

Lecture 8: Line Charts

April 1, 2019

Overview

Course Administration

Good, Bad and Ugly

Line Charts

Few on Stories

R Notes

Course Administration

1. Next week: in-class workshop
2. Workshop instructions online under Lecture 6
3. You need to post graphics by April 7 at 3:30
4. Presentation dates are assigned: if your group is not grouped, let me know
5. Anything else?

Class 8, April 1: Good Bad and Ugly

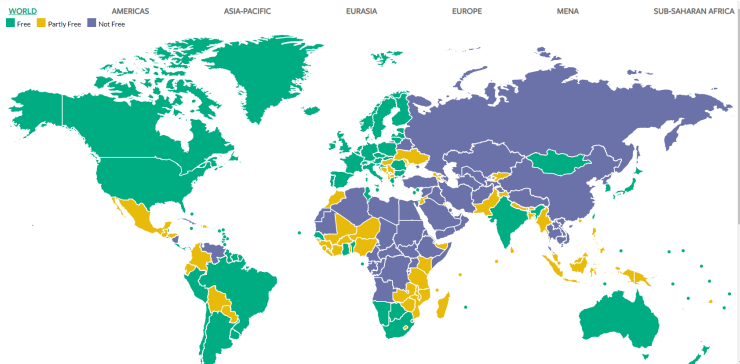
Send by 9 am next Monday. Look for a line chart.

