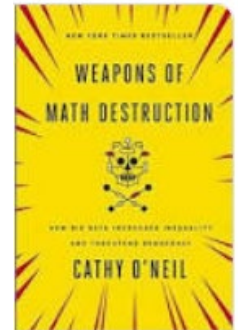
## Algorithmic Bias

Please watch this short TED talk by Dr. Joy Buolamwini, “How I am Fighting Bias in Algorithms”, where she discusses how human intelligence and human bias creates artificial intelligence algorithms, that are then innately biased.



[https://www.ted.com/talks/joy\\_buolamwini\\_how\\_i\\_m\\_fighting\\_bias\\_in\\_algorithms/discussion?.com&utm\\_medium=social&utm\\_campaign=tedspread](https://www.ted.com/talks/joy_buolamwini_how_i_m_fighting_bias_in_algorithms/discussion?.com&utm_medium=social&utm_campaign=tedspread)

She also refers to Cathy O’Neal’s book, “Weapons of Math Destruction,” which I highly recommend you read. It is a fast, engaging, enlightening, and somewhat frightening read.



## Data Journalism (Peter Aldhous)

The data we will use today ([found in this google drive](#))

Download the data for this session and save it to your datasets folder. It contains the following files, used in reporting [this story](#), which revealed that some of the doctors paid as “experts” by the drug company Pfizer had troubling disciplinary records:

- `pfizer.csv` Payments made by Pfizer to doctors across the United States in the second half on 2009. Contains the following variables:
  - `org_indiv` Full name of the doctor, or their organization.
  - `first_plus` Doctor’s first and middle names.
  - `first_name last_name` First and last names.
  - `city state` City and state.
  - `category of payment` Type of payment, which include Expert-led Forums, in which doctors lecture their peers on using Pfizer’s drugs, and ‘Professional Advising.
  - `cash` Value of payments made in cash.
  - `other` Value of payments made in-kind, for example purchase of meals.
  - `total value of payment`, whether cash or in-kind.
- `fda.csv` Data on warning letters sent to doctors by the U.S. Food and Drug Administration, because of problems in the way in which they ran clinical trials testing experimental treatments. Contains the following variables:
  - `name_last name_first name_middle` Doctor’s last, first, and middle names.
  - `issued` Date letter was sent.
  - `office` Office within the FDA that sent the letter.

# Load and view data

## Load data

Use the `read_csv` function. Copy the following code into your script and Run:

```
# load data of pfizer payments to doctors and warning letters sent by food and drug administration
pfizer <- read_csv("pfizer.csv")
fda <- read_csv("fda.csv")
```

Notice that the `Environment` now contains two objects, of the type `tbl_df`, a variety of the standard R object for holding tables of data, known as a **data frame**:

## Data 110 Data Journalism

Rachel Saidi

## Remember to set your working directory

Recall that you set the working directory to this folder by selecting from the top menu `Session>Set Working Directory>Choose Directory`. Then select the folder where you are keeping all your datasets for this class. By doing this, we can load the files in this directory without having to refer to the full path for their location, and anything we save will be written to this folder.

Once you set your working directory for where you will access your stored data

## Load the “tidyverse” then read in the data

```
library(tidyverse)
setwd("C:/Users/rsaidi/Dropbox/Rachel/MontColl/Datasets/Datasets")
pfizer <- read_csv("pfizer.csv")
fda <- read_csv("fda.csv")
```

Notice that the `Environment` now contains two objects, of the type `tbl_df`.

## Comment your code

Anything that appears on a line after `#` will be treated as a comment, and will be ignored when the code is run. Get into the habit of commenting your code: Don't trust yourself to remember what it does! Data journalism should ideally be fully documented and reproducible.

## Update your R packages

Each time you start R, it's a good idea to click on `Update` in the `Packages` panel to update all your installed packages to the latest versions. Installing a package makes it available to you, but to use it in any R session you

need to load it. You can do this by checking its box in the Packages tab. However, we will enter the following code into our script, then highlight these lines of code and run them:

## Manipulate the pfizer and fda data

Recall that the pfizer dataset contains information about Pfizer payments to doctors and warning letters sent by food and drug administration

### Examine the data

We can View data at any time by clicking on its table icon in the Environment tab in the Grid view.

