

# Lecture 6: Scatter Plots and R Skills

March 5, 2018

# Overview

Course Administration

Good, Bad and Ugly

Few, Chapter 13

Line Charts in R

# Course Administration

1. Rosa has graded problem sets – thank you
  - block group numbering
2. Grading google sheet now set up: missing later problem sets
3. Have revised syllabus to start with maps next week
4. Schedule consultations: sign up on form, will post online.  
Bring graph sketches at a minimum.
5. Would you prefer tutorial in html?
6. Missing anything else from me?

# Next Week's Good Bad and Ugly

Monday by 9 am. Earlier is ok.

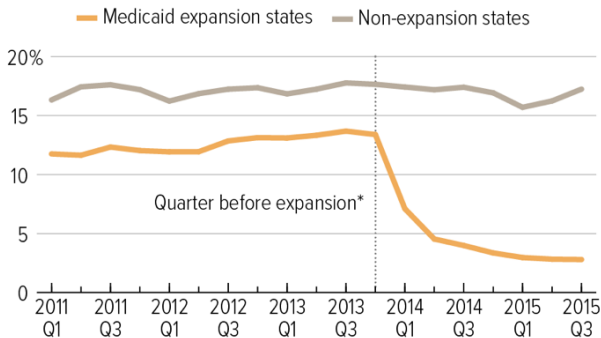
- Raphe Breit
- Meghan Demeter

# This Week's Good Bad and Ugly

- Francisca Alba
- Julia Robertson
- Adam Brooks

## Fran's Example

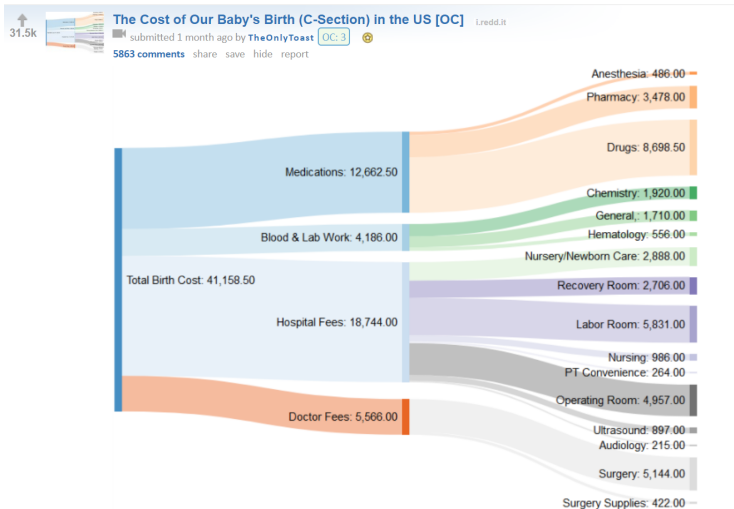
### ACA Medicaid Expansion Reduced Share of Opioid-Related Hospitalizations in Which Patient Was Uninsured



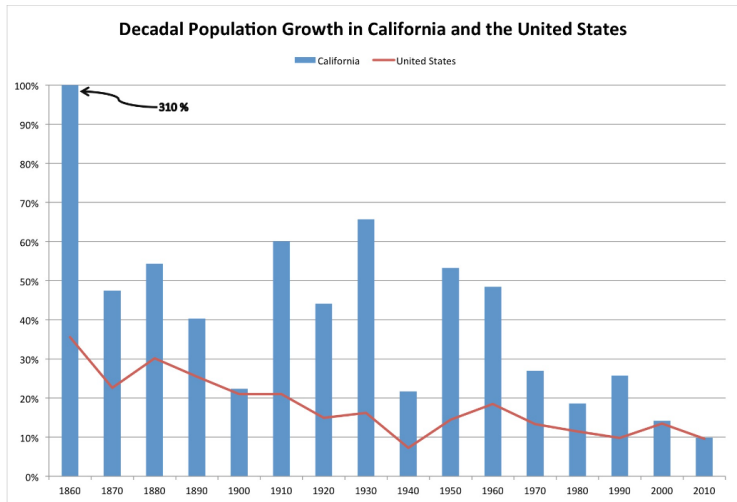
\*The Affordable Care Act (ACA) gave states the option to expand Medicaid to adults with income up to 138 percent of the poverty line starting in 2014.

