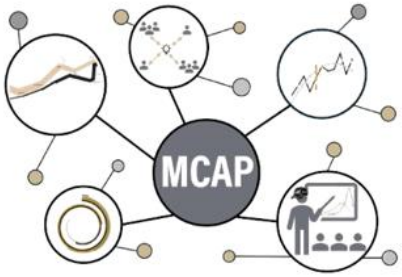
Data Visualization

Trenton D. Mize

Sociology & Statistics (by courtesy)



THE METHODOLOGY CENTER

PURDUE UNIVERSITY

Contents

INTRODUCTION 1

Assessing models via visualizations 2
Good data visualization balances art and science 6

1. LOOK AT YOUR DATA 7

Histograms 8
Kernel density plots 10
Box (and whisker) plots 15
Box and whisker plot rules 16

2. USE GRAPH TYPES PEOPLE READ ACCURATELY 19

Pie charts 20
Accuracy of perceptual tasks 22
Stacked bar charts 25
Bar charts 29

3. MAXIMIZE THE DATA-INK RATIO, WITHIN REASON (MAYBE) 31

The data-ink ratio 32
Dot plots 34
Advantages of dot plots 35

4. SHOW UNCERTAINTY 39

Moving forward: building blocks 43
Break 1: group task 46

5. MAKE GROUPS EASY TO TELL APART 49

General advice: color 53
General advice: color *54*
Nominal scheme color limits *56*
Creating figures considerate of colorblindness *58*
Nominal and ordinal color palettes *59*
6. CHOOSE AXES THAT FIT THE DATA 61
Should your y-axis always start at zero? 62
Are the "y-axis zero fundamentalists" right? *63*
Why the y-axis doesn't have to cover the range of the data *67*
Guidelines for choosing axis values *71*
7. DESIGN FOR YOUR AUDIENCE 73
Titles and axis labels 74
General advice: talks vs articles 75
Talks vs articles 76
General advice: fonts 77
Serif and sans serif fonts *78*
Graphics schemes in Stata *81*
Graphics themes in R *82*
General advice: graphics file formats 86
8. USE GRAPHS, NOT TABLES, FOR MODEL RESULTS 87
Why visualize model results? 88
Giant regression tables in talks = ☹ *93*
Coefficient plots 94

Coefficient plots: not just for coefficients *98*
Break 2: group task 100
9. SHOW WHAT THE MODEL PREDICTS 103
Plots of Model Predictions 104
Plotting predictions for nominal vs continuous variables *108*
Confidence intervals and statistical significance 112
Advice on confidence intervals and standard errors *119*
10. DON'T ASSUME LINEARITY 121
Scatterplots of two continuous variables 122
Binned scatterplots 125
LOWESS: locally weighted scatterplot smoothing *129*
PRINCIPLES OF GOOD DATA VISUALIZATION 134
CONCLUSION 139
Go forth and make pretty graphs 140
REFERENCES & RECOMMENDATIONS 141
General recommendations (books) 141
General recommendations (articles) 142
APPENDIX 143
Assessing Nonlinearities & Model Specification Using Predictions 143
Treating continuous predictors as nominal 144
Comparing nominal and continuous specifications *146*
Specifying ordinal predictors 148

Choosing confidence intervals or standard errors *151*
Use logical colors *152*
Graphics themes in R and color palettes 153
The `cleanplots` R package *154*
Stata graphics and the `cleanplots` graphics schemes *156*
Presenting multiple model results *157*
Incorporating distributional information 158
Misleading example: area ≠ rate *159*
Misleading examples: area ≠ rate or price *160*
General advice: 3D graphics *162*
3D graphs can be misleading *163*

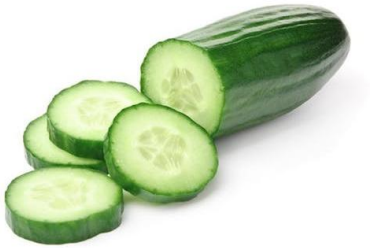
Introduction

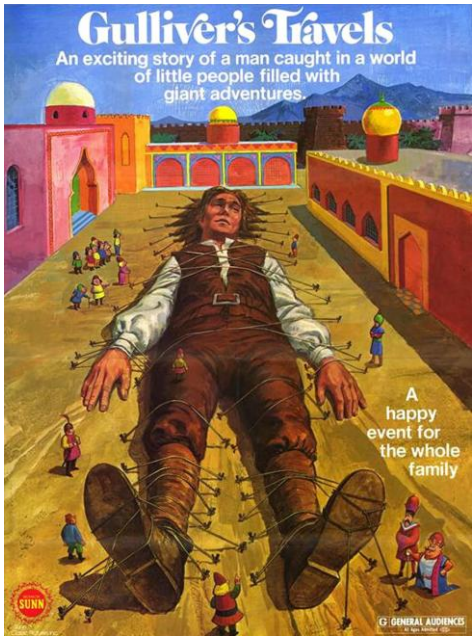
1. Vital to understand data before analyzing it
    a. Is coding correct? Is distribution skewed? Are there clear relationships between variables? Etc.

2. These types of questions are easy to answer/process visually
    a. The human brain can much more quickly understand visual information than numerical or textual information (Medina 2008)

"The graphical method has considerable superiority for the exposition of statistical facts over the tabular. A heavy bank of figures is grievously wearisome to the eye, and the popular mind is as incapable of drawing any useful lessons from it as of extracting sunbeams from cucumbers." (*Farquhar and Farquhar 1891*)







Assessing models via visualizations

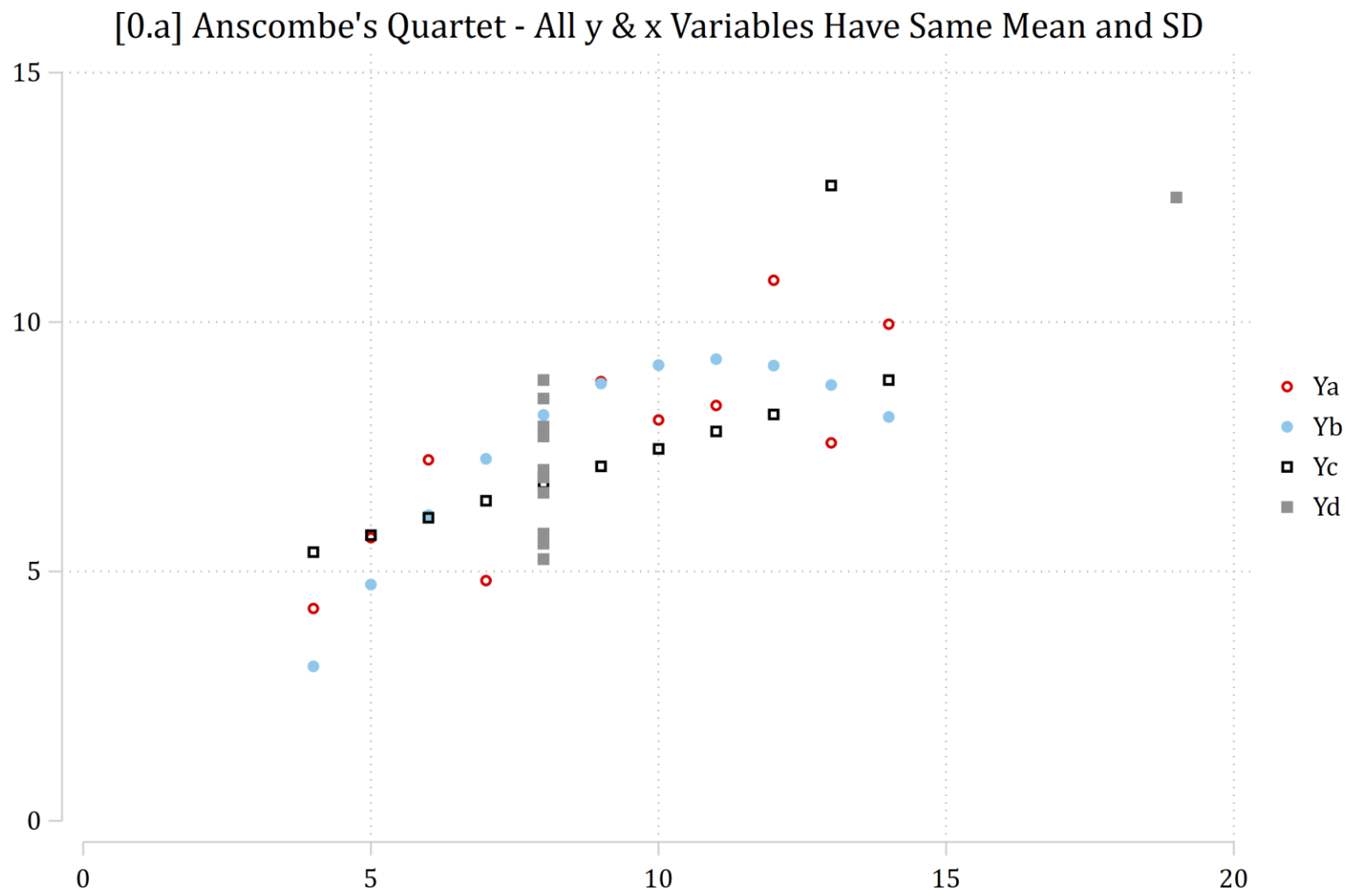
Anscombe's (1973) quartet

Variable	Mean	SD	Min	Max
Yb	7.50	2.03	3.10	9.26
Yc	7.50	2.03	5.39	12.74
Yd	7.50	2.03	5.25	12.50
Ya	7.50	2.03	4.26	10.84
Xabc	9.00	3.32	4.00	14.00
Xd	9.00	3.32	8.00	19.00

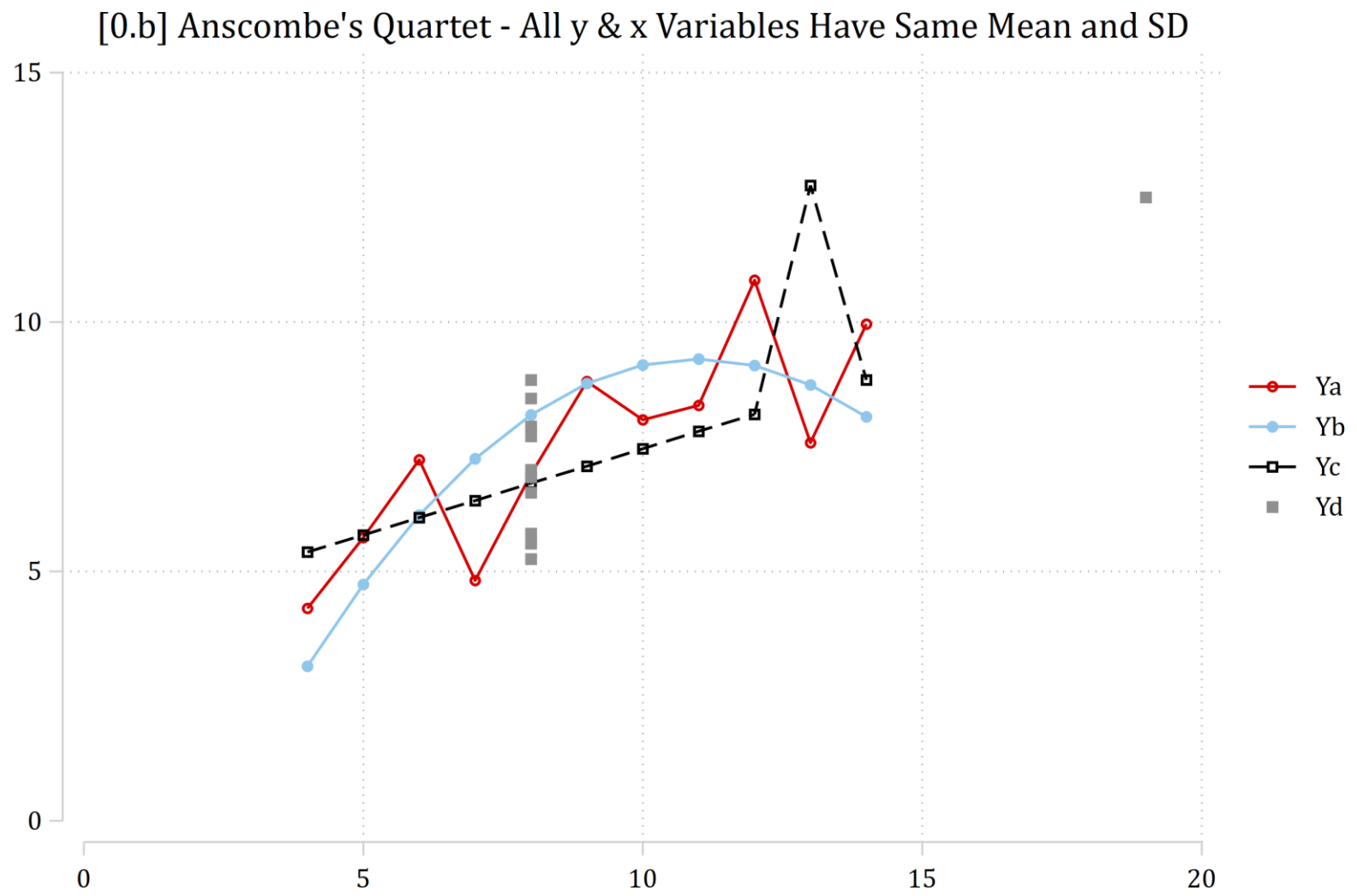
Linear Regression Estimates

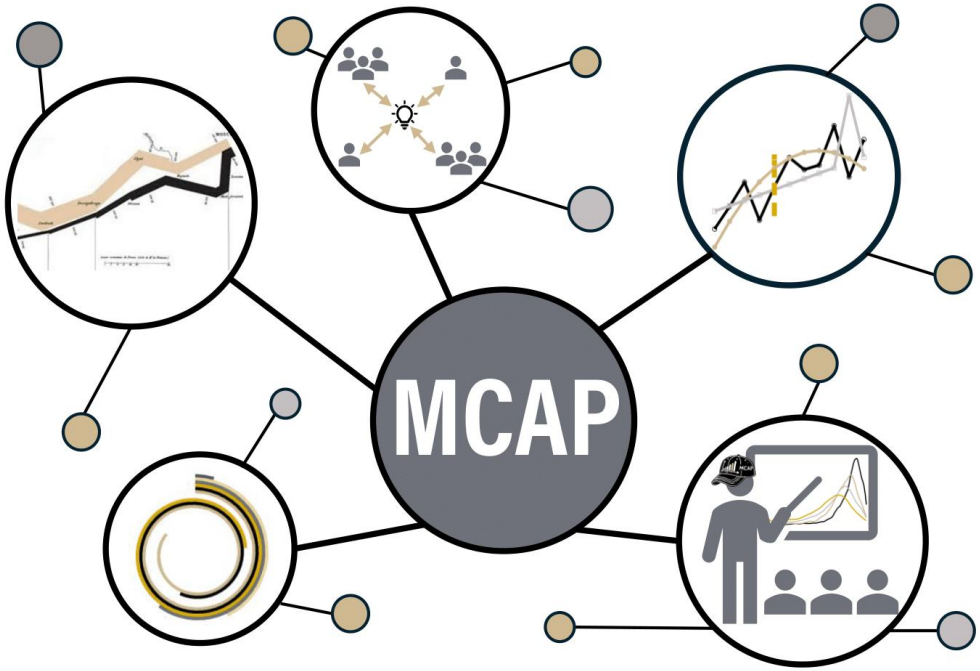
	(1) Ya	(2) Yb	(3) Yc	(4) Yd
Xabc	0.50**	0.50**	0.50**	
	(0.12)	(0.12)	(0.12)	
Xd				0.50**
				(0.12)
_cons	3.00*	3.00*	3.00*	3.00*
	(1.12)	(1.13)	(1.12)	(1.12)
N	11	11	11	11

```
ggplot(anscombe) + geom_point(aes(x = Xabc, y = Ya)) + …
twoway (scatter Ya Xabc) (scatter Yb Xabc) (scatter Yc Xabc) (scatter Yd Xd), …
```



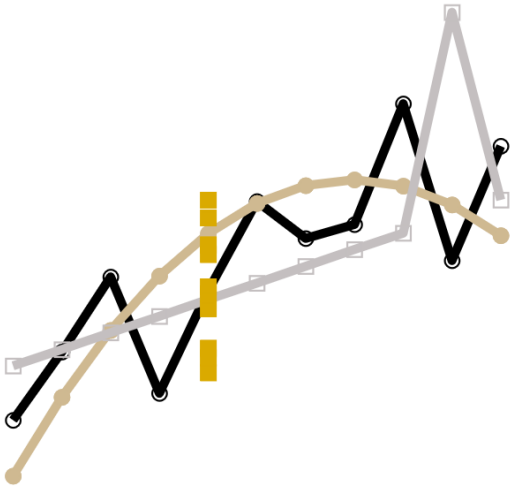
```
ggplot(anscombe) + geom_line(aes(x = Xabc, y = Ya)) + …
twoway (connected Ya Xabc) (connected Yb Xabc) (connected Yc Xabc) …
```





THE METHODOLOGY CENTER

PURDUE UNIVERSITY



*Read about The Methodology Center graphic

Good data visualization balances art and science

"Data visualization is part art and part science. The challenge is to get the art right without getting the science wrong, and vice versa." *(Wilke 2019 p.1)*

1. Too many data visualization guides lean on aesthetic opinions only
    a. Aesthetics are important…

2. But there is a large scientific literature on what makes an **effective** data visualization

3. Goal is to learn general principles to guide any data or model visualization
    a. To make visualizations that are both effective and beautiful

4. Along the way, add to your toolbox of visualization strategies
    a. Learn what tools are effective for what types of data and situations

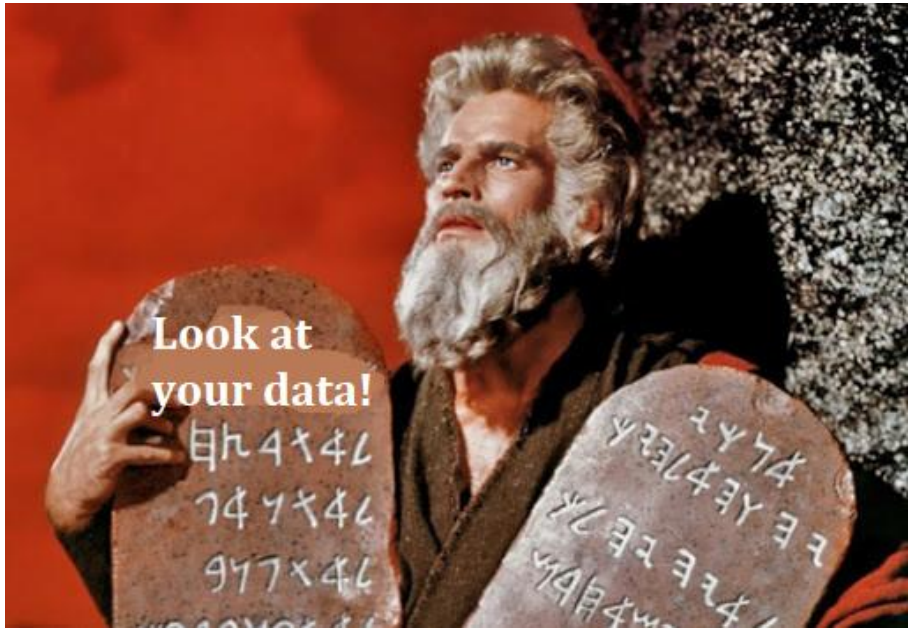




Principles of good data visualization

1. Look at your data

a. The first commandment of data analysis is look at your data
   i. Problems and/or quirks with the data are easily ascertained
   ii. Modeling strategies often become clear only after examining the data



Histograms

1. Histograms have the advantage of being intuitive and widely used
    a. Skew is easy to ascertain via a histogram

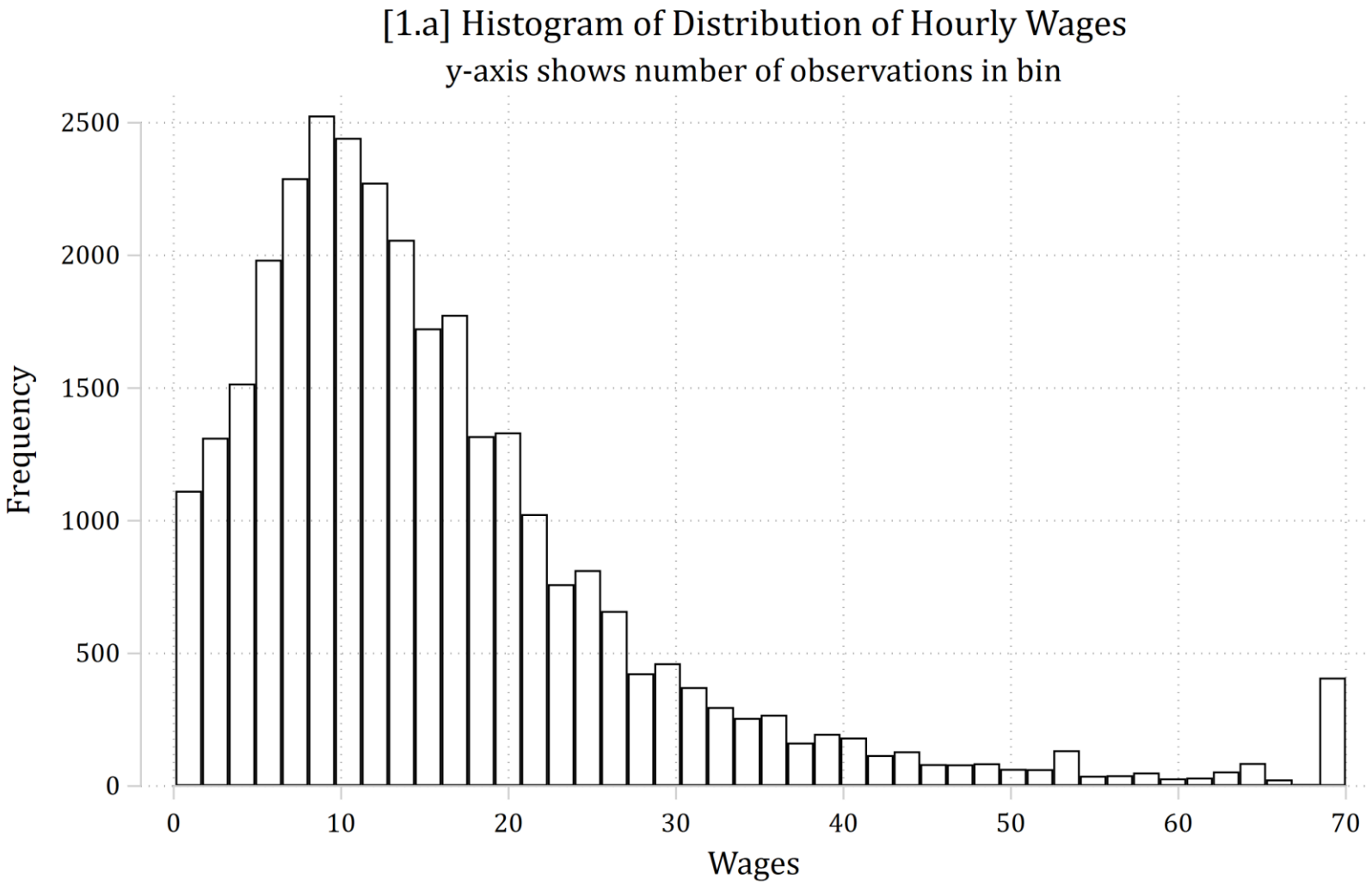
2. Histograms "bin" values within a certain range and plot the quantity within that bin as height: higher bars represent ranges that are more frequent in the data

Figure A. Distribution of the Log of Hourly Wages for All Employed Women (GSS)



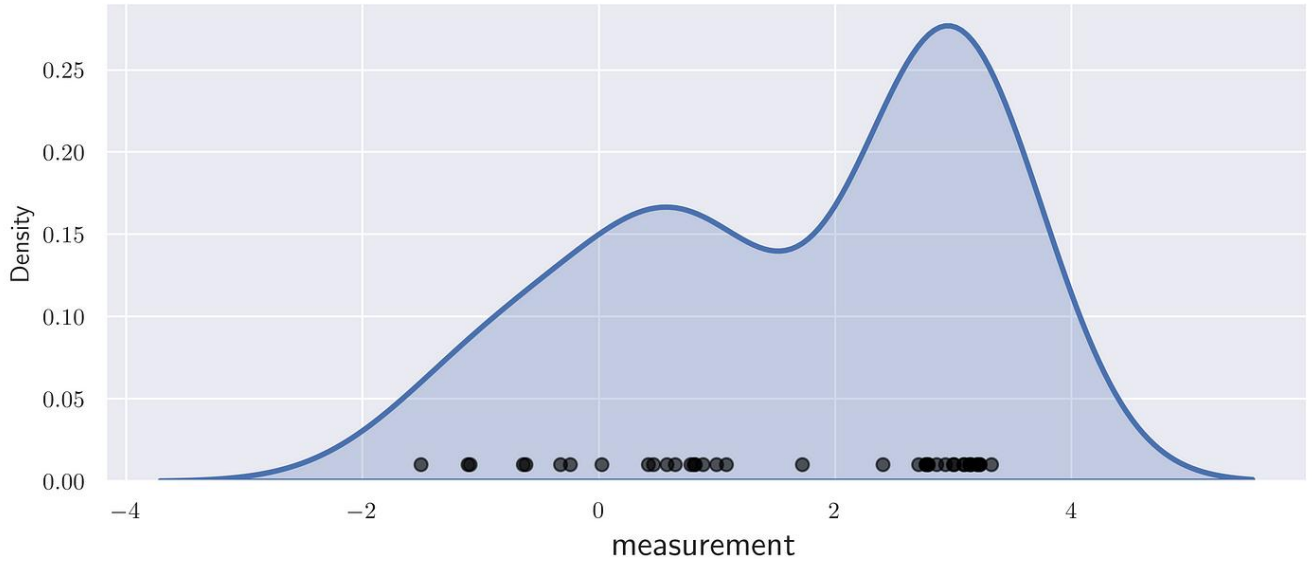
From Mize (2016)

```
ggplot(gss, aes(x = wages)) + geom_histogram(fill = NA, color = "gray20")
hist wages, freq fcol(none) xlab(0(10)70) …
```

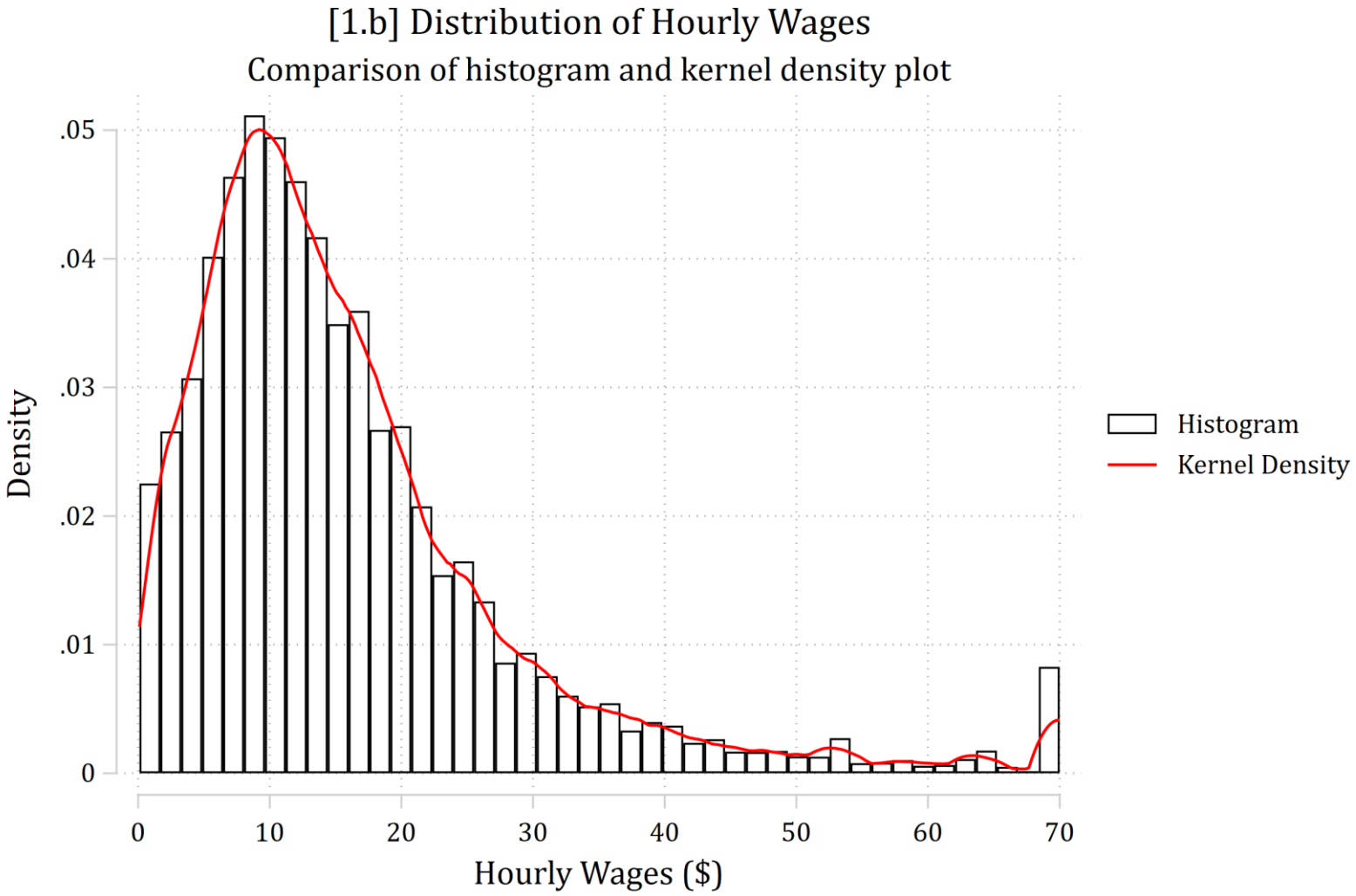


Kernel density plots

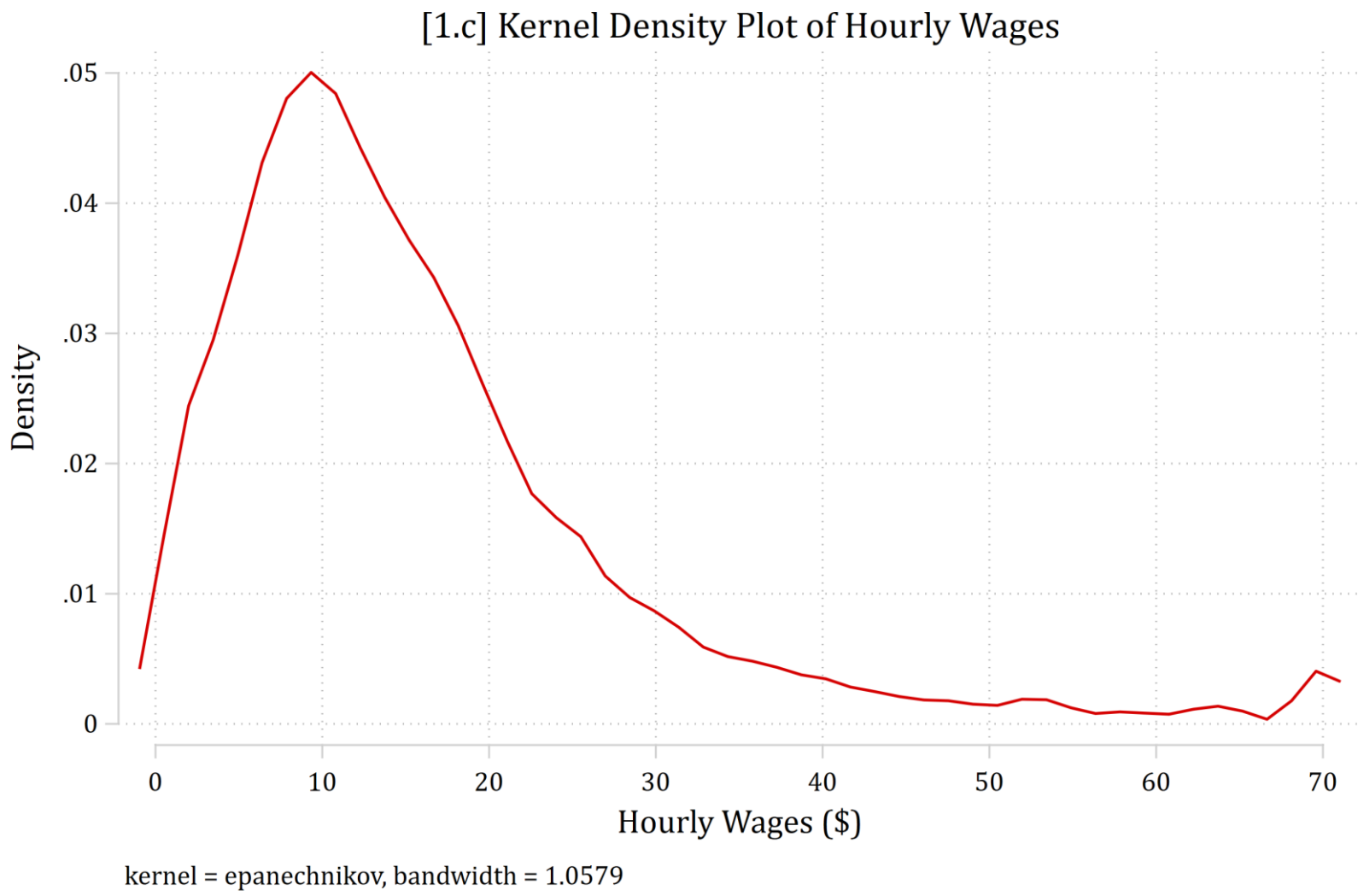
1. Kernel density plots provide smoothed lines showing the distribution of a variable
    a. Think of these as a smoothed histogram
    b. Rather than the height of individual bins at different values showing frequency/percent, the height of the smooth line illustrates density of the data at that value
        i. NOTE: density ≈ proportion of observations near that value



```
ggplot(gss, aes(x = wages)) + geom_histogram(aes(y = after_stat(density))) + geom_density( )
twoway hist wages, fcol(none) || kdensity wages, col(red) || , …
```



```
ggplot(gss, aes(x = wages)) + geom_density( )
kdensity wages, xlab(0(10)70) …
```



Published example

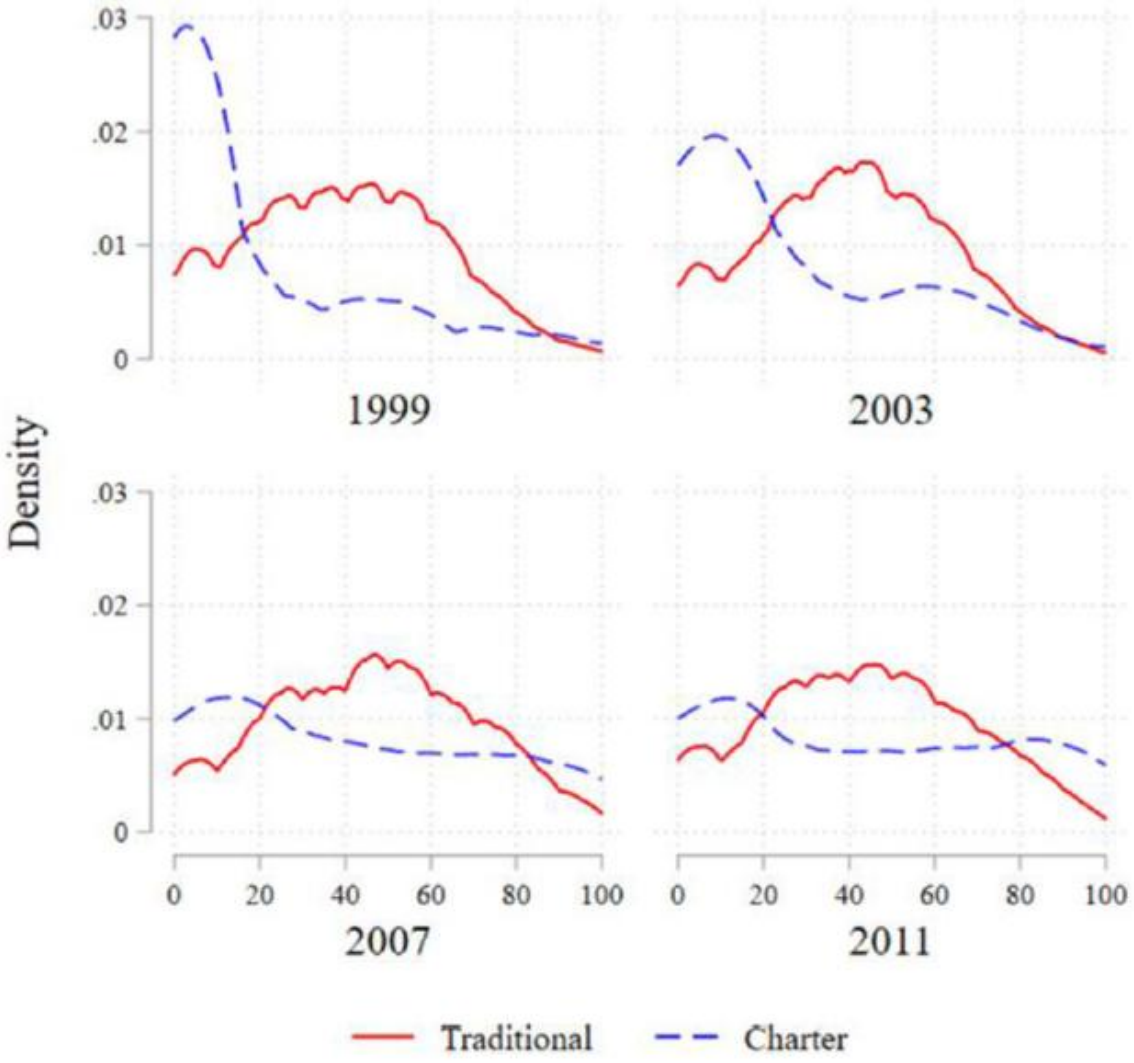
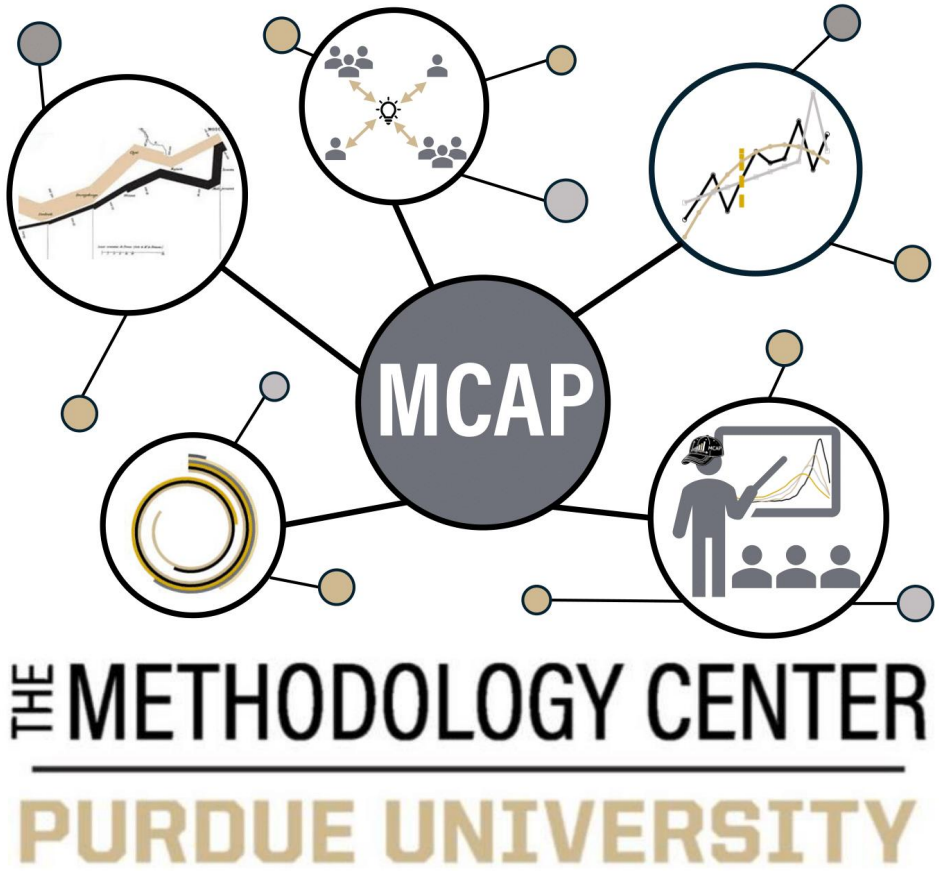


Fig. 2. Distribution of college-Going Rates, by Sector and Year.

