Lecture 8: Line Charts

March 23, 2020

Overview

Course Administration

Good, Bad and Ugly

Line Charts

Few on Stories

R Notes

Course Administration

1. Next week: guest speaker
   - Luis Melgar from *WSJ* will join online
   - you look at his stuff in advance
   - come prepared with questions
2. Next week: in-class workshop
   - Workshop instructions online under Lecture 6
3. You need to post your work by March 29 at 3:30
4. Presentations
   - you'll record your presentation
   - I'd rather wait to finalize details until we see how the online stuff goes
5. Anything else?

Class 10, April 6: Good Bad and Ugly

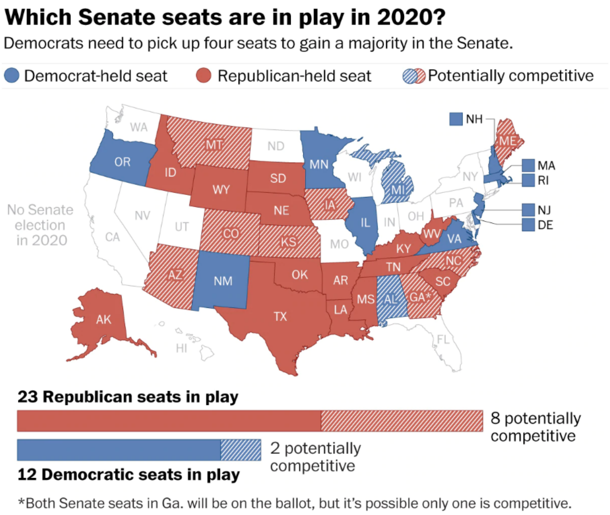
Just post this week by Wednesday noon before you forget. Look for a line chart.

Finder	Commenter
Lydia G.	Aaron K.
Kaila C.	Dallas C.
David N.	Basia D.

This Week's Good Bad and Ugly

Finder	Commenter
Betsy K.	Didem B.
Erik C.	Lydia G.
Josh F.	Neha M.

Betsy's Example, Comments by Didem



Which Senate seats are in play in 2020?

Democrats need to pick up four seats to gain a majority in the Senate.

● Democrat-held seat ● Republican-held seat ▨ Potentially competitive

23 Republican seats in play

8 potentially competitive

2 potentially competitive

12 Democratic seats in play

*Both Senate seats in Ga. will be on the ballot, but it's possible only one is competitive.