Source: CBPP analysis of Healthcare Cost and Utilization Project data from the Agency for Healthcare Research and Quality. Analysis includes 26 states for which data are available for all of 2011-2015 and which either expanded Medicaid in January 2014, or had not expanded as of October 2015.

# Julia's Example



# Adams's Example





# Few:

## Telling Compelling Stories with Numbers

# Today

1. Few Chap 10: Non-Data Components
2. Few Chap 13: Stories with Numbers

# 1. Non-Data Components

- Axes
- Aspect Ratio
- Data region

# Axes

“... the only non-data components that are routinely useful in graphs.” (p. 247)

- Two is ok; one if possible
- Box the region when you need to separate it from other things, such as text

# Aspect Ratio

- Ratio of data region's height to width
- Usually defined by space on slide or paper
- But it may be worth it to sometimes think a bit more about this
- Time series convention is to have width greater than height

# Choices of Aspect Ratio

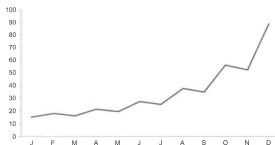


FIGURE 10.82 This graph has an aspect ratio of 2 to 1, or 2.

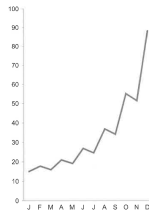


FIGURE 10.85 This graph has an aspect ratio of 1 to 1.5, or 0.67.

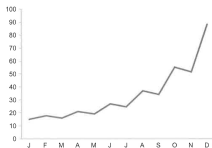


FIGURE 10.83 This graph has an aspect ratio of 1.5 to 1, or 1.5.

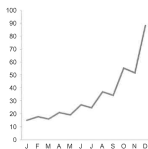


FIGURE 10.84 This graph has an aspect ratio of 1 to 1, or 1.

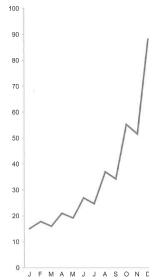


FIGURE 10.86 This graph has an aspect ratio of 1 to 2, or 0.5.

# Data Region

- Just go with white
- Unless you want to draw readers' attention to the region
  - use gray or light yellow
  - or lightly color non-data region (Stata's default)
- Don't put pictures or weird gradients behind

## 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

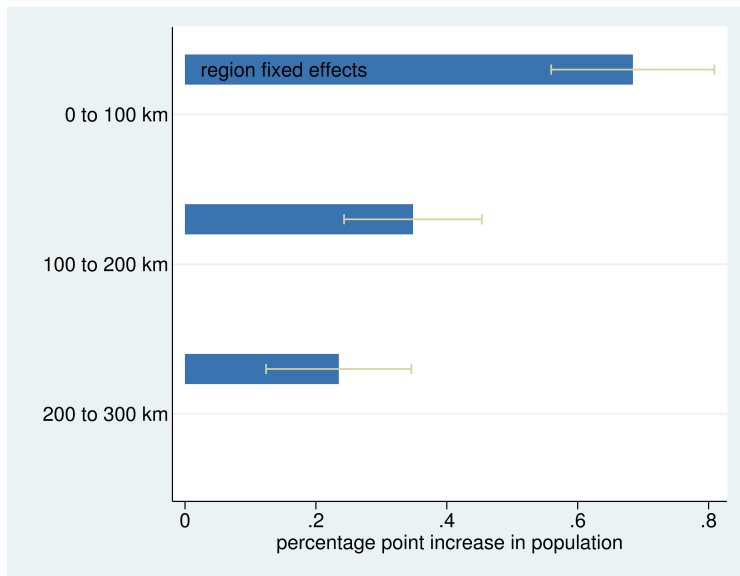
# Contextual

- 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
- Example: Cellini’s presentation on higher education advertising