Alternatively, you can use code we learned in the last unit - `head(data)`. Notice the variable names and types.

```
head(pfizer)

# A tibble: 6 x 10
  org_indiv      first~1 first~2 last_~3 city  state categ~4  cash other total
  <chr>         <chr>   <chr>   <chr>   <chr> <chr> <chr>   <dbl> <dbl> <dbl>
1 3-D MEDICAL SER~ STEVEN~ STEVEN DEITEL~ NEW ~ LA    Profes~ 2625    0 2625
2 AA DOCTORS, INC. AAKASH~ AAKASH AHUJA  PASO~ CA    Expert~ 1000    0 1000
3 ABBO, LILIAN MA~ LILIAN~ LILIAN ABBO   MIAMI FL    Busine~    0  448  448
4 ABBO, LILIAN MA~ LILIAN~ LILIAN ABBO   MIAMI FL    Meals    0  119  119
5 ABBO, LILIAN MA~ LILIAN~ LILIAN ABBO   MIAMI FL    Profes~ 1800    0 1800
6 ABDULLAH RAFFEE~ ABDULL~ ABDULL~ RAFFEE  FLINT MI    Expert~  750    0  750
# ... with abbreviated variable names 1: first_plus, 2: first_name,
# 3: last_name, 4: category
```

### Now view the fda data

```
head(fda)

# A tibble: 6 x 5
  name_last name_first name_middle issued      office
  <chr>     <chr>     <chr>     <chr>     <chr>
1 ADELGLASS JEFFREY    M.        5/25/1999 Center for Drug Evaluation and Re~
2 ADKINSON  N.         FRANKLIN  4/19/2000 Center for Biologics Evaluation a~
3 ALLEN     MARK      S.        1/28/2002 Center for Devices and Radiologic~
4 AMSTERDAM DANIEL   <NA>      11/17/2004 Center for Biologics Evaluation a~
5 AMSTUTZ   HARLAN    C.        7/19/2004 Center for Devices and Radiologic~
6 ANDERSON  C.        JOSEPH    2/25/2000 Center for Devices and Radiologic~
```

Notice that issued has been recognized as a Date variable. Other common data types include num, for numbers that may contain decimals and POSIXct for full date and time.

To specify an individual column use the name of the data frame and the column name, separated by \$. Determine the class for the variable “total”

```
class(pfizer$total)

[1] "numeric"
```



If you need to change the data type for any column, use the following functions:

- . as.character converts to a text string
- . as.numeric converts to a number
- . as.factor converts to a categorical variable
- . as.integer converts to an integer
- . as.Date converts to a date
- . as.POSIXct convets to a full date and time

(Conversions to full dates and times can get complicated, because of timezones.

The summary function will run a quick statistical summary of a data frame, calculating mean, median and quartile values for continuous variables:

```
summary(pfizer) # summary of pfizer data
```

org_indiv	first_plus	first_name	last_name
Length:10087	Length:10087	Length:10087	Length:10087
Class :character	Class :character	Class :character	Class :character
Mode :character	Mode :character	Mode :character	Mode :character
city	state	category	cash
Length:10087	Length:10087	Length:10087	Min. : 0
Class :character	Class :character	Class :character	1st Qu.: 0
Mode :character	Mode :character	Mode :character	Median : 0
			Mean : 3241
			3rd Qu.: 2000
			Max. :1185466
			NA's :1
other	total		
Min. : 0.0	Min. : 0		
1st Qu.: 0.0	1st Qu.: 191		
Median : 41.0	Median : 750		
Mean : 266.5	Mean : 3507		
3rd Qu.: 262.0	3rd Qu.: 2000		
Max. :27681.0	Max. :1185466		
NA's :3			

## Manipulate and analyze data

Now we will use dplyr to manipulate the data, using operations and functions:

- . Sort: Largest to smallest, oldest to newest, alphabetical etc.