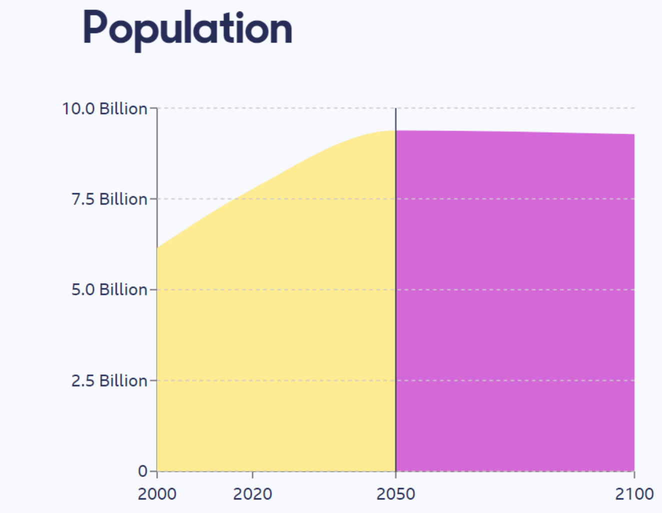
Erik’s Example, Comments by Lydia

In

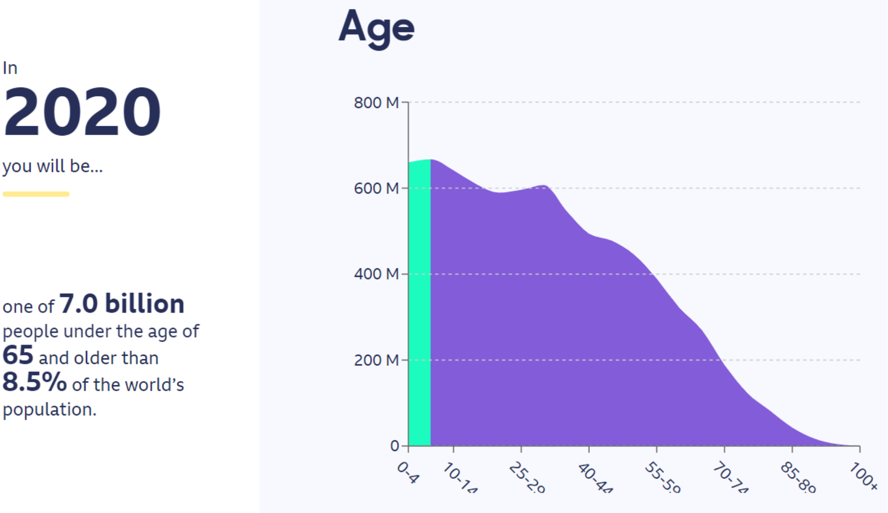
2050

you will be...

one of **9.4 billion** people in the world, with the global population increasing **21%** since 2020.



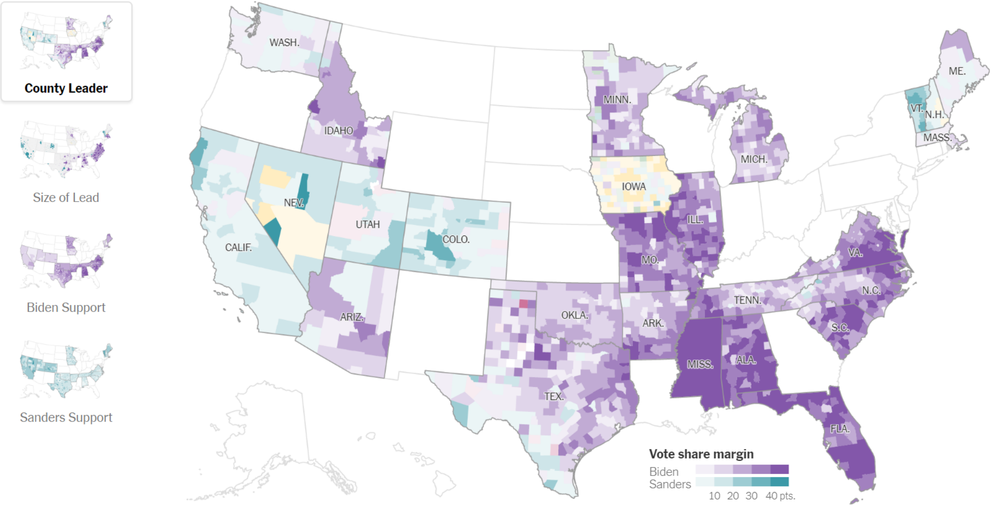
Erik's Example, Comments by Lydia



In
2020
you will be...

one of **7.0 billion** people under the age of **65** and older than **8.5%** of the world's population.

Josh’s Example, Comments by Neha



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

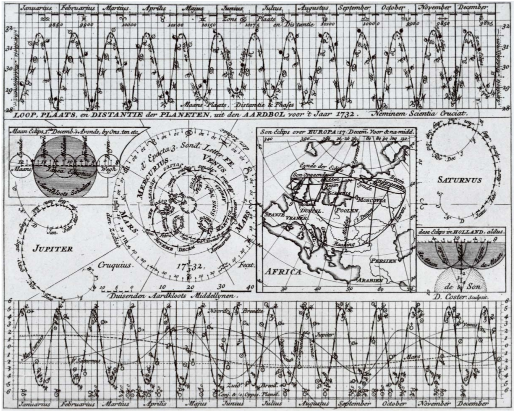
- Have time on the horizontal axis
  - **Always** have consistent time units
- Values on the vertical axis
  - usually start at zero
- Should you put dots for points?
  - Con: Noisy, may add little info
  - Pro: When data are sparse, readers assume full line is data

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: rate of change
- More than a few lines is too much