From Renzulli, Boylan, and Paino 2023

Note, to make multi-panel graphs like this use **`graph combine`** in Stata and `ggplot2's` faceting features in R

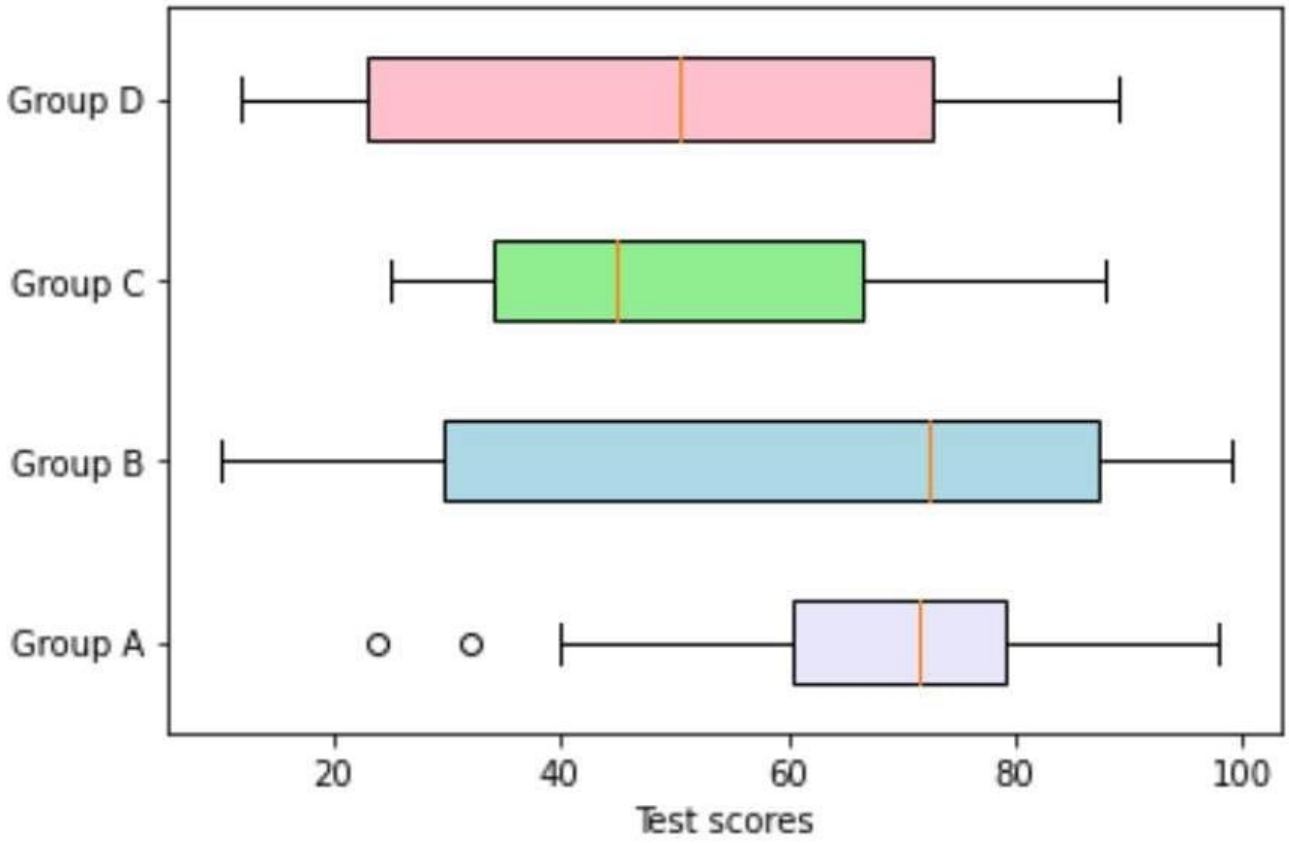




*Read about The Methodology Center's graphic

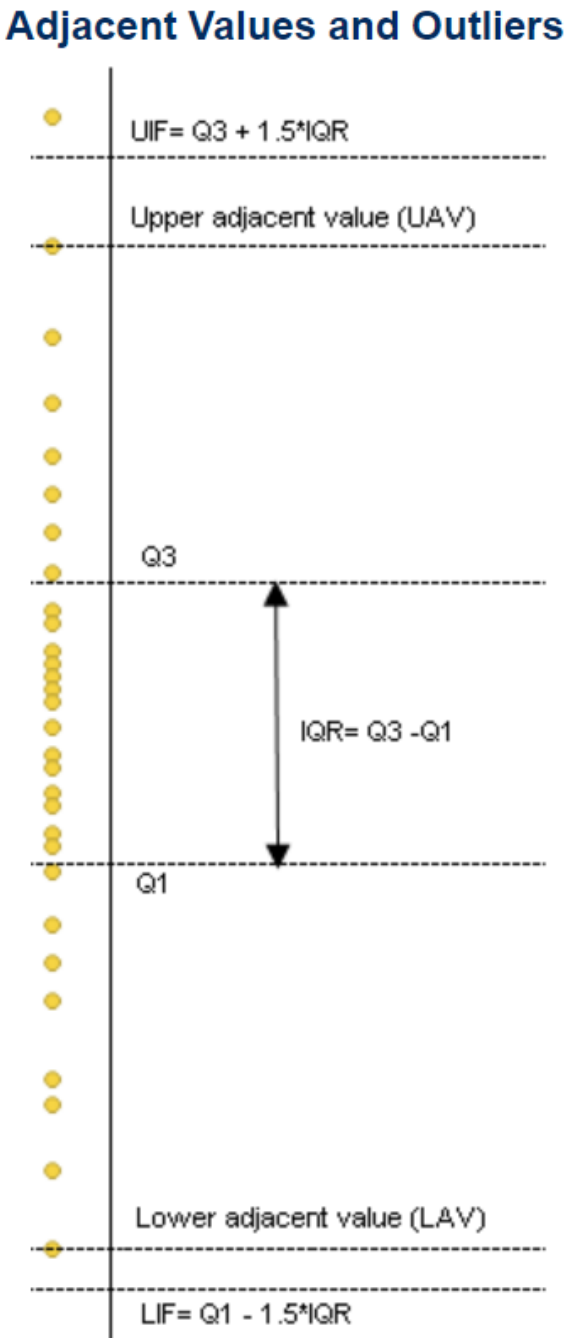
Box (and whisker) plots

1. Box plots are useful for comparing distributions across variables or across waves (or groups)
    a. Standardized rules make comparing distributions easy

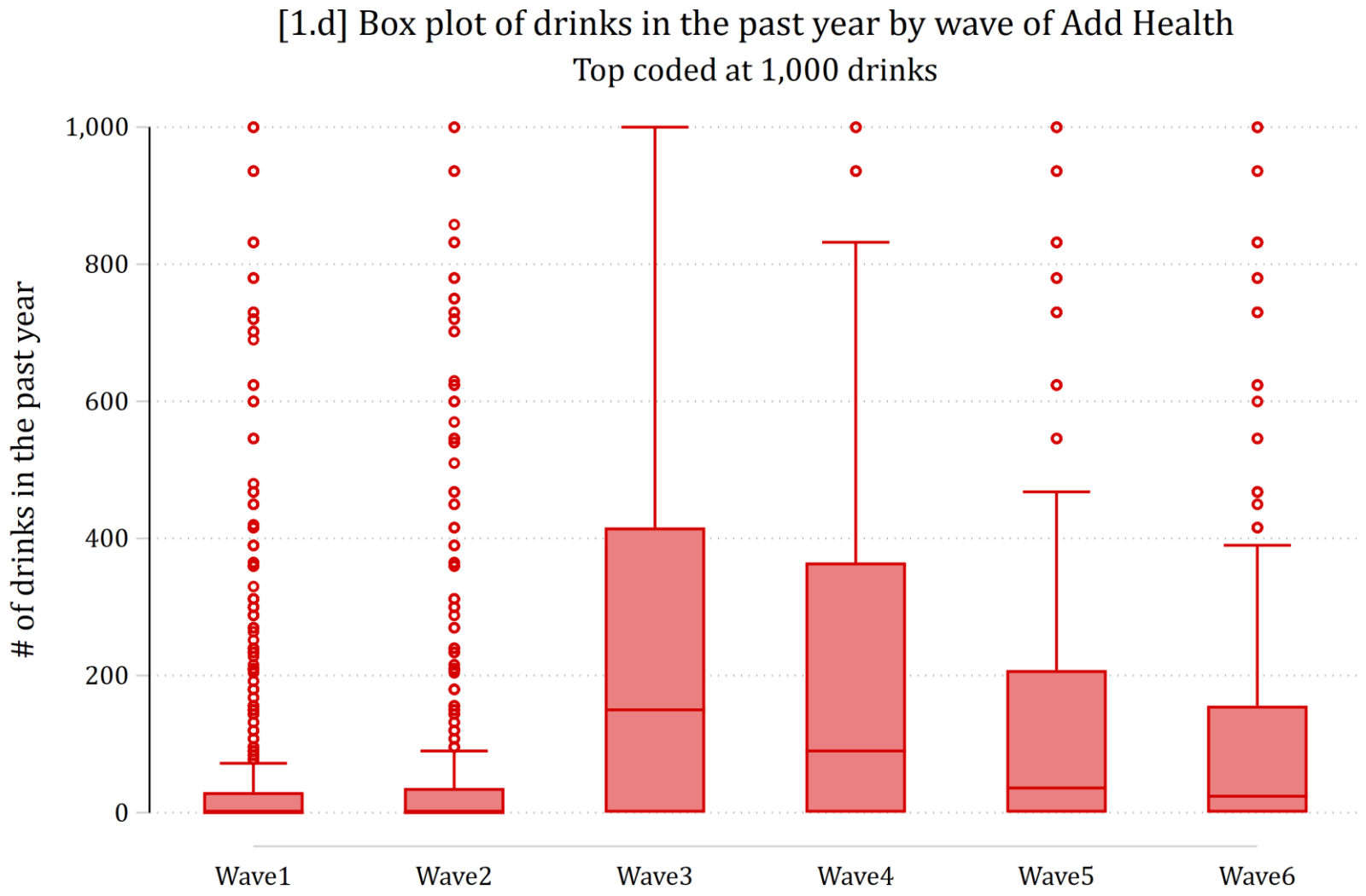


Box and whisker plot rules

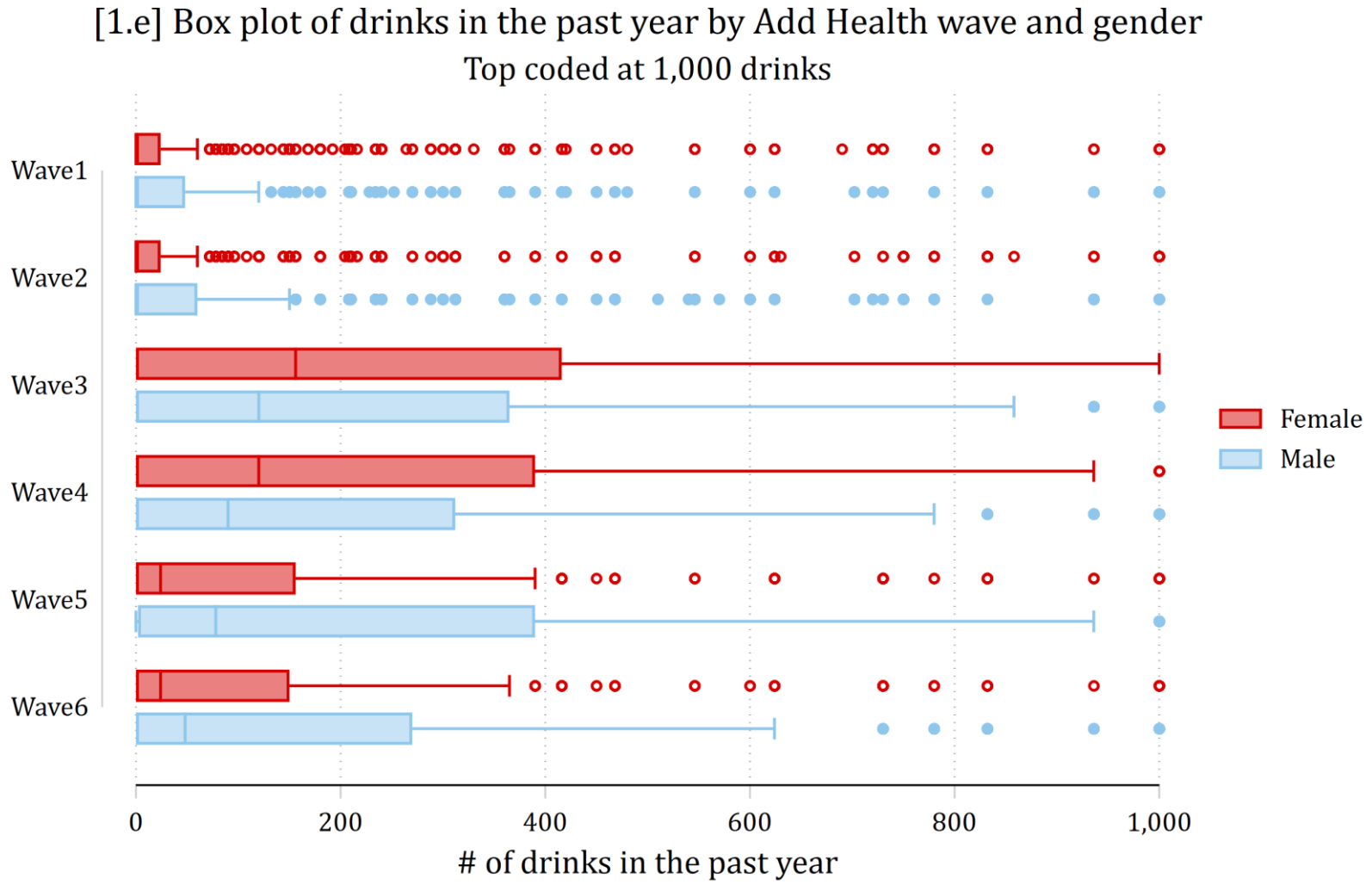
1. Rules:
    a. Line break in middle of box = median
    b. Box boundaries = interquartile range (IQR)
        i. IQR: 75th percentile – 25th percentile
    c. Whiskers = range of data without outliers*
        i. I.e., Where the vast majority of the data falls
    d. Dots = outliers
        i. NOTE: Following Tukey (1977), the length of the whiskers (and thus, the cutoff for outliers) is:
            1. Upper cutoff: 75th percentile + 1.5 * IQR
            2. Lower cutoff: 25th percentile - 1.5 * IQR



```
ggplot(ah, aes(x = wave, y = numdrinksyrT)) + geom_boxplot( )
graph box numdrinksyrT, over(wave) …
```



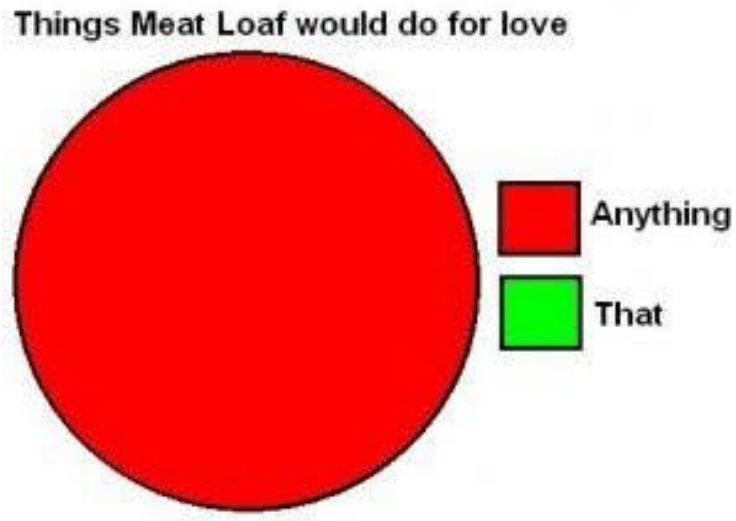
```
ggplot(ah, aes(x = wave, y = numdrinksyrT, fill = male)) + geom_boxplot( ) + coord_flip( )
graph box numdrinksyrT, over(male) over(wave) asyvars horizontal …
```



Principles of good data visualization

1. Look at your data
2. **Use graph types people read accurately**
   a. Pie charts, bar charts, and dot plots can often display the same information: usually, the univariate distribution of a categorical variable
      a. But, most can also be used for bivariate relationships and we will use these as building blocks for visualizing more complicated relationships

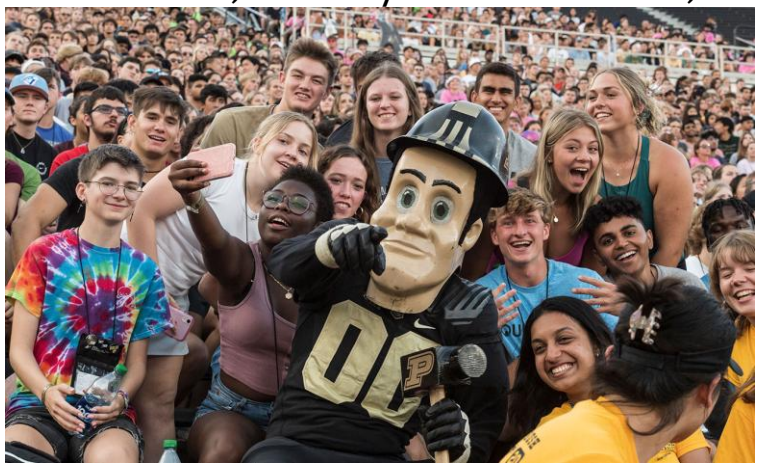
   b. Despite their similarities, they can differ dramatically in their effectiveness



Meatloaf music video (highly recommended)

Pie charts

1. Pie charts present the proportion of a variable represented in each category
    a. The fundamental rule of a pie chart is it should show **parts of a whole**
    b. E.g., Among the Purdue student population, what percent major in engineering, social science, health/medical fields, humanities, etc.?

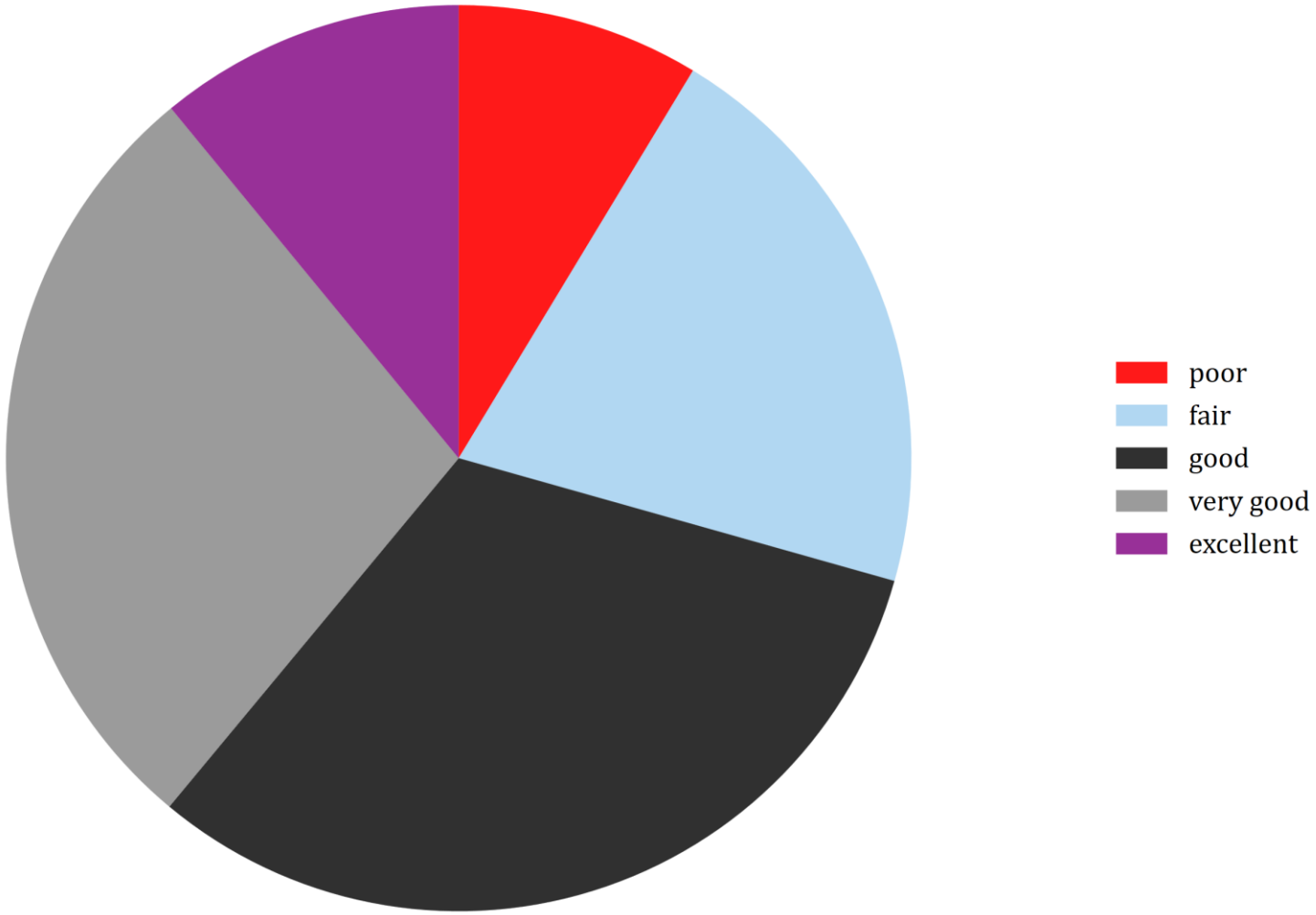


2. Pie charts have the advantage of being intuitive to understand and widely used—so your audience is probably familiar with them

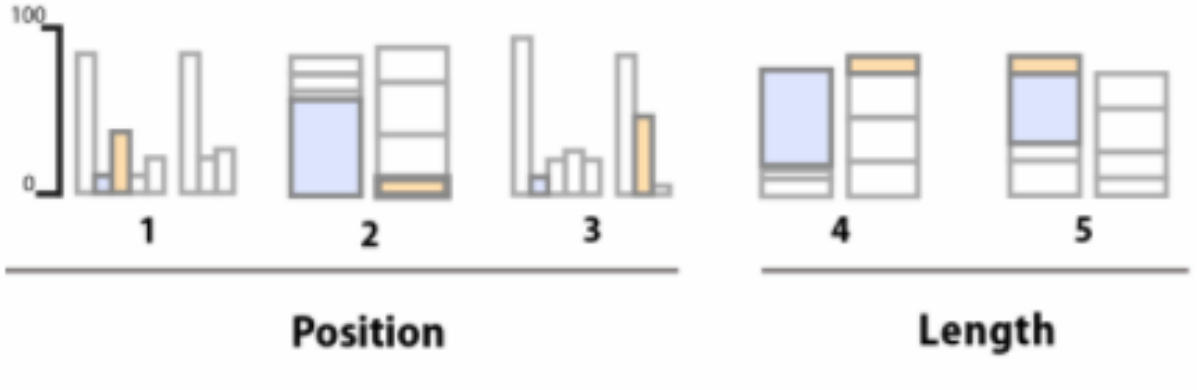
3. The downside of a pie chart is that judging area size and angles is hard
    a. The reader receives more of a general sense of the size of different categories rather than a precise estimate
    b. There are no axis labels that can be referred to

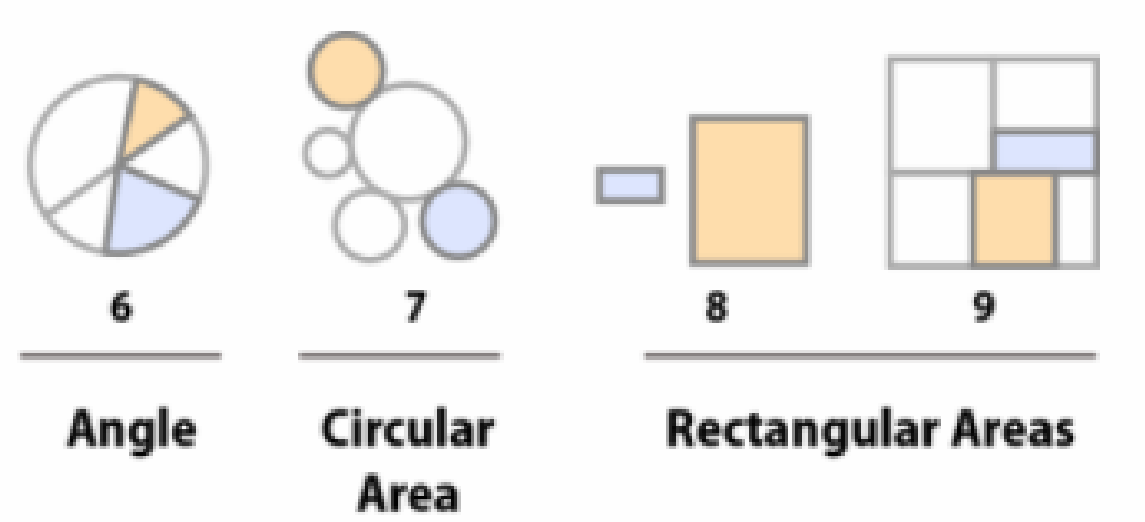
```
ggplot(hrs, aes(x = "", fill = srhealth)) + geom_bar( ) + coord_polar(theta = "y")
graph pie, over(srhealth) …
```

[2.a] Pie chart for distribution of self-rated health among older adults



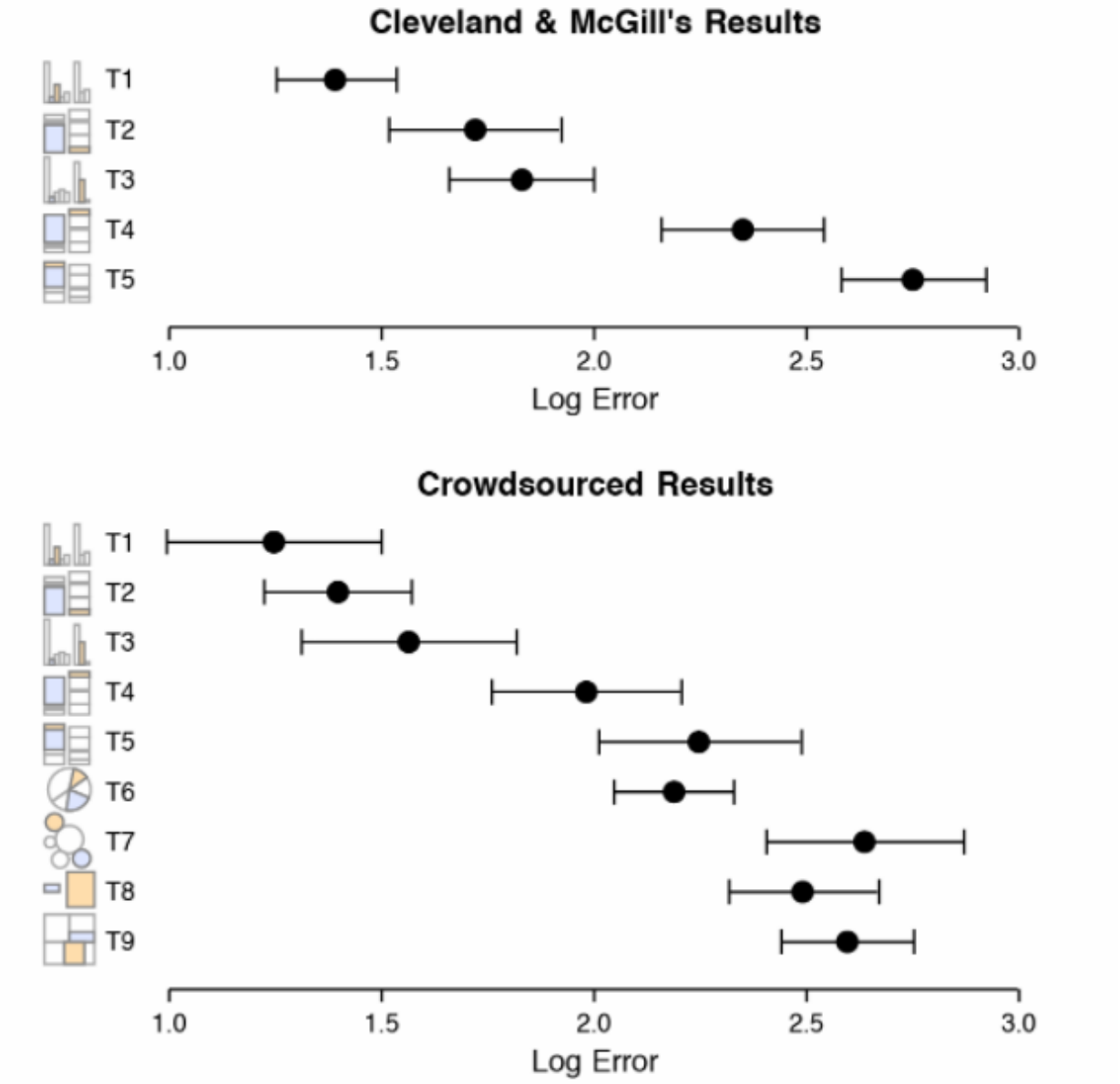
Accuracy of perceptual tasks



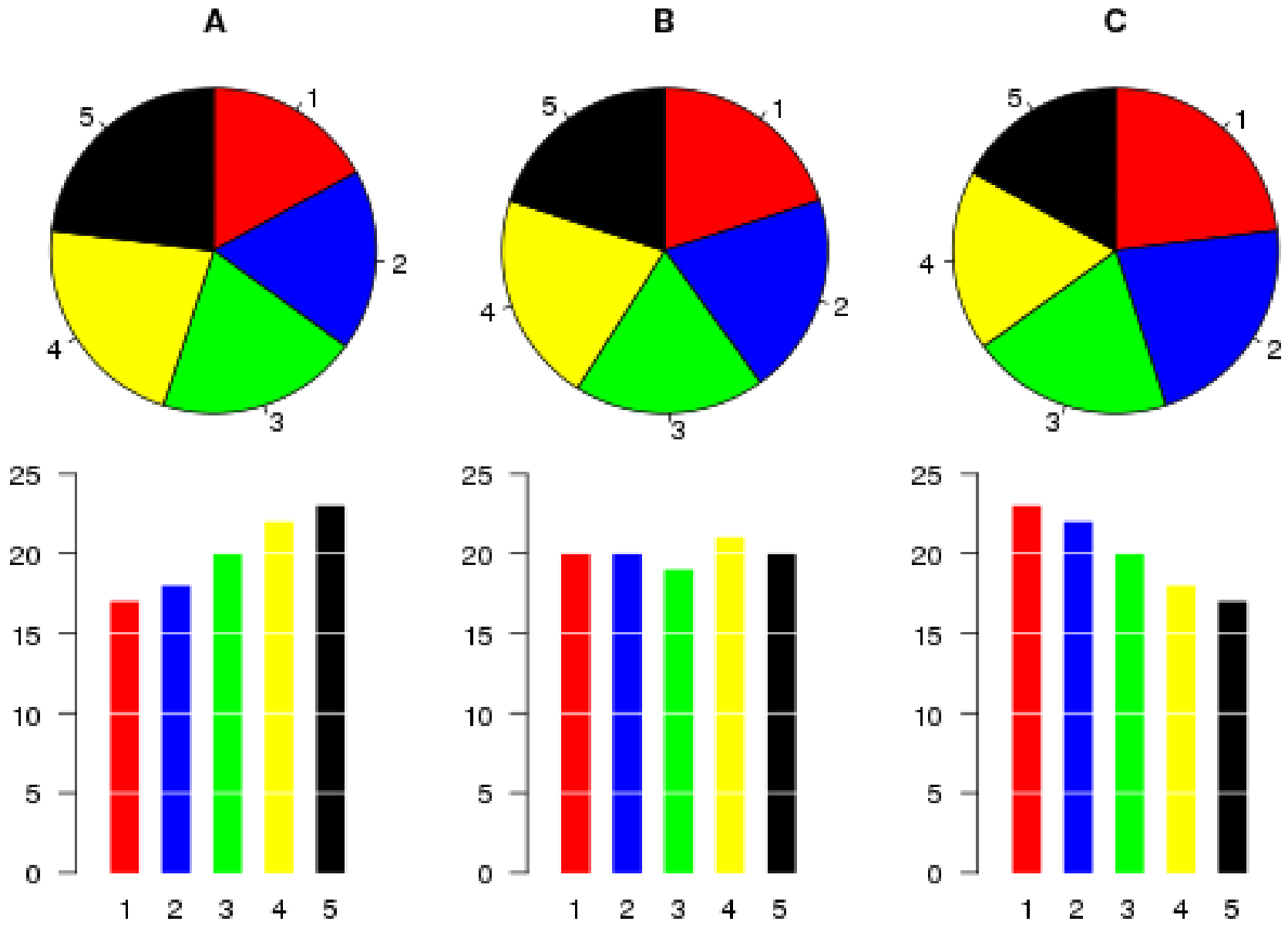


Findings and figures from:

Cleveland and McGill 1984;
Heer & Bostock 2010;
Healy 2018



Note: "crowdsourced" results refer to Heer & Bostock (2010)



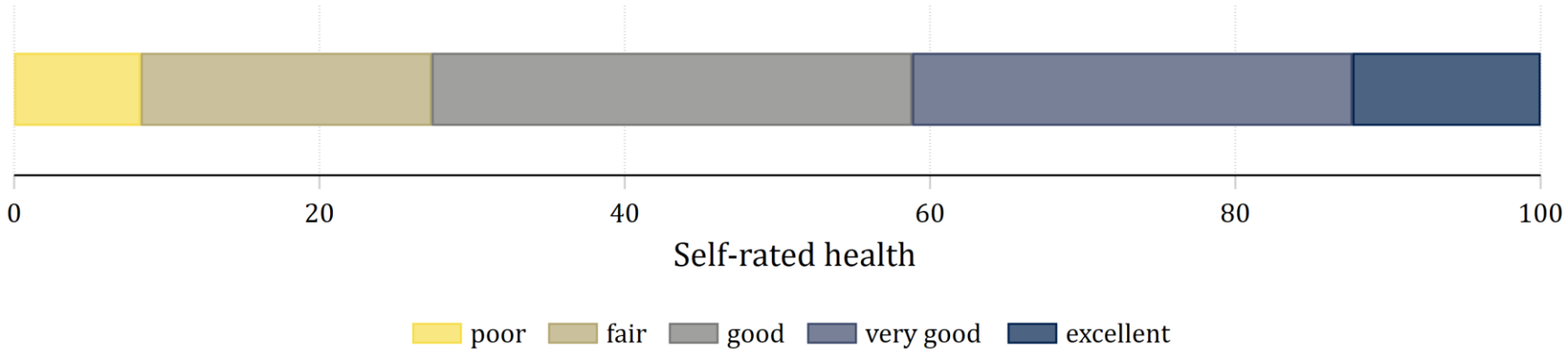
*Bar charts are simple but very effective for comparing amounts across groups

Stacked bar charts

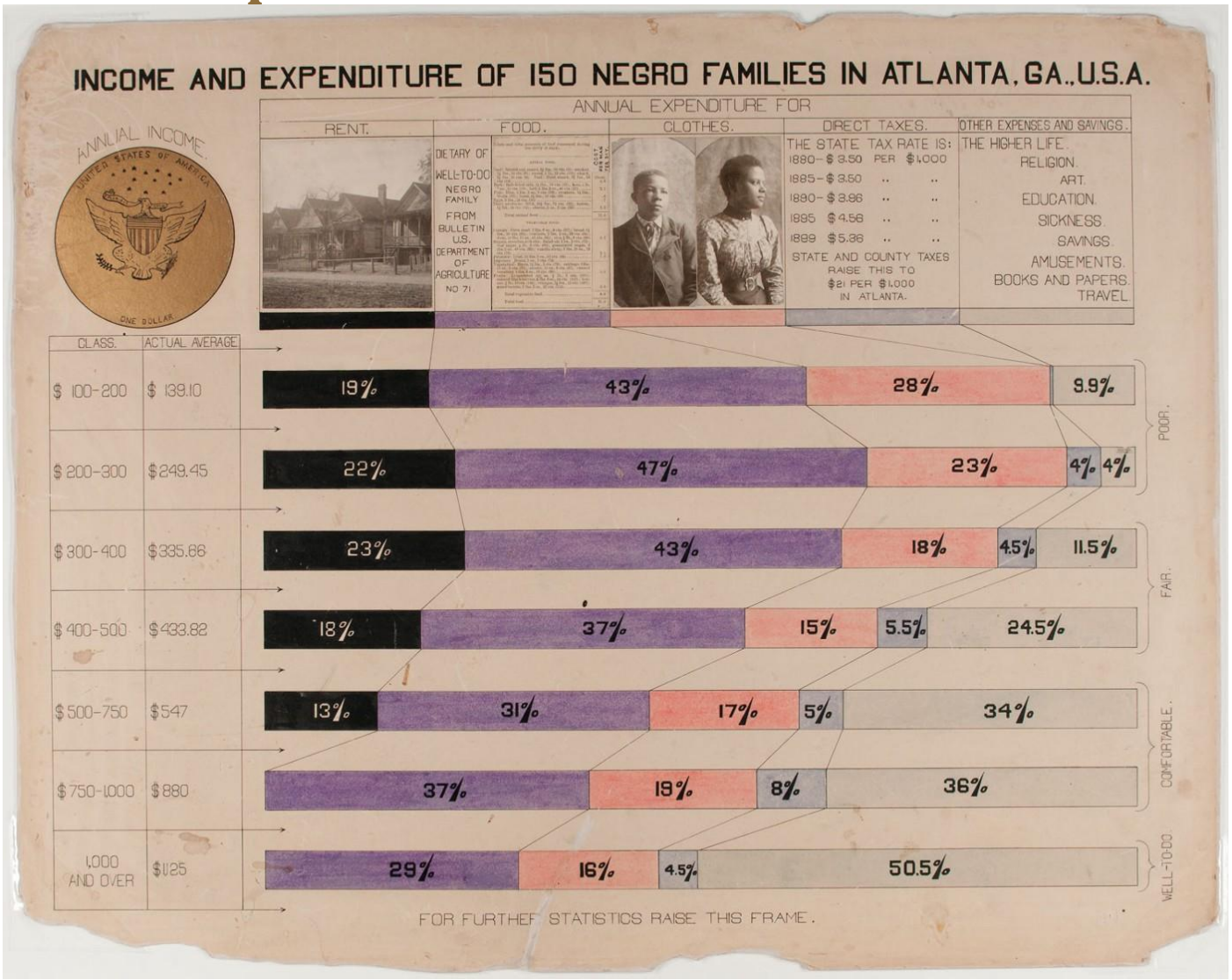
1. Stacked bar charts are a useful alternative to pie charts
    a. Also, make it easy to compare across time/waves or groups (which is very difficult with a pie chart)

2. Stacked bar charts can be used when you want a bar to fill the entire space of an axis (compare to how much space on the *y*-axis is covered by the bars on the last page)

3. Stacked bar charts are especially useful when you have categorical outcome (*y*) variables and/or when the total of multiple variables is of interest
    a. Works well for binary dependent variables
    b. Especially useful for nominal/ordinal dependent variables



Historical example: W.E.B DuBois

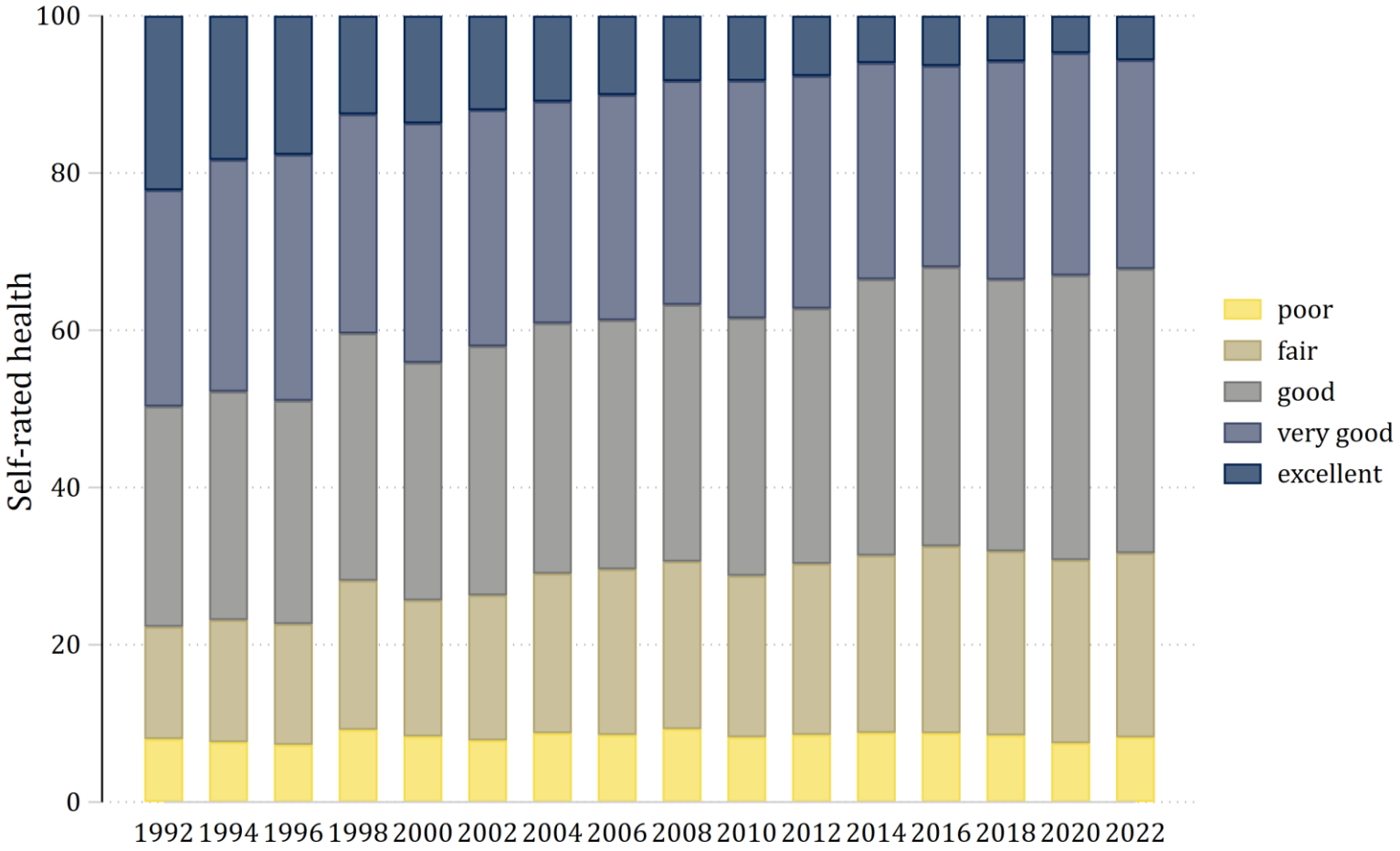


*Read more about W.E.B. DuBois's pioneering data visualizations here (and a whole book here)

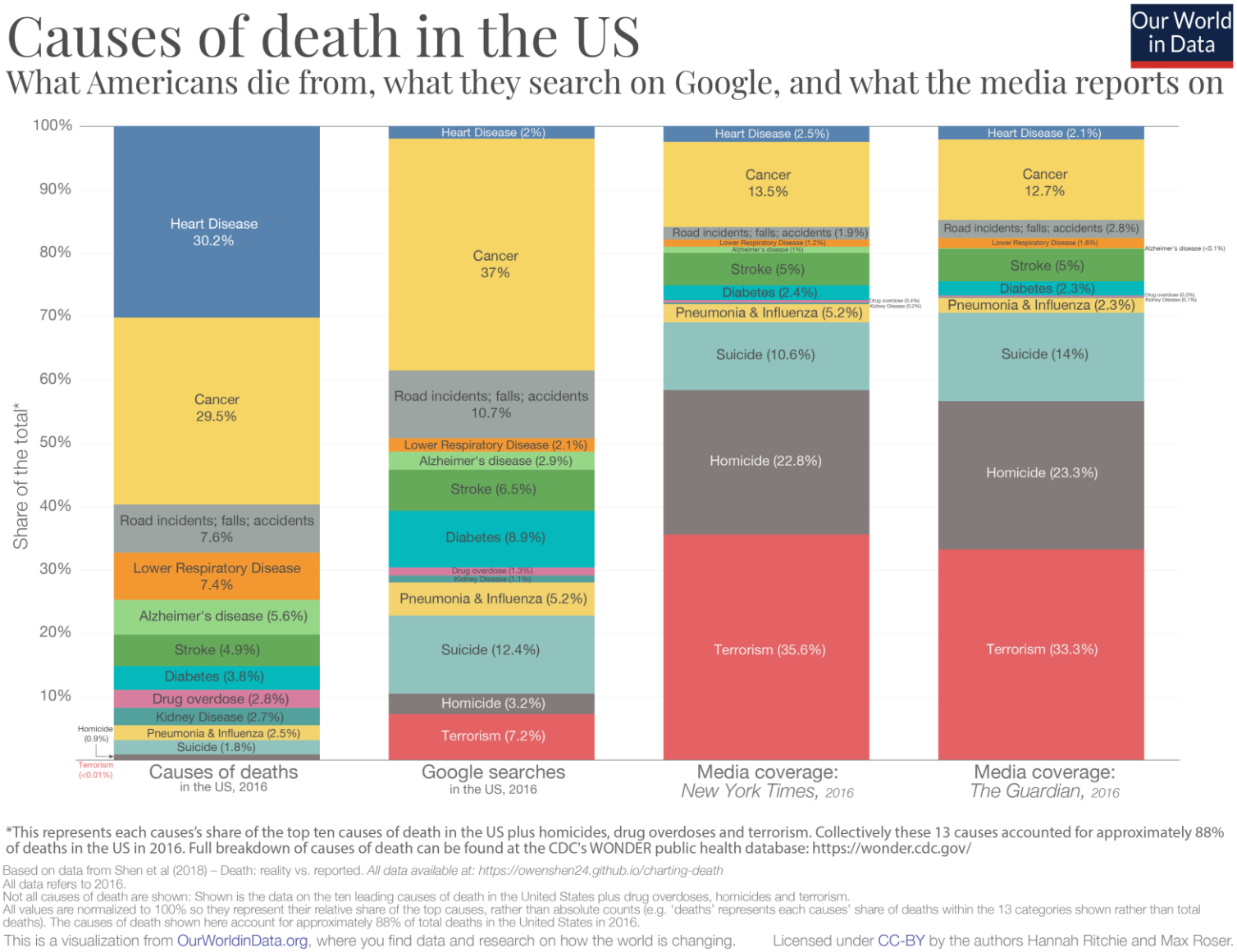
```
ggplot(hrs_filtered, aes(fill = srhealth, x = year)) + geom_bar(position = "fill") +
  scale_fill_viridis_d(option = "cividis", direction = -1)
```

```
graph bar (count) if in1992, over(srhealth) over(year) asyvars stack ///
    percentages scheme(cleanplots5) …
```

[2.c] Stacked bar chart of self-rated health among older adults over time (HRS)



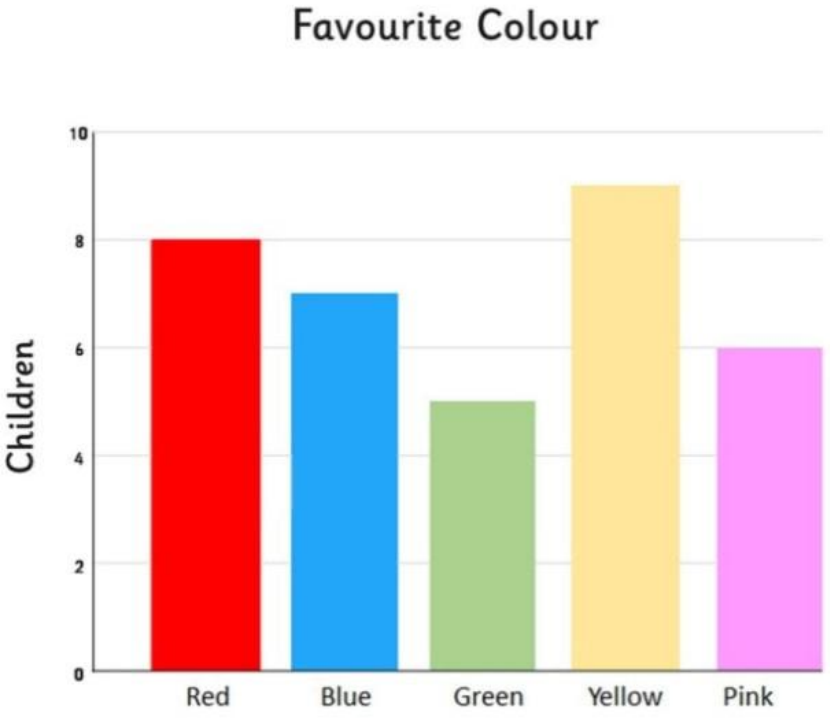
Published stacked bar chart example



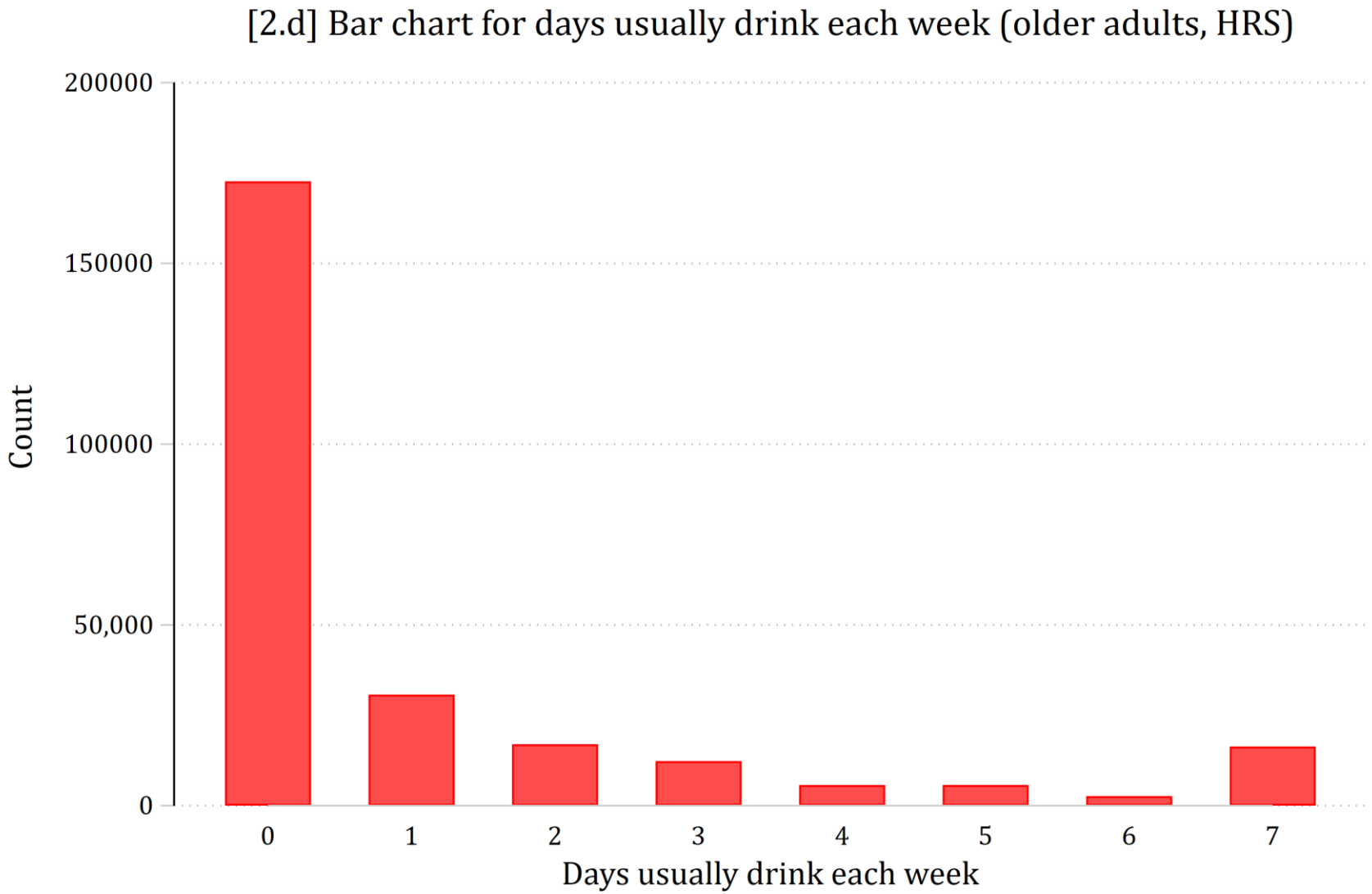
Bar charts

1. The findings from the visual perception experiments pioneered by Cleveland and colleagues lead to some straightforward conclusions about effective graphics:
    a. Angles are difficult to perceive accurately
        i. Pie charts are not effective
    b. Area is difficult to judge—especially when multiple aspects are varying at once (e.g., both length and width)
    c. A common scale is necessary to compare across groups

2. Bar charts—where the width stays the same and only the length varies based on a quantitative metric—are among the most effective tools for comparing quantities



```
ggplot(hrs, aes(x = dayweekdrink)) + geom_bar( )
graph bar (count), over(dayweekdrink) …
```



Principles of good data visualization

1. Look at your data
2. Use graph types people read accurately
3. **Maximize the data-ink ratio, within reason (maybe)**

The data-ink ratio

*"Maximize the data-ink ratio, **within reason**." (Tufte 1983; emphasis added)*

1. Excessive decoration/color should be avoided (but see my caveat below)
    a. All aspects of the graph should aid in presenting the data
        i. The "data ink" ratio should be as high as possible (Tufte 1983)
            1. The basic idea behind this principle: all aspects of a graphic should serve the purpose of showing the data
            2. Consider whether superfluous additional pieces/colors on your graphic can be removed
    b. My less strict rule: **all color should convey some information**



The data-ink ratio: my caveat

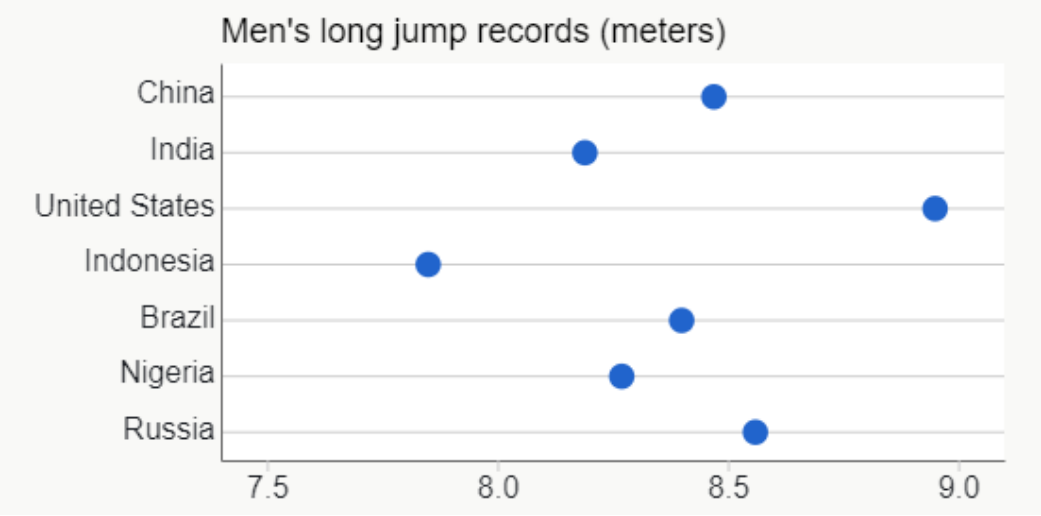
"… 'remove clutter and strive for clean and elegant designs'… is reasonable advice. But… 'do everything you can to remove non-data ink'…will result in poor design choices." *(Wilke 2019 p.277-278)*

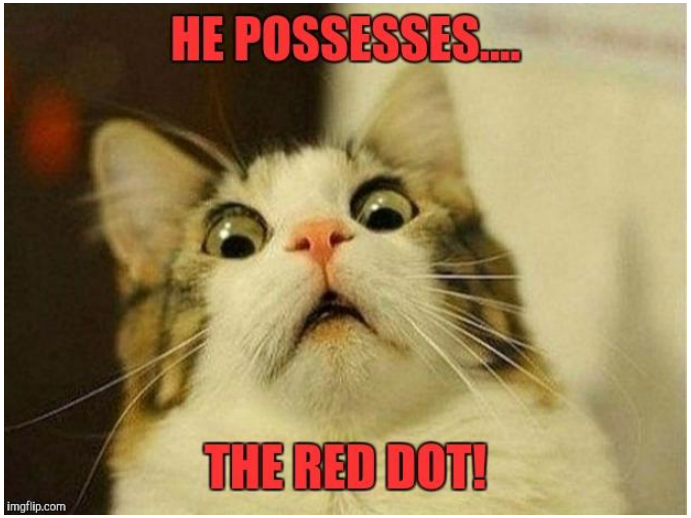
c. I am much less of a fundamentalist about the maximize data ink rule
    i. In the age of PDFs and online journals/reports/websites, it isn't "wasteful" to include extra color or other data ink
d. Sometimes the color displays useful information (e.g., your brain registers amount better with a bar chart than a dot plot because the blocks of color carry visual information on quantity)
e. Color can help you quickly identify different parts of the graph (e.g., *the blue line shows…*)
f. Sometimes the color… looks nice
    i. Having aesthetically pleasing graphics is important

Dot plots

1. Dot plots can often be used interchangeably with bar charts
    a. But, instead of a colored in bar, you simply place a dot at the value being plotted (would be the top of the bar)
    b. Dot plots can similarly easily incorporate measures of uncertainty in the form of SEs or CIs

2. Dot plots were invented to take advantage of the knowledge gleaned from Cleveland and colleague's visual perception experiments and Tufte's opinions about excess decoration



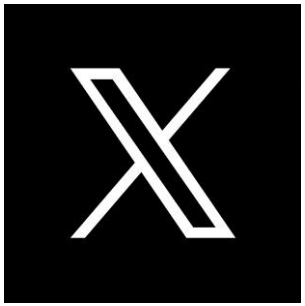


Advantages of dot plots

3. An advantage of dot plots is that they waste less data ink when compared to a bar chart (Tufte 1983)
    a. This also keeps them from becoming crowded and unreadable

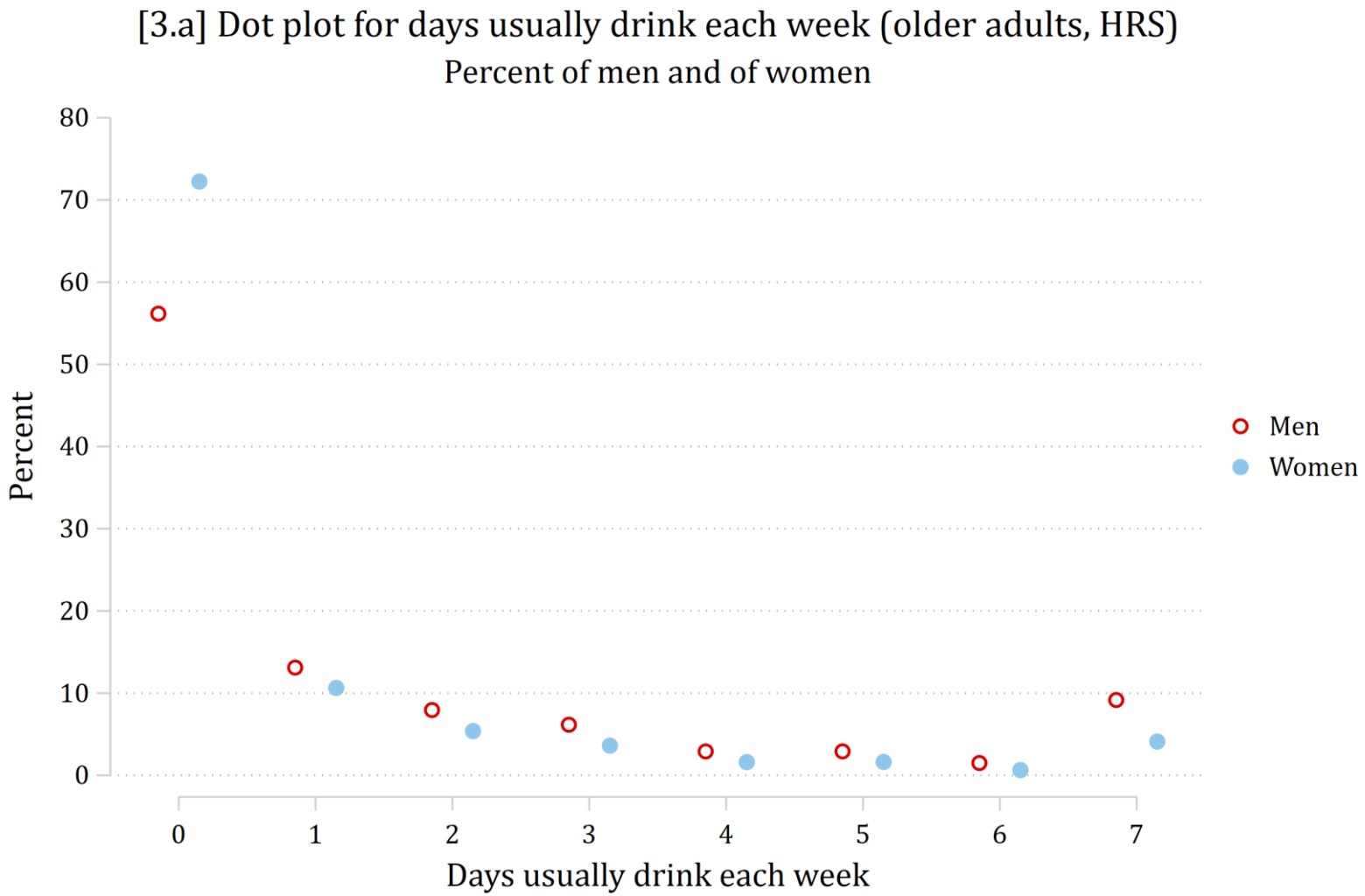
4. A purely utilitarian reason to use a dot plot is that people will give you a hard time for having a bar chart that does not start at zero—while no one seems to care if a dot plot starts at zero
    a. There are good reasons for this concern as bar charts display information both as length and area—and the area can be misleading without a zero baseline
        i. More on zero baselines later in these notes



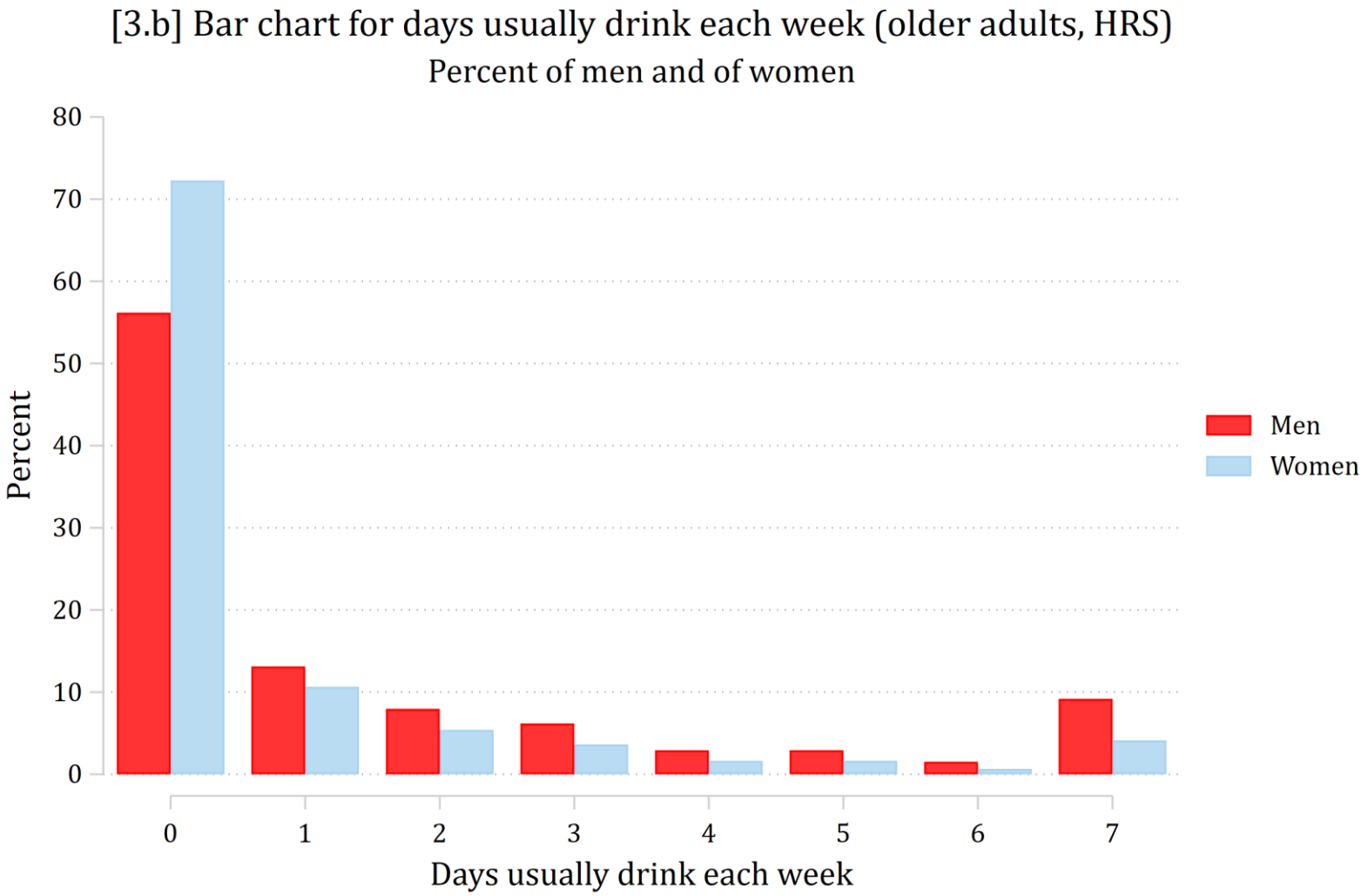




```
ggplot(drink_pct, aes(x = factor(dayweekdrink), y = pct, color = woman)) + geom_point( )
proportion dayweekdrink if woman == 0       // and again for women
coefplot drink_men drink_wom, vertical noci rescale(100) …
```



```
ggplot(drink_pct, aes(x = factor(dayweekdrink), y = pct, fill = woman)) + geom_col(…)
coefplot drink_men drink_wom, vertical noci rescale(100) recast(bar) barwidth(0.4) …
```



Americans overestimate the size of minority groups and underestimate the size of most majority groups

Estimated proportions are calculated by averaging weighted responses (ranging from 0% to 100%, rounded to the nearest whole percentage) to the question "If you had to guess, what percentage of American adults..." True proportions were drawn from a variety of sources, including the U.S. Census Bureau, the Bureau of Labor Statistics, and polls by YouGov and other polling firms.

