Lecture 6: Line Charts

June 22, 2022

Overview

Course Administration

Good, Bad and Ugly

Line Charts

Few on Stories

Power of Annotations

R Notes

Course Administration

1. Next week (7): in-class workshop
   - Workshop instructions online under Lecture 6
   - Come prepared to talk about your policy brief
2. Ask for help!
   - Use Piazza
   - Bianca and I are here to help

Course Administration

1. Next week (7): in-class workshop
   - Workshop instructions online under Lecture 6
   - Come prepared to talk about your policy brief
2. Ask for help!
   - Use Piazza
   - Bianca and I are here to help
   - In this vein: what is a block group?

Course Administration

1. Next week (7): in-class workshop
   - Workshop instructions online under Lecture 6
   - Come prepared to talk about your policy brief
2. Ask for help!
   - Use Piazza
   - Bianca and I are here to help
   - In this vein: what is a block group?
3. Presentations and final brief
   - see handout posted under today's lecture

Course Administration

1. Next week (7): in-class workshop
   - Workshop instructions online under Lecture 6
   - Come prepared to talk about your policy brief
2. Ask for help!
   - Use Piazza
   - Bianca and I are here to help
   - In this vein: what is a block group?
3. Presentations and final brief
   - see handout posted under today's lecture
4. Zoom policy
   - Once or twice is fine
   - You don't need to tell me first
   - If you miss >2 classes, I am concerned about your ability to participate fully

Course Administration

1. Next week (7): in-class workshop
   - Workshop instructions online under Lecture 6
   - Come prepared to talk about your policy brief
2. Ask for help!
   - Use Piazza
   - Bianca and I are here to help
   - In this vein: what is a block group?
3. Presentations and final brief
   - see handout posted under today's lecture
4. Zoom policy
   - Once or twice is fine
   - You don't need to tell me first
   - If you miss >2 classes, I am concerned about your ability to participate fully
5. Anything else? (chart feedback on next slide)

Fully Composed Chart Feedback

- Biggest improvement
  - figure out the point of the chart
  - make this the title

Fully Composed Chart Feedback

- Biggest improvement
  - figure out the point of the chart
  - make this the title
  - For example, "Exports up 250% Since 2005"
- Subtitle should tell you what the chart does
  - For example, "Value of US Exports in 2020 Dollars"
- In grading folder, see standardized comments
- May be helpful to review all of them

Fully Composed Chart Feedback

- Biggest improvement
  - figure out the point of the chart
  - make this the title
  - For example, "Exports up 250% Since 2005"
- Subtitle should tell you what the chart does
  - For example, "Value of US Exports in 2020 Dollars"
- In grading folder, see standardized comments
- May be helpful to review all of them
- Use the fully composed chart list as a checklist for your final graphics

Class 7, June 29: Good Bad and Ugly

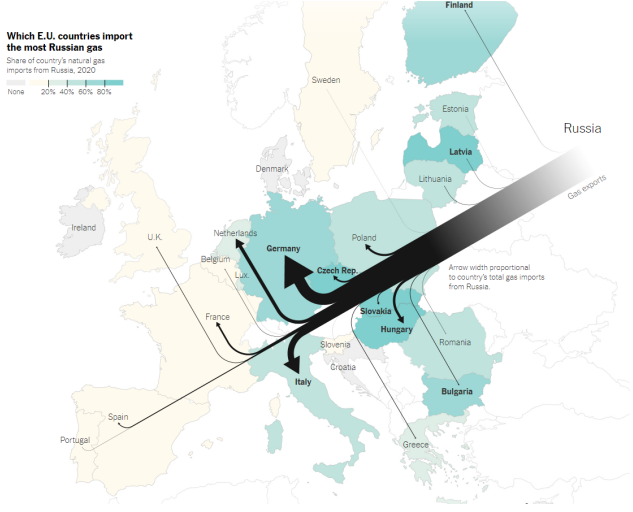
Post this week by Friday noon. Look for a line chart.
Let me know if you chart doesn't appear in a timely fashion.

Finder	Commenter
Mary	David
Andres	Jarred
Ensold	Jack

This Week's Good Bad and Ugly

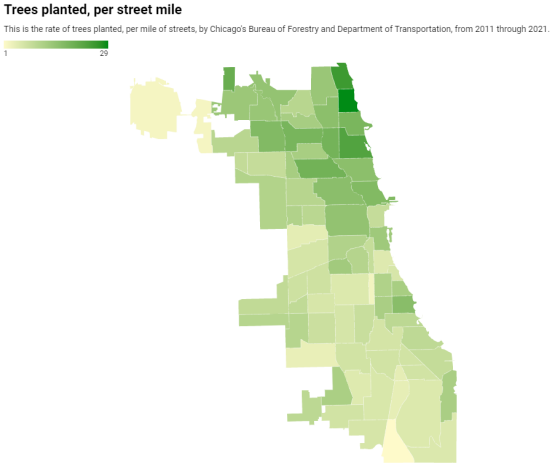
Finder	Commenter
David	Dylan
Danielle	Christina
Richa	Brady
Brady	Evan

David's Example, Comments by Dylan



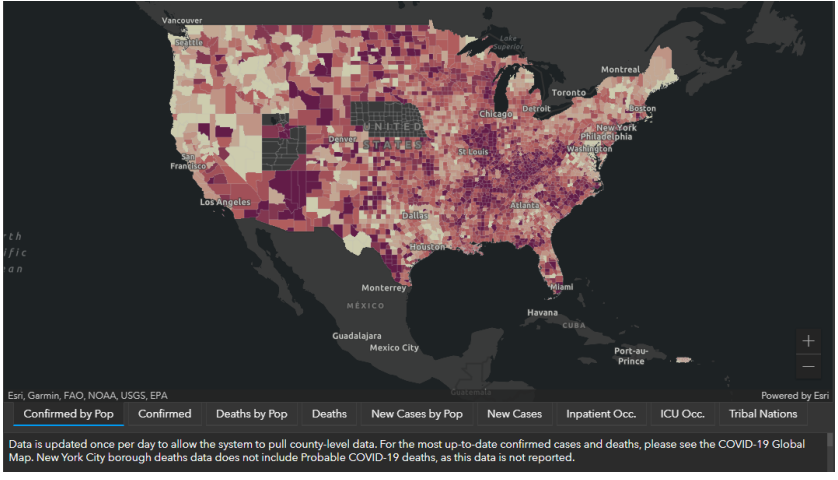
"How a Ukraine Conflict Could Reshape Europe's Reliance on Russia," *New York Times*, Feb. 15, 2022