- . select - Choose which columns to include.
- . filter - Filter the data.
- . arrange - Sort the data, by size for continuous variables, by date, or alphabetically.
- . group\_by - Group the data by a categorical variable.
- . mutate - Create new column(s) in the data, or change existing column(s).
- . rename - Rename column(s).
- . bind\_rows - Merge two data frames into one, combining data from columns with the same name.

## Summarize has special additional functions associated with it

. summarize - Summarize, or aggregate (for each group if following group\_by). Often used in conjunction with functions including:

- mean Calculate the mean, or average
- median Calculate the median
- max Find the maximum value
- min Find the minimum value
- sum Add all the values together
- n Count the number of records

There are also various functions to join data, which we will explore below.

## Explore the pfizer dataset

We will explore this dataset, filtering in many ways, in order to see if there are doctors who have behaved unethically by taking money from Pfizer Pharma and then we will join that dataset with the FDA dataset that reveals doctors who have received warnings for this unethical behavior.

## Filter and sort data

Filter and sort the data in specific ways. For each of the following examples, copy the code that follows into your script, and view the results. Notice how we create a new objects to hold the processed data.

Find doctors in California paid \$10,000 or more by Pfizer to run “Expert-Led Forums.”

```
# doctors in California who were paid $10,000 or more by Pfizer to run "Expert-Led Forums"
.  
ca_expert_10000 <- pfizer %>%  
  filter(state == "CA" & total >= 10000 & category == "Expert-Led Forums")
```

Notice the use of == to find values that match the specified text, >= for greater than or equal to, and the Boolean operator &.

## Add a sort to the end of the code to list the doctors in descending order by the payments received:

```
# doctors in California who were paid $10,000 or more by Pfizer to run "Expert-Led Forums"
ca_expert_10000 <- pfizer %>%
  filter(state == "CA" & total >= 10000 & category == "Expert-Led Forums") %>%
  arrange(desc(total))
```

If you arrange without the desc function, the sort will be from smallest to largest.

## Boolean Operators (&, |, ==, >, <, >=, <=, !=)

Notice the use of the | Boolean operator, and the brackets around that part of the query. This ensures that this part of the query is run first. See what happens if you exclude them.

```
# Find doctors in states other than California who were paid $10,000 or more by Pfizer to run "Expert-Led Forums."
not_ca_expert_10000 <- pfizer %>%
  filter(state != "CA" & total >= 10000 & category=="Expert-Led Forums") %>%
  arrange(desc(total))
```

## Notice the use of the != operator to exclude doctors in California.

```
# Find the 20 doctors across the four largest states (CA, TX, FL, NY) who were paid the most for professional advice.
ca_ny_tx_fl_prof_top20 <- pfizer %>%
  filter((state=="CA" | state == "NY" | state == "TX" | state == "FL") & category == "Professional Advising") %>%
  arrange(desc(total)) %>%
  head(6)
```

Notice the use of head, which grabs a defined number of rows from the start of a data frame. Here, it is crucial to run the sort first! See what happens if you change the order of the last two lines.