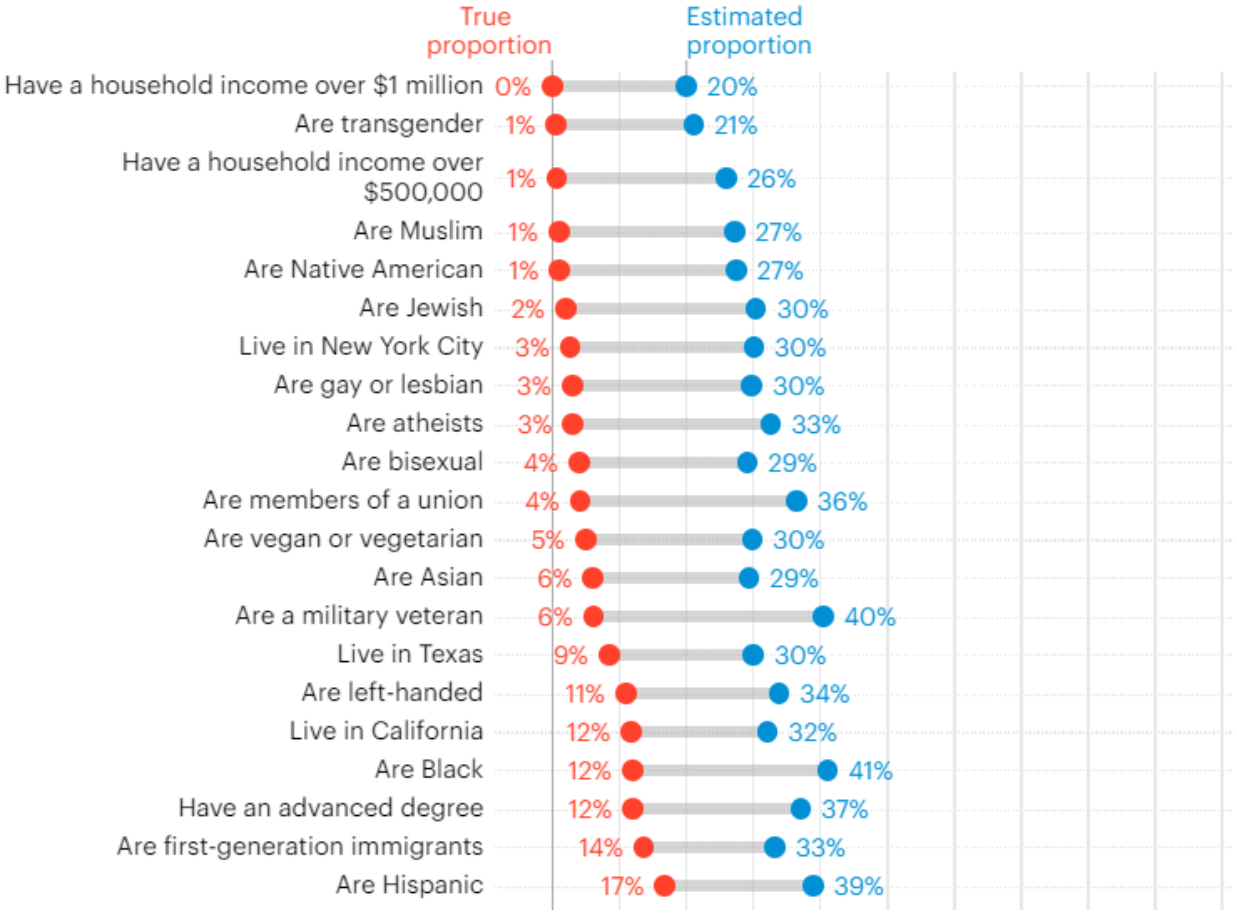


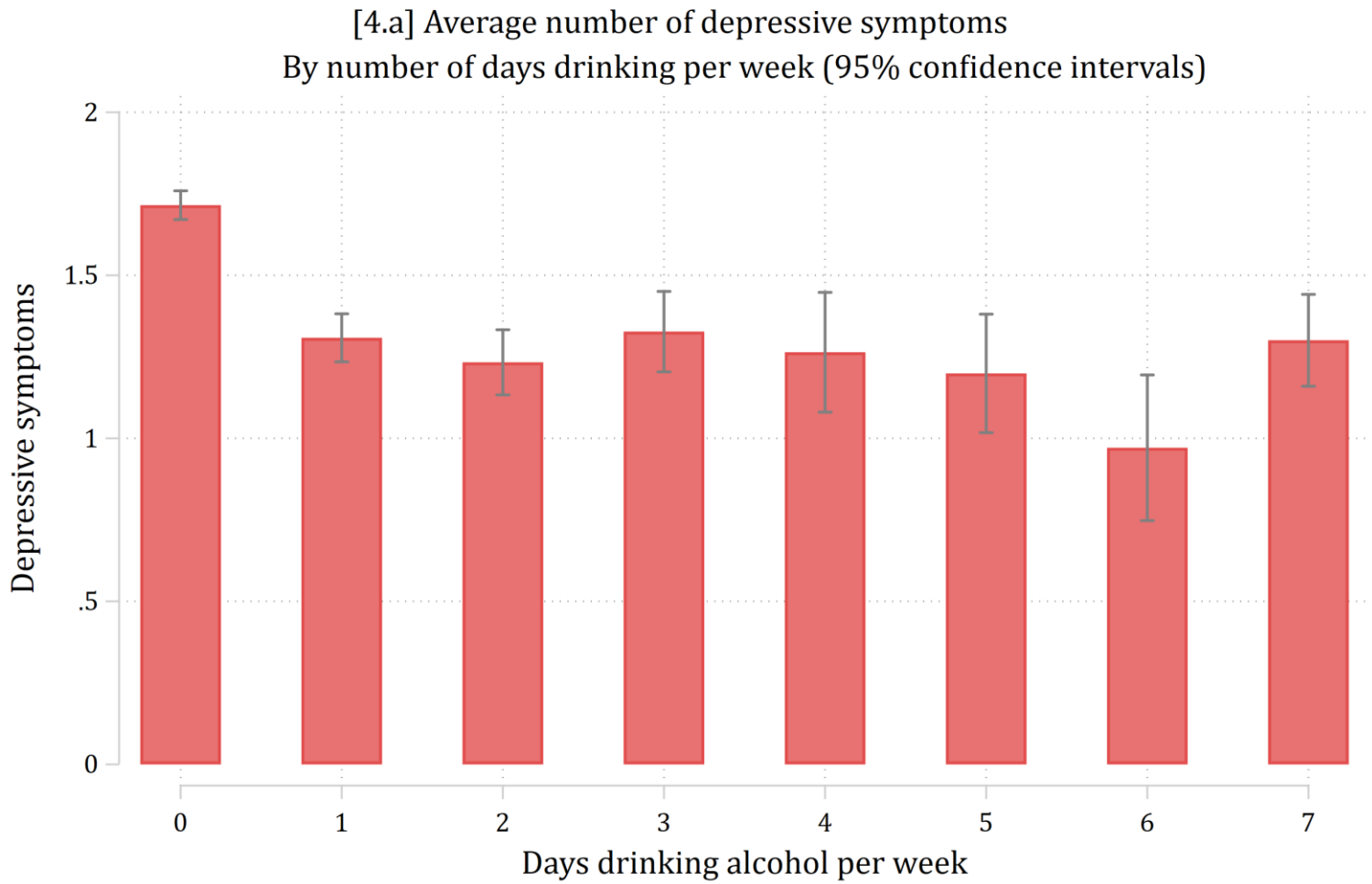
Figure from YouGov 2022

Principles of good data visualization

1. Look at your data
2. Use graph types people read accurately
3. Maximize the data-ink ratio, within reason (maybe)
4. **Show uncertainty**
    a. Uncertainty should be incorporated visually when possible
        i. Confidence intervals and standard errors should be included for model estimates / predictions

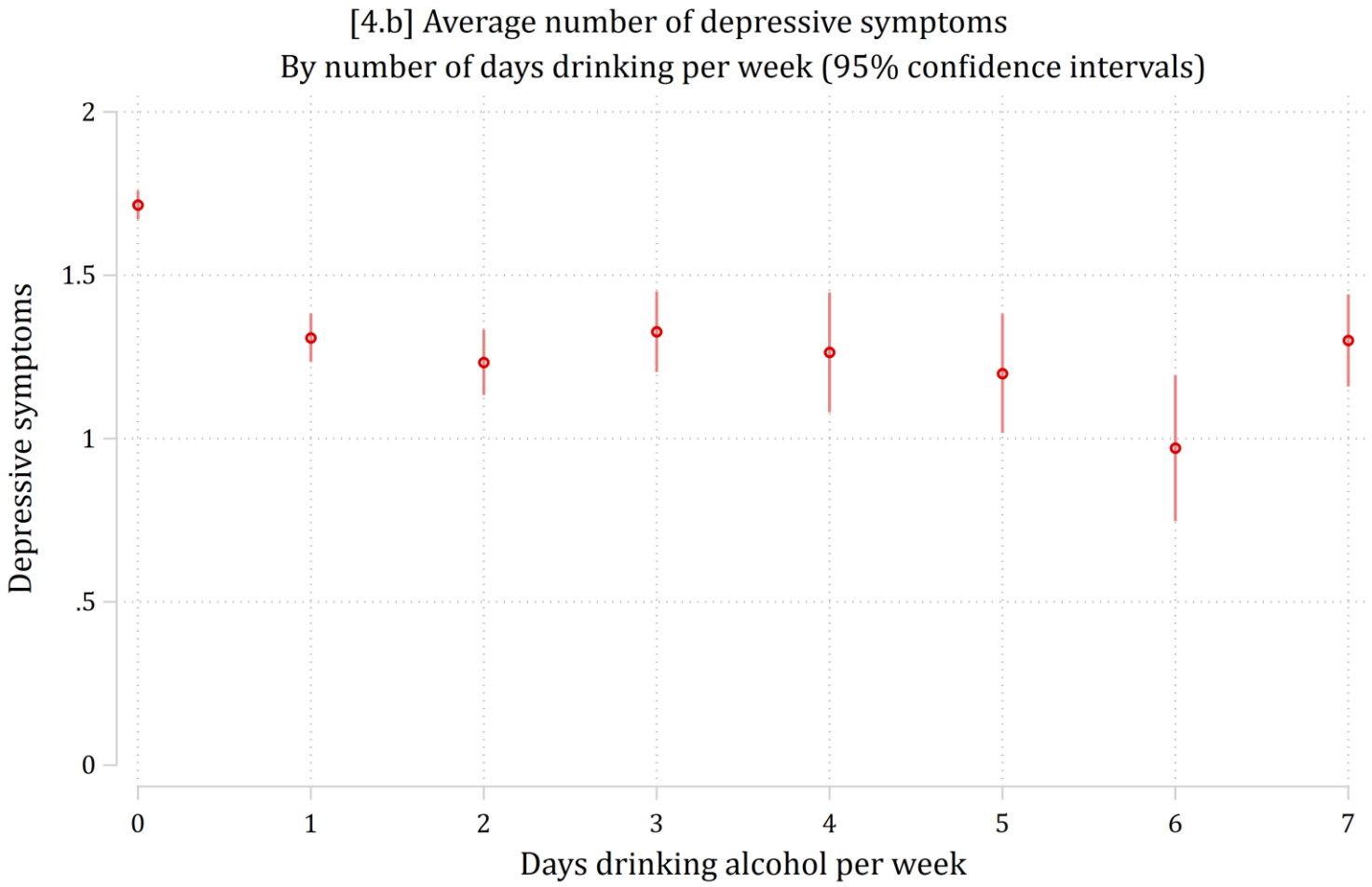
```
ggplot(filter(hrs, year == 2022), aes(x = dayweekdrink, y = cesd)) +
  stat_summary(fun = mean, geom = "bar") +
  stat_summary(fun.data = mean_cl_normal, geom = "errorbar")
```

```
mean cesd if year == 2022, over(dayweekdrink)
marginsplot, recast(bar) …
```

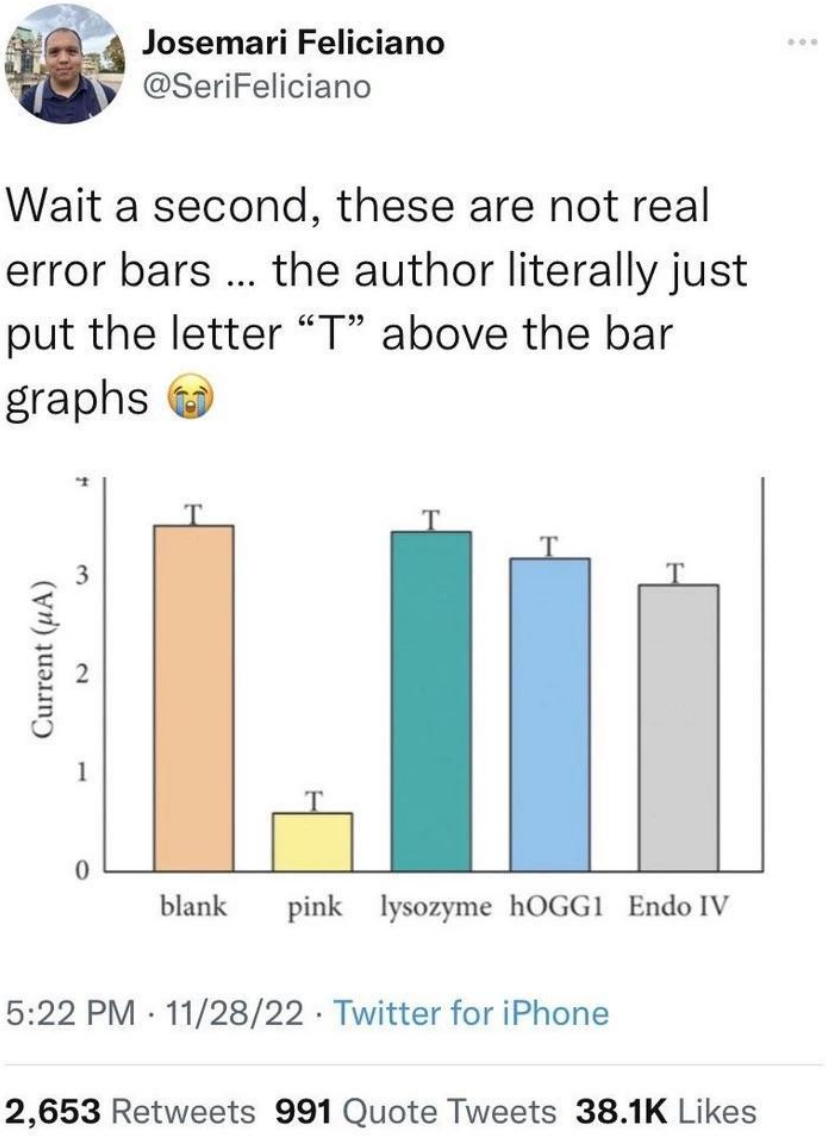


```
ggplot(filter(hrs, year == 2022), aes(x = dayweekdrink, y = cesd)) +
  stat_summary(fun = mean, geom = "point") +
  stat_summary(fun.data = mean_cl_normal, geom = "linerange")
```

```
mean cesd if year == 2022, over(dayweekdrink)
marginsplot, recast(scatter) …
```



Bad graph levity



Moving forward: building blocks

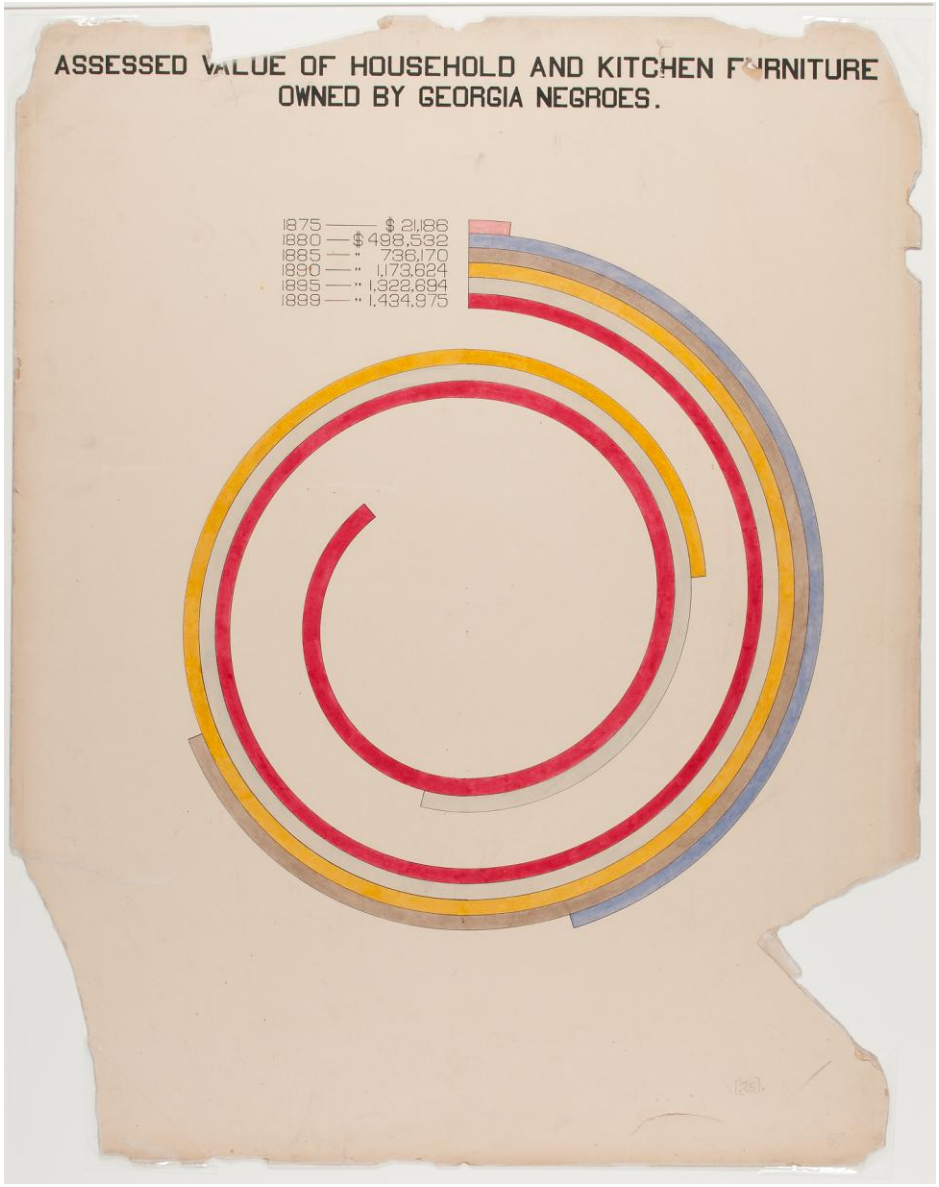
"There is a strand of the data viz world that argues everything could be a bar chart. That's possibly true but also possibly a world without joy." *(Amanada Cox, Editor, The Upshot, quoted in HBR 2013)*

1. As we move into more advanced data and model visualizations the basic tools we learned in this section will often reappear

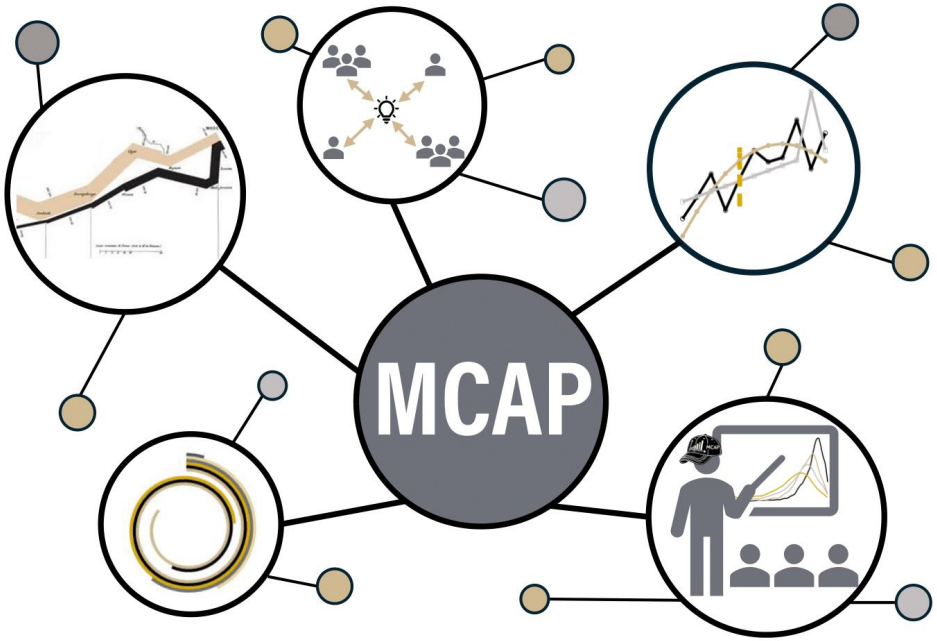
2. Aspects of both bar charts and dot plots are used very often in displaying complex model results
    a. They both have good visualization properties and work nicely as the building blocks of more complicated graphs—where you can layer on additional aspects



Bar chart that didn't catch on



Read more about this spiral chart and other pioneering visualizations of W.E.B. DuBois





*Read about The Methodology Center's graphic

Break 1: group task

1. Run the code below in R or Stata: it loads the data and makes a quick graph

2. Discuss with your group:
    a. What does the graph show? What do you learn?
    b. What would you need to change to make the graph better (more effective and/or nicer to look at), beyond the defaults?

```
library(foreign); library(ggplot2); library(dplyr)

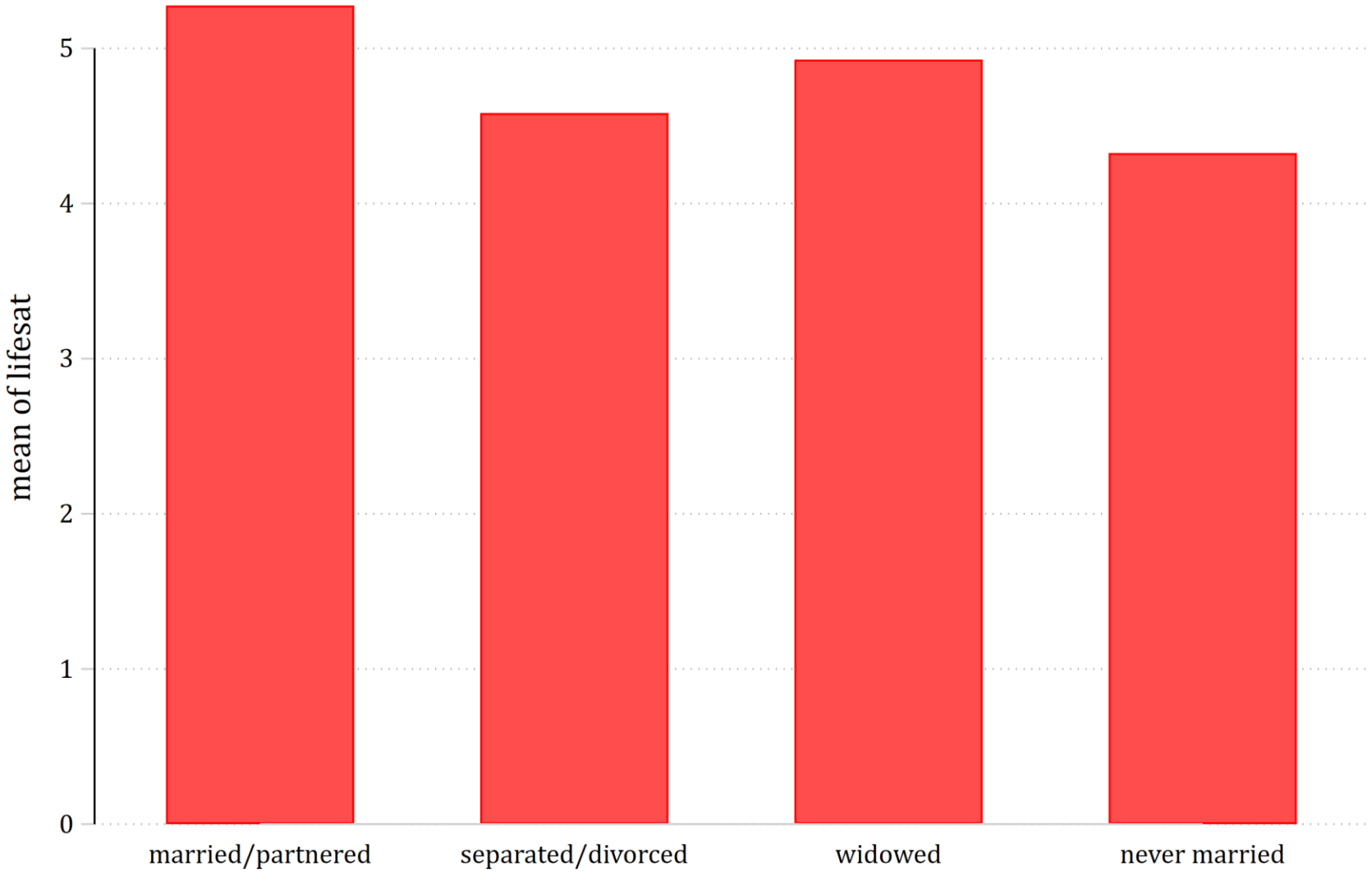
hrs <- read.dta("https://tdmize.github.io/data/data/hrs_si.dta")

ggplot(filter(hrs, !is.na(marstat), year == 2022),
       aes(x = marstat, y = lifesat)) +
  stat_summary(fun = mean, geom = "bar")
```

```
use "https://tdmize.github.io/data/data/hrs_si.dta", clear

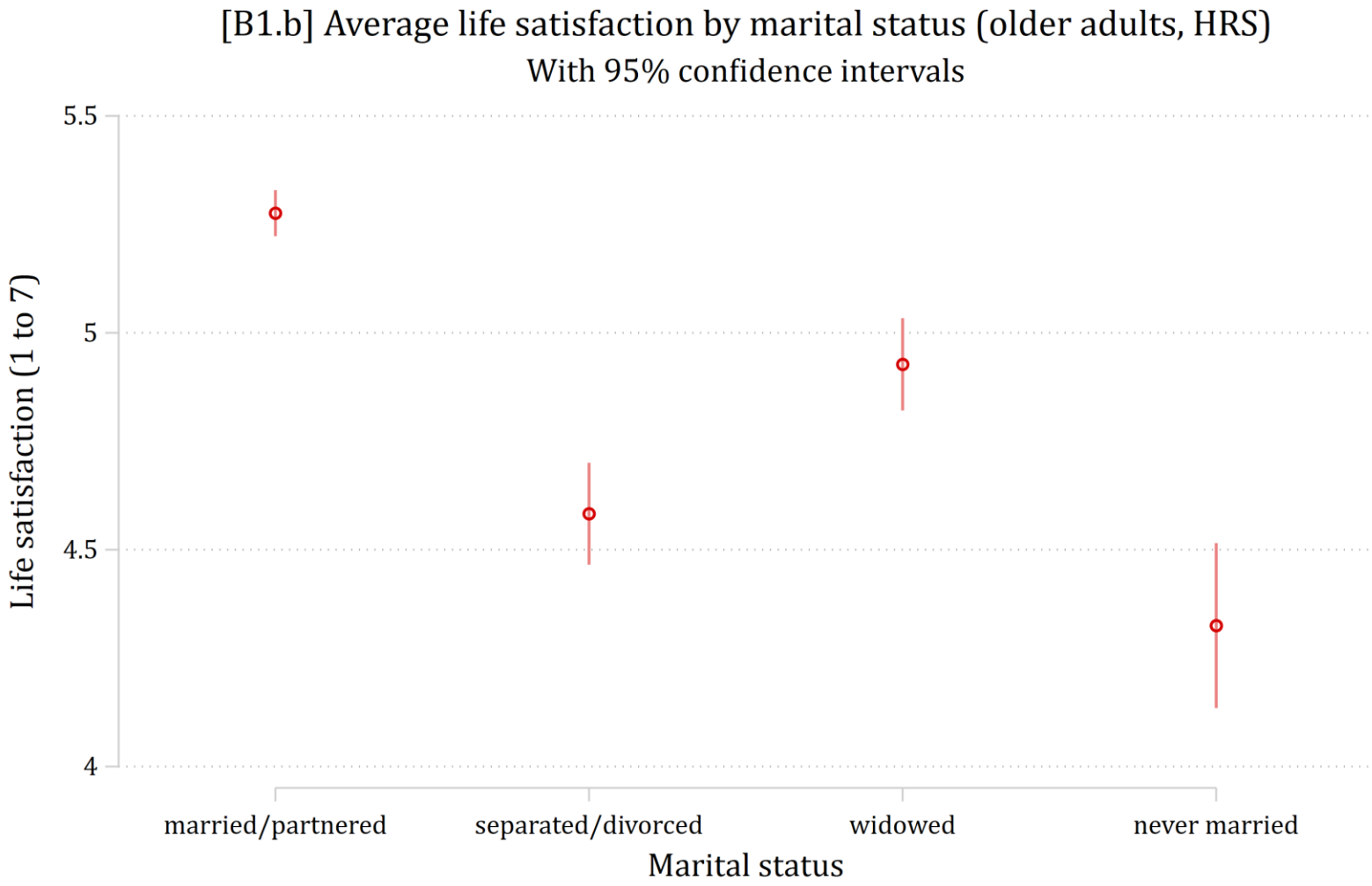
graph bar (mean) lifesat if year == 2022, over(marstat)
```

```
ggplot(filter(hrs, !is.na(marstat), year == 2022), aes(x = marstat, y = lifesat)) + stat_summary(…)
graph bar (mean) lifesat if year == 2022, over(marstat)
```



```
ggplot(…, aes(x = marstat, y = lifesat)) + stat_summary(fun = mean, geom = "point") +
  stat_summary(fun.data = mean_cl_normal, geom = "linerange") + …
```

```
mean lifesat if year == 2022, over(marstat)
coefplot, vertical …
```

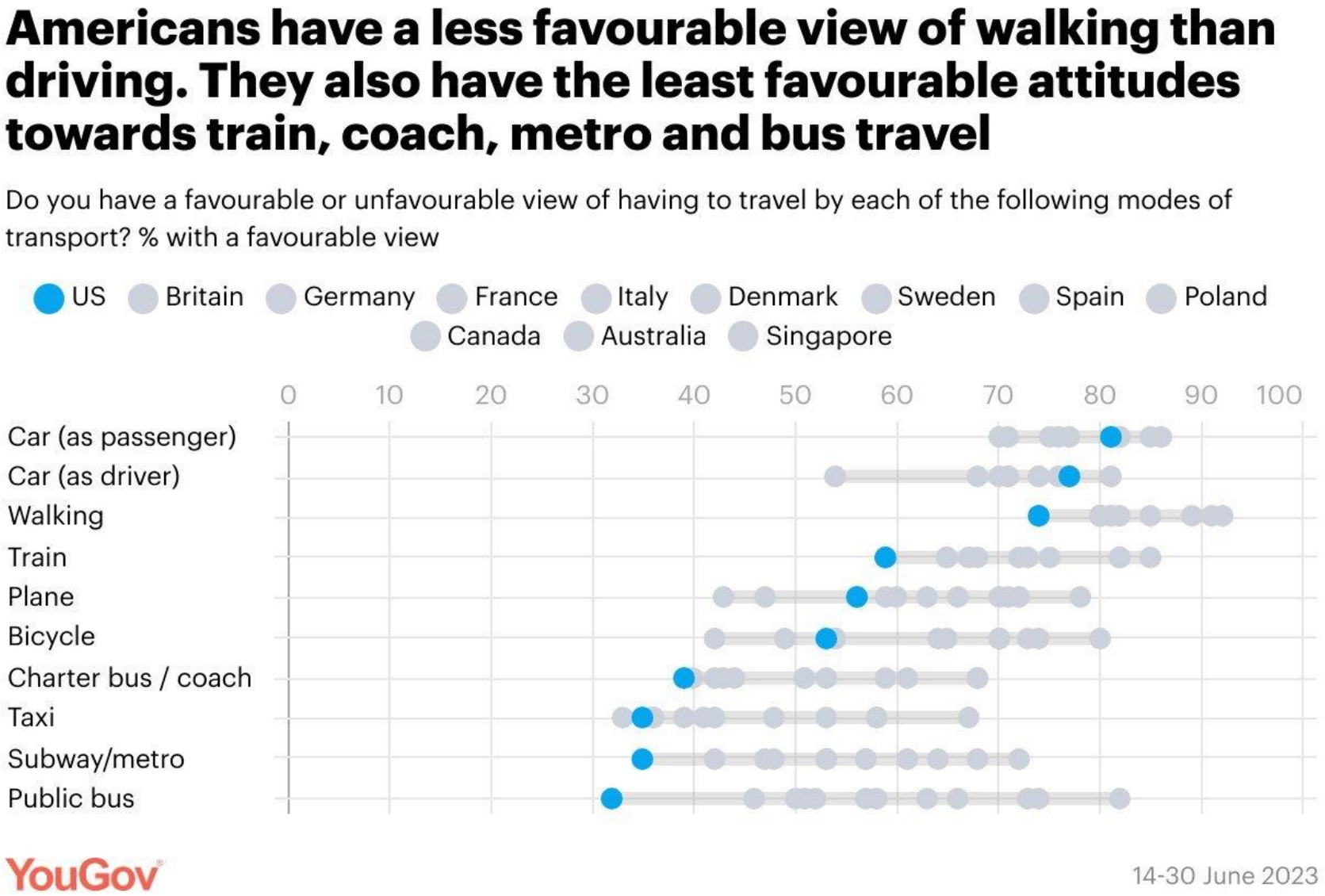


Principles of good data visualization

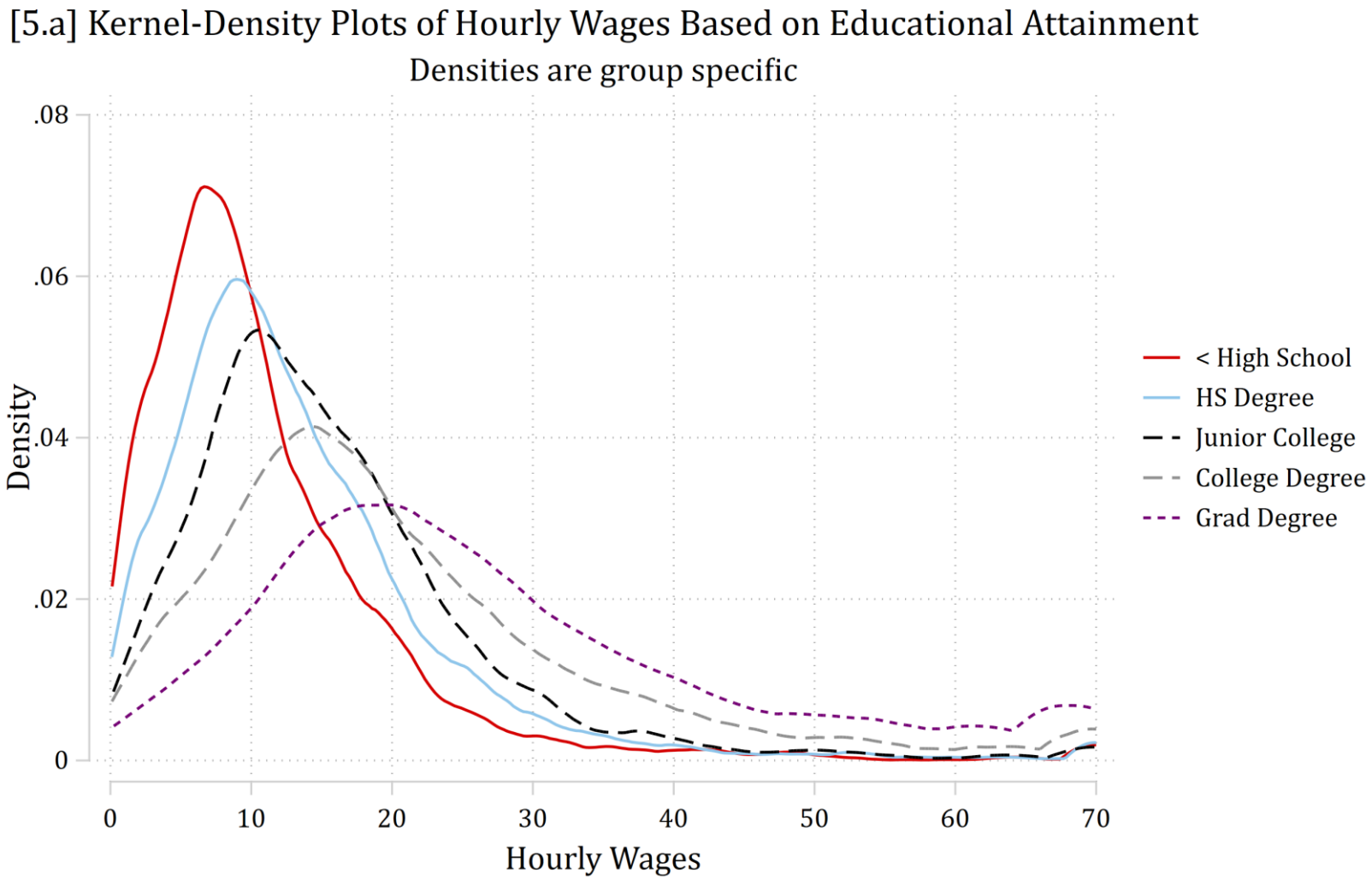
1. Look at your data
2. Use graph types people read accurately
3. Maximize the data-ink ratio, within reason (maybe)
4. Show uncertainty
5. **Make groups easy to tell apart**
    a. If there are multiple groups, a legend should clearly delineate between the groups or data should be labeled directly
        i. Legend should be close to the data itself when possible
        ii. Legend should be in logical (e.g., no high school, high school, college, graduate degree) and/or consistent order (e.g., in order on graph)

    b. The colors and/or symbols on the graph should clearly delineate between different lines/groups