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

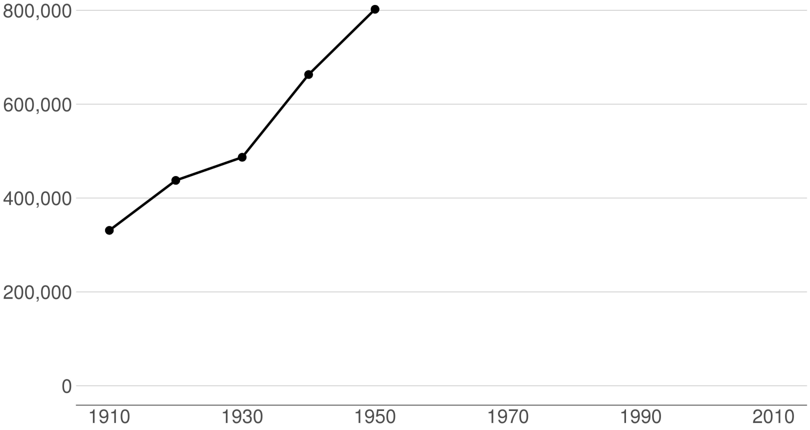
How to Call Things out in a Line Chart

Think back to preattentive processing

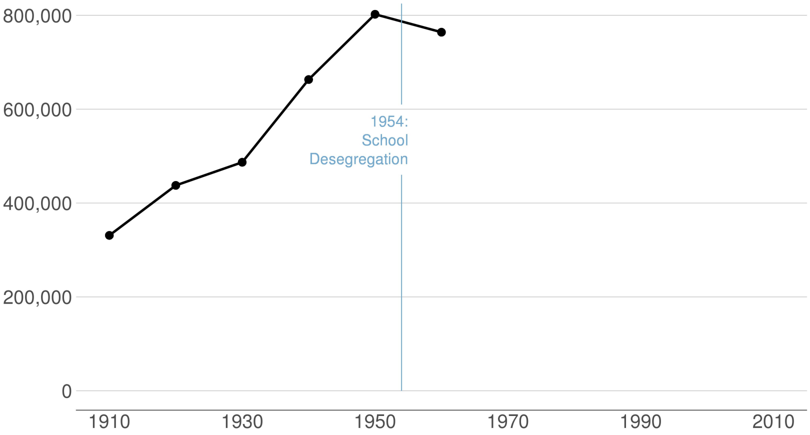
- color
- size
- timing

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

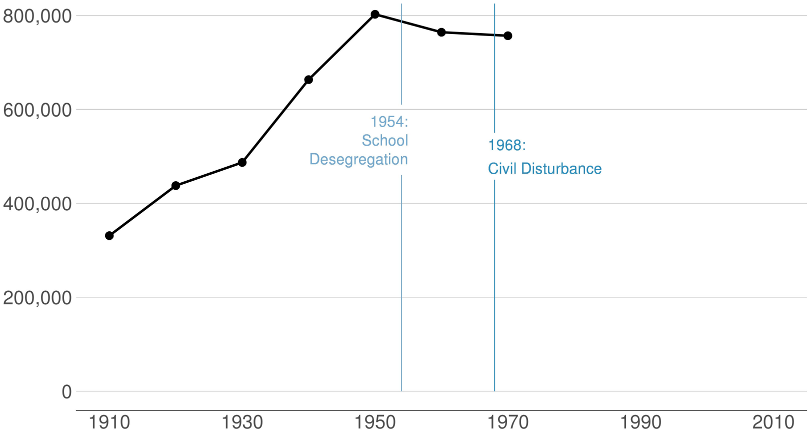
DC Gains Population Through 1950



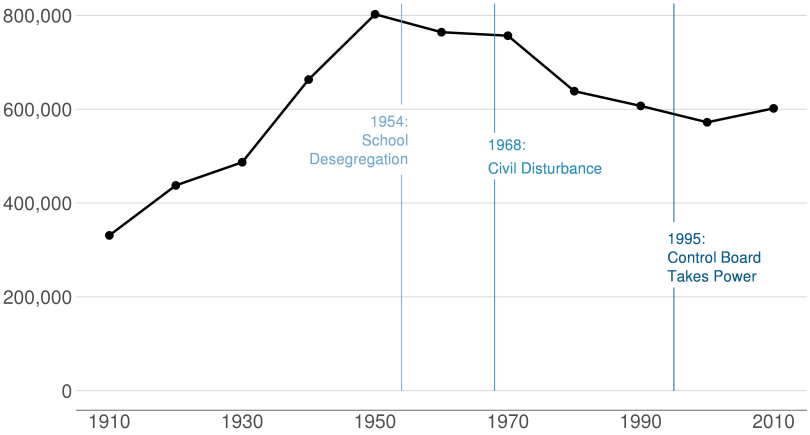
Population Loses Start with Desegregation