- KE
- AF

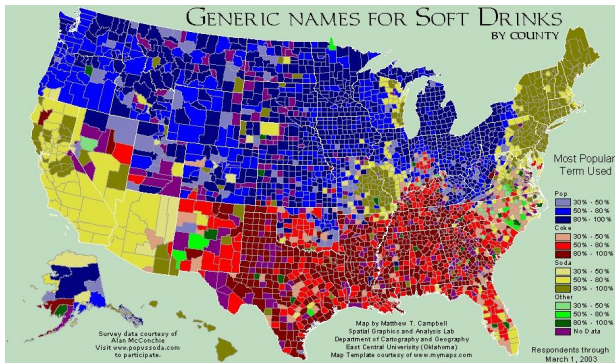
This Week's Good Bad and Ugly

- MD
- GM
- IT

Melissa's Example



Ian's Example



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

How to Call Things out in a Line Chart

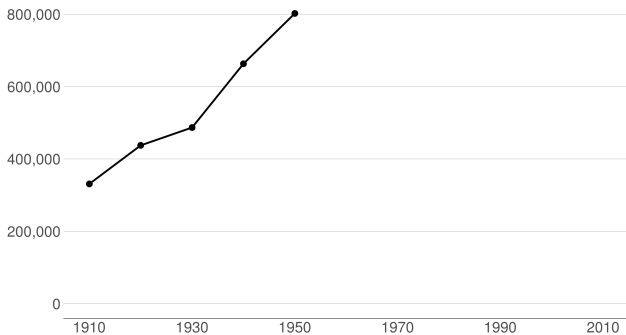
How to Call Things out in a Line Chart

Think back to preattentive processing

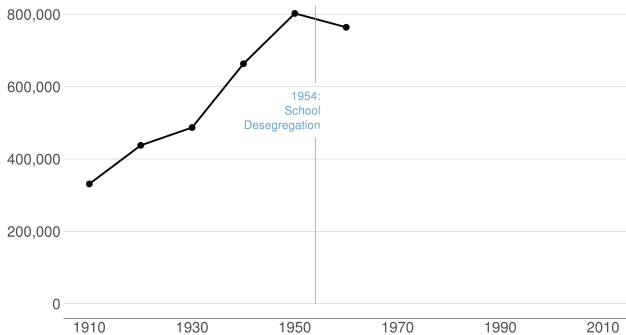
- color
- size
- timing

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

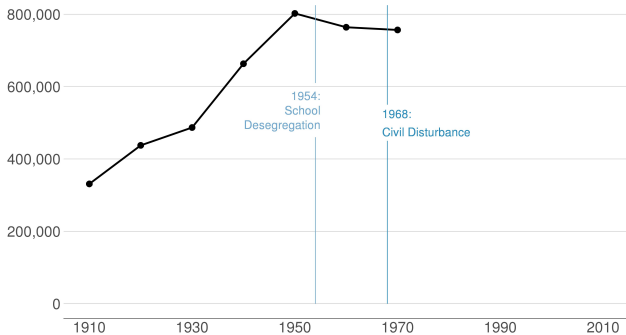
DC Gains Population Through 1950



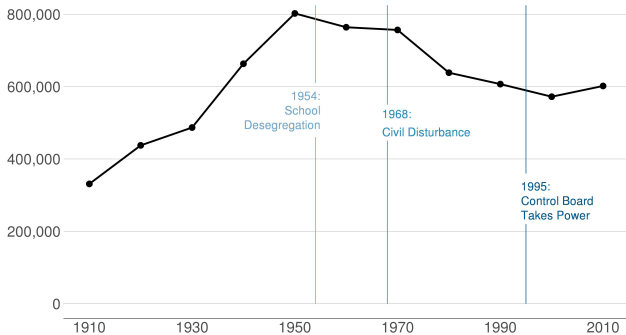
Population Losses Start with Desegregation



Continue After Civil Disturbance



Population Turns Up After 2000



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

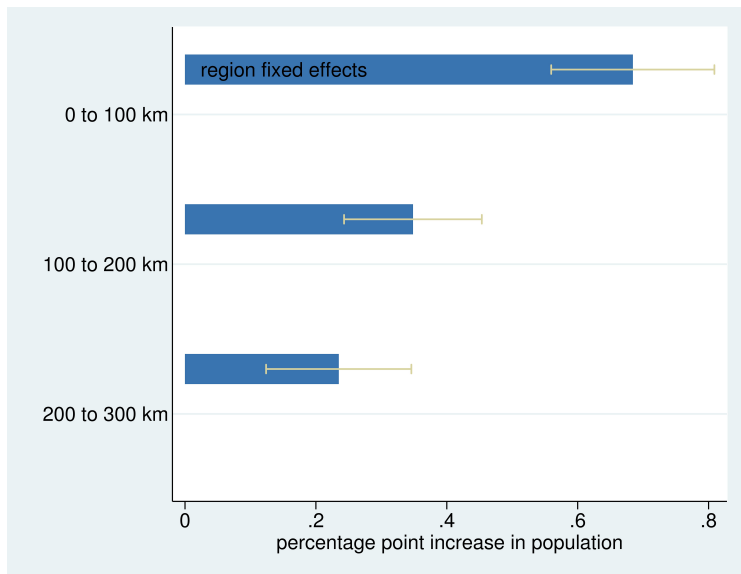
- Very important for magnitudes with which people are not familiar
- Helps us answer “so what” question
- Regression tables should have dependent variable means
- Visuals can put in context
 - dates
 - comparative categories
 - baseline mean
 - standard deviation

What does this mean for your policy brief?