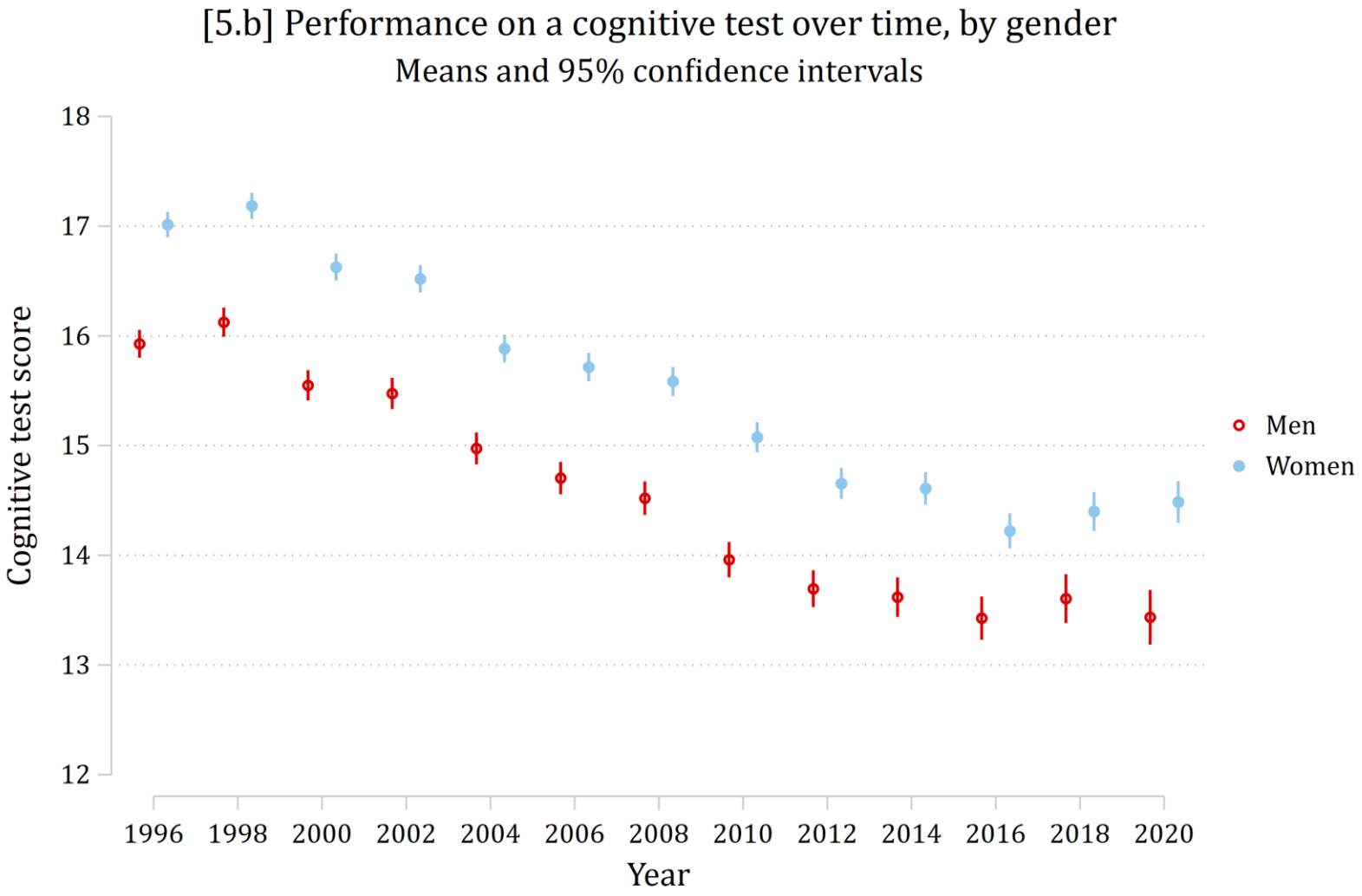
Bad graph example: cannot distinguish between symbols



```
ggplot(gss, aes(x = wages, color = degree)) + geom_density( )
twoway kdensity wages if degree==0 || kdensity wages if degree==1 || … ||, …
```



```
ggplot(…, aes(x = factor(year), y = cog, color = woman)) + stat_summary(…, position = dodge)
regress cog ibn.year if in1992 & woman == 0, nocons   // and again for women
coefplot (cog_men, label("Men")) (cog_wom, label("Women")), vertical …
```



General advice: color

The color scheme (sequence of colors) on your graph should work for you—not against you

Nominal (unordered) color schemes are best when you want distinct visual separation for each aspect of the plot

- E.g., For a nominal variable such as race/ethnicity or religion

Ordinal (sequential) color schemes work best for representing data that has a natural order

- E.g., Age groups (e.g., generations) or Likert response options

Diverging color schemes have a natural midpoint and diverge to both extremes

- E.g., Political views



*Color chart from Healy 2018

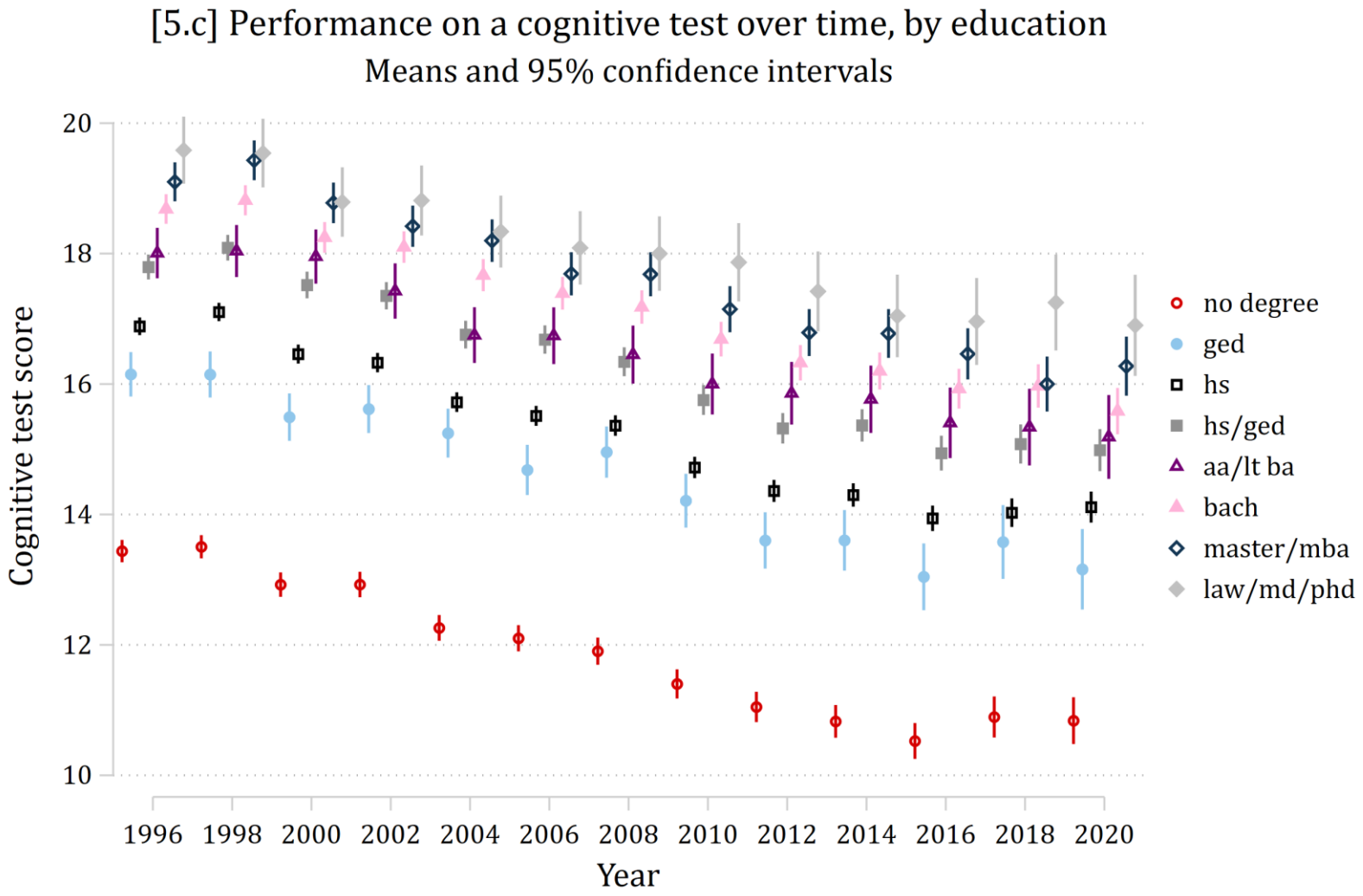
General advice: color

1. Color alone can usually be used to distinguish between approximately 4–5 different lines/markers (see color palette section below)
    a. Once you have 5+, you should also distinguish between lines/markers with different patterns (e.g., solid/dashed) and/or symbols (e.g., o, △, ☐)

2. The main colors on a graphic should be distinguishable even when printed in black & white
    a. Most journals allow color figures for the online version
        i. However, most only print in black & white
    b. But if your color scheme also works in black & white, no need to make two figures (more in the Graphics Themes section)

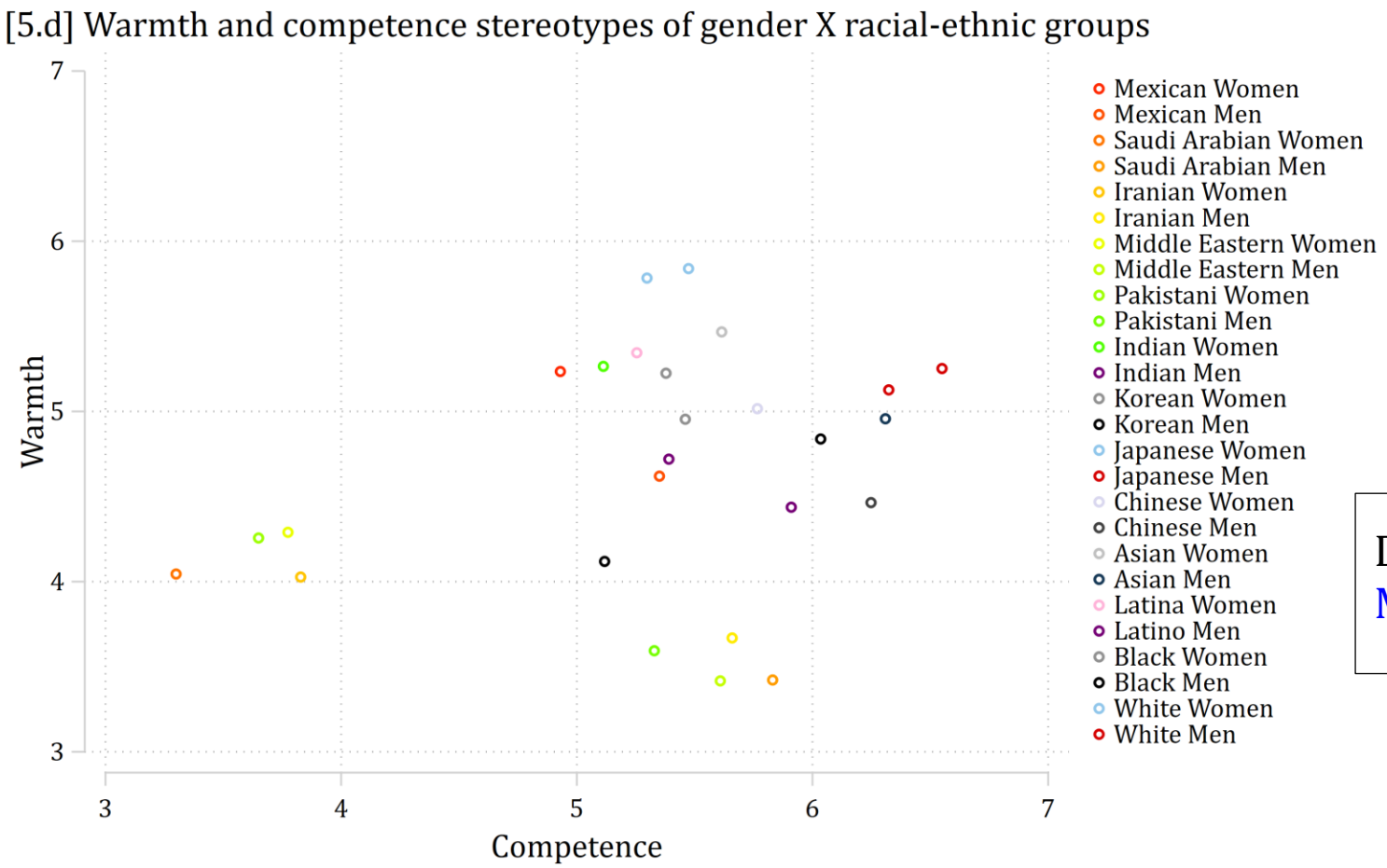
3. Avoid combining red and green
    a. Red-green colorblindness is by far the most common type of colorblindness, affecting up to 8% of men (more discussion of colorblind friendly colors in the Graphics Themes section)

```
ggplot(…, aes(x = factor(year), y = cog, color = degree, shape = degree)) + stat_summary(…)
regress cog ibn.year if in1992 & degree == `d', nocons   // one model per degree
coefplot (cog_deg0) (cog_deg1) (cog_deg2) … (cog_deg7), vertical …
```



Nominal scheme color limits

1. With a nominal scheme, it is hard to distinguish more than about 6 colors and *"…the task of matching colors to categories becomes too burdensome to be useful…"* *(Wilke 2019 p.234)*

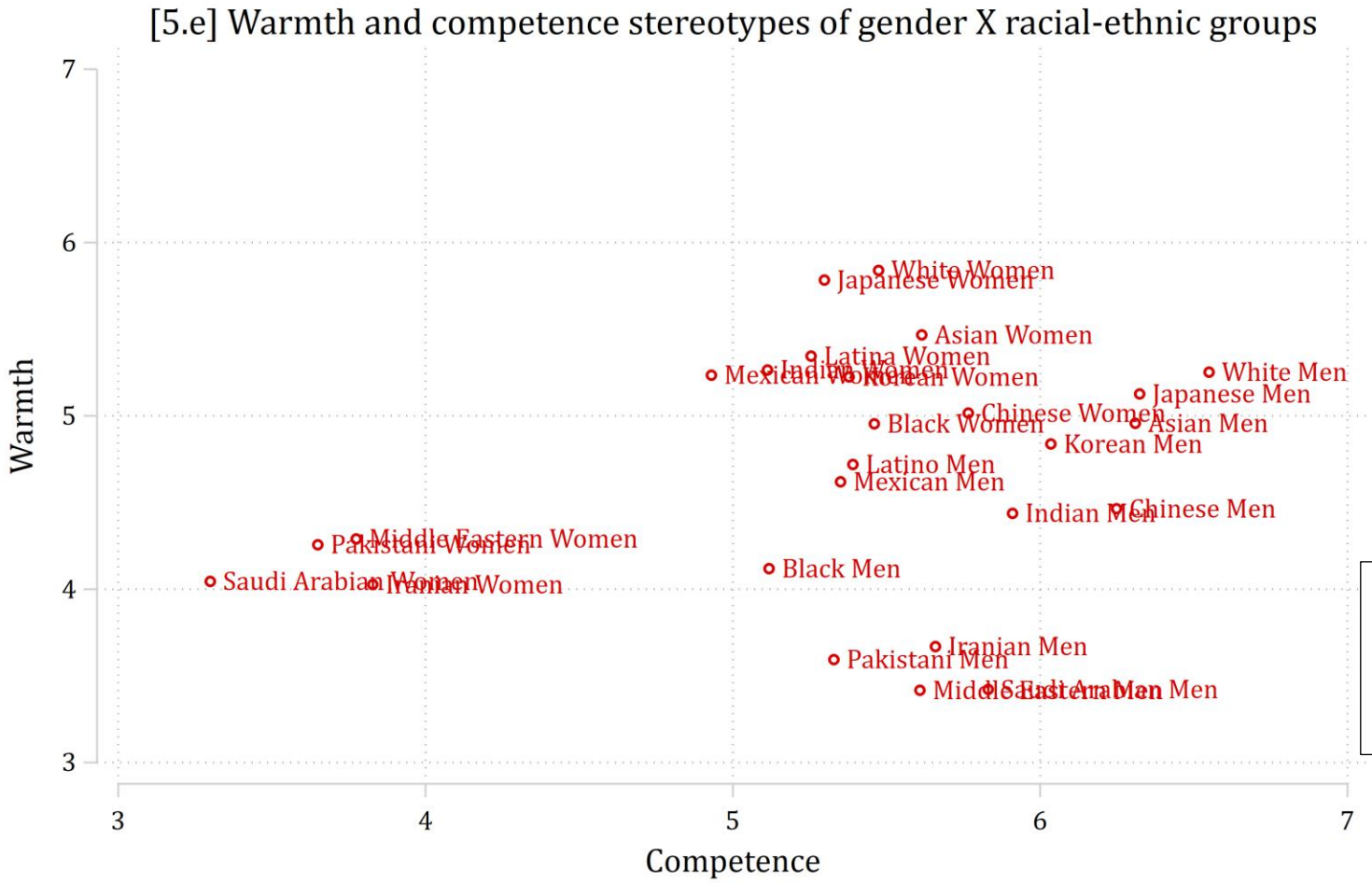


Data from Mize 2025

Nominal scheme color limits

1. With many categories, can be more effective to directly label the markers/lines

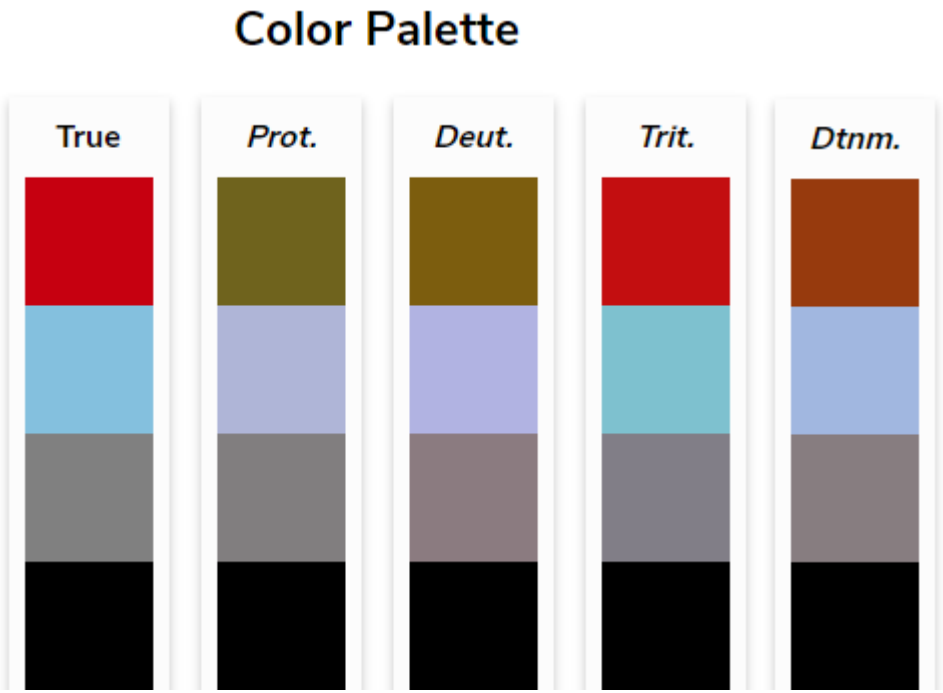
```
ggplot(means_by_condition, aes(x = mean_competence, y = mean_warmth, label = cond)) + geom_text( )
twoway scatter warmSS compSS, mlabel(cond) …
```



Data from Mize 2025

Creating figures considerate of colorblindness

1. About 5% of the population has some form of colorblindness (red/green being the most common which is why you should always avoid having red and green together on your figure… looking at you **`stcolor`**)
    a. This excellent website Coloring for Colorblindness lets you check your color schemes to see how they will be viewed for those who are colorblind
        i. Below is an example for how the first four default **`cleanplots`** colors look to individuals with various types of colorblindness

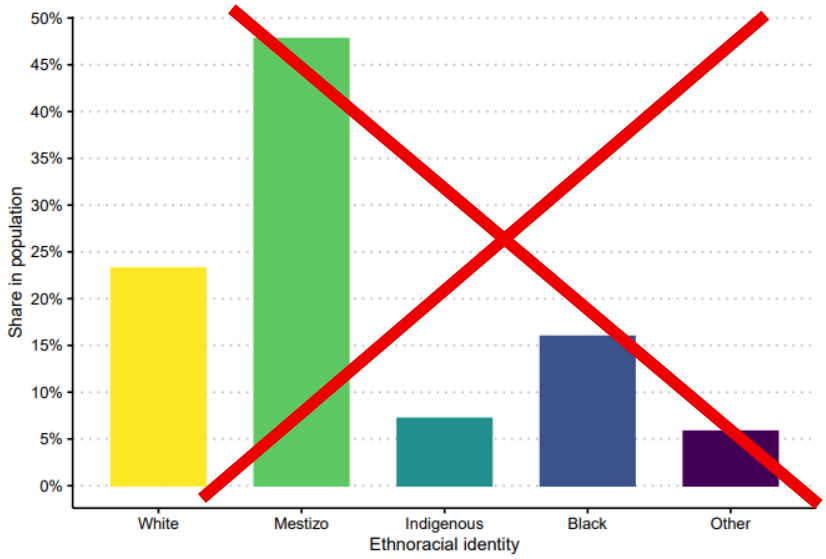


Nominal and ordinal color palettes

1. Most graphics themes' palettes are nominal
    a. I.e., They do not imply ordering by the color choices
    b. `okabe_ito` and `cleanplots` are nominal palettes

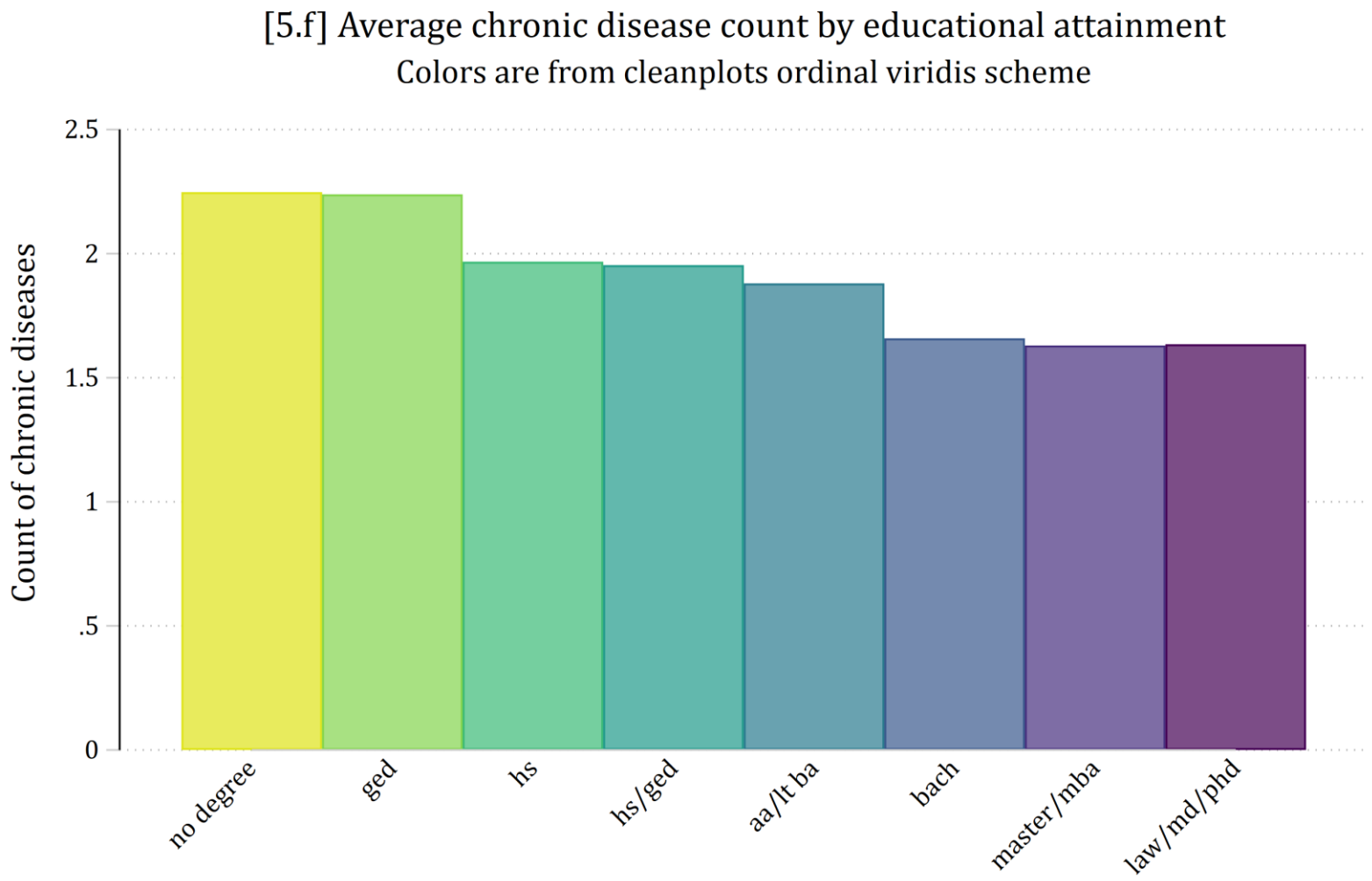
2. It is sometimes useful to use an ordinal palette to have the visual reflect the ordering in the variable (e.g., for educational degrees)
    a. The viridis and cividis palettes are my preferred ordinal palettes
    b. Note you should not use an ordinal palette for a nominal variable (falsely implies ordering)





Should use a nominal palette for a race-ethnicity variable

```
ggplot(hrs, aes(x = degree, y = dzcount, fill = degree)) + … + scale_fill_viridis_d(direction = -1)
graph bar (mean) dzcount, over(degree) asyvars showyvars … scheme(cleanplots8v)
```



*Note ordering of colors. Logic = darker, more saturated colors = more of construct

Principles of good data visualization

1. Look at your data
2. Use graph types people read accurately
3. Maximize the data-ink ratio, within reason (maybe)
4. Show uncertainty
5. Make groups easy to tell apart
6. **Choose axes that fit the data**
    a. The “ticks” and gaps on the axes should be clear and logical
        i. Balance the readability of the axes with precision/number of ticks
        ii. More than 2 or 3 decimal places are rarely necessary

Should your y-axis always start at zero?

The Rise of the Y-Axis-Zero Fundamentalists

It's OK not to start your y-axis at zero

Shut up about the y-axis. It shouldn't always start at zero.

YOUR AXIS DOESN'T ALWAYS HAVE TO START AT ZERO

*Headlines above from: (1) Justin Fox, (2) Quartz, (3) Vox, and (4) Axis.

Are the "*y*-axis zero fundamentalists" right?

"Charts should convey information and make a point. We make charts to illustrate ideas that have context beyond their x- and y-axes. Forcing the y-axis to start at zero can do just as much to obscure and confuse the point as the opposite." *(Yanofsky 2015 in Quartz)*

1. Graphs are tools for *showing the data*
    a. The data should always be *accurately and honestly* represented
    b. Including zero points on axes does not guarantee accuracy—in fact, it may be less accurate and honest than excluding it!

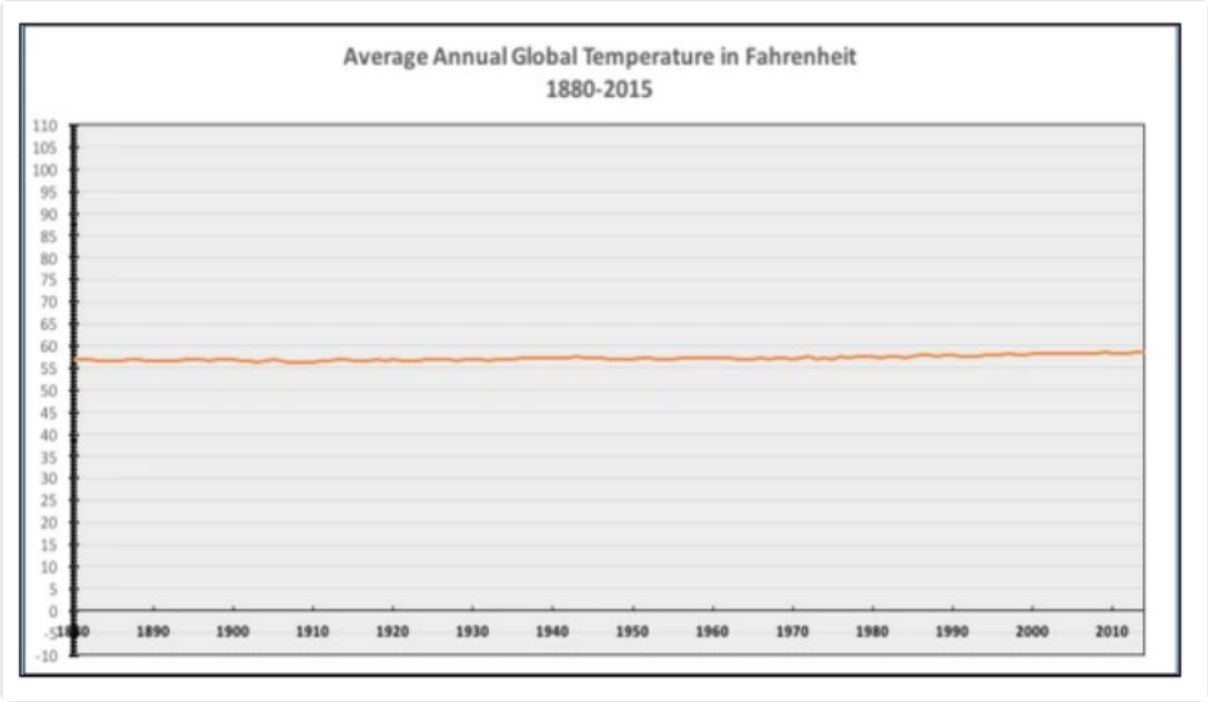


National Review
@NRO

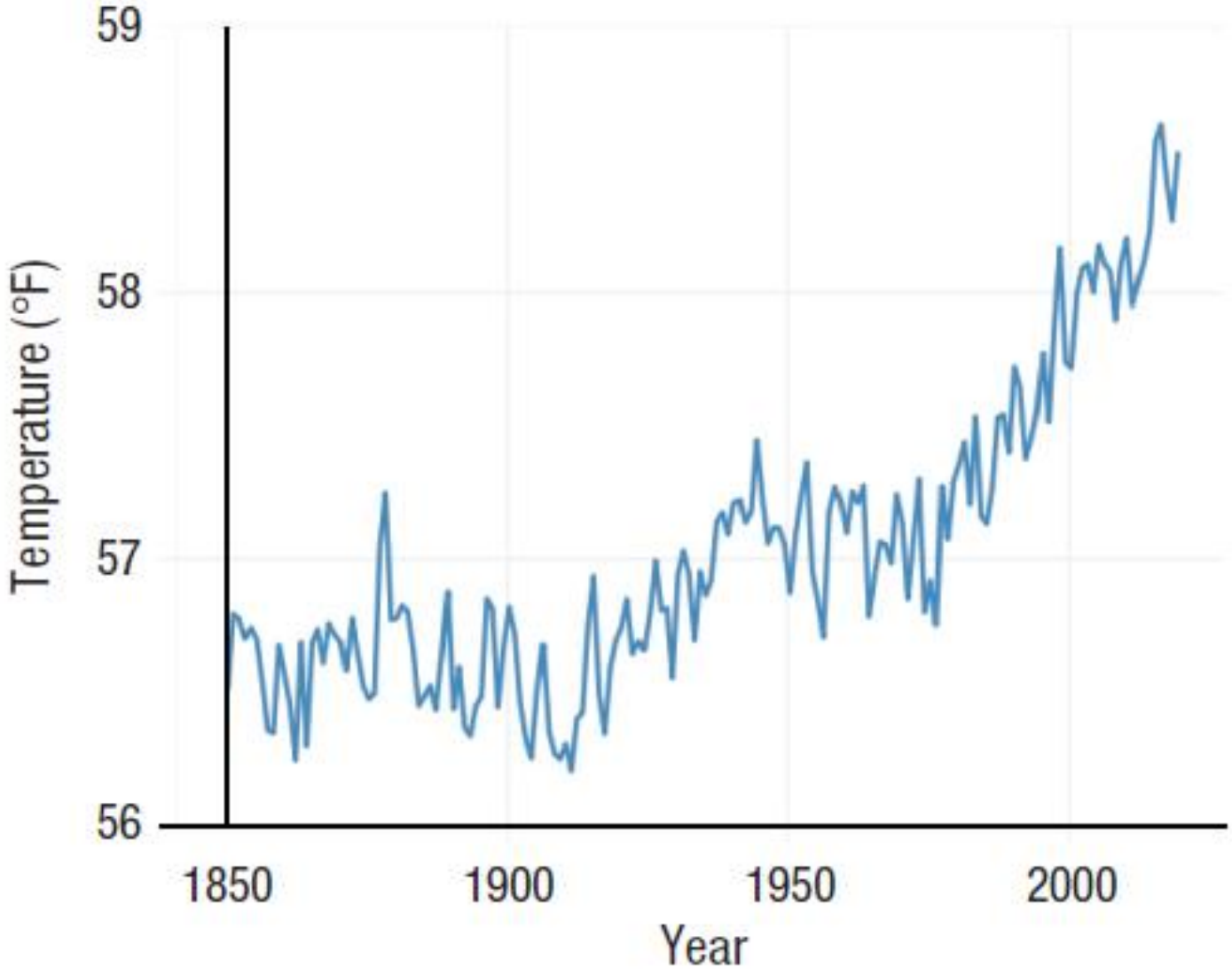
Follow

The only #climatechange chart you need to see. natl.re/wPKpro

(h/t @powerlineUS)



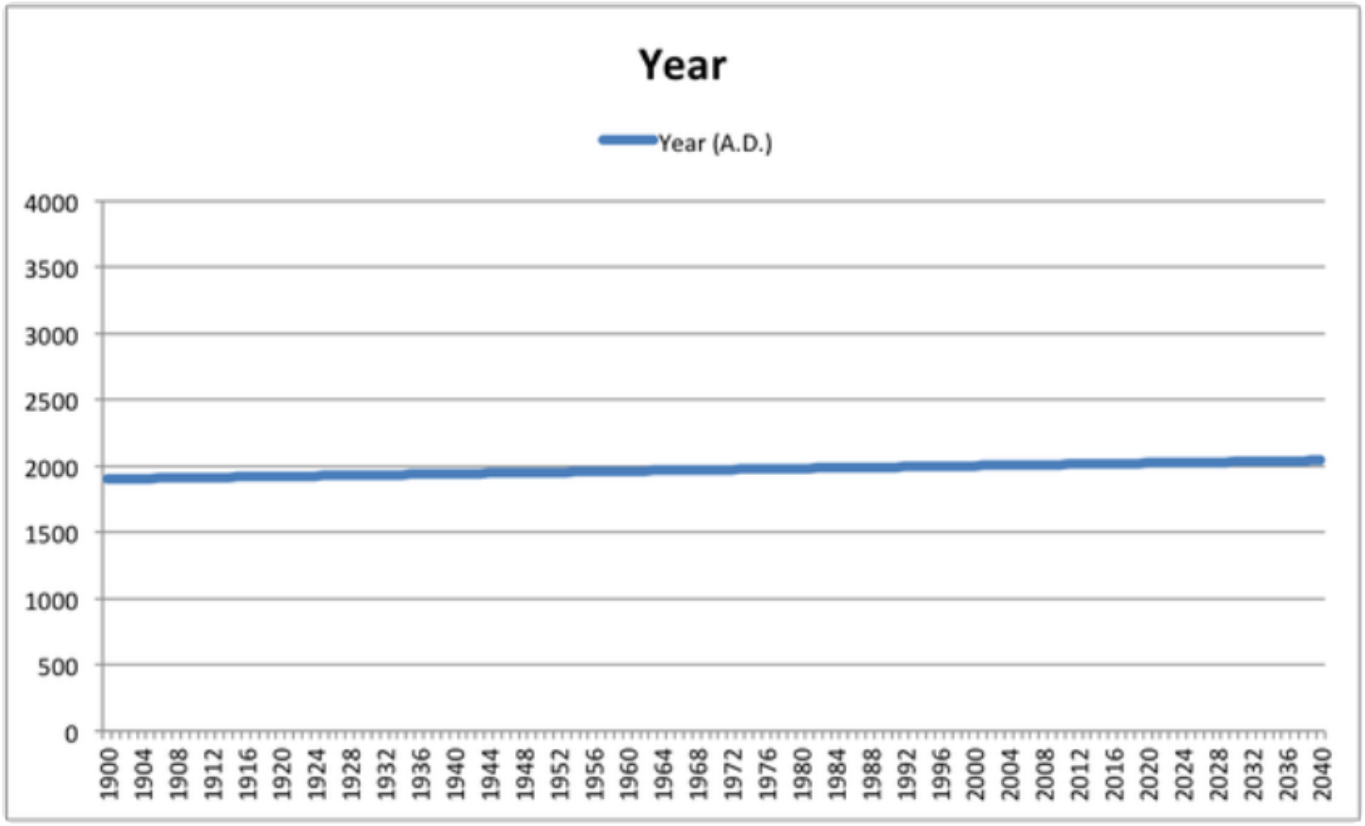
4:36 PM - 14 Dec 2015


59
58
57
56
Temperature (°F)
1850
1900
1950
2000
Year

Bloomberg GFX Teens
@bizweekgraphics

Follow

#NROgraphics



5:46 PM - 14 Dec 2015

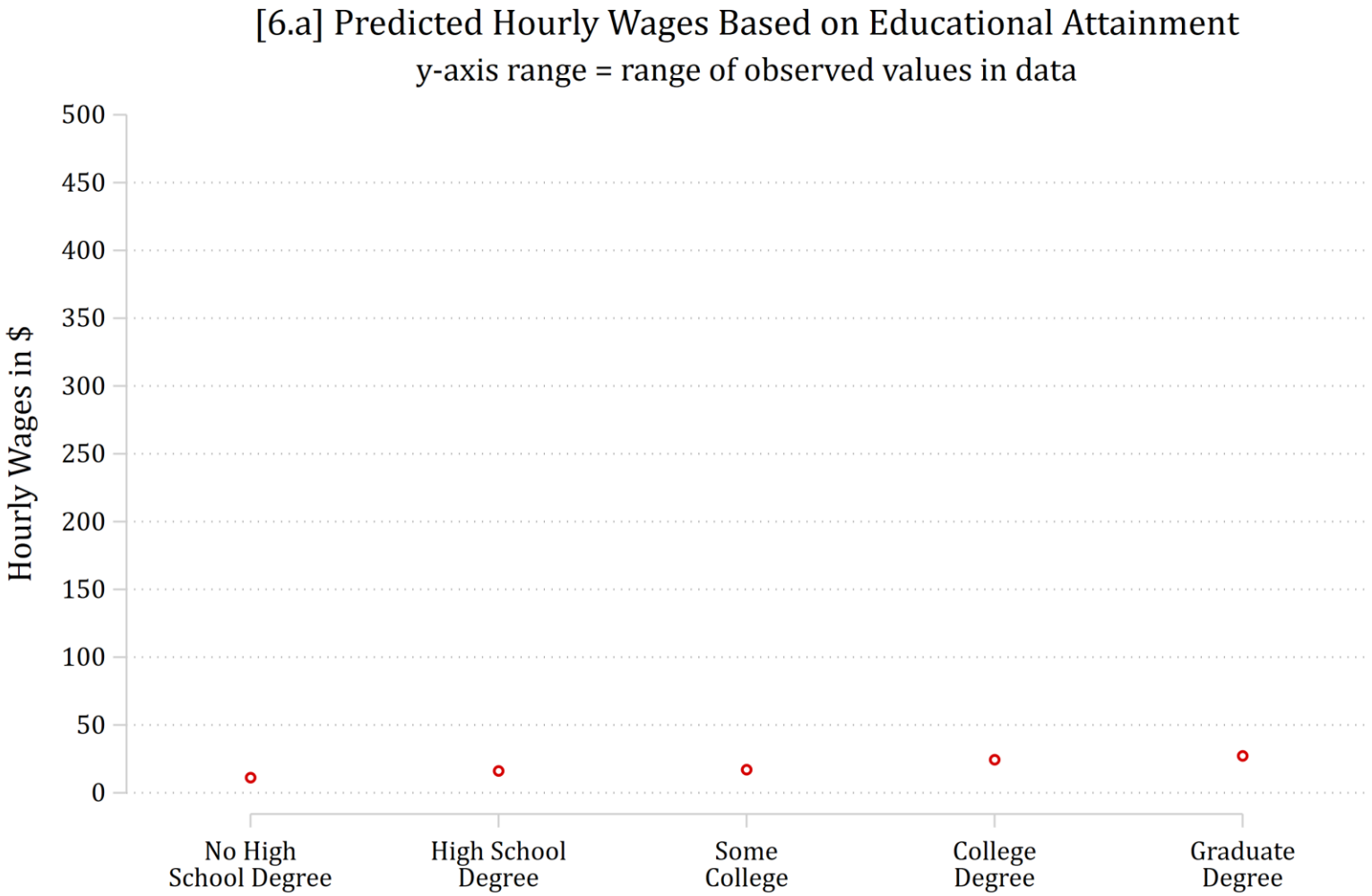
Why the y-axis doesn't have to cover the range of the data