Danielle's Example, Comments by Christina



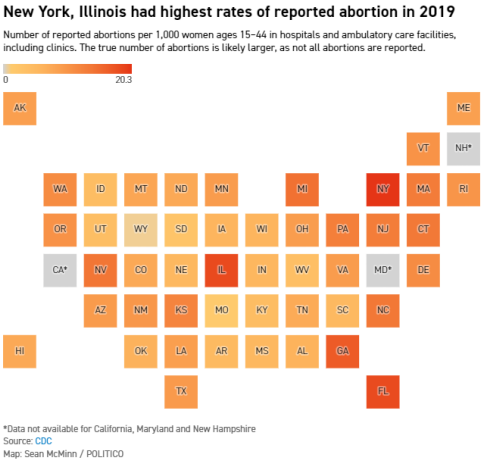
"In extreme heat, trees can help cool neighborhoods. But a Tribune investigation found the city has planted more trees in wealthier, whiter areas," *Chicago Tribune*, June 15, 2022

Richa's Example, Comments by Brady



Title says "COVID-19 United States Cases by County." Johns Hopkins Coronavirus Resource Center, Accessed June 22, 2022.

Brady's Example, Comments by Evan



Goldberg, Dan. "Abortion Statistics by state: Maps, trigger laws, and possible bans," *Politico*, May 3, 2022.

Line Charts

Line Charts

- Have time on the horizontal axis
  - **Always** have consistent time units

Line Charts

- Have time on the horizontal axis
  - **Always** have consistent time units
- Values on the vertical axis
  - usually start at zero

Line Charts

- Have time on the horizontal axis
  - **Always** have consistent time units
- Values on the vertical axis
  - usually start at zero
- Should you put dots for points?

Line Charts

- Have time on the horizontal axis
  - **Always** have consistent time units
- Values on the vertical axis
  - usually start at zero
- Should you put dots for points?
  - Con: Noisy, may add little info
  - Pro: When data are sparse, readers assume full line is data

Line Charts

- Have time on the horizontal axis
  - **Always** have consistent time units
- Values on the vertical axis
  - usually start at zero
- Should you put dots for points?
  - Con: Noisy, may add little info
  - Pro: When data are sparse, readers assume full line is data
- Slope has meaning:

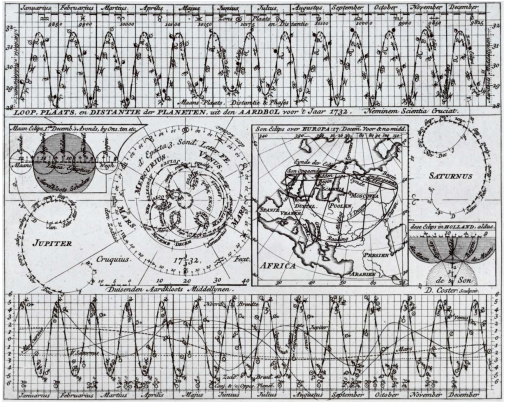
Line Charts

- Have time on the horizontal axis
  - **Always** have consistent time units
- Values on the vertical axis
  - usually start at zero
- Should you put dots for points?
  - Con: Noisy, may add little info
  - Pro: When data are sparse, readers assume full line is data
- Slope has meaning: rate of change
- More than a few lines is too much

Look at the Values on the Vertical Axis

Line Chart, c. 1732

Nicolaas Kruik (1678-1754) "land surveyor, cartographer, astronomer and weatherman" who "liked to measure things"



Thanks to Wikipedia.

How to Call Things out in a Line Chart

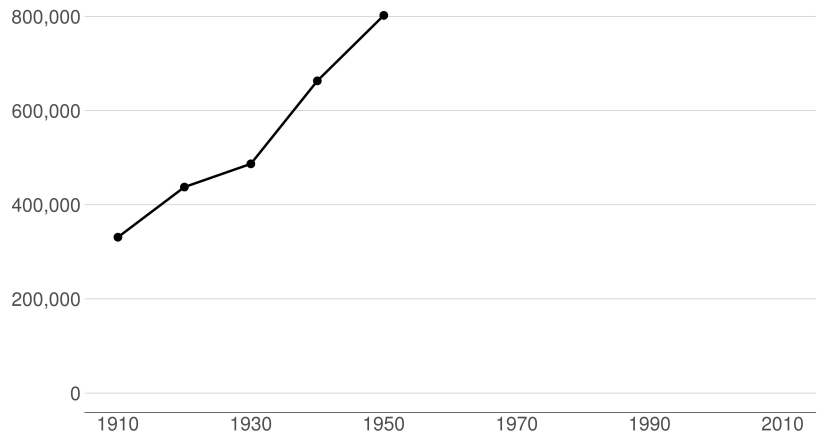
How to Call Things out in a Line Chart

Think back to preattentive processing

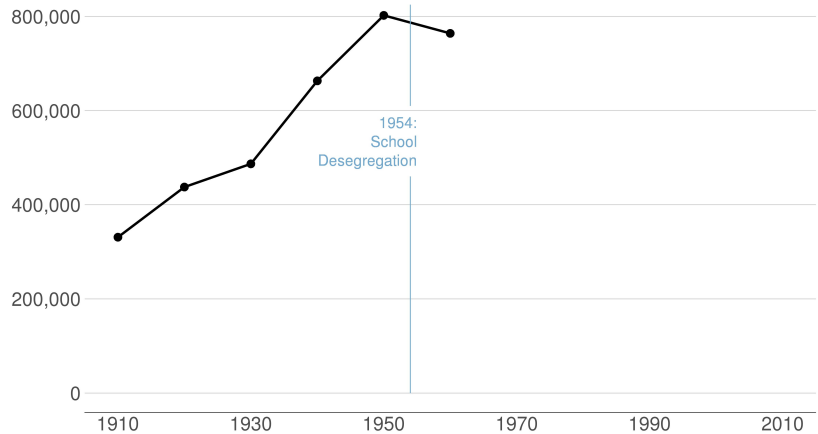
- color
- size
- timing

My example with this; think how to re-do for a report.