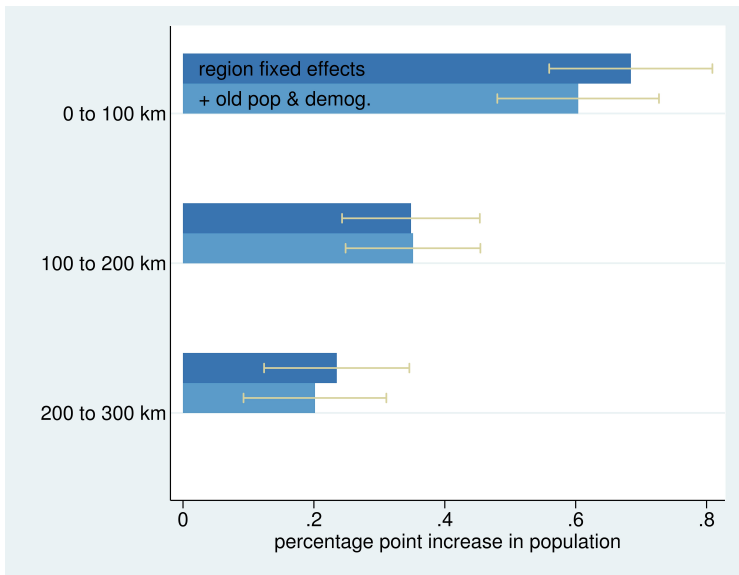
# Sequential

- 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

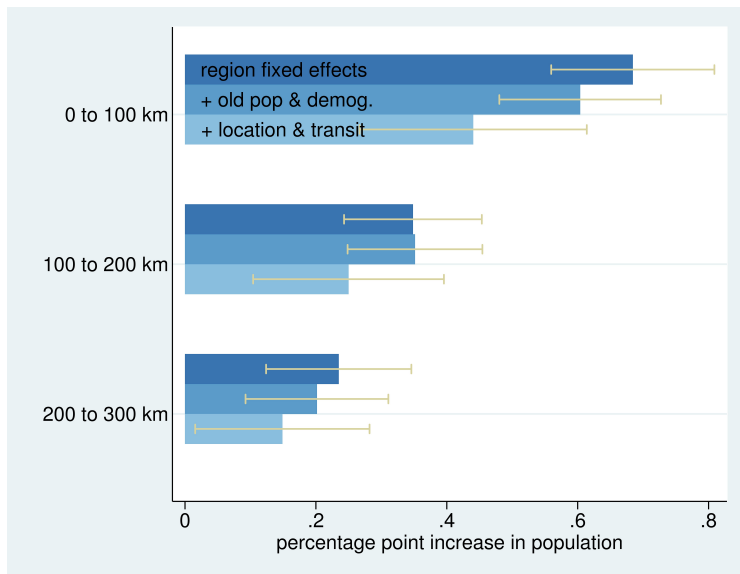
## Bars with Error Bars, Building



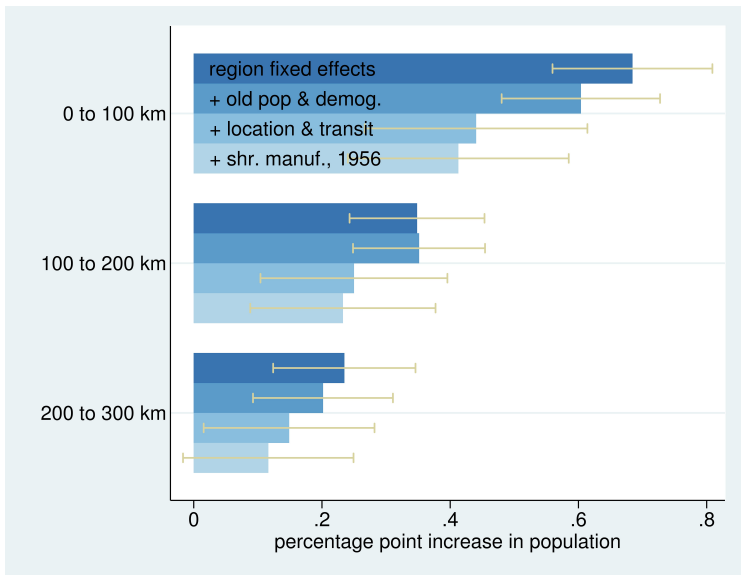
## Bars with Error Bars, Building



## Bars with Error Bars, Building

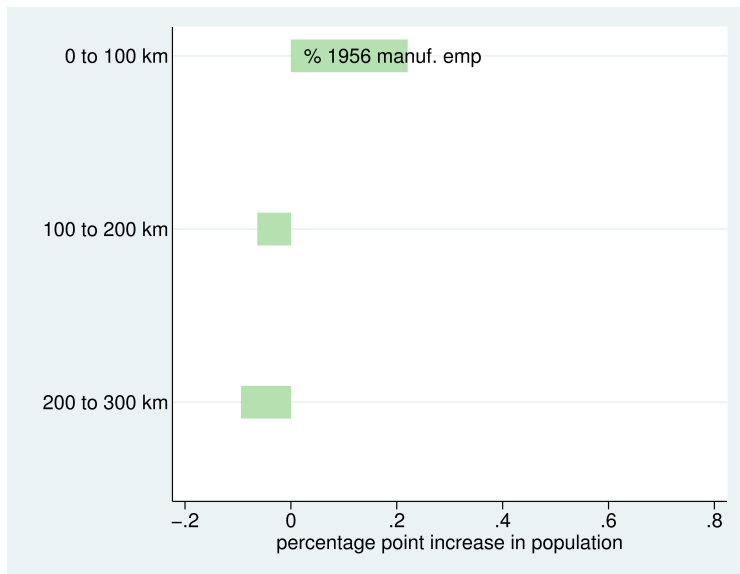


## Bars with Error Bars, Building

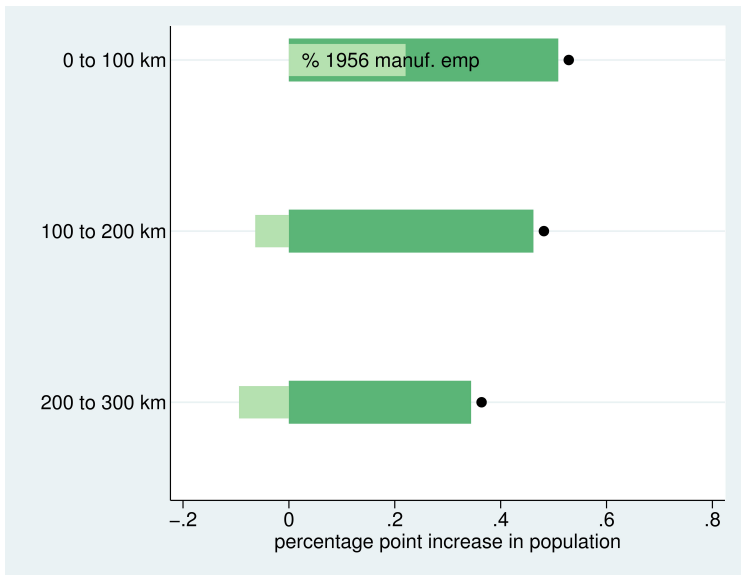




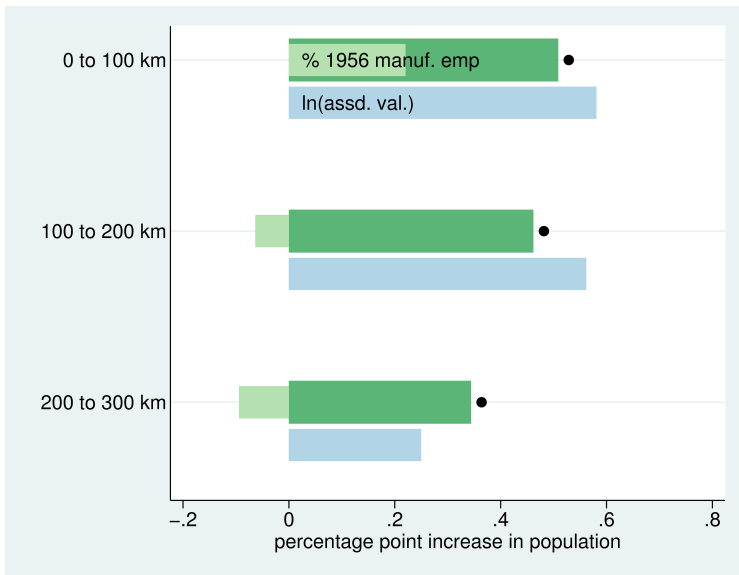
# Interaction Effects



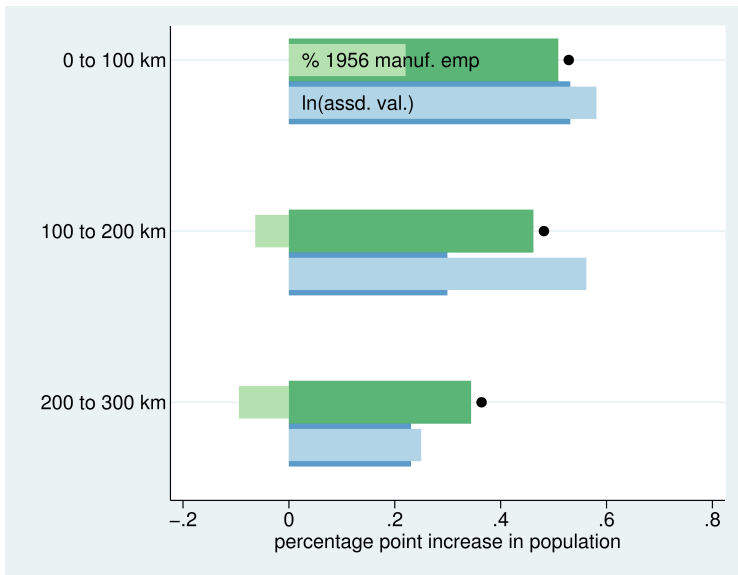
## Interaction Effects



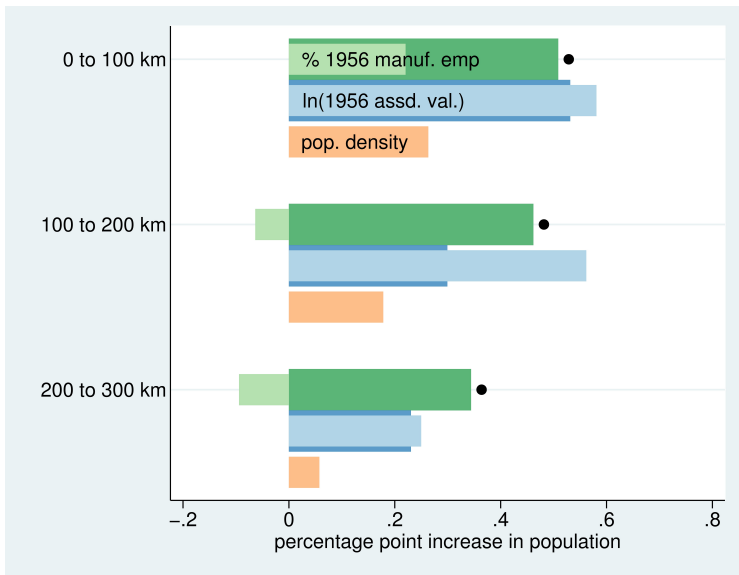
## Interaction Effects



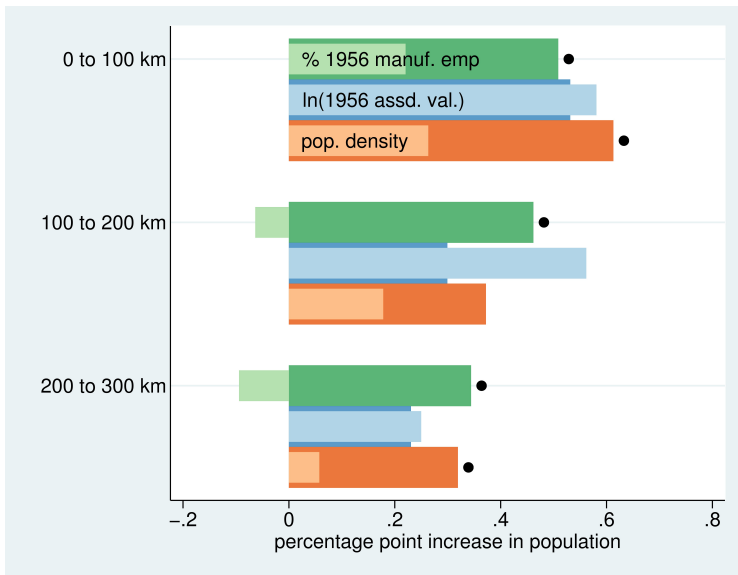
## Interaction Effects



## Interaction Effects



## Interaction Effects



# Scatter Charts in R

# Today's Goals

- Data Manipulation
  - Alternative methods for subset
  - `substr()`
  - Binning data
- Scatter graphs, via `ggplot`
  - `geom_point()`