"… there is a lot to be said in favor of showing the data over the range we observe it, rather than forcing every scale to encompass its lowest and highest theoretical value. Many otherwise informative visualizations would become useless if it was mandatory to include a zero point on the x- or y-axis." (Healy 2018)

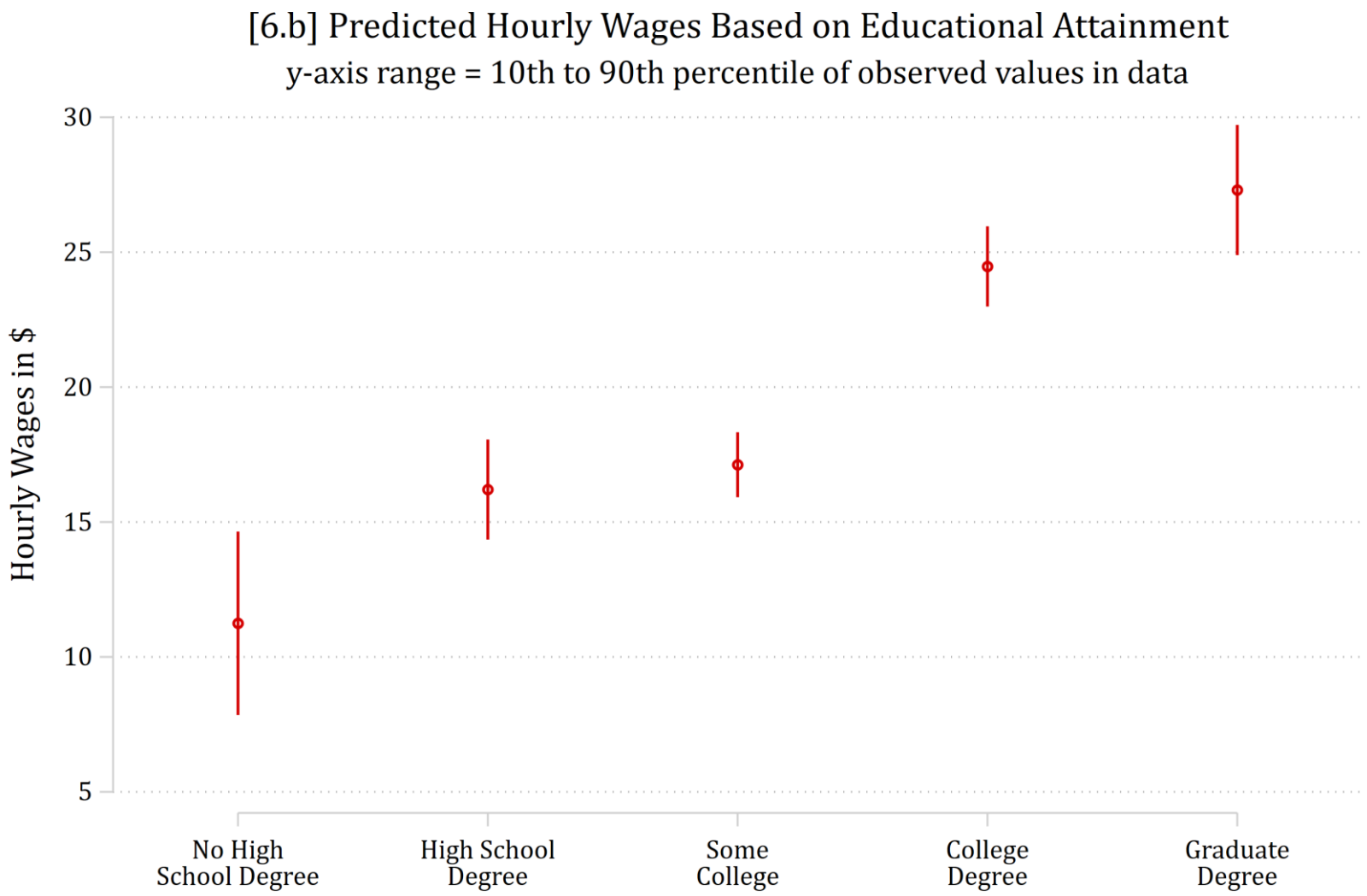
1. The most popular (sometimes misguided) rule is to always include zero on your axis
    a. The second most popular (sometimes misguided) rule is to have your axis cover the entire observed range in your data

2. But consider the following examples predicting hourly wages based on educational attainment
    a. Which axis range do you think is most **"truthful"** to represent the story in the data?

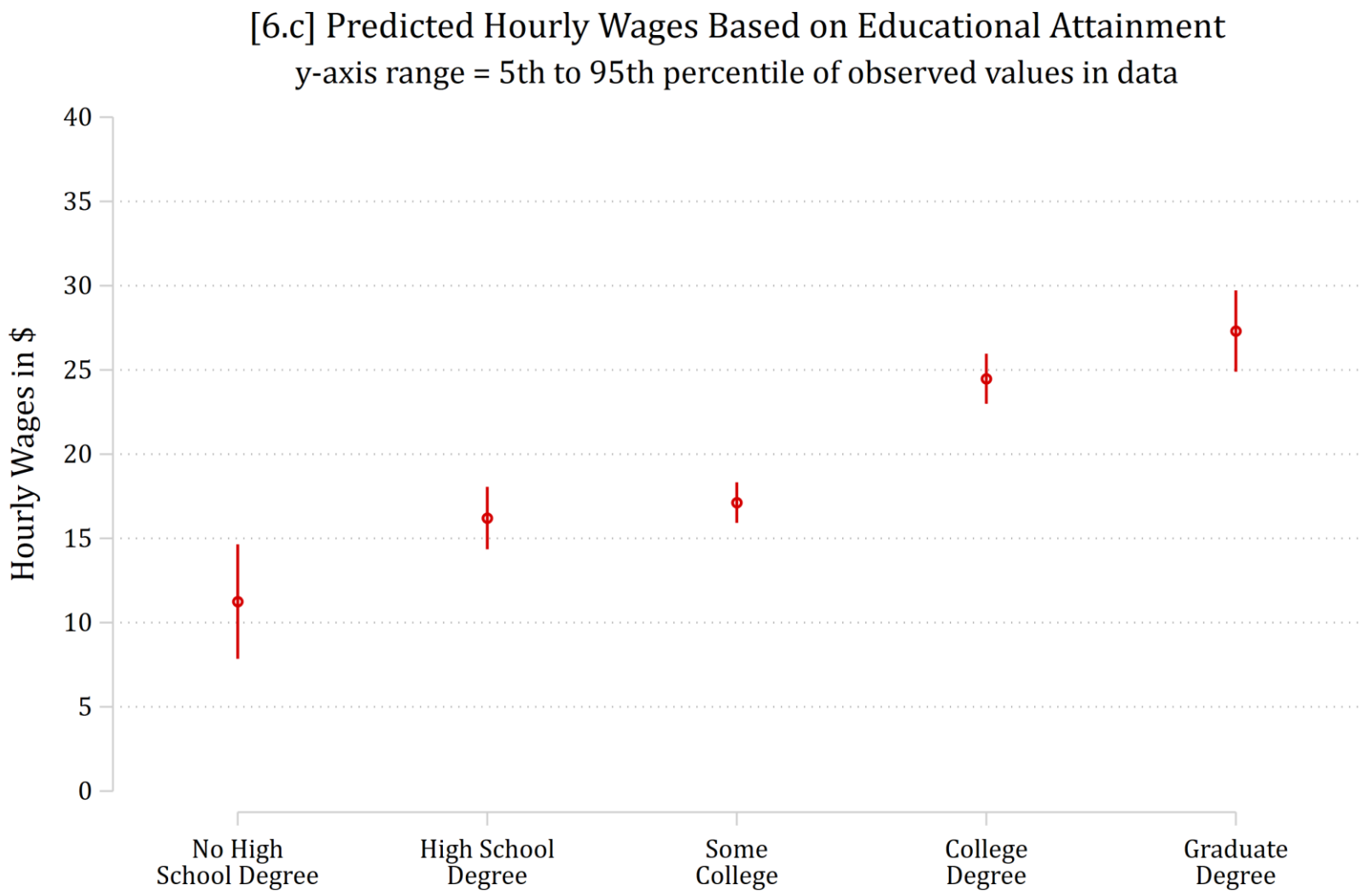
```
ggplot(ah4_emp, aes(x = educ, y = wages)) + … + scale_y_continuous(limits = c(0, 500))
reg wages ibn.educ if employed, nocon
coefplot, noci vertical ylab(0(50)500) …
```



```
ggplot(ah4_emp, aes(x = educ, y = wages)) + … + coord_cartesian(ylim = c(5, 30))
coefplot, vertical ylab(5(5)30) …
```



```
ggplot(ah4_emp, aes(x = educ, y = wages)) + … + coord_cartesian(ylim = c(0, 40))
coefplot, vertical ylab(0(5)40) …
```



Guidelines for choosing axis values

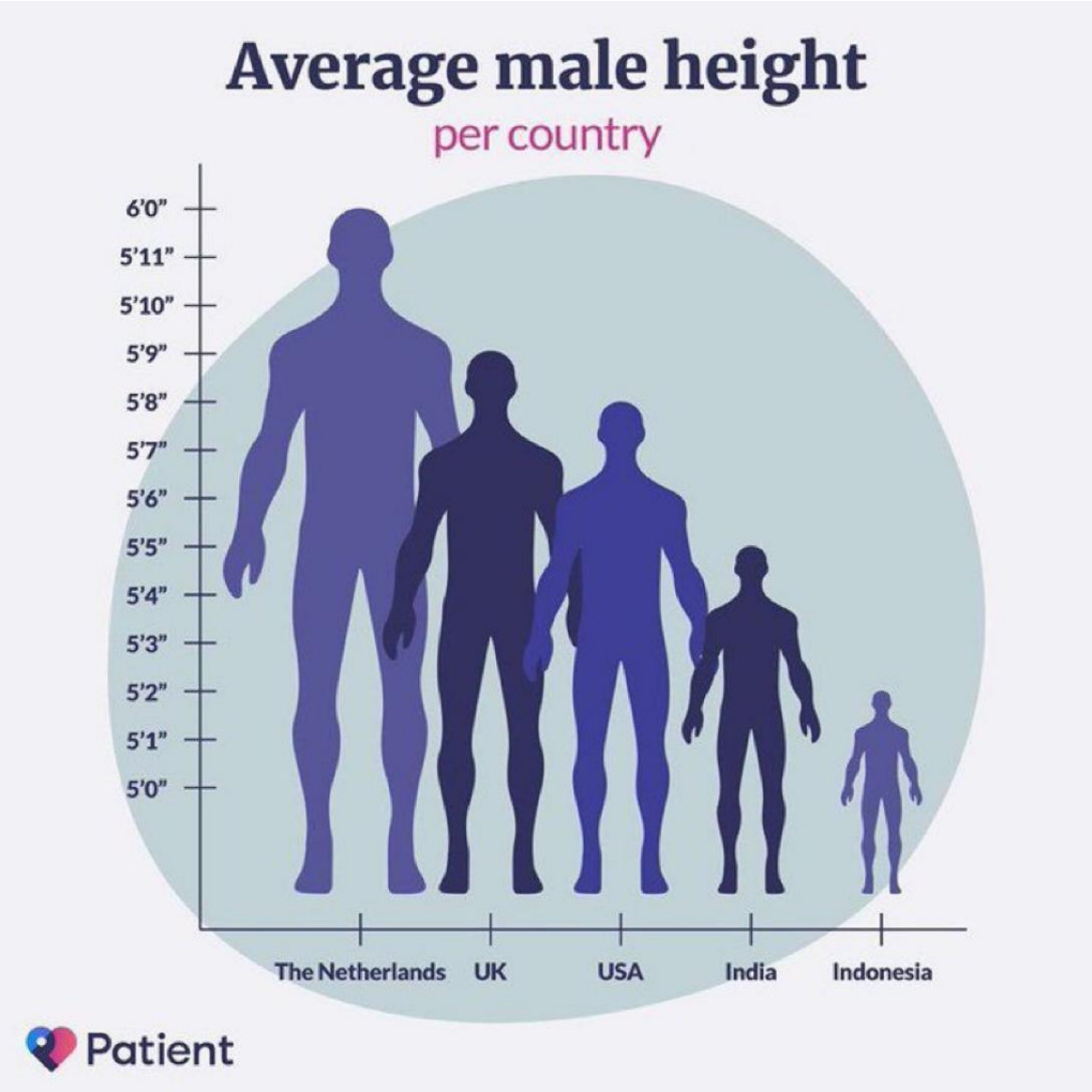
"There is emerging research that suggests that perhaps starting bar charts at something other than zero does not bias our perception… participants were better able to assess the sensitivity of the results (e.g., no effect, small effect, medium effect, or big effect) and more accurate (e.g., the size of the effect) when the vertical axis was set at a range more consistent with variation of the data." *(Schwabish 2021)*

1. My personal rule: the *y*-axis (and usually the *x*-axis as well) should cover at least the 10th to the 90th percentile of observed values
    a. 5th to the 95th percentile is sometimes more appropriate

2. Often, what we are trying to communicate is an effect or a difference across groups—not necessarily the exact value of the statistic (tables better for this)

3. All of these rules are mostly based on the logic of exploratory visualization of all sample values
    a. Unclear that these rules work well when applied to model results
        i. Sometimes, even more restrictive ranges better represent the data
        ii. Consider these "starting rules:" a good default to choose first—but that should be updated if a different choice better represents the data

Bad graph levity



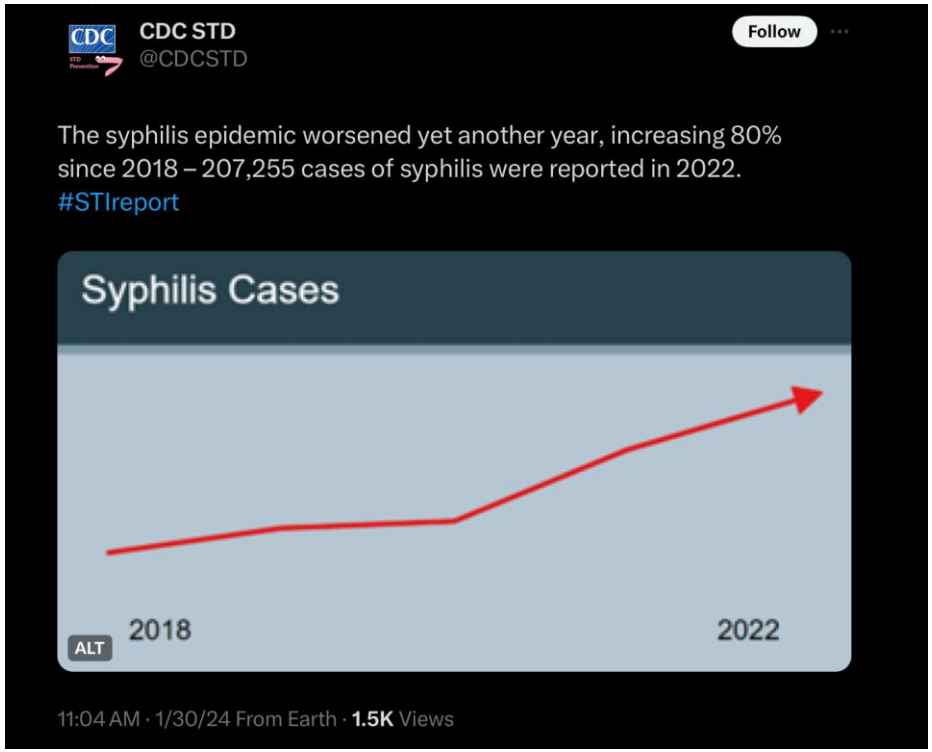
Principles of good data visualization

1. Look at your data
2. Use graph types people read accurately
3. Maximize the data-ink ratio, within reason (maybe)
4. Show uncertainty
5. Make groups easy to tell apart
6. Choose axes that fit the data
7. **Design for your audience**

Titles and axis labels

1. All graphs should have a descriptive, informative title

2. Both the *x* and *y*-axis should have a substantive label



General advice: talks vs articles

1. Font size
    a. Talks: the font in the graph (titles, labels, axis titles, etc.) should be large enough to be easily read from afar
        i. This is larger than you think!
    b. Articles and Slides: the font size should flow seamlessly with the text of the article/slides*
        i. Smaller, and the graph will be unreadable
        ii. Larger, and the graph will look out of place

2. Titles
    a. Talks: a clear and descriptive title should be placed within the graph itself (usually at the top of the graph)
    b. Articles: titles are generally typeset as part of the article—not as part of the graphics file itself, and are generally placed under the graph

*In R's ggplot2, use **`base_size = #`**; in Stata, use **`scale(#)`** where # > 1 to increase all font sizes

Talks vs articles

"The clearest conclusion from most of these studies is that tables are best suited for looking up specific information, and graphs are better for perceiving trends and making comparisons and predictions… Subjects who indicated a preference for tables were generally novices: laypersons unused to statistical information or beginning college students. Studies using more sophisticated subjects showed a preference for graphs..."
(Gelman, Pasarica, and Dodhia 2002)

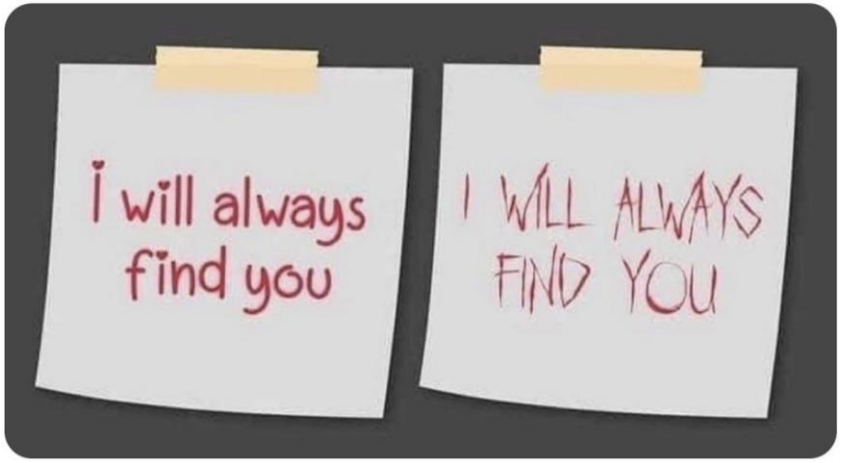
1. Tables vs graphs
    a. Talks: keep your audience awake (and aid more rapid interpretation): use graphics whenever possible
    b. Articles: graphics are more effective for observing patterns and overall relationships; tables for specific numbers
        i. I find graphics in the main text and tables in the Online Appendix useful
            1. Always helpful to include the table (e.g., a regression table) somewhere just in case a reviewer or editor wants to see specifics (and, useful for meta-analyses)

General advice: fonts

1. Fonts within figures should present the information clearly and not distract from the important part of the figure: the data

2. Fonts for graphics should not be too "light" or too thin or they will be difficult to read – e.g., when there is a single number on an axis: 12345
    a. They also should not be too "heavy" or compressed, or text can run together: **12345**
        i. Note that letters can look very different than numbers for some fonts: ABCabc 12345 (Georgia font)

FONT MATTERS!



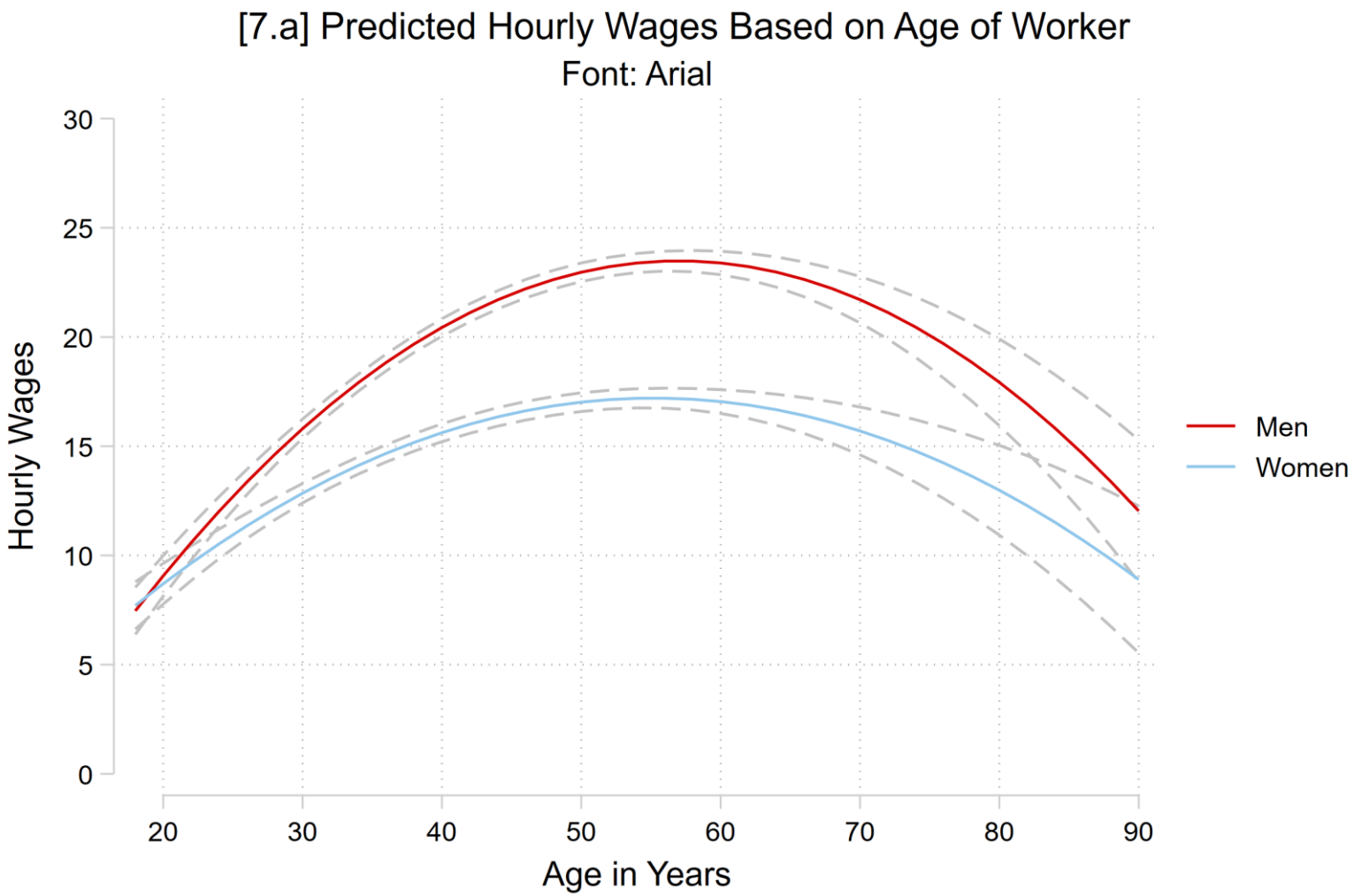
Serif and sans serif fonts

1. I often see recommendations for sans serif fonts with the rationale they are more accessible; this is largely a myth (APA Style Guide)
    a. There are accessible serif fonts and inaccessible sans serif fonts
    b. Tufte (1983) recommends serif fonts
        i. Most journals use serif fonts
            1. *"…unify the elements of your presentation. This means consistency in your use of colors and fonts…(etc.)."* *(Schwabish 2017 p.4)*
    c. If using a serif font for graphics: fonts that are relatively minimal/unadorned/have small serifs tend to be more readable

2. Some fonts I personally like for graphics*, that are accessible to read:
    a. Serif: Cambria (download for Mac), Times New Roman, Palatino
    b. Sanserif: Segoe UI (download for Mac), Helvetica (download for Windows), Arial*, Trebuchet MS

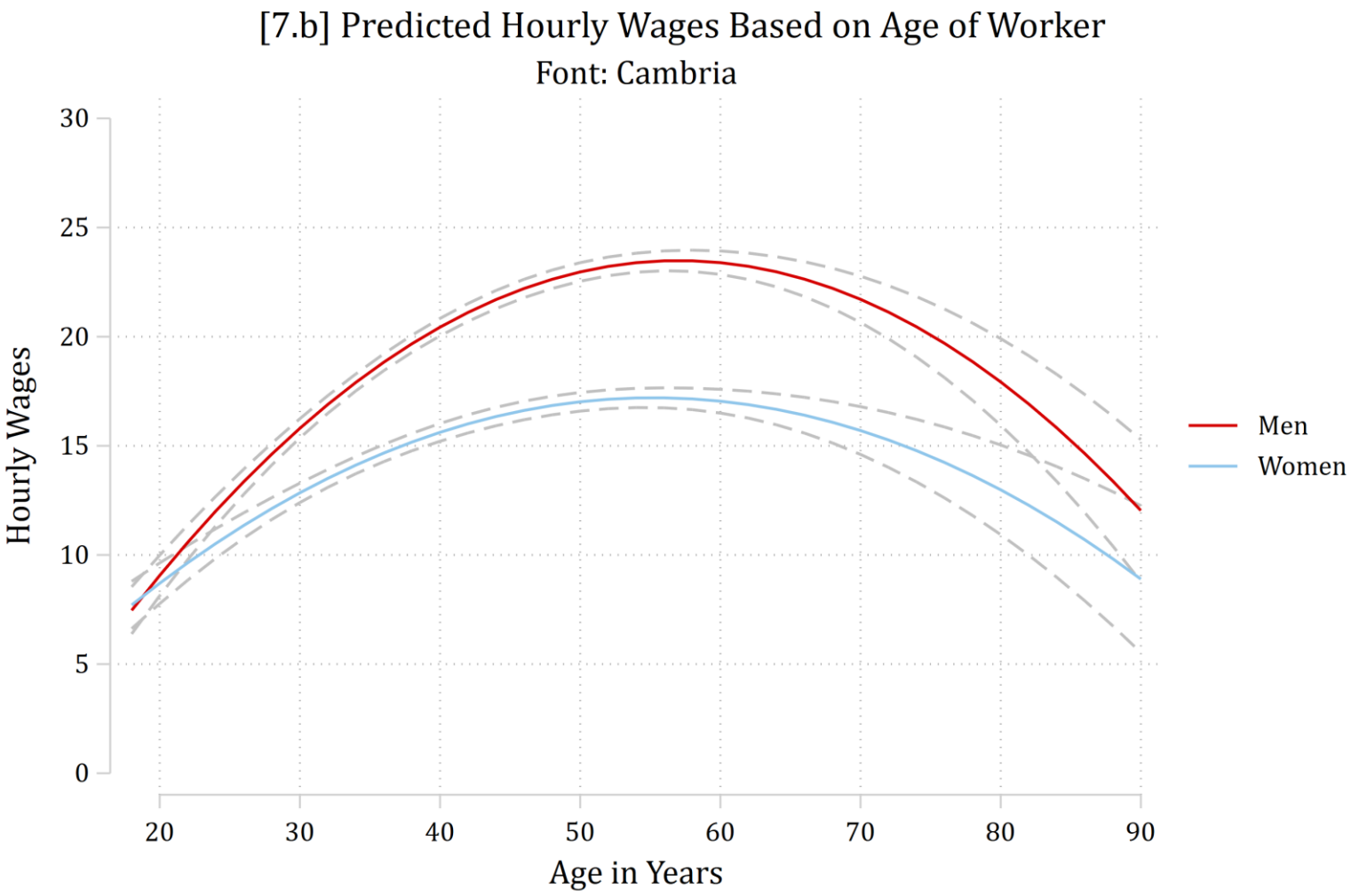
3. To make more fonts available in R, use `extrafont` or `windowsFont`

```
fig7 + theme(text = element_text(family = "Arial"))
graph set window fontface "Arial"
```



*Arial (or Helvetica – of which it is a clone) looks nice – but very “default”

```
fig7 + theme(text = element_text(family = "Cambria"))
graph set window fontface "Cambria"
```



*Cambria is my normal go-to font for slides and graphics

Graphics schemes in Stata

1. Stata's graphics unfairly have a bad reputation
    a. Stata's graphical capabilities are *impressive* and *flexible*
    b. Stata's *default* graphics choices (pre-Stata 18) *were not great*

2. Stata has many graphics schemes other than its pre-18 default (**`s2color`**)
    a. Stata 18's default scheme (**`stcolor`**) is much better
        i. See examples of additional built-in schemes here

3. Some nice user-written schemes exist that work well
    a. The **`blindschemes`** package comes with multiple schemes with nice properties (these are primarily modeled after R's ggplot2's themes)
    b. The **`schemepack`** package has many excellent schemes

4. **`cleanplots`** is a scheme I wrote to implement the best practices outlined in this seminar
    a. It is available for free on my website
    b. Almost all of the graphics made for this workshop use **`cleanplots`**

Graphics themes in R

1. .Base R's default graphics theme leaves much to be desired
    a. **ggplot2**'s default theme (**theme_gray**) is much better, but still imperfect
    b. The good news is there are good additional options built into **ggplot2**

2. You can change a theme on a single graph by specifying it as an element on a **ggplot2** graph. E.g.,

```
fig1 + theme_gray( )
```

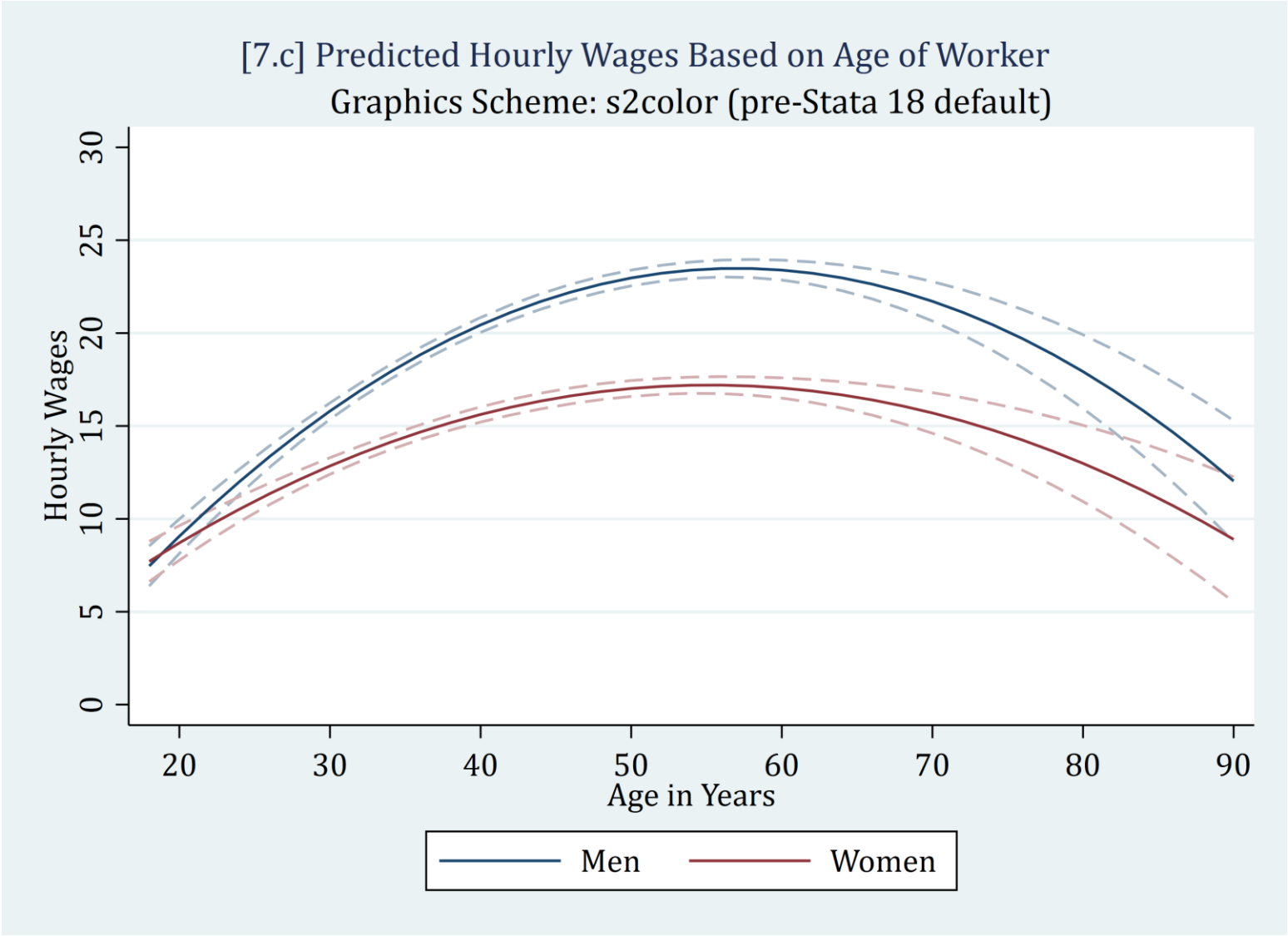
3. I prefer to set the theme at the top of my R file using **theme_set( )** so the same theme will apply to all graphs

```
theme_set(theme_minimal())
```