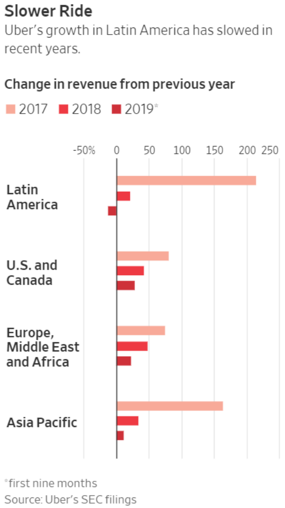
Continue After Civil Disturbance



Population Turns Up After 2000



Something That Should be a Line Chart



Slower Ride

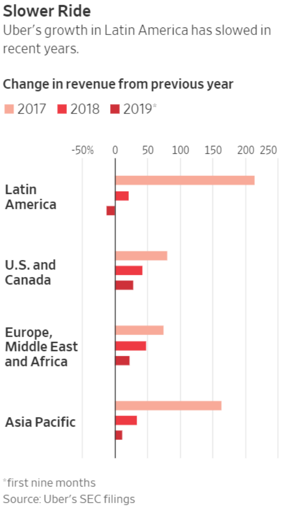
Uber's growth in Latin America has slowed in recent years.

Change in revenue from previous year

*first nine months

Source: Uber's SEC filings

Something That Should be a Line Chart



Slower Ride

Uber's growth in Latin America has slowed in recent years.

*first nine months

Source: Uber's SEC filings

- 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
- Regression tables should have dependent variable means
- Visuals can put in context
    - dates
    - comparative categories
    - baseline mean
    - standard deviation

Contextual

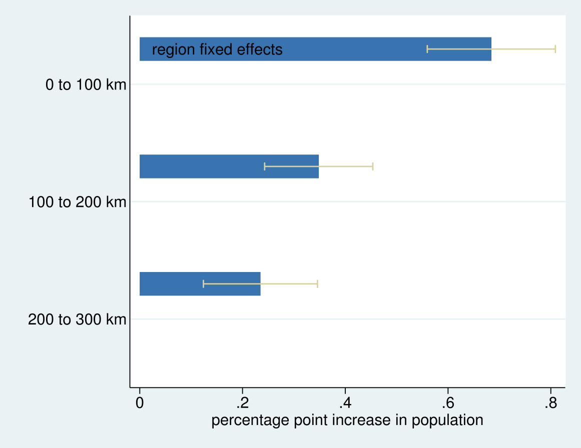
- Very important for magnitudes with which people are not familiar
- Helps us answer "so what" question
- Regression tables should have dependent variable means
- Visuals can put in context
  - dates
  - comparative categories
  - baseline mean
  - standard deviation

What does this mean for your policy brief?