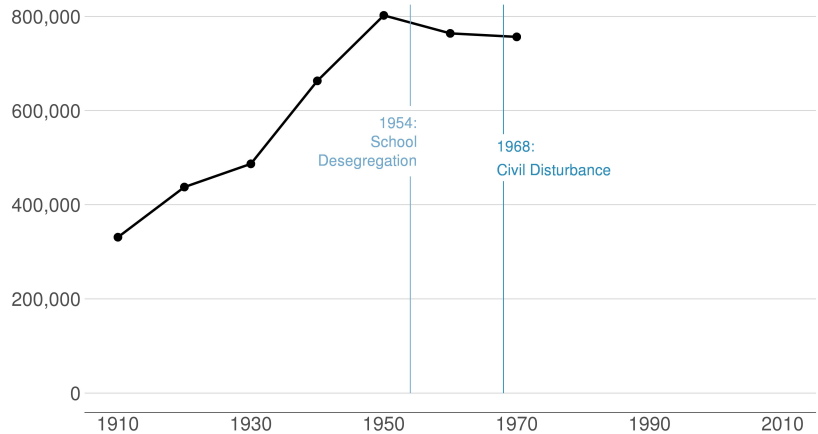
DC Gains Population Through 1950



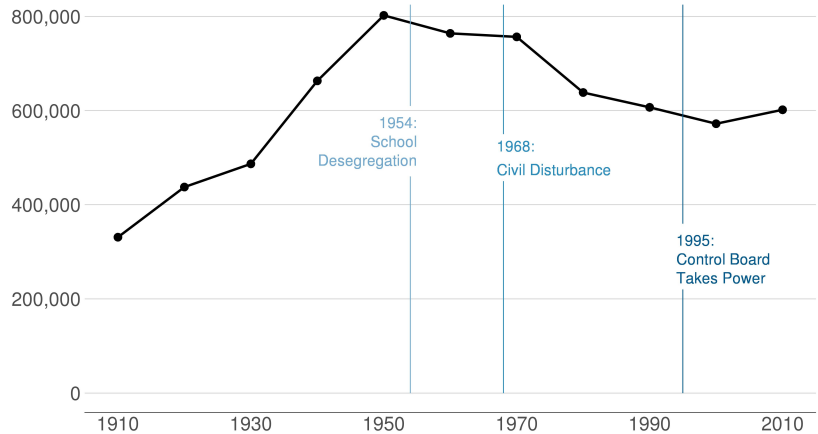
Population Loses Start with Desegregation



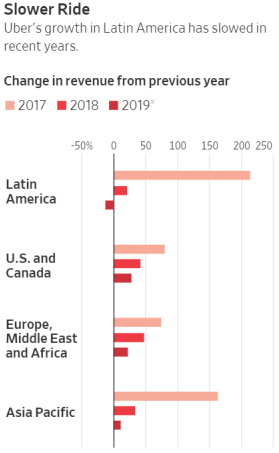
Continue After Civil Disturbance



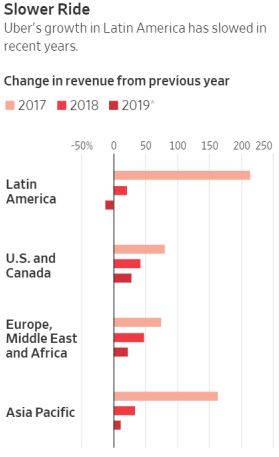
Population Turns Up After 2000



Something That Should be a Line Chart



Something That Should be a Line Chart



- We use lines to show change over time
- Lines make pace of change obvious
- These bars have to point out years
- Vertical alignment of lines would show that they are the same year

From this very interesting *WSJ* article about Soft Bank's funding of Uber and its competitors

Few on Stories

Chap 13: Telling Compelling Stories with Numbers

- Answer to "Is it a good chart?" depends on the story you're trying to tell
- The graphic can tell you about the story
- But the story can also lead you to the graphic
- Make sure you know the point that the graphic should make

Few's Components of a Compelling Story

- **Simple**
- Seamless
- Informative
- True
- **Contextual**
- Familiar
- Concrete
- Personal
- Emotional
- Actionable
- **Sequential**

Simple

- Always present the simplest possible version of your analysis first
- Summary statistics preferred to regression coefficients

Contextual

- Very important for magnitudes with which people are not familiar
- Helps us answer "so what" question
- Visuals can put in context
  - dates
  - comparative categories
  - comparative values
  - baseline mean
  - standard deviation

Contextual

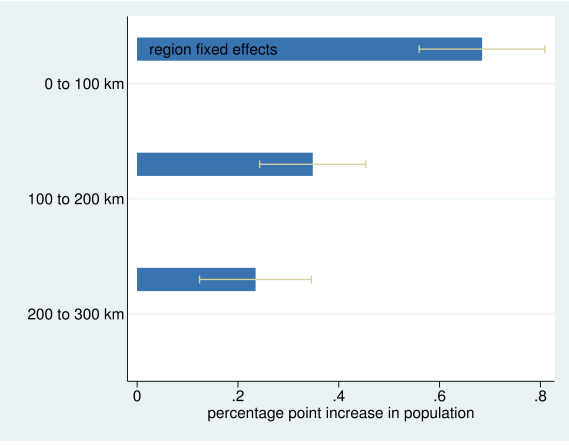
- Very important for magnitudes with which people are not familiar
- Helps us answer "so what" question
- Visuals can put in context
  - dates
  - comparative categories
  - comparative values
  - baseline mean
  - standard deviation

What does this mean for your policy brief?