## Filter the data for all payments for running Expert-Led Forums or for Professional Advising, and arrange alphabetically by doctor (last name, then first name)

```
expert_advice <- pfizer %>%
  filter(category == "Expert-Led Forums" | category == "Professional Advising") %>%
  arrange(last_name, first_name) %>%
  head(20)
expert_advice
```

```
# A tibble: 20 x 10
  org_indiv first~1 first~2 last~3 city state categ~4 cash other total
  <chr>      <chr>   <chr>   <chr>   <chr> <chr> <chr>   <dbl> <dbl> <dbl>
1 ABBO, LILIAN M~ LILIAN~ LILIAN ABBO MIAMI FL Profes~ 1800 0 1800
```

```

2 ABEBE, SHEILA Y SHEILA~ SHEILA ABEBE INDI~ IN Expert~ 825 0 825
3 NEW YORK UNIVE~ JUDITH~ JUDITH ABERG NEW ~ NY Profes~ 1750 0 1750
4 ABOLNIK, IGOR Z IGOR Z IGOR ABOLNIK PROVO UT Expert~ 1750 0 1750
5 ABRAKSIA, SAMIR SAMIR SAMIR ABRAKS~ BEAC~ OH Expert~ 2000 0 2000
6 ABRAKSIA, SAMIR SAMIR SAMIR ABRAKS~ BEAC~ OH Profes~ 2500 0 2500
7 ABRAMSON, STEV~ STEVEN~ STEVEN ABRAMS~ NEW ~ NY Profes~ 4400 0 4400
8 ABUZZAHAB, FAR~ FARUK S FARUK ABUZZA~ MINN~ MN Profes~ 1750 0 1750
9 ABUZZAHAB, MAR~ MARY J~ MARY ABUZZA~ SAIN~ MN Expert~ 1000 0 1000
10 ACCACHA, SIHAM~ SIHAM ~ SIHAM ACCACHA MINE~ NY Expert~ 1250 0 1250
11 ACEVEDO MARTY,~ IRIS A~ IRIS ACEVED~ CAGU~ PR Expert~ 750 0 750
12 ACKERMAN, IVAN~ IVAN F~ IVAN ACKERM~ BRAN~ FL Expert~ 1250 0 1250
13 PAIN MEDICINE ~ WILLIA~ WILLIAM ACKERM~ LITT~ AR Expert~ 1000 0 1000
14 ACOSTA, LUIS S~ LUIS S~ LUIS ACOSTA HOUS~ TX Expert~ 1000 0 1000
15 ADAMS, SANDRA ~ SANDRA~ SANDRA ADAMS SAN ~ TX Profes~ 12840 0 12840
16 ADDONA, TOMMASO TOMMASO TOMMASO ADDONA NEW ~ NY Expert~ 750 0 750
17 HEALTH RESEARC~ ALEX A~ ALEX ADJEI BUFF~ NY Profes~ 2000 0 2000
18 ADLER, DAVID E~ DAVID ~ DAVID ADLER PORT~ OR Profes~ 71 0 71
19 ADLER, JEREMY A JEREMY~ JEREMY ADLER ENCI~ CA Expert~ 850 0 850
20 ADMANI, ARIFF ARIFF ARIFF ADMANI PARA~ NJ Expert~ 2000 0 2000
# ... with abbreviated variable names 1: first_plus, 2: first_name,
# 3: last_name, 4: category

```

Notice that you can sort by multiple variables, separated by commas. Use pattern matching to filter text.

## Use the `grepl` function to find values containing a particular string of text. This can simplify the code used to filter based on text.

```

# use pattern matching with grepl to filter text
expert_advice <- pfizer %>%
  filter(grepl("Expert|Professional", category)) %>%
  arrange(last_name, first_name)

not_expert_advice <- pfizer %>%
  filter(!grepl("Expert|Professional", category)) %>%
  arrange(last_name, first_name)

```

This code differs only by the `!` Boolean operator. Notice that it has split the data into two, based on categories of payment.

Append one data frame to another.

## Use the `bind_rows` function to append one data frame to another, which recreates unfiltered data from the two data frames above

```

# merge/append data frames
pfizer2 <- bind_rows(expert_advice, not_expert_advice)

```

## Write data to a CSV file

readr can write data to CSV and other text files.

```
# write expert_advice data to a csv file
write_csv(expert_advice, "expert_advice.csv", na="")
```

na="" ensures that any empty cells in the data frame are saved as blanks - R represents null values as NA, so if you don't include this, any null values will appear as NA in the saved file.

## Group and summarize data

Calculate the total payments, by state

```
# calculate total payments by state
state_sum <- pfizer %>%
  group_by(state) %>%
  summarize(sum = sum(total)) %>%
  arrange(desc(sum))
```

Notice the use of group\_by followed by summarize to group and summarize data, here using the function sum.

## Calculate some additional summary statistics, by state

```
# As above, but for each state also calculate the median payment, and the number of payments
state_summary <- pfizer %>%
  group_by(state) %>%
  summarize(sum = sum(total), median = median(total), count = n()) %>%
  arrange(desc(sum))
```

Notice the use of multiple summary functions, sum, median, and n. (You don't specify a variable for n because it is simply counting the number of rows in the data.)