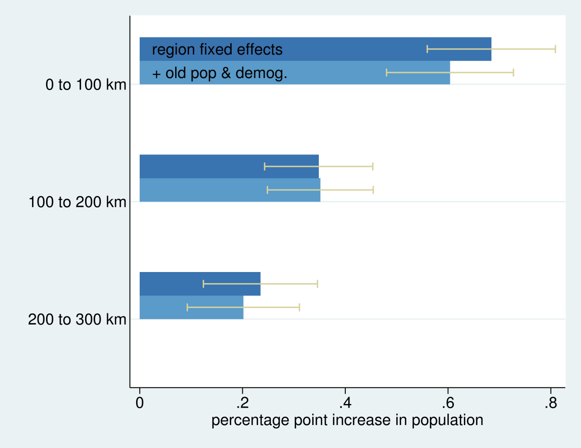
Sequential

- 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

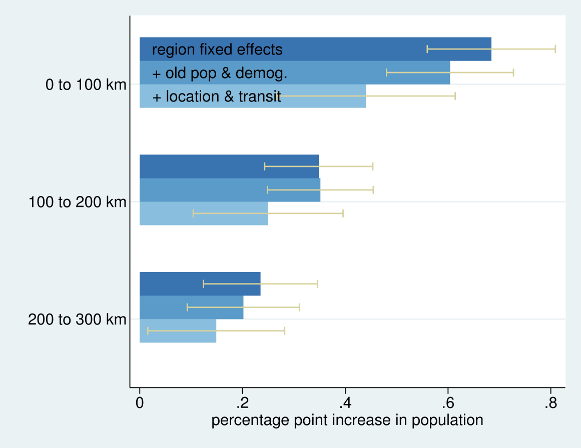
Bars with Error Bars, Building



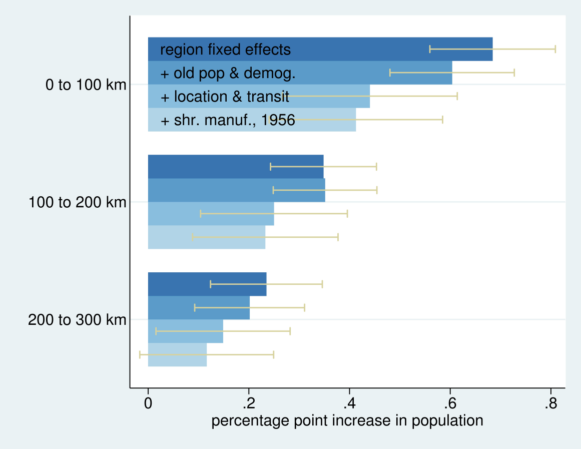
Bars with Error Bars, Building



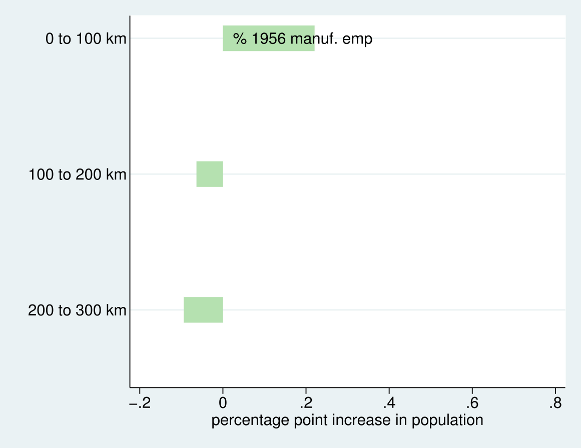
Bars with Error Bars, Building



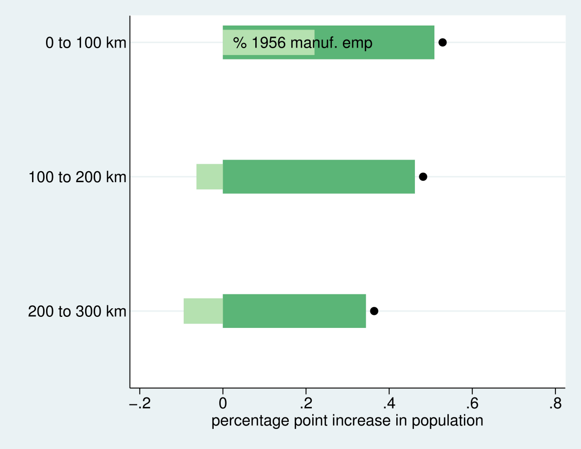
Bars with Error Bars, Building



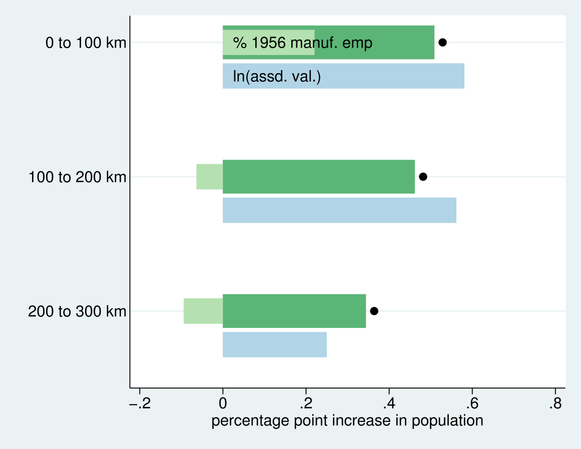
Interaction Effects



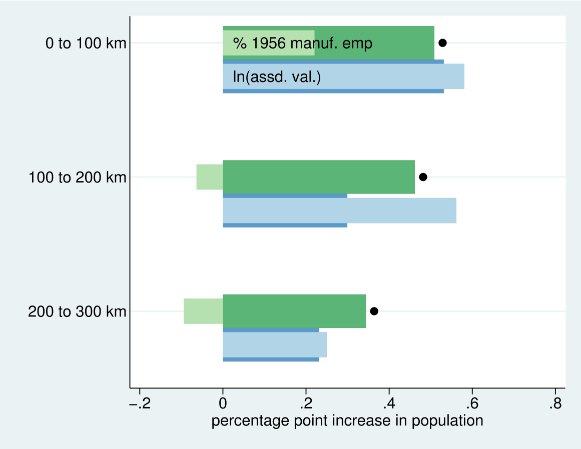
Interaction Effects



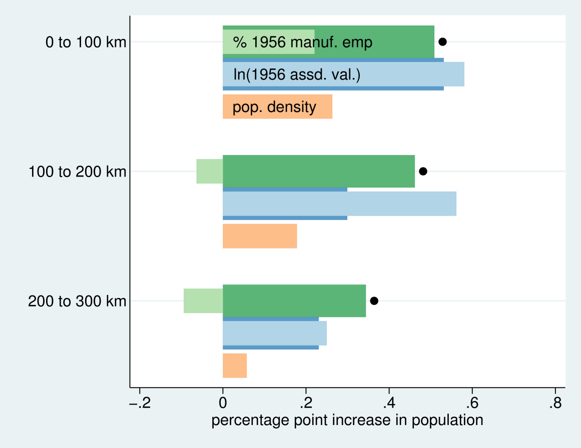
Interaction Effects



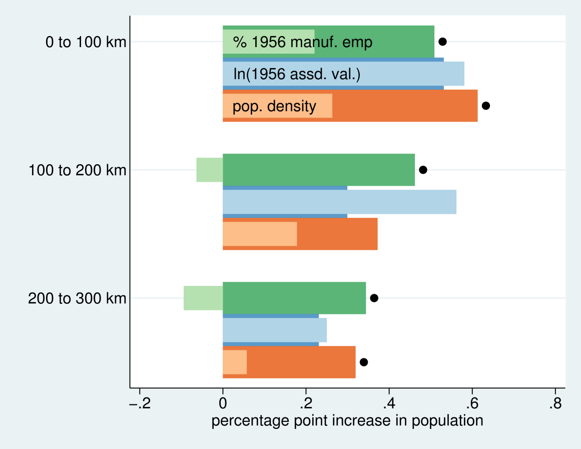
Interaction Effects



Interaction Effects



Interaction Effects



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

```r
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

Multiple Lines

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