4. My favorite built-in **ggplot** theme is **theme_minimal**

5. My **cleanplots** theme is also available for R and works with **ggplot**

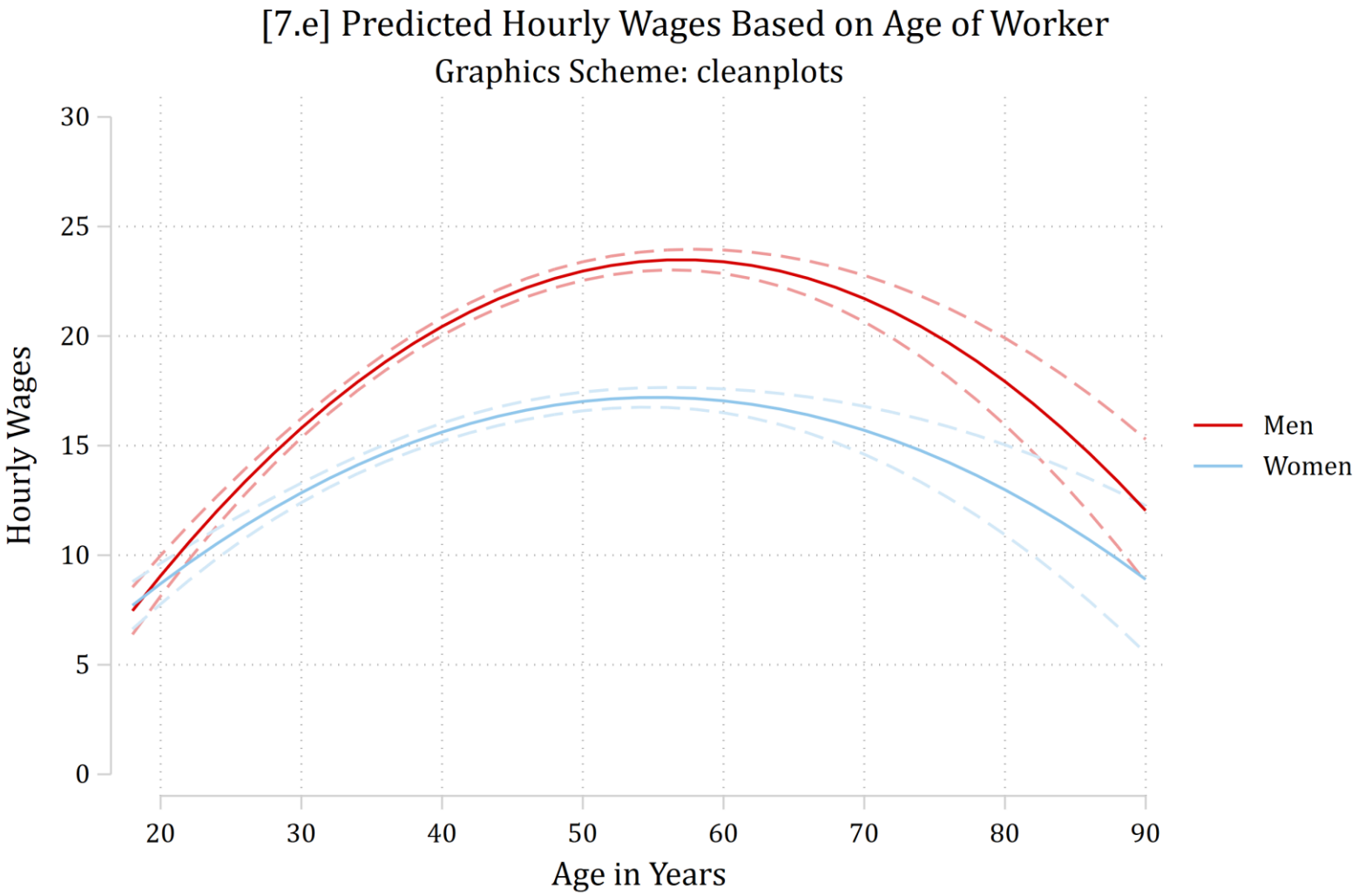
```
fig7 + theme_gray( )
marginsplot, … scheme(s2color)
```



```
fig7 + theme_minimal( )
marginsplot, … scheme(stcolor)
```



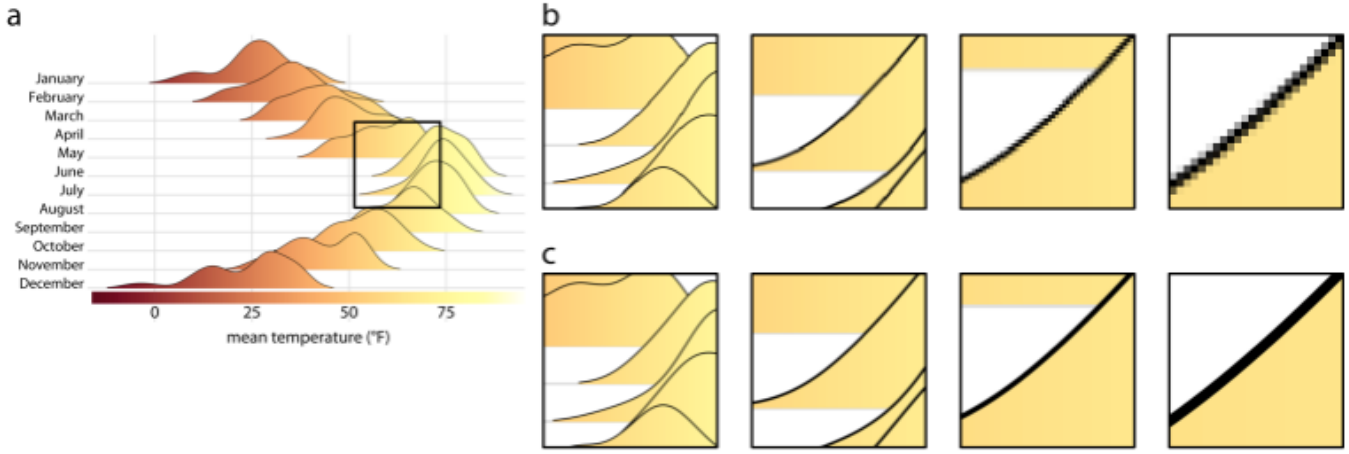
```
cleanplots_defaults( )
set scheme cleanplots
```



General advice: graphics file formats

1. I generally recommend PNG files
    a. PNG works well across platforms (Word, PowerPoint, websites; Windows, Mac; etc.)
    b. Avoid JPEG: JPEGs are optimized for photographs, not graphs

2. Vector graphics formats—PDF, EPS, and EMF—have some advantages but also some disadvantages
    a. Advantage: they rescale without pixelating
    b. Disadvantage: they are redrawn each time they are opened, so can look different across platforms (e.g., the original font may not be available)



Wilke 2019

Principles of good data visualization

1. Look at your data
2. Use graph types people read accurately
3. Maximize the data-ink ratio, within reason (maybe)
4. Show uncertainty
5. Make groups easy to tell apart
6. Choose axes that fit the data
7. Design for your audience
8. **Use graphs, not tables, for model results**

Why visualize model results?

"One great virtue of good graphical representation is that it can serve to display clearly and effectively a message carried by quantities whose calculation or observation is far from simple." *(Tukey and Wilk 1966)*

1. Visualizing data can simplify complex relationships
    a. Visualizations often synthesize large amounts of information succinctly; relationships/patterns become clear visually

2. Visualizations make your data more accessible and memorable (bonus: nice visualizations make your audience less likely to fall asleep!)
    a. *"The more visual the input becomes, the more likely it is to be recognized and recalled."* *(Medina 2008)*

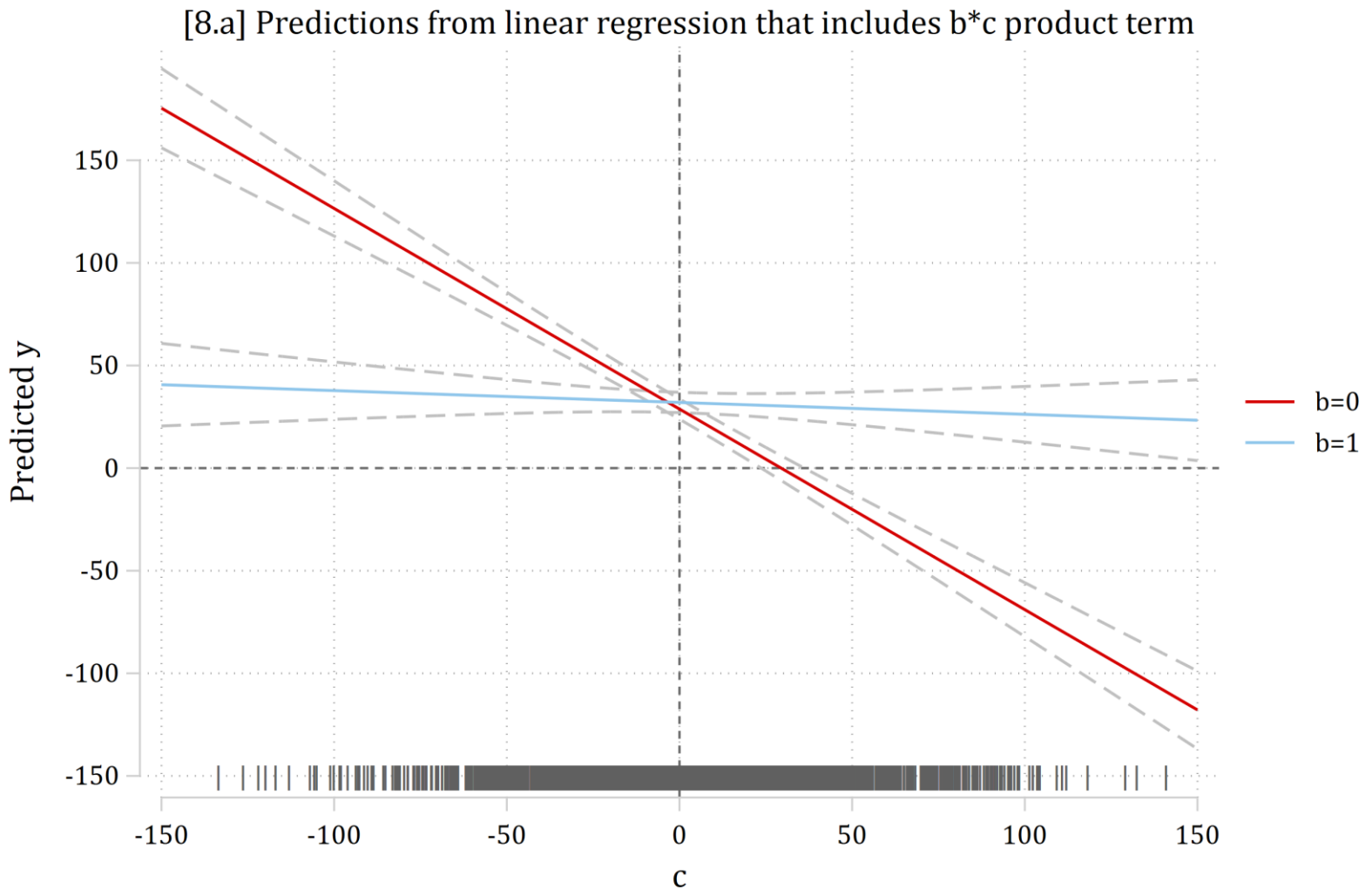
Can you extract the sunbeams from these cucumbers?

Linear Regression Results - b*c Interaction

	(1)	(2)	(3)
b	4.600		
	(3.657)		
c	-0.537***	-0.978***	-0.978***
	(0.0465)	(0.0629)	(0.0629)
1.b		3.107	3.199
		(3.571)	(3.574)
1.b#c.c		0.920***	0.920***
		(0.0908)	(0.0908)
x1			-1.204
			(1.786)
_cons	28.19***	28.86***	28.81***
	(2.587)	(2.525)	(2.526)
N	2000	2000	2000

Some sunbeams of interest: (1) Is there an interaction effect between *b* and *c*? (2) Is the effect of *c* (when *b* = 0) positive or negative? (3) Is the effect of *c* (when *b* = 1) positive or negative? (4) Where across the range of *c* is the difference between *b* = 0 and *b* = 1 statistically significant? (5) Where is the observed data?

```
plot_predictions(mod1c, condition = c("c", "b"))
qui margins, at(c=(-150(5)150) b=(0 1))
marginsplot, recastci(rline) …
```



Can you extract the sunbeams from these cucumbers?

```
Logistics Regression Results - b*c Interaction Effect
------------------------------------------------------------
                      (1)             (2)             (3)
------------------------------------------------------------
b                  -4.225***
                  (0.262)

c                   0.487***        0.147           0.148
                  (0.0832)         (0.313)         (0.313)

1.b                                -6.363**        -6.366**
                                   (1.950)         (1.950)

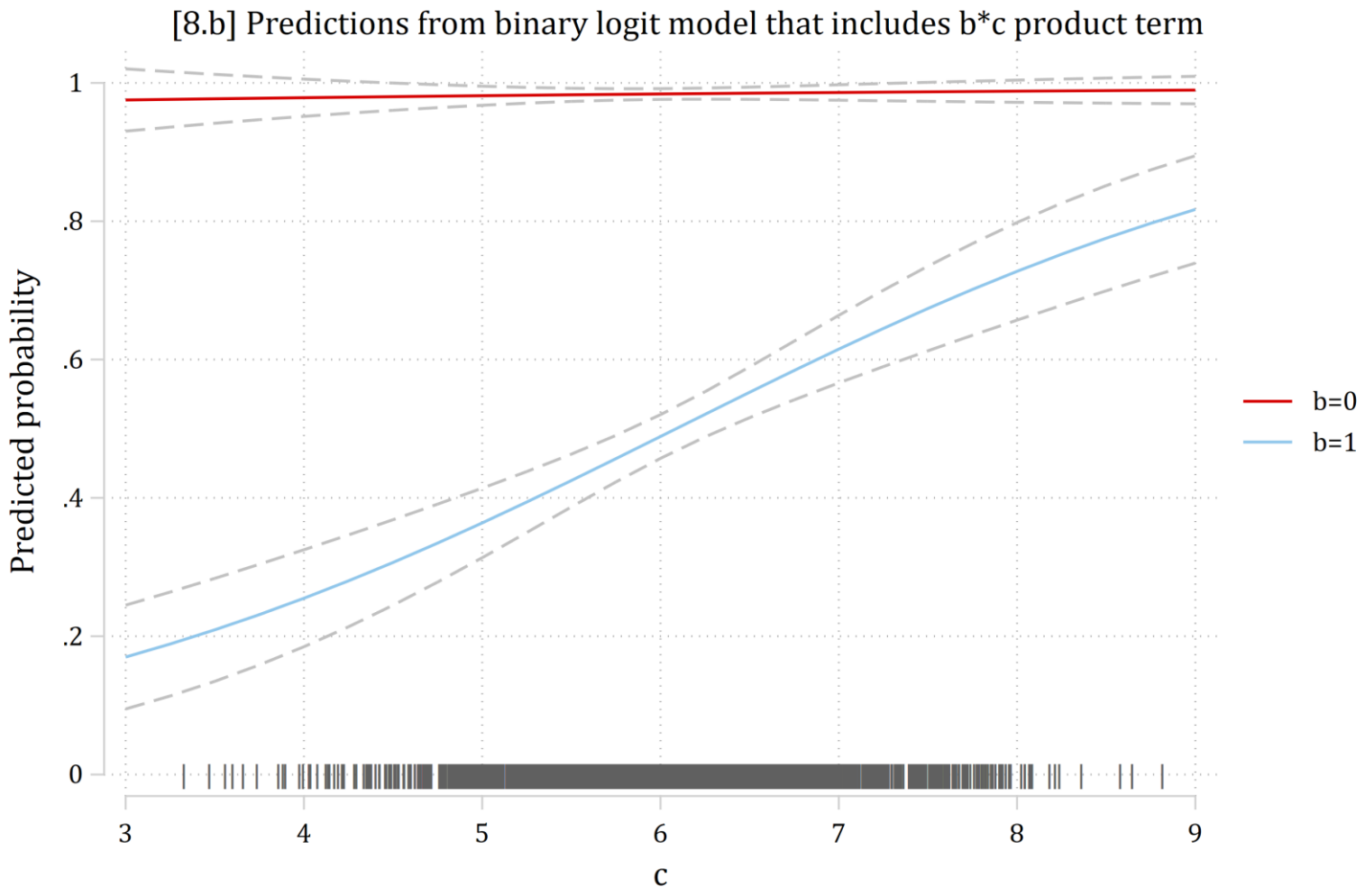
1.b#c.c                             0.366           0.367
                                   (0.325)         (0.325)

x1                                                 -0.0575
                                                  (0.0622)

_cons               1.257*          3.238           3.235
                   (0.539)         (1.877)         (1.877)
------------------------------------------------------------
N                    2000            2000            2000
------------------------------------------------------------
```

Some sunbeams of interest: (1) Is there an interaction effect between *b* and *c*? (2) Is the effect of *c* (when *b* = 0) positive or negative? (3) Is the effect of *c* (when *b* = 1) positive or negative? (4) Where across the range of *c* is the difference between *b* = 0 and *b* = 1 statistically significant? (5) Where is the observed data?

```
plot_predictions(mod2c, condition = c("c", "b"), rug = TRUE)
qui margins, at(c=(3(.25)9) b=(0 1))
marginsplot, recastci(rline) …
```



*Curious how this is possible? See Mize 2019 on nonlinear interaction effects

Giant regression tables in talks = ☹

[11.0] Linear Regression Results - Log Hourly Wages

	(1)		(2)		(3)	
Woman R	-0.37***	(0.01)	-0.37***	(0.01)	-0.36***	(0.01)
high school				(0.01)	0.35***	(0.01)
junior college				(0.02)	0.56***	(0.02)
bachelor				(0.02)	0.76***	(0.02)
graduate				(0.02)	0.98***	(0.02)
black				(0.01)	-0.03*	(0.01)
other				(0.02)	-0.07***	(0.02)
age of respondent				(0.00)	0.01***	(0.00)
Currently Married					0.15***	(0.01)
Number of Children					-0.00	(0.00)
catholic					0.05***	(0.01)
jewish					0.17***	(0.03)
none					0.01	(0.01)
other					0.01	(0.03)
buddhism					0.06	(0.07)
hinduism					0.25*	(0.10)
other eastern					0.38*	(0.17)
moslem/islam					-0.29**	(0.10)
orthodox-christian					0.07	(0.10)
christian					-0.01	(0.04)
native american					0.01	(0.20)
inter-nondenominat~l					-0.05	(0.08)
Lives in Metro Area					0.05***	(0.01)
Midwest					-0.09***	(0.01)
South					-0.13***	(0.01)
West					-0.04**	(0.01)
Constant	2.65***	(0.01)	1.61***	(0.02)	1.61***	(0.02)
Observations	31102		31021		30876	

Standard errors in parentheses
* p<0.05, ** p<0.01, *** p<0.001



Coefficient plots

1. Coefficient plots are very effective for replacing regression tables (of coefficients) in talks and (sometimes) in articles/reports
    a. Coefficient plots also display multiple model results better than tables—making it easy to compare estimates across models
        i. Measures of uncertainty (e.g., CIs) are also easily incorporated and visually easily processed

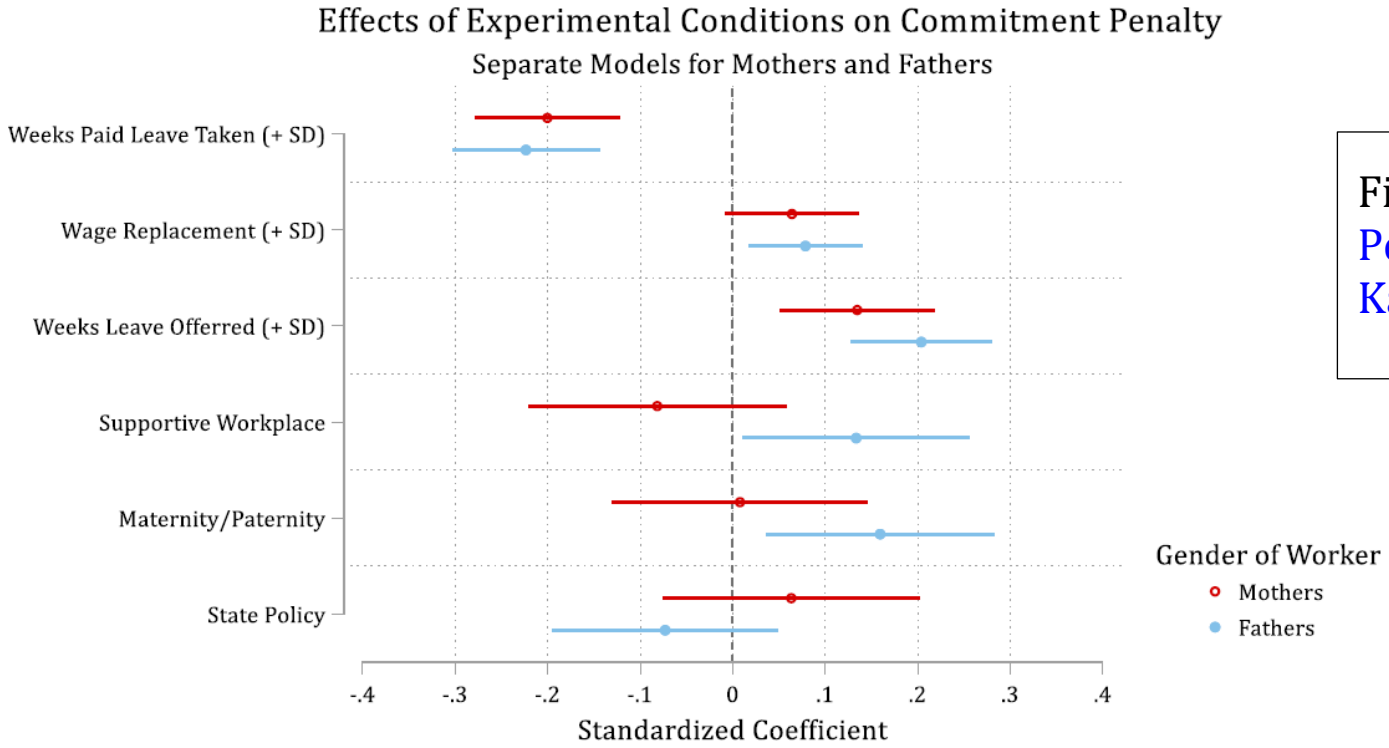
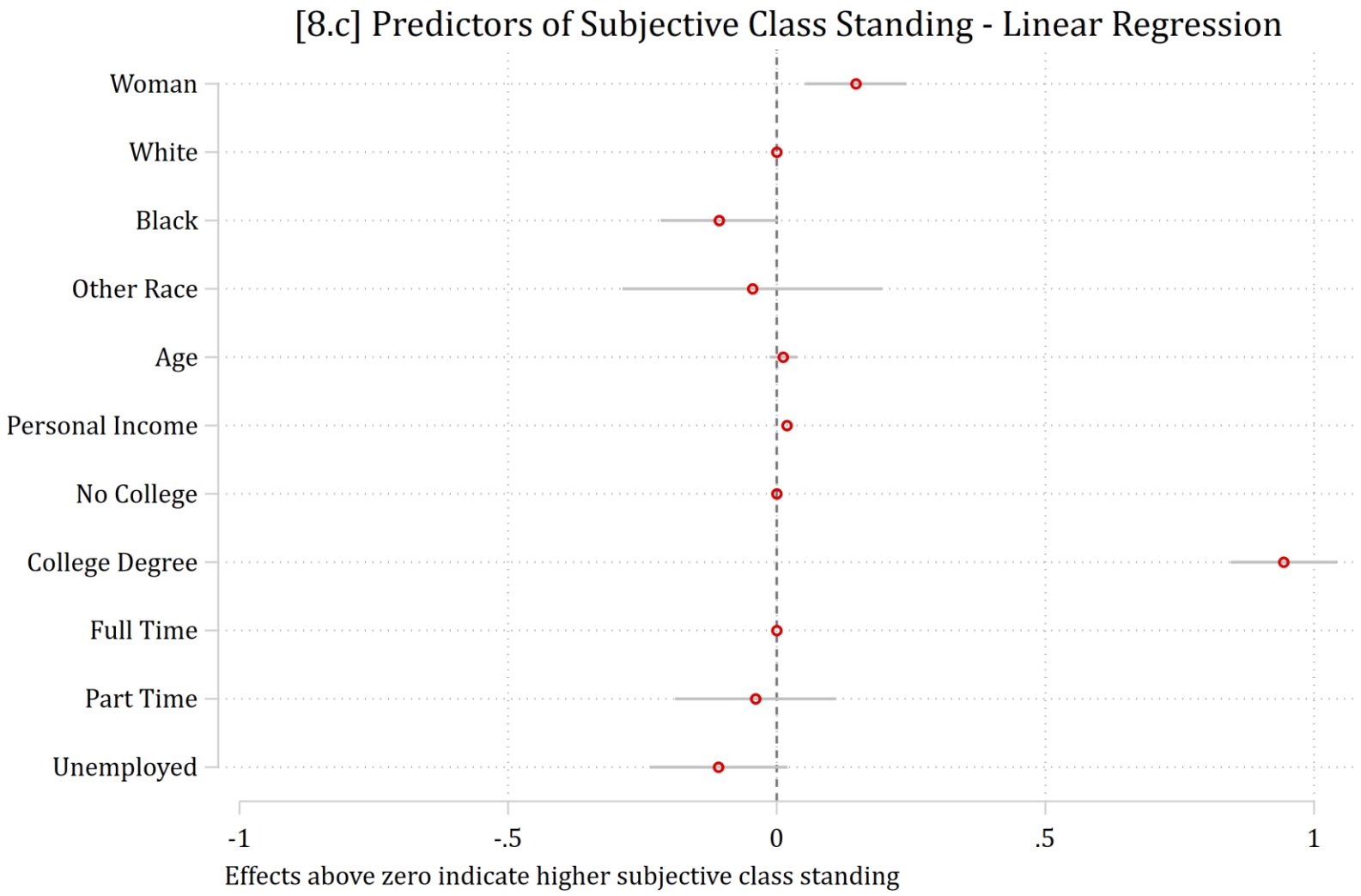


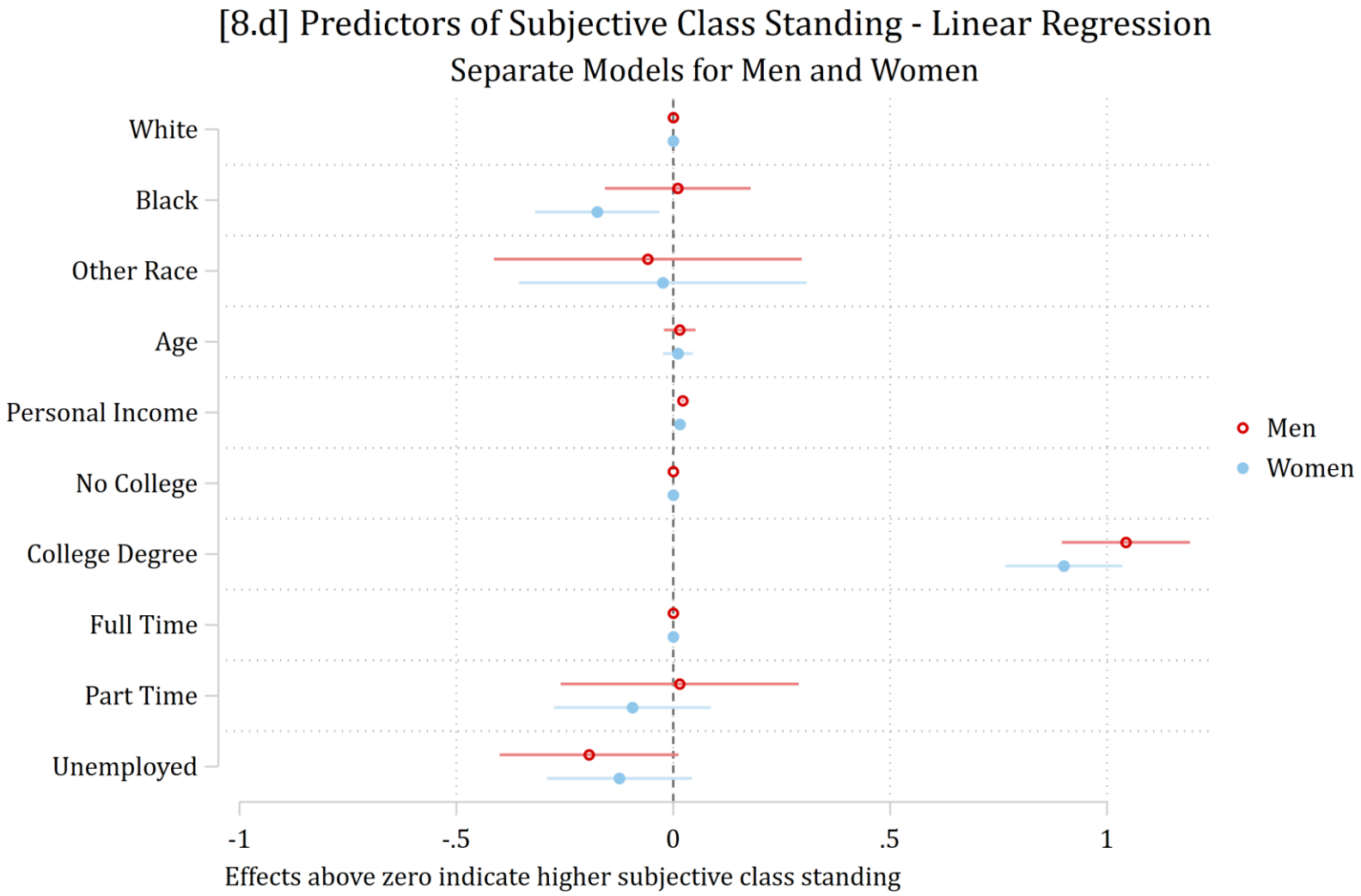
Figure 1 from Petts, Mize, and Kaufman 2022

```
modelplot(reg_sc, coef_omit = "Intercept") + geom_vline(xintercept = 0) + …
regress subjclass i.woman i.race3 c.age c.income i.college i.workstat
coefplot, xline(0) baselevels drop(_cons 0.woman) …
```



```
ggplot(modelplot(list("Men" = reg_men, "Women" = reg_wom), draw = FALSE), …) +
  geom_linerange(…, alpha = .5) + geom_point( ) + …

coefplot (scmen, label("Men")) (scwom, label("Women")), ciop(col(*.5)) …
```



Published example

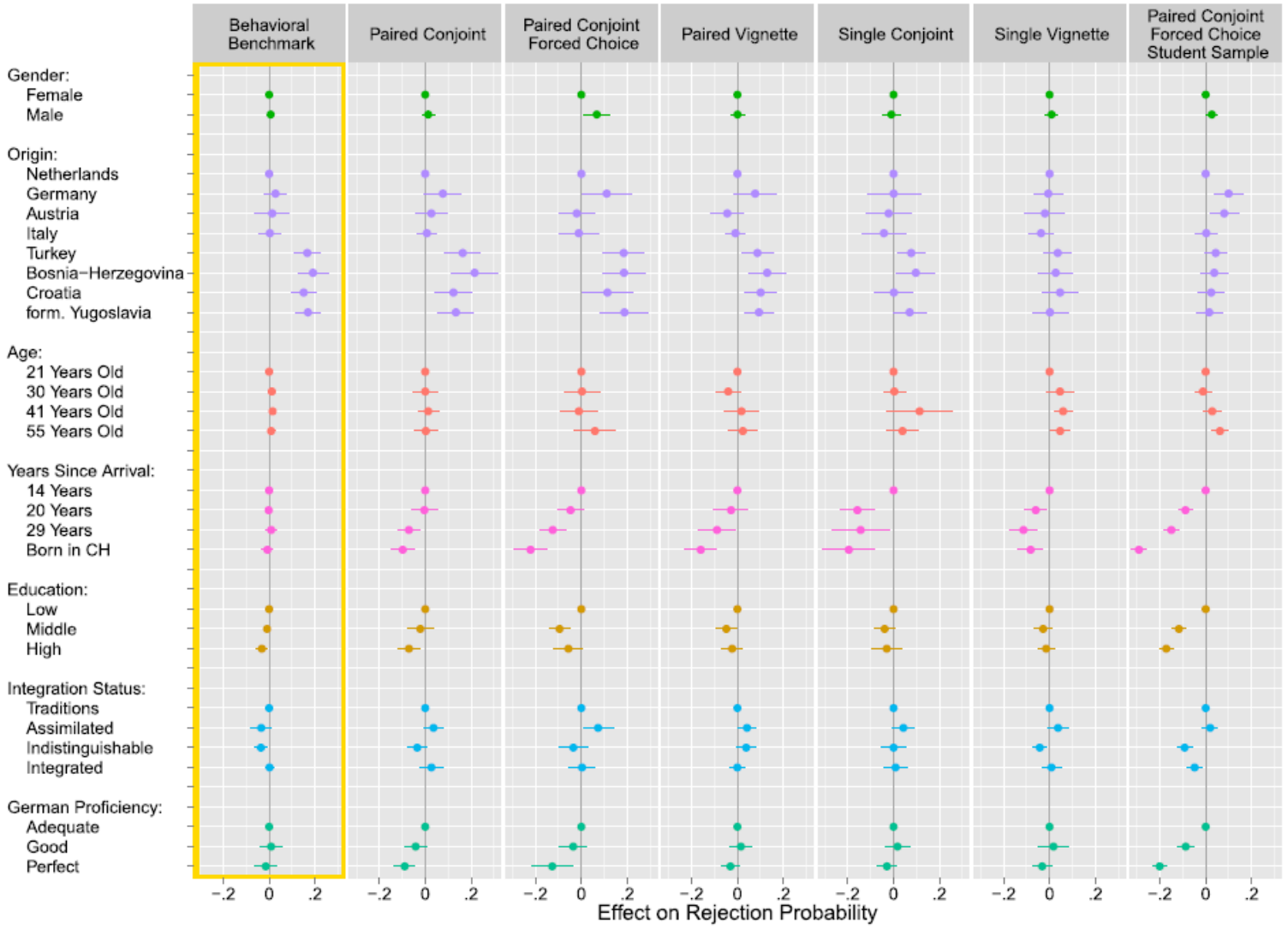


Figure 1 from Hainmueller, Hangartner, and Yamamoto 2015

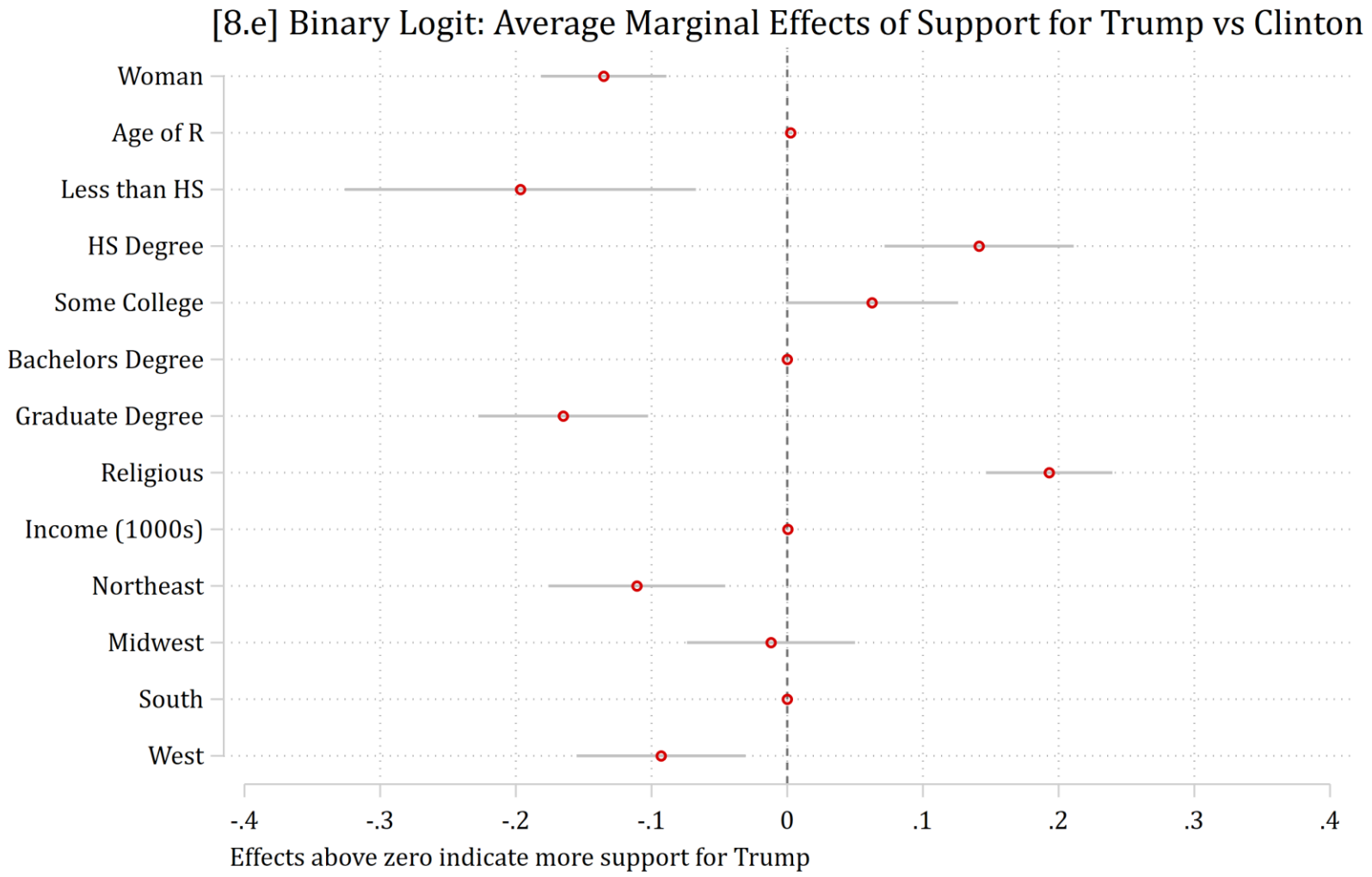
Coefficient plots: not just for coefficients

1. Increasingly, most health and social science fields are moving towards reporting marginal effects for categorical models instead of coefficients

2. A “coefficient plot” can be used to plot any effect

```
ames <- avg_slopes(logit_trump)
modelplot(ames) + geom_vline(xintercept = 0) + …
```

```
margins, dydx(*) post
coefplot, xline(0) baselevels …
```



Break 2: group task

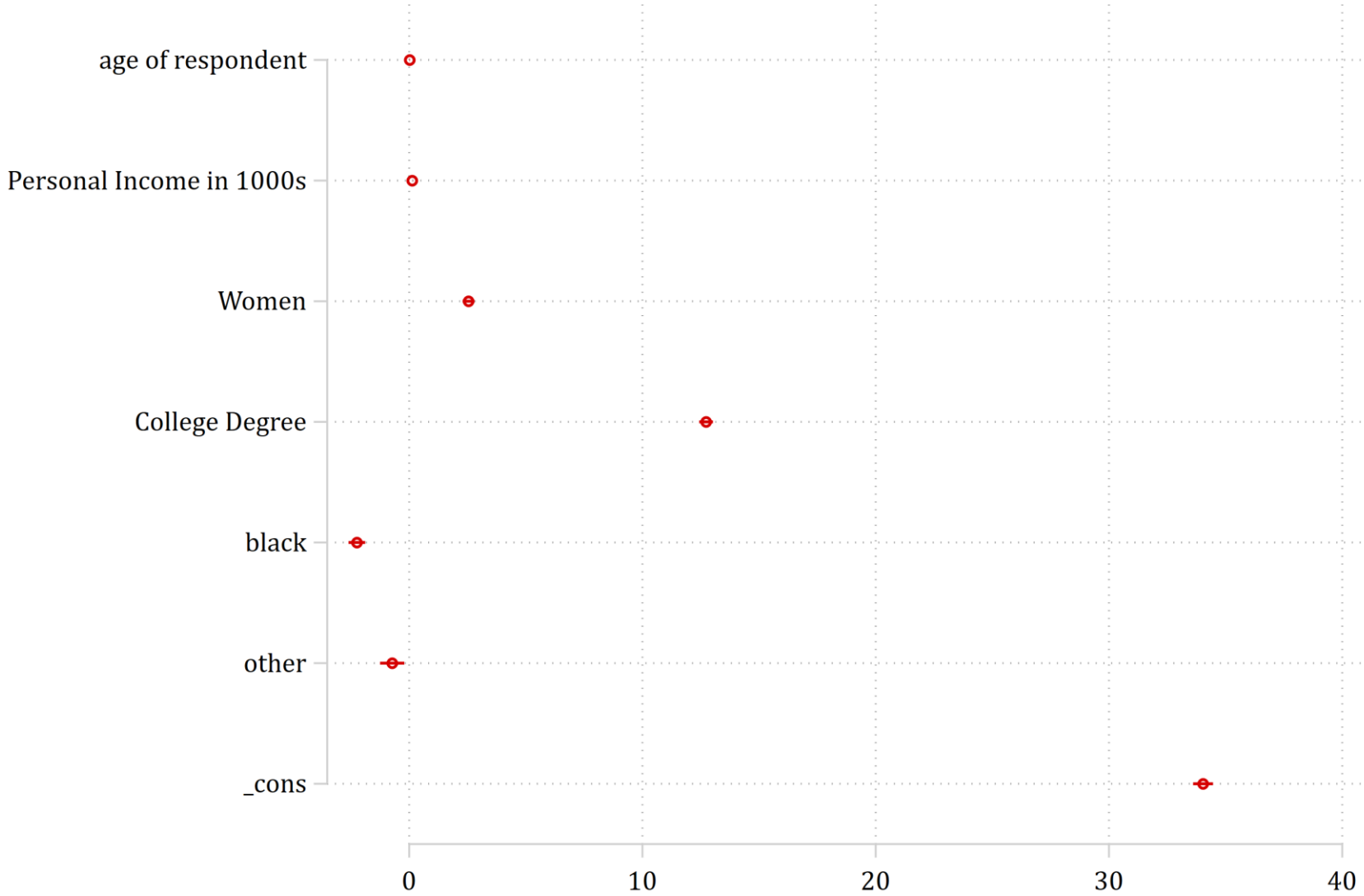
1. Run the code below in R or Stata: it loads the data and makes a quick graph

2. Discuss with your group:
    a. What does the graph show?
    b. What would you need to change to make it better, beyond the defaults?