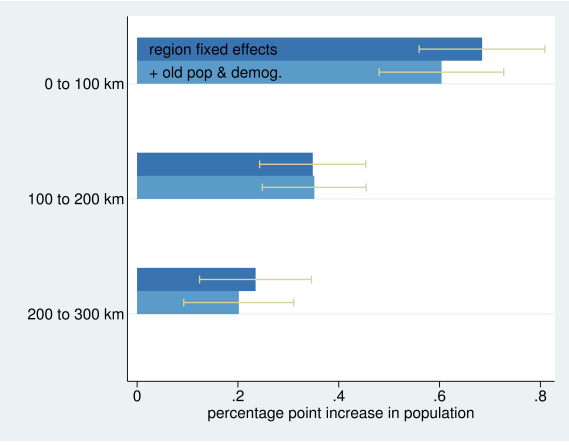
Sequential

- It is possible to present relatively complex graphics
- With proper groundwork
- Can be easer in a presentation than in a paper
- Paper/screen visuals need to be sequential differently
  - dance on screen vs dance in person

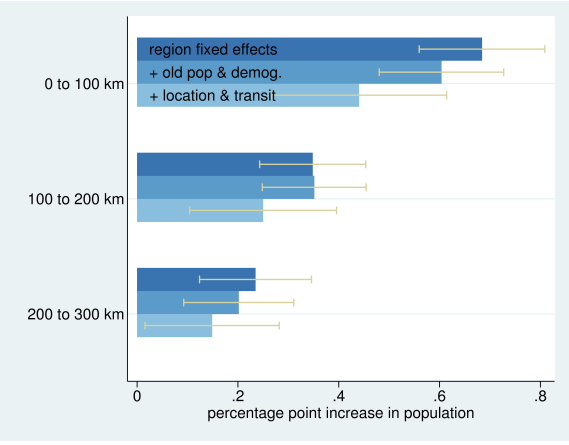
Bars with Error Bars, Building



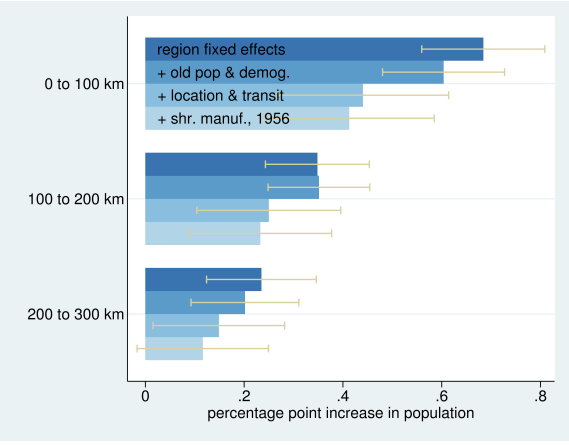
Bars with Error Bars, Building



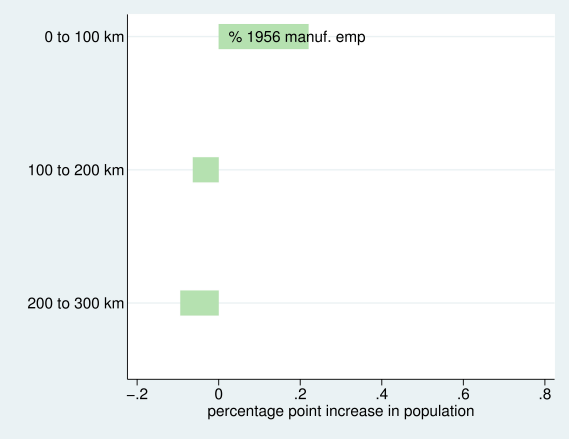
Bars with Error Bars, Building



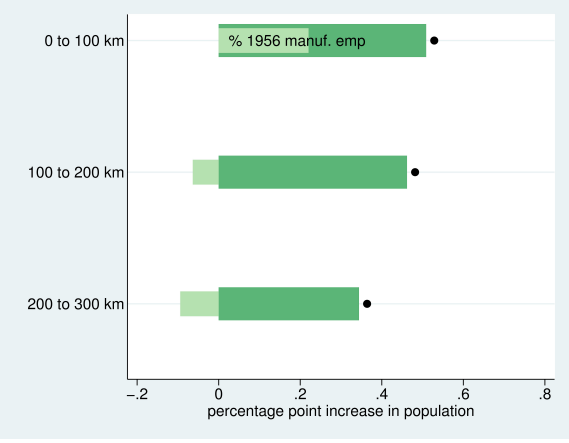
Bars with Error Bars, Building



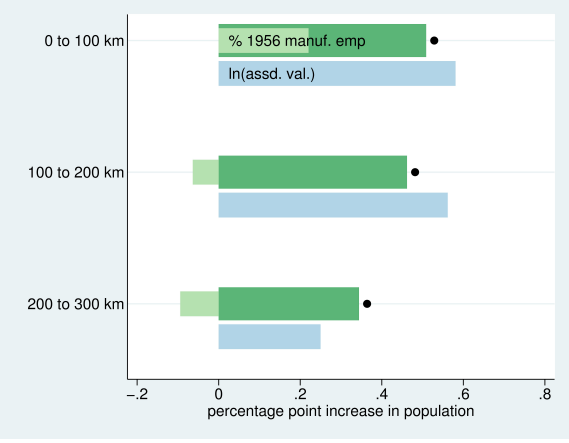
Interaction Effects



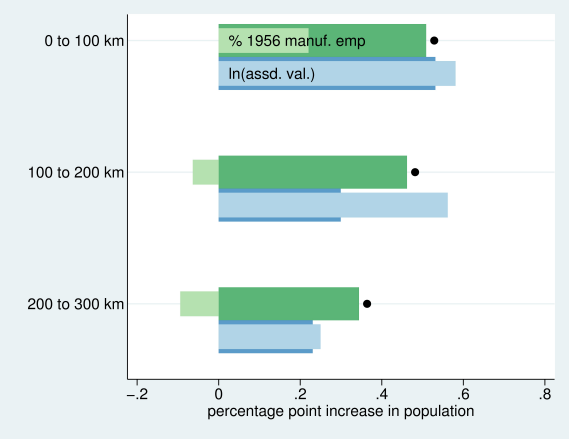
Interaction Effects



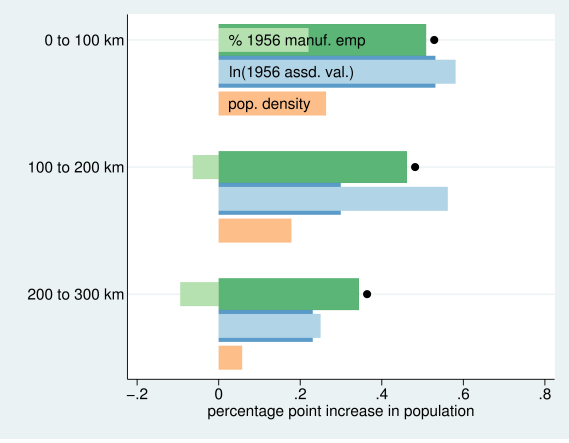
Interaction Effects



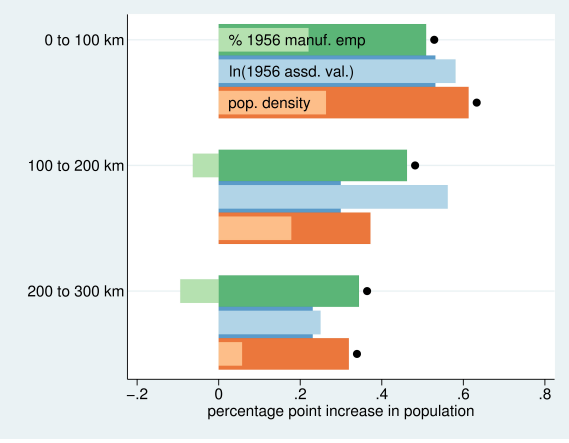
Interaction Effects



Interaction Effects

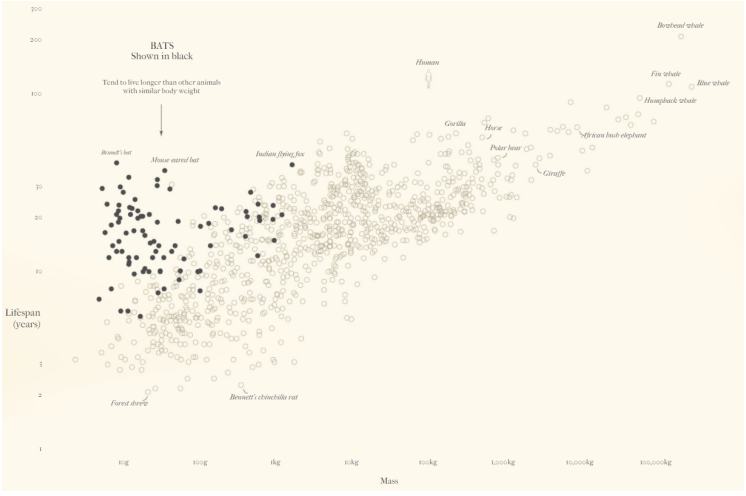


Interaction Effects



Annotations

A Great Use of Annotations



From Reuters

Today in R

Today in R: Line Charts and De-Bugging

1. Line charts and `ggplot`