- groupvar should be a variable that identifies the type
- Be wary of using too many 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
  - → 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
  - → 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

Annotations are best done on a chart, rather than as a label on the side.

General logic is

```
np <- already.existing.plot +
      annotate(geom = "text",
               x = [x location],
               y = [y location],
               other options -- size, color, etc)
```

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

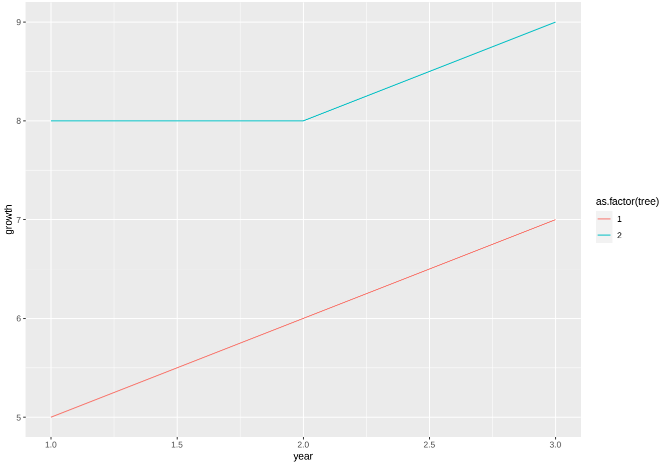
```r
trees <- data.frame(year = c(1,2,3,1,2,3),
                    tree = c(1,1,1,2,2,2),
                    growth = c(5,6,7,8,8,9))
```

3. Line plot of trees

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

3. Line plot of trees