## Subset generally

- ▶ `Subset()` may not work well in more complex commands
- ▶ But you can subset more simply
- ▶ choose row or column you'd like to keep and tell R

## Subset example, old data frame

```
old.df <- data.frame(c1 = c("1","2","3","4"),  
                     c2 = c("5","6","7","8"),  
                     c3 = c("a","b","c","d"),  
                     c4 = c("e","f","g","h"),  
                     c5 = c("9","10","11","12"))  
old.df
```

```
##   c1 c2 c3 c4 c5  
## 1  1  5  a  e  9  
## 2  2  6  b  f 10  
## 3  3  7  c  g 11  
## 4  4  8  d  h 12
```

## Subset example: Keep only columns 1, 3, and 5

```
old.df
```

```
##   c1 c2 c3 c4 c5
## 1  1  5 a  e  9
## 2  2  6 b  f 10
## 3  3  7 c  g 11
## 4  4  8 d  h 12
```

```
new.df <- old.df[,c(1,3,5)]
new.df
```

```
##   c1 c3 c5
## 1  1  a  9
## 2  2  b 10
## 3  3  c 11
## 4  4  d 12
```

## Subset with which

What if you want to keep only based on a condition in a row?

- ▶ Keep only observations where the state is California
- ▶ Keep only people > 25 years old
- ▶ Keep only redheads

```
new.df <- old.df[which(old.df$state == "06"),]
```

# Substring

- ▶ To extract a bit of string from a longer variable
- ▶ Can be very useful
- ▶ Need to know where you start cutting
- ▶ And where you stop cutting
- ▶ Syntax is

```
substr([variable],[start],[stop])
```

# Binning data

Basic idea here is to

- ▶ decide where you want to make the bins
- ▶ assign each observation to a bin
- ▶ take the mean by bin
- ▶ plot these binned means

# Binning data in R

## Steps

- ▶ make a vector that has the discrete breaks for bins
- ▶ assign each observation (row) to a bin
- ▶ take the mean by bin (`ddply`)
- ▶ plot these binned means (`ggplot`)

You know the last 2, so we'll focus on the first two

## 1. Set up Boundaries for Intervals/Bins

You could type them in directly

```
breaks <- c(1,2,3)
breaks
```

```
## [1] 1 2 3
```

Or make a sequence:

```
breaks <- c(seq(8.4,9.6,0.2))
breaks
```

```
## [1] 8.4 8.6 8.8 9.0 9.2 9.4 9.6
```



## 2. Assign each observation to a row

cut() function

- ▶ designate the variable that you'd like to work on
- ▶ specify breaks
- ▶ tell R whether you want to include observations with values equal to the lowest break
- ▶ tell R whether you want to include observations with values equal to the highest break

```
block.groups$ln.inc.bin <- cut(block.groups$ln.med.hh.inc,  
                               breaks,  
                               include.lowest = T,  
                               right=FALSE)  
table(block.groups$ln.inc.bin)
```

# Try Today's Tutorial

- Ask questions if the command doesn't make sense
- Go forth!

# Next Lecture

- Next week: spring break
- For March 19: Read mapping notes
- Next policy brief deadline: April 9 for draft