## Group and summarize for multiple categories

```
# as above, but group by state and category
state_category_summary <- pfizer %>%
  group_by(state, category) %>%
  summarize(sum = sum(total), median = median(total), count = n()) %>%
  arrange(state, category)
```

As for arrange, you can group\_by by multiple variables, separated by commas.

## Working with dates

Now let's see how to work with dates, using the FDA warning letters data.

Filter the data for letters sent from the start of 2005 onwards. FDA sent warning letters from the start of 2005 onwards

You will have to fix "issued" to be read as a date. If you look back at str(fda), it was read in as a chr (character). To coerce it to be a date, use the command,

```
fda$issued <- as.Date(fda$issued, "%m/%d/%Y")
class(fda$issued)

[1] "Date"

post2005 <- fda %>%
  filter(issued >= "2005-01-01") %>%
  arrange(issued)
```

Notice that operators like >= can be used for dates, as well as for numbers.

## Count the number of letters issued by year

```
# count the letters by year
letters_year <- fda %>%
  mutate(year = format(issued, "%Y")) %>%
  group_by(year) %>%
  summarize(letters=n())
```

This code introduces dplyr's mutate function to create a new column in the data. The new variable year is the four-digit year "%Y" (see here for more on time and date formats in R), extracted from the issued dates using the format function. Then the code groups by year and counts the number of letters for each one.

Add columns giving the number of days and weeks that have elapsed since each letter was sent

```
# add new columns showing many days and weeks elapsed since each letter was sent
fda <- fda %>%
  mutate(days_elapsed = Sys.Date() - issued,
         weeks_elapsed = difftime(Sys.Date(), issued, units = "weeks"))
```

Notice in the first line that this code changes the fda data frame, rather than creating a new object. The function Sys.Date returns the current date, and if you subtract another date, it will calculate the difference in days. To calculate date and time differences using other units, use the difftime function.

Notice also that you can mutate multiple columns at one go, separated by commas.

## Join data from two data frames

Here is an animation for the different types of joins: <https://github.com/gadenbuie/tidyexplain>

There are a number of join functions in dplyr to combine data from two data frames. Here are the most useful:

- . inner\_join() returns values from both tables only where there is a match
- . left\_join() returns all the values from the first-mentioned table, plus those from the second table that match
- . semi\_join() filters the first-mentioned table to include only values that have matches in the second table
- . anti\_join() filters the first-mentioned table to include only values that have no matches in the second table.

To illustrate, these joins will find doctors paid by Pfizer to run expert led forums who had also received a warning letter from the FDA:

## Join to identify doctors paid to run Expert-led forums who also received a warning letter

```
expert_warned_inner <- inner_join(pfizer, fda, by=c("first_name" = "name_first", "last_name" = "name_last")) %>%  
  filter(category=="Expert-Led Forums")  
  
expert_warned_semi <- semi_join(pfizer, fda, by=c("first_name" = "name_first", "last_name" = "name_last")) %>%  
  filter(category=="Expert-Led Forums")
```

The code in `by=c()` defines how the join should be made. If instructions on how to join the tables are not supplied, `dplyr` will look for columns with matching names, and perform the join based on those.

The difference between the two joins above is that the first contains all of the columns from both data frames, while the second gives only columns from the `pfizer` data frame.

In practice, you may wish to `inner_join` and then use `dplyr`'s `select` function to select the columns that you want to retain, for example:

## Select desired columns from data

```
expert_warned <- inner_join(pfizer, fda, by=c("first_name" = "name_first", "last_name" = "name_last")) %>%  
  filter(category=="Expert-Led Forums") %>%  
  select(first_plus, last_name, city, state, total, issued)  
  
expert_warned <- inner_join(pfizer, fda, by=c("first_name" = "name_first", "last_name" = "name_last")) %>%  
  filter(category=="Expert-Led Forums") %>%  
  select(2:5, 10, 12)
```

Notice that you can select by columns' names, or by their positions, where 1 is the first column, 3 is the third, and so on.