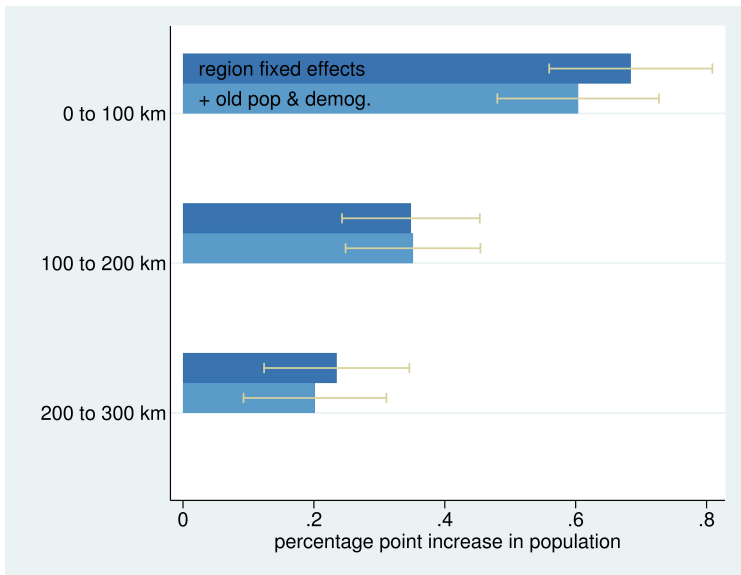
Sequential

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

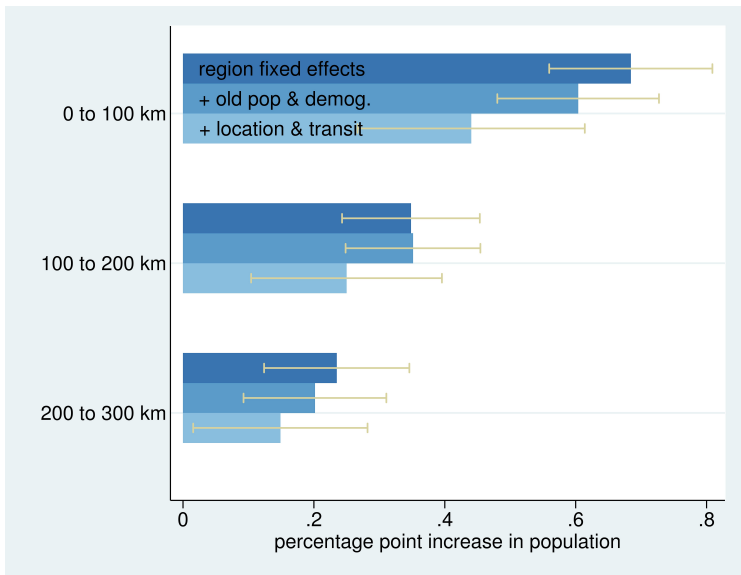
Bars with Error Bars, Building



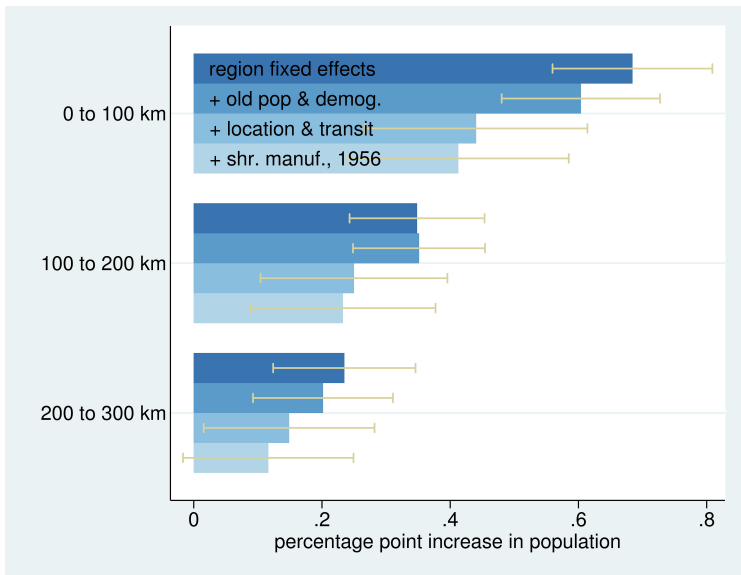
Bars with Error Bars, Building



Bars with Error Bars, Building



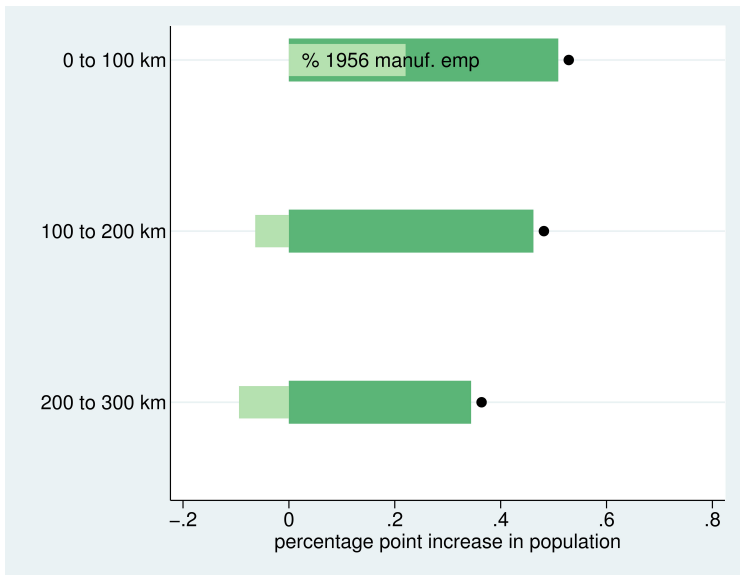
Bars with Error Bars, Building



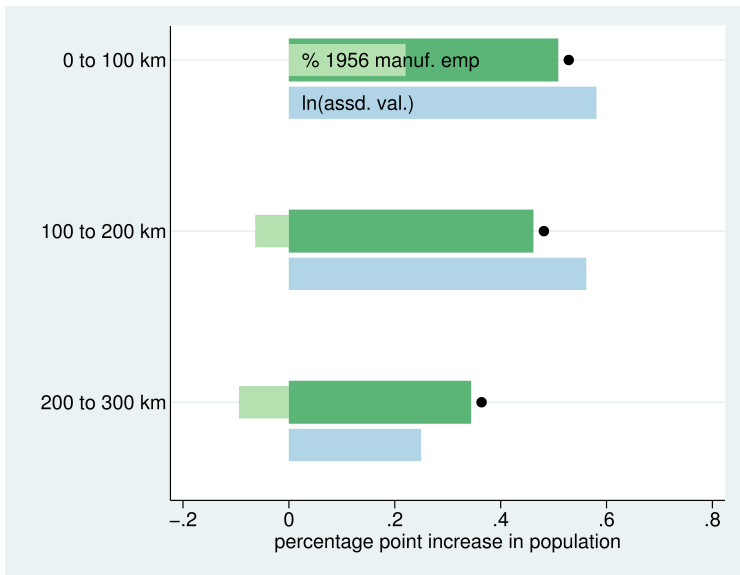
Interaction Effects



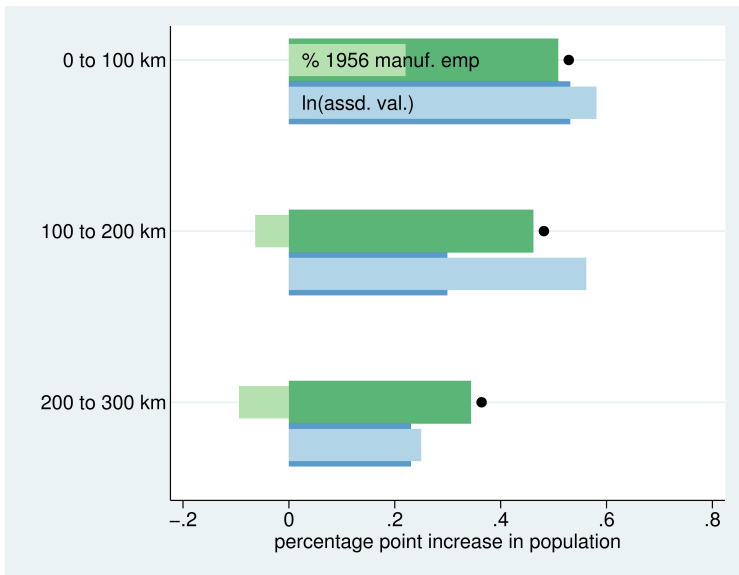
Interaction Effects



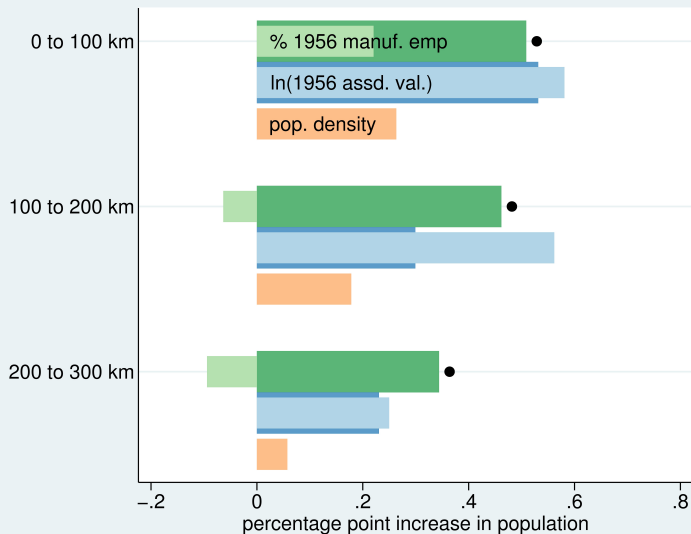
Interaction Effects



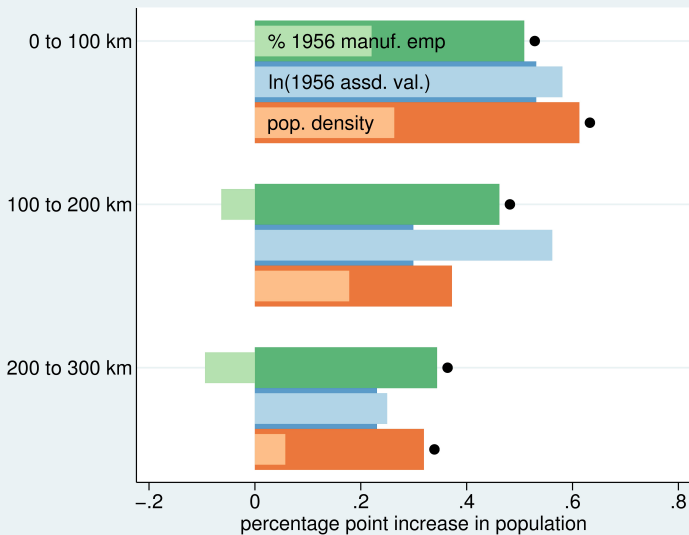
Interaction Effects



Interaction Effects



Interaction Effects



Today in R

Today in R: Line Charts and De-Bugging

1. Line charts and `ggplot`
2. A better function with `ggplot`
3. Simple loop
4. 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

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. A Better ggplot Function