2. Summarizing data
3. Annotations
4. Making data long
5. De-bugging

1. Line charts

```
p1 <- ggplot() +
  geom_line(data = polys,
            mapping = aes(x = xvar, y = yvar))
```

1. Line charts

```
p1 <- ggplot() +
  geom_line(data = polys,
            mapping = aes(x = xvar, y = yvar))
```

- R does not require `xvar` to be time
- But your readers will assume it is

Hurricanes by Year

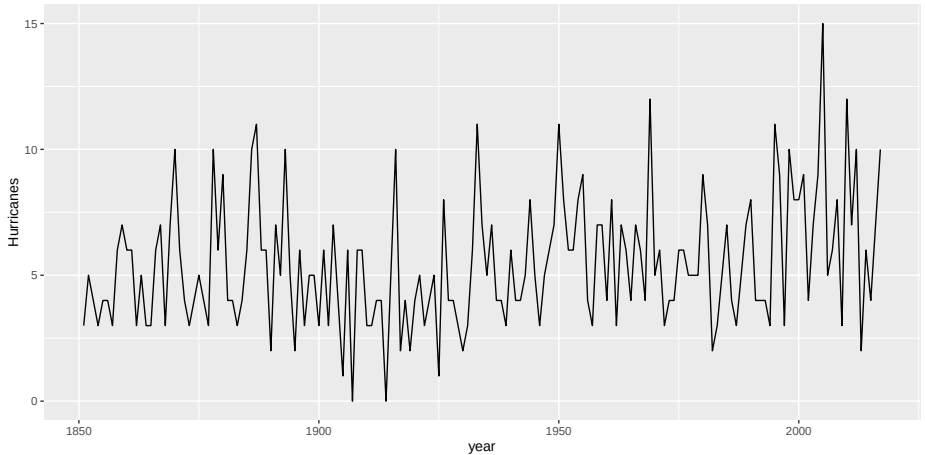
```
# remember the hurricanes
hurr <- read.csv("H:/pppa_data_viz/2019/tutorial_data/lecture03/2019-02-02
# look at it
head(hurr)
```

```
##   year Named.Storms Hurricanes Major
## 1 1851            6          3     1
## 2 1852            5          5     1
## 3 1853            8          4     2
## 4 1854            5          3     1
## 5 1855            5          4     1
## 6 1856            6          4     2
```

Graphing Hurricanes by Year

```
# plot it
hurr.by.year <- ggplot() +
                geom_line(data = hurr,
                          mapping = aes(x = year, y = Hurricanes))
```