## (Additional Information) Merging Two Datasets in R

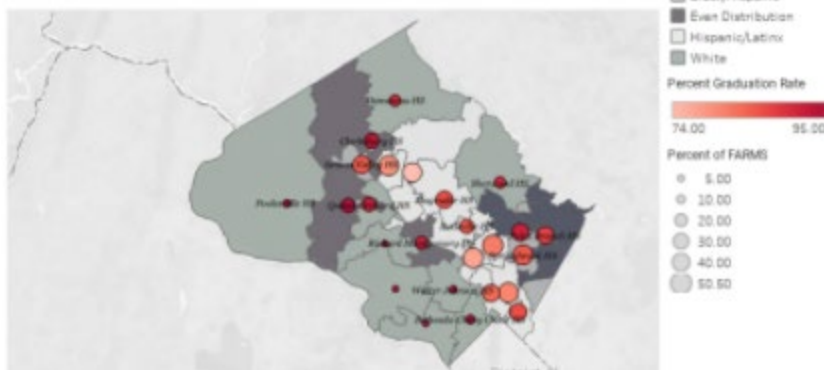
We can do the same joining process in R. So now we will re-do this using R code.

This page has some simple (but easy-to-understand!) animations on how various types of joins work.

<https://github.com/gadenbuie/tidyexplain>

[Here is a useful reference](#) for managing joins with `dplyr`.

## Introduction to Tableau



Follow the tutorial, [MoCo High School District Tableau Tutorial](#) (also posted as a separate document on Blackboard in this week's material). Recreate the visualization along with information in the tooltips, title and caption. Save your work in tableau public and submit the link in this week's assignment dropbox.

In addition, write a brief essay (minimum 500 words) on a separate Word document. ***Be sure to describe relationships (similarities and/or differences) you notice that are displayed by color delineations, sizes of the circles, and other variable measurements listed in the tooltips.***

## Week 4 Homework Assignment

1. **(Ungraded)** Complete copying the Markdown code from these notes to explore the pfizer and fda data (anything you did not complete in class). You may opt to publish your RMD file in Rpubs, but it is not required or graded. Remember - you are responsible for trying and learning all code in these notes. Also revisit the Working with Dates Notes.
2. **(Worth up to 10 points)** Follow the tutorial, MoCo High School District Tableau Tutorial (posted as a separate document on Blackboard in this week's material). Recreate the visualization along with information in the tooltips, title and caption. Save your work in tableau public and submit the link in this week's assignment dropbox.

*Submit the link for your published tableau visualization in the appropriate assignment dropbox by **11:59 pm on Sunday, Sept 28<sup>th</sup>**.*

3. **(Worth up to 10 points)** Watch [this video](#) FIRST (by Veritasium) about the reproducibility crisis. Then select one of the provided articles on the Reproducibility Crisis.

The Reproducibility Crisis Articles (use this link to find all these articles: <http://bit.ly/Data110Articles> )

Article: Statisticians, roll up your sleeves - there is a crisis to be solved (Significance 8/21)

Article: Something is off-base with this title (Significance 3/24)

Article: Cargo Cult Statistics and Scientific Crisis

Article: ASA Statement on P-Values (and P-hacking)

Article: Cornell's Top Food Researcher has Had 13 Studies Retracted

Article: 1500 Scientists Could Not Reproduce their Own Studies (Nature News)

Article: Replicability or reproducibility? On the replication crisis in computational neuroscience and sharing only relevant detail (Journal of Computational Neuroscience)

Article: [Investigators found plagiarism and data falsification from prominent cancer lab](#) (Nature)



Article: [Blots on a Field? A neuroscience image sleuth finds signs of fabrication in scores of Alzheimer's articles...](#) (Science July 2022)

Article: [Fault Found \(Again\) with Conflicts of Interest](#) (MedPage Today)

Article: We're All P-Hacking Now, Wired <https://www.wired.com/story/were-all-p-hacking-now/>

**For your essay submission, include the following:**

- Title and author of article you select.
- Write a paragraph or two to summarize the article
- Write a paragraph response to the article, including your opinion on how statisticians/data scientists can work to eliminate this problem.

**Your essay should be a total word count of around 250-500 words.**

*Submit this assignment in the assignment dropbox by 11:59 pm on Sunday, Sept 28<sup>th</sup>. You will present your findings to the class the following week in a socratic circle forum.*