This doesn't work

```
func.a <- function(var1){  
  ploto <- ggplot() +  
    geom_line(data = polys,  
              mapping = aes(x = time, y = var1))  
  ploto  
}  
tester <- func.a(var1 = number.of.ppl)
```

2. A Better ggplot Function

This does work

```
func.a <- function(var1){  
  ploto <- ggplot() +  
    geom_line(data = polys,  
              mapping = aes(x = time, y = !! var1))  
  ploto  
}  
tester <- func.a(var1 = quo(number.of.ppl))
```

What is R Doing?

- ▶ `ggplot` and the whole tidyverse use “non-standard evaluation”
- ▶ This is a custom interpretation of R code
 - ▶ why you can write `geom_line(aes(x = vara, y = varb))` instead of `geom_line(aes(x = df$vara, y = df$varb))`
 - ▶ `quo()` gives a “quosure”, which you can take into the command
- ▶ Then use `!!` to tell R to unquote the thing

3. Simple Loop

```
for(i in 1:3){  
  print(paste0("Hello Mr. ",i))  
}
```

```
## [1] "Hello Mr. 1"
```

```
## [1] "Hello Mr. 2"
```

```
## [1] "Hello Mr. 3"
```

What else can you loop over?

- ▶ You can loop over any list
- ▶ You can create a numeric list with `seq()`
- ▶ You can make a list of things with strings

What else can you loop over?

- ▶ You can loop over any list
- ▶ You can create a numeric list with seq()
- ▶ You can make a list of things with strings

```
things <- c("thing1","thing2","thing3")
for(i in things){
  print(paste0("Look, it's ",i))
}
```

```
## [1] "Look, it's thing1"
## [1] "Look, it's thing2"
## [1] "Look, it's thing3"
```

Loop Disadvantages

- ▶ Loops loop through 1 thing
- ▶ So you can't change variable and label at the same time
- ▶ R discourages you from using loops
- ▶ We'll introduce `lapply()` shortly: a loop but more compact

4. 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
- Check presentation dates