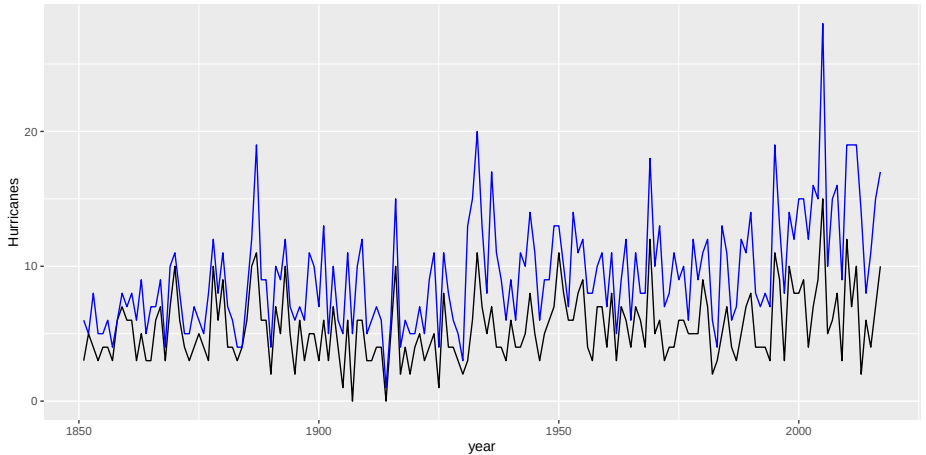
Graphing Hurricanes by Year



Graphing Hurricanes and Named Storms by Year

```
# plot it
hurr2.by.year <- ggplot() +
                 geom_line(data = hurr,
                           mapping = aes(x = year, y = Hurricanes)) +
                 geom_line(data = hurr,
                           mapping = aes(x = year, y = Named.Storms),
                           color = "blue")
```

Graphing Hurricanes and Named Storms by Year



Using Group= to Make Multiple Lines

You need long data:

```
df <- data.frame(year = c(1, 2, 3, 1, 2, 3, 1, 2, 3),
                 type = c(1, 1, 1, 2, 2, 2, 3, 3, 3),
                 outcome = c(1, 2, 3, 2, 3, 3.1, 3, 4, 5))
df
```

```
##   year type outcome
## 1    1    1     1.0
## 2    2    1     2.0
## 3    3    1     3.0
## 4    1    2     2.0
## 5    2    2     3.0
## 6    3    2     3.1
## 7    1    3     3.0
## 8    2    3     4.0
## 9    3    3     5.0
```

Using Group= to Make Multiple Lines

You need long data:

```
# plot it
df.by.year <- ggplot() +
                geom_line(data = df,
                          mapping = aes(x = year, y = outcome,
                                        group = type))
```

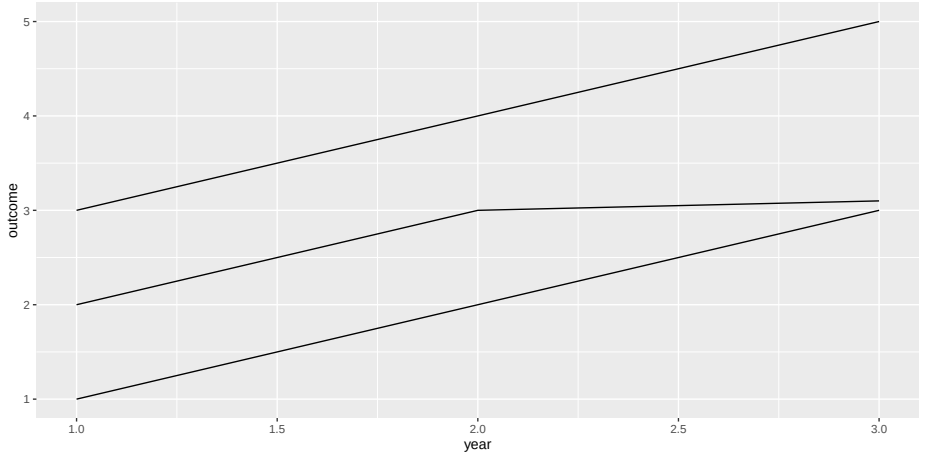
Using Group= to Make Multiple Lines

You need long data:

```
# plot it
df.by.year <- ggplot() +
                geom_line(data = df,
                          mapping = aes(x = year, y = outcome,
                                        group = type))
```

- Be wary of using too many lines

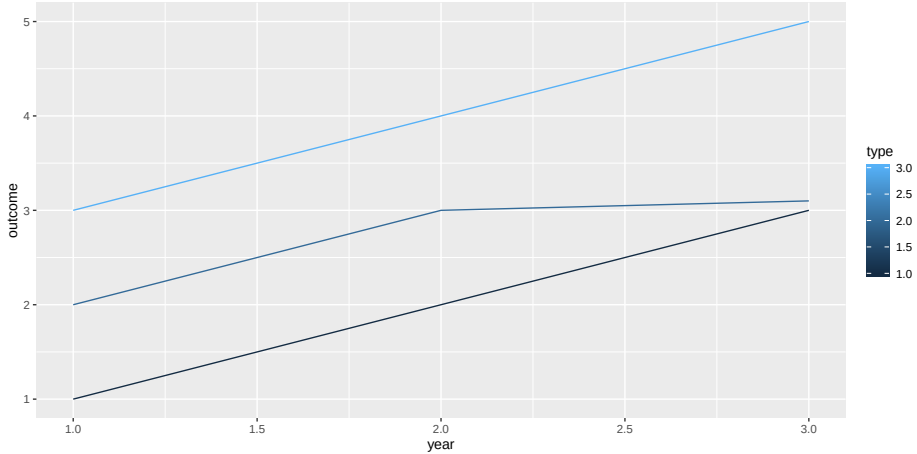
Using `Group=` to Make Multiple Lines



Using Group= to Make Multiple Lines and color = to color lines

```
df.by.year2 <- ggplot() +
                geom_line(data = df,
                          mapping = aes(x = year, y = outcome,
                                        group = type, color = type))
```

Using `Group=` to Make Multiple Lines and `color =` to color lines



2. Summarizing data

In today's tutorial, you'll use bikeshare data

- these come at the level of the individual ride
- we describe them by hour
  - $\rightarrow$ summarize by hour (`group_by` first)
- for the homework, you describe them by minute

2. Summarizing data

In today's tutorial, you'll use bikeshare data

- these come at the level of the individual ride
- we describe them by hour
  - $\rightarrow$ summarize by hour (`group_by` first)
- for the homework, you describe them by minute

Remember that `group_by` and then `summarize` take you from one unit of observation to another.