```
tp
```

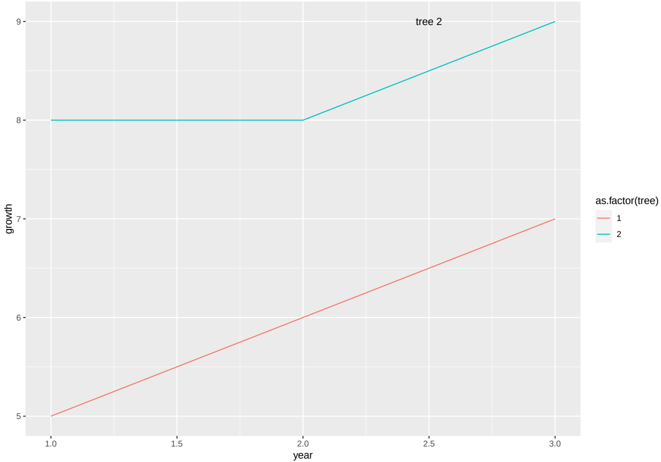


3. Add annotation

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

3. Line plot of trees with annotation

```
tp2
```



3. Alternative, worse ways to annotate

`geom_text()`

- a worse way to put text on one specific point on the plot
- a better way to put text at multiple x/y points

`annotate()`

- can be altered with other geoms – “segment”, “rectangle” and others
- you can change color and many other options

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

```r
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
- → just keep what you need and pivot

5. De-bugging

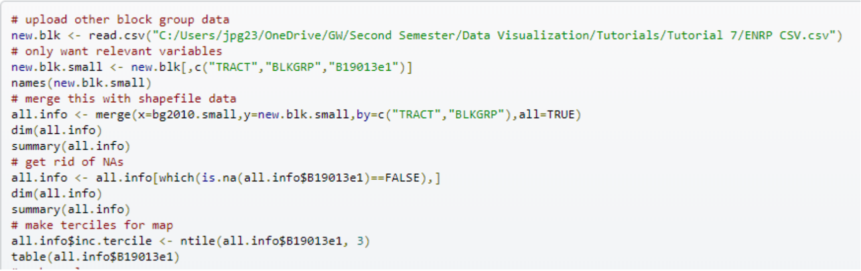
- Write a minimal reproducible example
- Doing this frequently solves your problem
- Two basic methods
  - A. start from scratch
  - B. Remove till problem disappears

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

4.a. Start from scratch method

- Problem: map is not plotting

  Map won't even load



```
# upload other block group data
new.blk <- read.csv("C:/Users/jpg23/OneDrive/GW/Second Semester/Data Visualization/Tutorials/Tutorial 7/ENRP CSV.csv")
# only want relevant variables
new.blk.small <- new.blk[,c("TRACT","BLKGRP","B19013e1")]
names(new.blk.small)
# merge this with shapefile data
all.info <- merge(x=bg2010.small,y=new.blk.small,by=c("TRACT","BLKGRP"),all=TRUE)
dim(all.info)
summary(all.info)
# get rid of NAs
all.info <- all.info[which(is.na(all.info$B19013e1)==FALSE),]
dim(all.info)
summary(all.info)
# make terciles for map
all.info$inc.tercile <- ntile(all.info$B19013e1, 3)
table(all.info$B19013e1)
```

4.a. How to implement start from scratch?

4.a. How to implement start from scratch?

- Are data ok?
- Plot map by itself
- Plot data by themselves
- Plot merged data
- These should help you narrow down the problematic portion of the code

4.b. Remove till problem appears

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

4.b. Remove till problem appears

- 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?

4.b. Remove till problem appears

- 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..

4.b. Remove till problem appears

- 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

Minimal Reproducible Example

- The smallest piece of code that generates your problem
- May need to include data
- Frequently, generating this solves your problem

Next Lecture

- Next week: Guest speaker from LMI, In-class workshop
- I will plan to “drop in” on each workshop group
- If today’s WebEx was not a disaster, I will send out group WebEx invites
- To join all groups I’ll go till 6 – let me know if 5:20 to 6 is no good for you