3. Annotations: Better than Legends

General logic is

```
np <- already.existing.plot +
      annotate(geom = [name of geom],
               x = [x location],
               y = [y location],
               [xmin = , xmax = ,
                ymin = , ymax =]
               other options -- size, color, etc)
```

3. Annotations: Better than Legends

General logic is

```
np <- already.existing.plot +
      annotate(geom = [name of geom],
               x = [x location],
               y = [y location],
               [xmin = , xmax = ,
                ymin = , ymax =]
               other options -- size, color, etc)
```

What you add is in `geom = [name of geom]` which can be

- text
- point
- segment (line)
- rectangle
- possibly others

3. Annotation example plan

- make a small dataframe to illustrate
- show the line graph of this small dataframe
- add an annotation

3. Small example dataframe

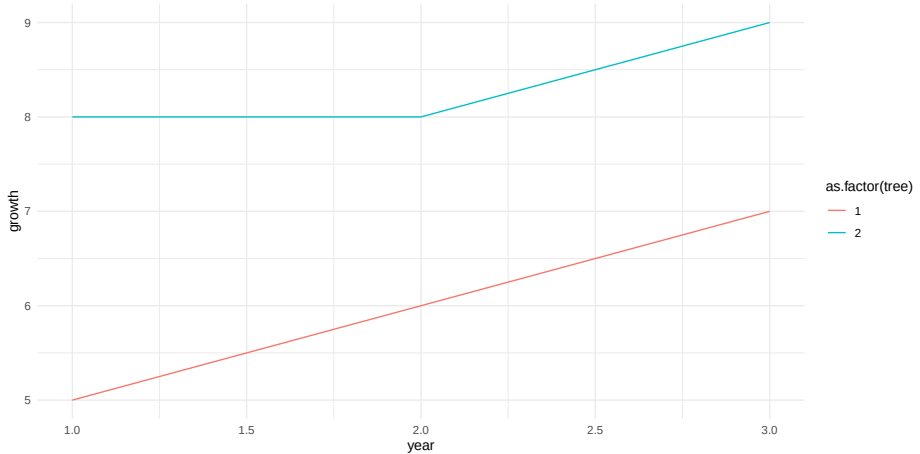
```
trees <- data.frame(year = c(1,2,3,1,2,3),
                    tree = c(1,1,1,2,2,2),
                    growth = c(5,6,7,8,8,9))
```

3. Line plot of trees

```
tp <- ggplot() +
      geom_line(data = trees,
                mapping = aes(x = year, y = growth,
                              group = tree,
                              color = as.factor(tree))) +
      theme_minimal()
```

3. Line plot of trees

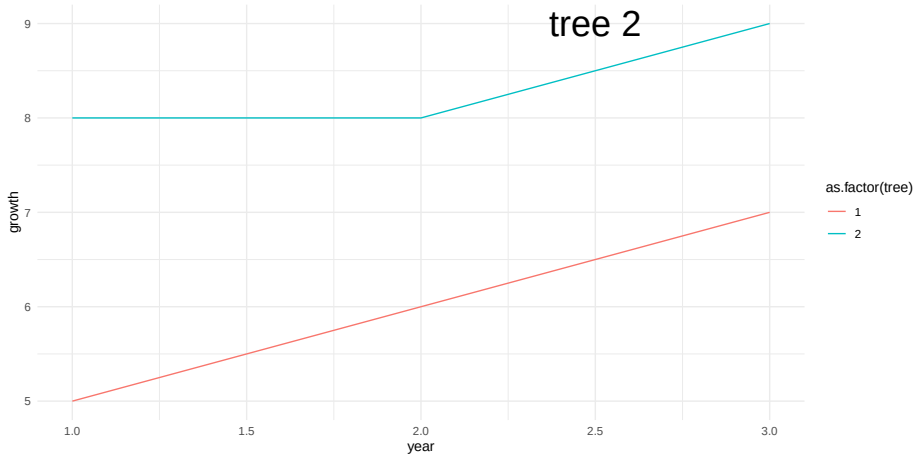
```
tp
```



3. Add text annotation

```
tp2 <- tp +
       annotate(geom = "text",
                x = 2.5,
                y = 9,
                label = "tree 2",
                size = 10)
```

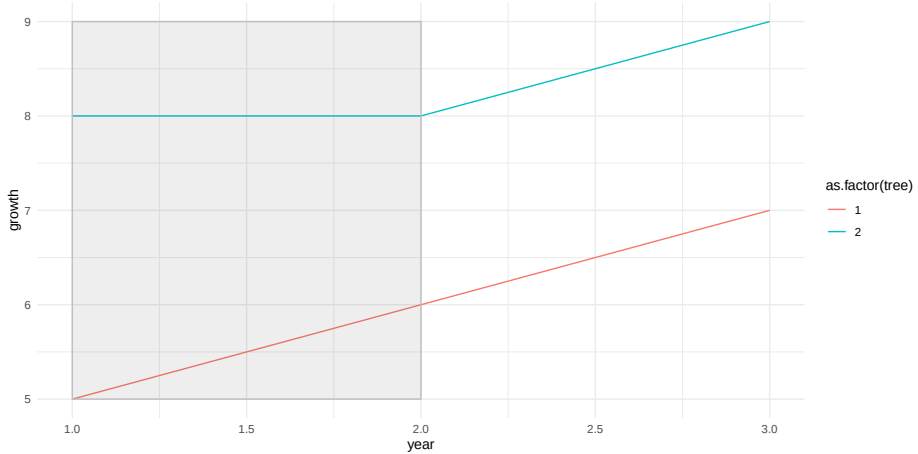
3. Line plot of trees with annotation



3. Add rectangle annotation

```
tp3 <- tp +
       annotate(geom = "rect",
                xmin = 1, xmax = 2,
                ymin = 5, ymax = 9,
                color = "grey",
                alpha = 0.1)
```

3. Add rectangle annotation



4. Making Data Long

`ggplot()` prefers long data

To think about this we will

- show wide data
- show long data
- show how to go wide to long

4. Wide data

```
wide <- data.frame(state = c("6","36","48"),
                   female_pop = c("10","12","14"),
                   male_pop = c("11","13","12"))
wide
```

```
##   state female_pop male_pop
## 1     6         10       11
## 2    36         12       13
## 3    48         14       12
```

4. Long data

```
long <- data.frame(state = c("6","36","48","6","36","48"),
                   pop = c("10","12","14","11","13","12"),
                   sex = c("female","female","female","male","male","male"))
long
```

```
##   state pop    sex
## 1     6  10 female
## 2    36  12 female
## 3    48  14 female
## 4     6  11   male
## 5    36  13   male
## 6    48  12   male
```

4. Going from wide to long

```
long2 <- pivot_longer(data = wide,
                      cols = c("female_pop","male_pop"),
                      names_to = "sex",
                      values_to = "pop")
long2
```

```
## # A tibble: 6 x 3
##   state sex        pop
##   <fct> <chr>      <fct>
## 1 6     female_pop 10
## 2 6     male_pop   11
## 3 36    female_pop 12
## 4 36    male_pop   13
## 5 48    female_pop 14
## 6 48    male_pop   12
```

4. Additional notes

- you can clean up the sex variable with a `substr()` command
- or there is even a way to do set this up in `pivot_longer()` itself

4. Additional notes

- you can clean up the sex variable with a `substr()` command
- or there is even a way to do set this up in `pivot_longer()` itself
- and there is `pivot_wider()` for going the other way

4. Additional notes

- you can clean up the sex variable with a `substr()` command
- or there is even a way to do set this up in `pivot_longer()` itself
- and there is `pivot_wider()` for going the other way
- be careful with data in the dataframe that you are not pivoting – frequently wrongly organized
- $\rightarrow$ just keep what you need and pivot

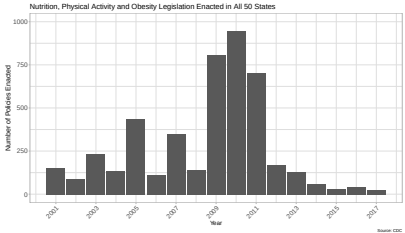
5. De-bugging

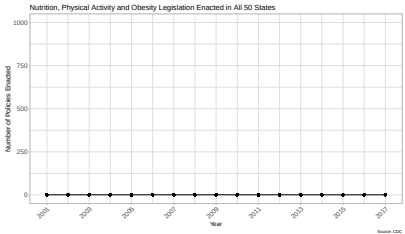
- Write a minimal reproducible example
- Doing this frequently solves your problem
- Two basic methods
  - A. start from scratch
  - B. Remove till problem disappears

Taken largely from Stack Overflow's advice. For Hadley Wickham's official advice, see here.

5.a. Start from scratch method

Problem: Jasmine is getting a wacky line graph





5.a Jasmine's code

```
enacted %>%
  group_by(Year, ProvisionID, HealthTopic) %>%
  slice(1) %>%
  summarize(num.policies = n()) %>%
  ggplot() + geom_col(aes(x = Year, y = num.policies))
```

5.a Jasmine's code

```
enacted %>%
  group_by(Year, ProvisionID, HealthTopic) %>%
  slice(1) %>%
  summarize(num.policies = n()) %>%
  ggplot() + geom_col(aes(x = Year, y = num.policies))
```

Ack! A lot of stuff together

5.a. My response

(i) do the data prep before the graph, so the graph command has just the graphing, and not the summarizing or slicing or any of that. So prep the data and then show me the head of the head, with all the variables that go into the graph

(ii) then the code for both geom_col and geom_line that provides different estimates.

It's possible that just doing (i) will figure out your problem, but we'll see.

LB

5.a. My response

(i) do the data prep before the graph, so the graph command has just the graphing, and not the summarizing or slicing or any of that. So prep the data and then show me the head of the head, with all the variables that go into the graph

(ii) then the code for both geom_col and geom_line that provides different estimates.

It's possible that just doing (i) will figure out your problem, but we'll see.

LB

$\rightarrow$ (i) did figure out the problem – data not formatted correctly

5.a. How to implement start from scratch?

- Are data ok?
- Look at data by themselves
- Plot bar only
- Plot line only
- These should help you narrow down the problematic portion of the code

5.b. Remove till problem appears method

- This is for less obvious serious problems
- Method:
  - Get rid of bottom half of your code
  - Problem still exist?

5.b. Remove till problem appears method

- This is for less obvious serious problems
- Method:
  - Get rid of bottom half of your code
  - Problem still exist?
  - Get rid of bottom half of your code
  - Problem still exist?

5.b. Remove till problem appears method

- This is for less obvious serious problems
- Method:
  - Get rid of bottom half of your code
  - Problem still exist?
  - Get rid of bottom half of your code
  - Problem still exist?
  - etc..

5.b. Remove till problem appears method

- This is for less obvious serious problems
- Method:
  - Get rid of bottom half of your code
  - Problem still exist?
  - Get rid of bottom half of your code
  - Problem still exist?
  - etc..
- Surely a second-choice method
- But sometimes necessary
- I use this most frequently for R Markdown, which is buggy

5.c. Minimal Reproducible Example

- The smallest piece of code that generates your problem
- May need to include data
- Frequently, generating this solves your problem
- stackoverflow has a great page on this

Next Lecture

- Next week: In-class workshop and storyboarding
- Also – functions!
- I will be around to answer questions until 8:45