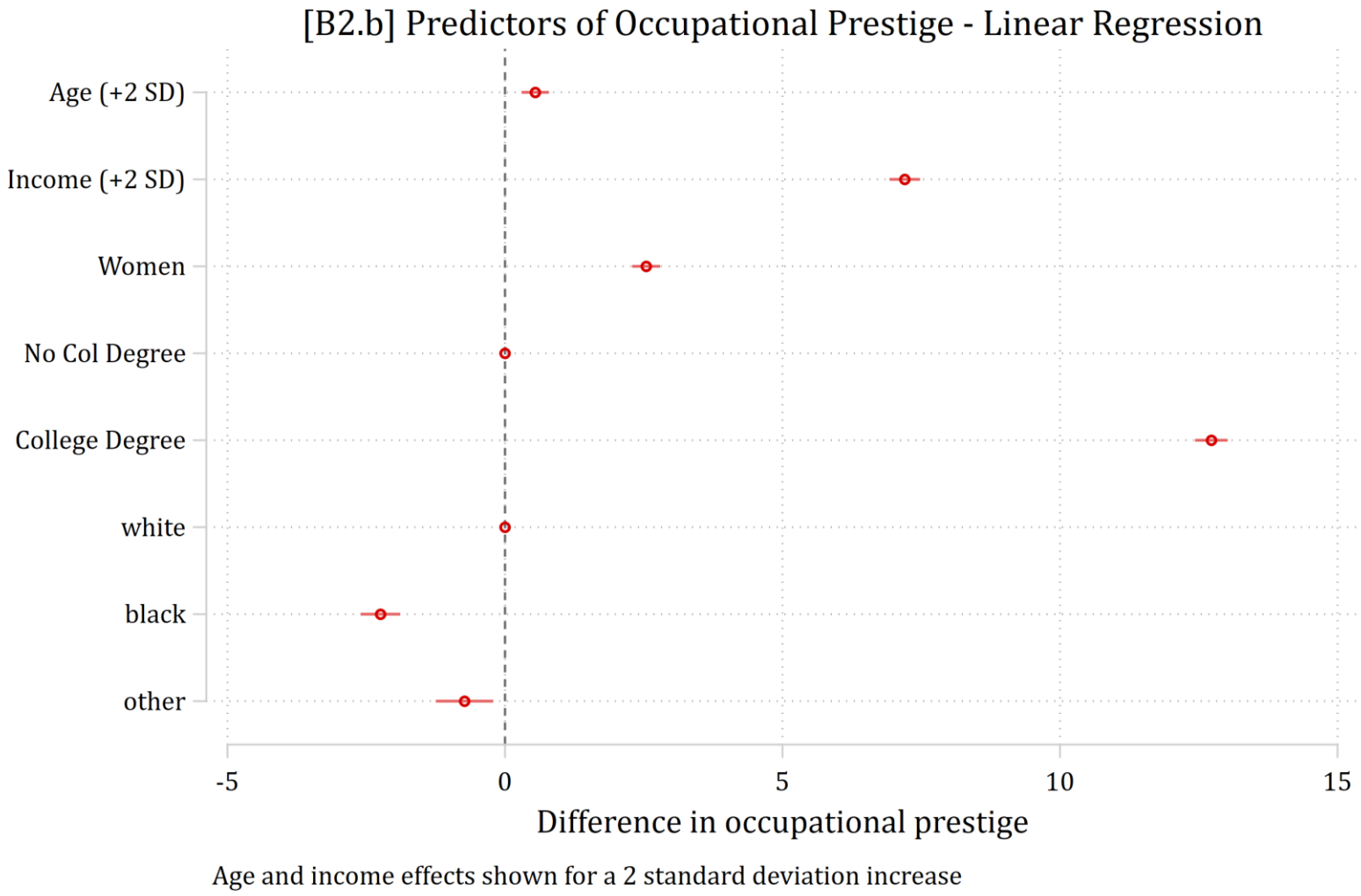
```
library(foreign); library(modelsummary)
gss <- droplevels(read.dta("https://tdmize.github.io/data/data/gss.dta"))
reg_op <- lm(occprest ~ age + income + woman + college + race, data = gss)
modelplot(reg_op)
```

```
use "https://tdmize.github.io/data/data/gss.dta", clear
regress occprest c.age c.income i.woman i.college i.race
coefplot
```

```
modelplot(reg_op)
coefplot
```



```
modelplot(reg_op2, coef_omit = "Intercept", coef_rename = c(…)) + …
  # age_2sd and income_2sd: age and income divided by 2 SDs
coefplot, xline(0) baselevels drop(_cons 0.woman) ///
    rescale(age = $age2sd income = $inc2sd) …
```

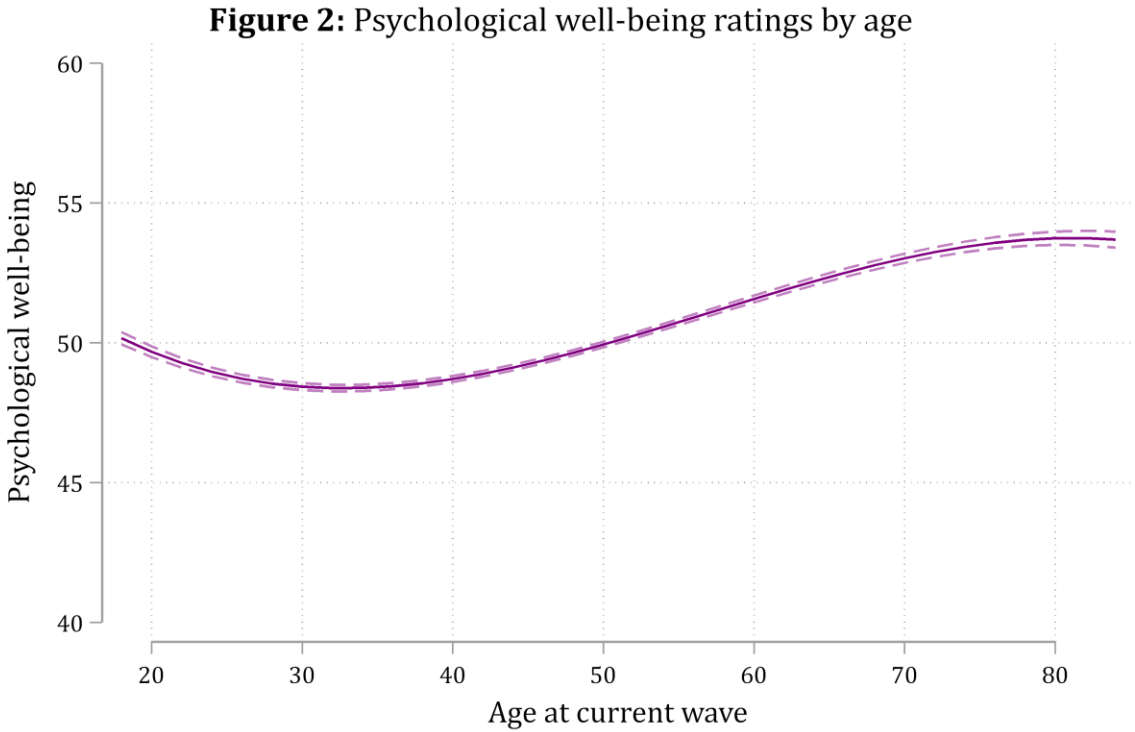


Principles of good data visualization

1. Look at your data
2. Use graph types people read accurately
3. Maximize the data-ink ratio, within reason (maybe)
4. Show uncertainty
5. Make groups easy to tell apart
6. Choose axes that fit the data
7. Design for your audience
8. Use graphs, not tables, for model results
9. **Show what the model predicts**

Plots of Model Predictions

1. Perhaps the most versatile and useful tool of the applied data analyst is the ability to plot model predictions
    a. The most effective way to understand a model is almost always to examine what the model predicts!
        i. Predictions are especially important when the model is nonlinear
            1. E.g., Because of a quadratic/polynomial term for an IV, because the DV is categorical, etc.

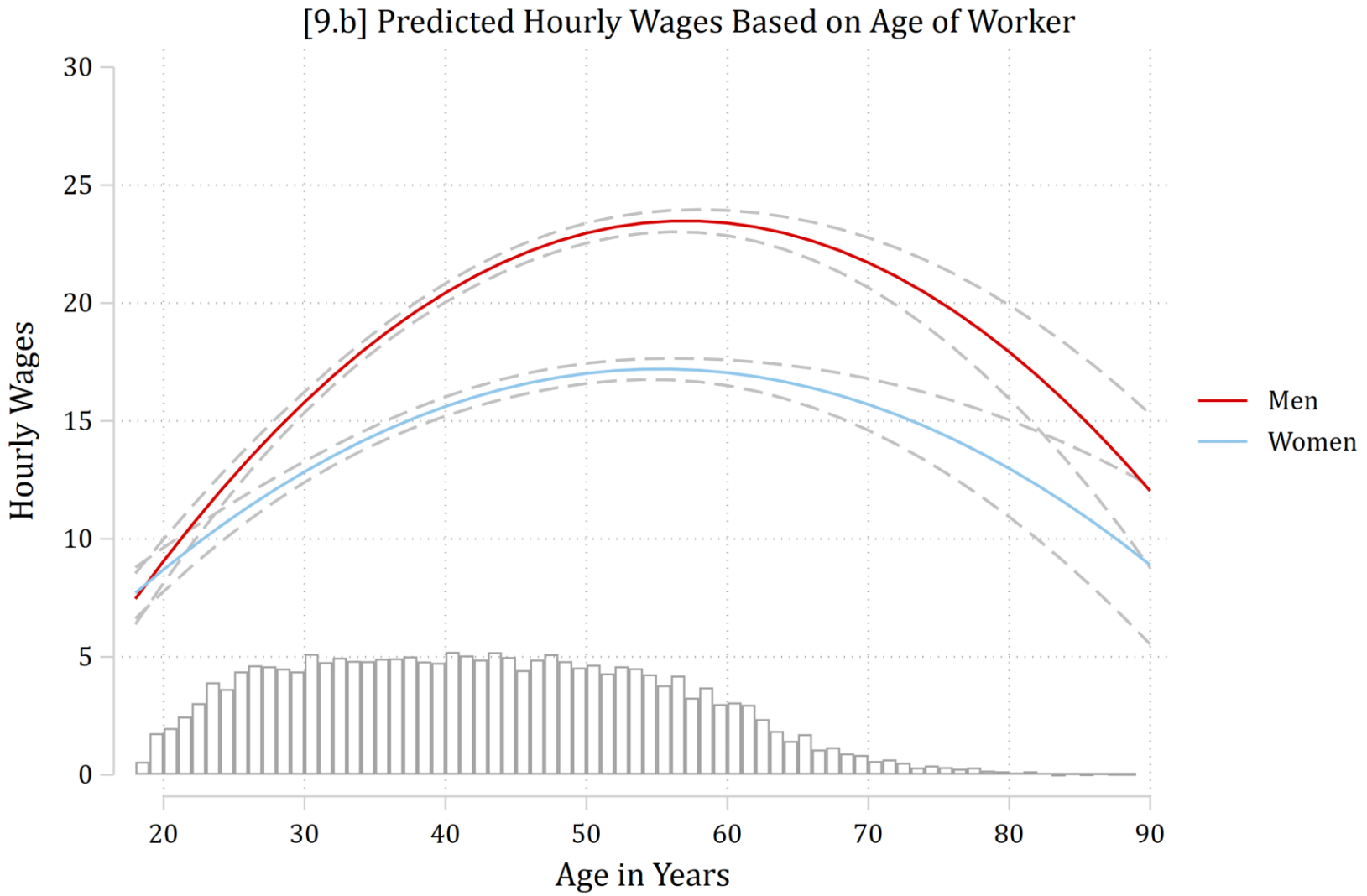


From Mize and Kincaid (2025)

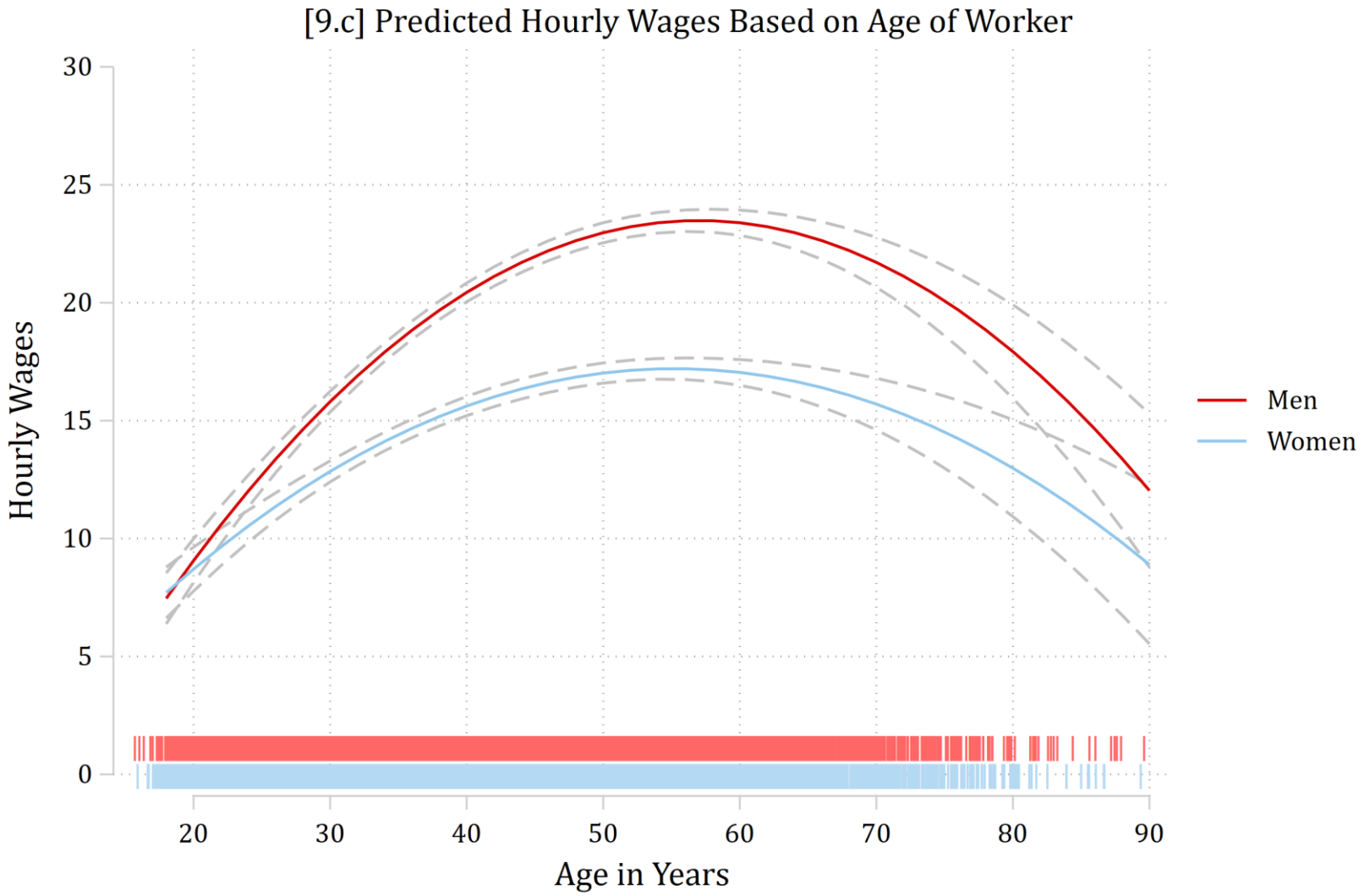
```
plot_predictions(wage_mod, condition = "age")
qui margins, at(age=(18(2)90))
marginsplot, recastci(rline) ciopts(lpat(dash)) …
```



```
plot_predictions(wage_int, condition = c("age", "woman")) + geom_histogram(…)
marginsplot, … addplot(hist age, yaxis(2) …)
```



```
plot_predictions(wage_int, condition = c("age", "woman")) + geom_rug(…, alpha = .06)
marginsplot, … addplot(scatter where agerand, mlabel(pipe) mlabcol(red*.6) …)
```



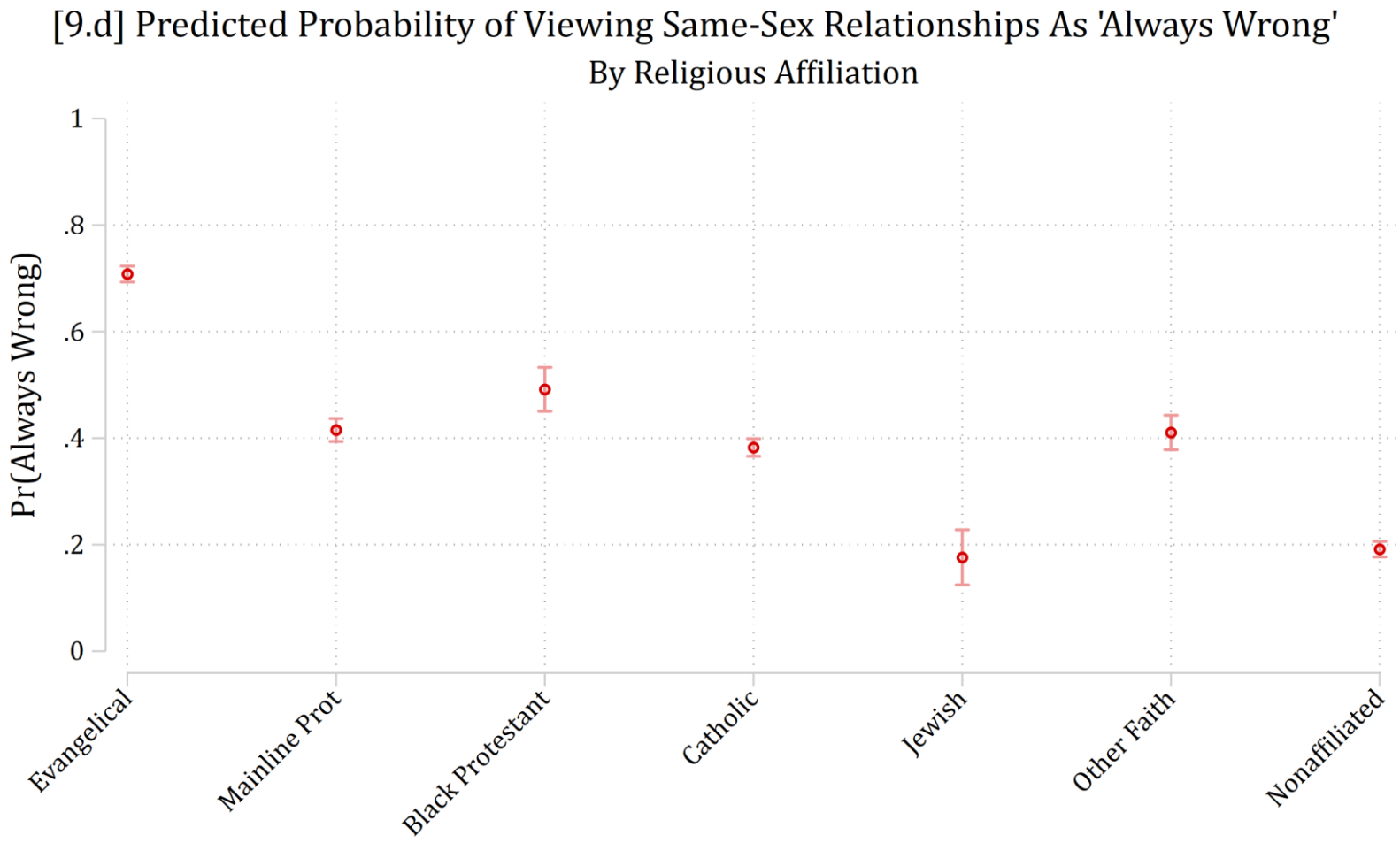
Plotting predictions for nominal vs continuous variables

1. For IVs treated as continuous: sort the *x*-axis left to right by values of the IV, make predictions at most or all levels of the IV, and connect the lines between the predictions
    a. This helps visually mirror the assumption of continuity in how you modeled the IV

2. For IVs treated as nominal: *x*-axis sorting is often arbitrary*, dots or bars should clearly represent distinct predictions being made for each grouping, and **there should be no lines connecting** the predictions
    a. Lines connecting predictions suggest continuity, whereas nominal IVs have no ordering information
    b. Be consistent in ordering across figures, tables, etc.

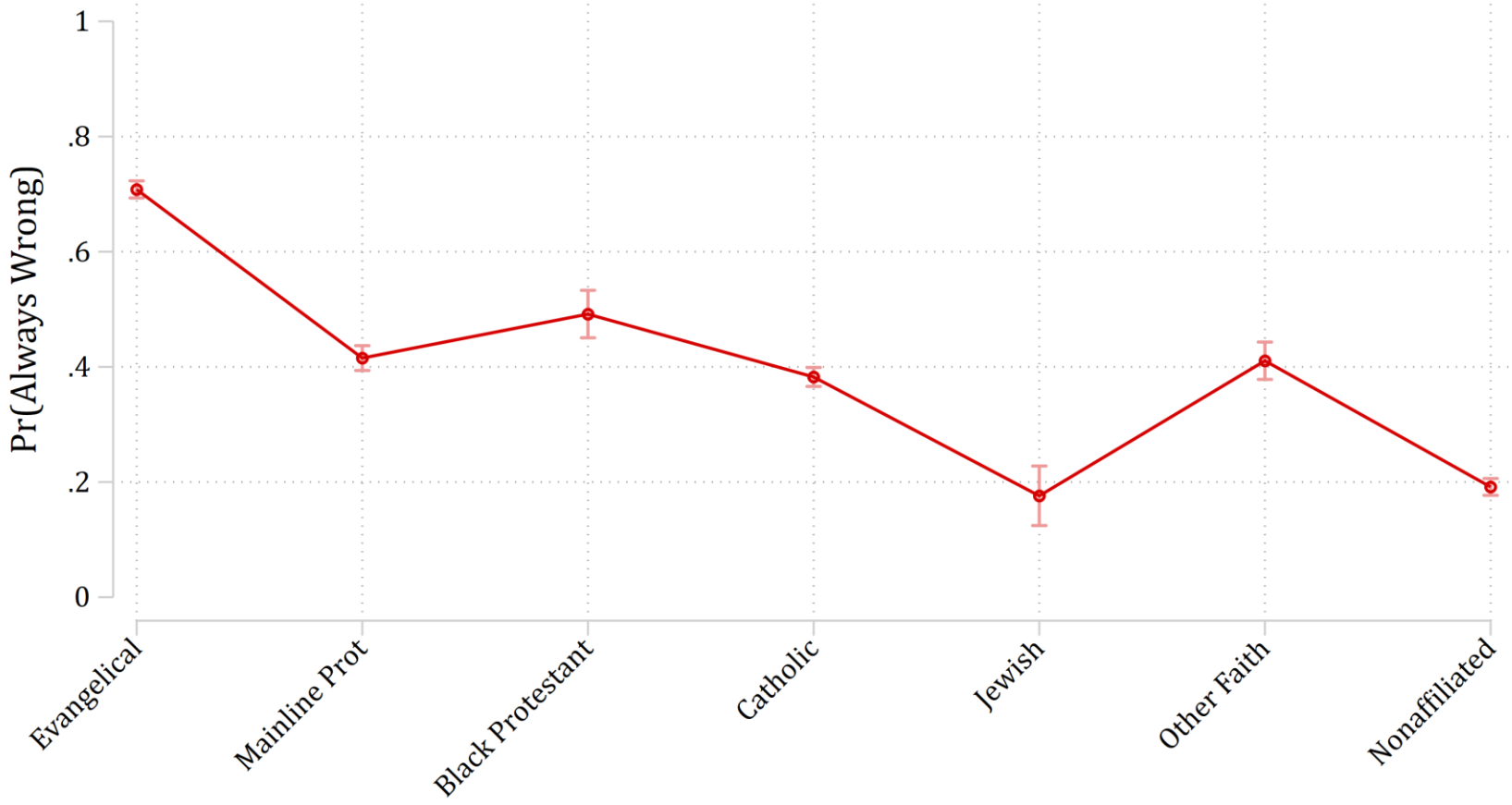
3. For ordinal variables modeled as nominal, use the logical order in tables and graphs (e.g., “high school degree” left of “college degree”)

```
plot_predictions(logit_rel, condition = "reltrad")
margins reltrad
marginsplot, recast(scatter) …
```

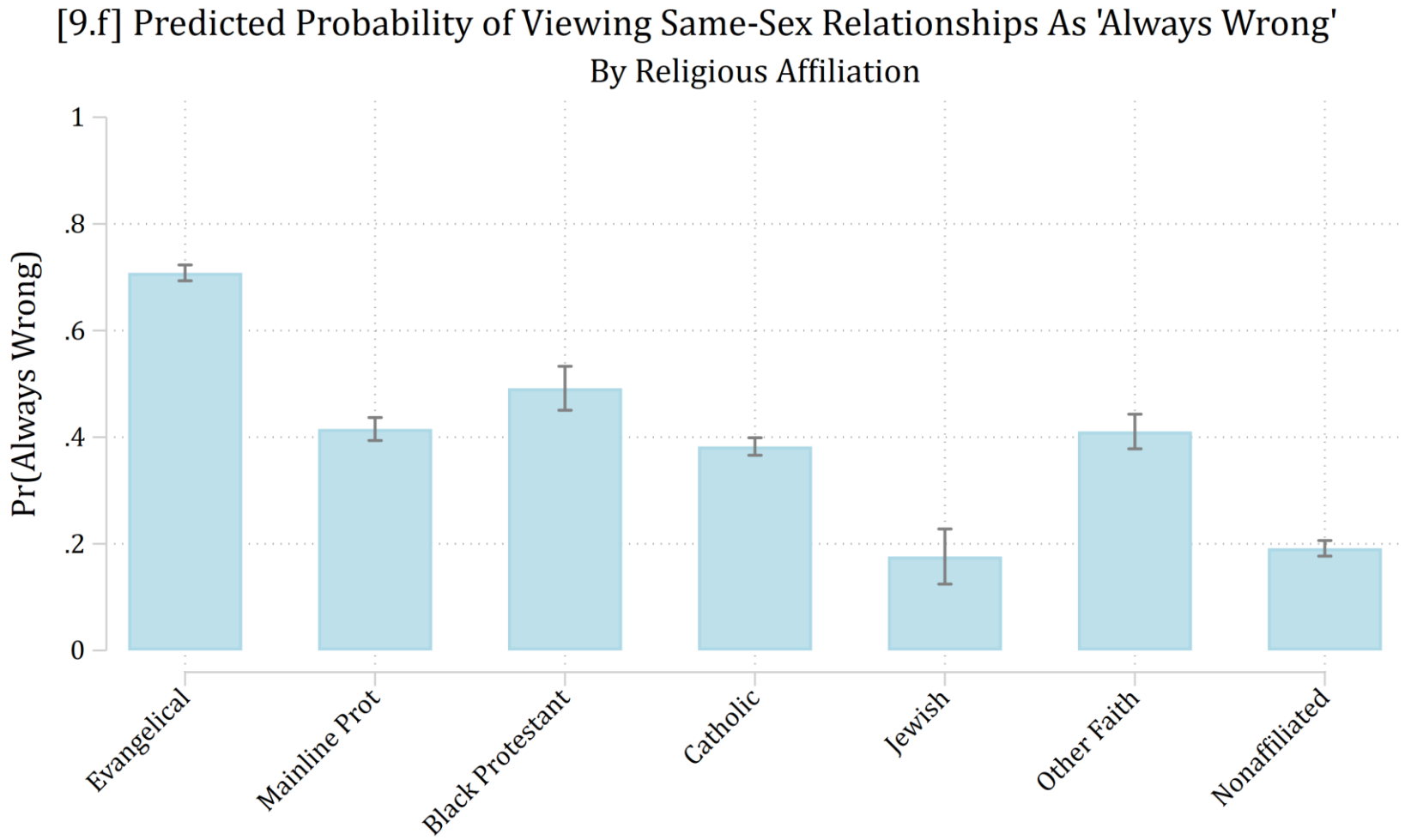


[9.e] **Don't Do This:** Do Not Connect Lines of Predictions for Nominal IVs

Connected lines suggest a continuous IV which this is not



```
ggplot(pred_rel, aes(x = reltrad, y = estimate)) + geom_col( ) + …
marginsplot, recast(bar) …
```



Confidence intervals and statistical significance

1. "Fun" game: guess which of the following group differences (b_0 vs. b_1) are significant at the 95% level (two-tailed test)

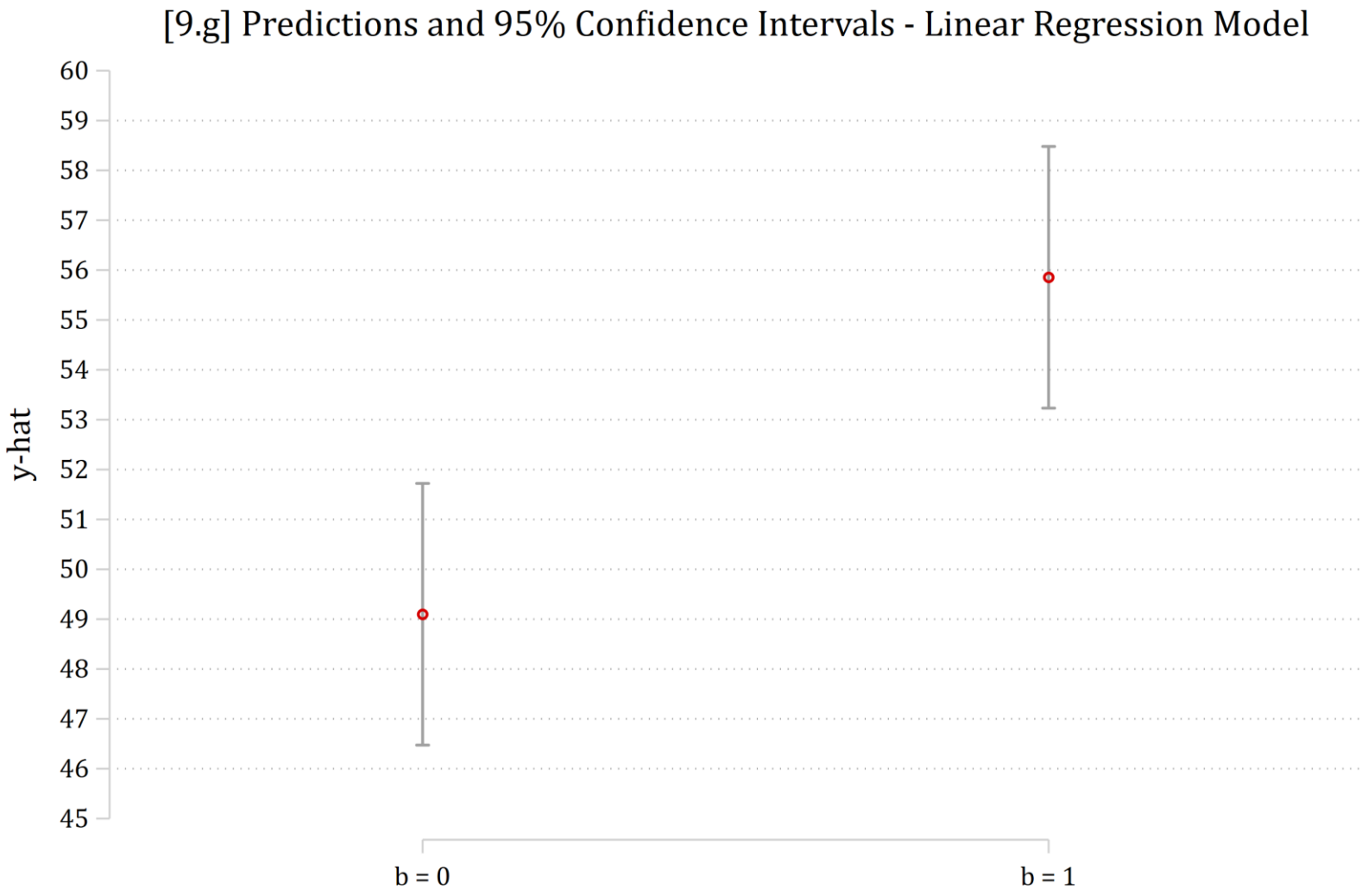
2. The dependent variable is continuous
    a. The independent variable (b) is binary with two values (0 and 1)

3. The test for significance can be*:
    a. A *t*-test
    b. An ANOVA
    c. A linear regression model
    d. A Wald test for equality of predictions

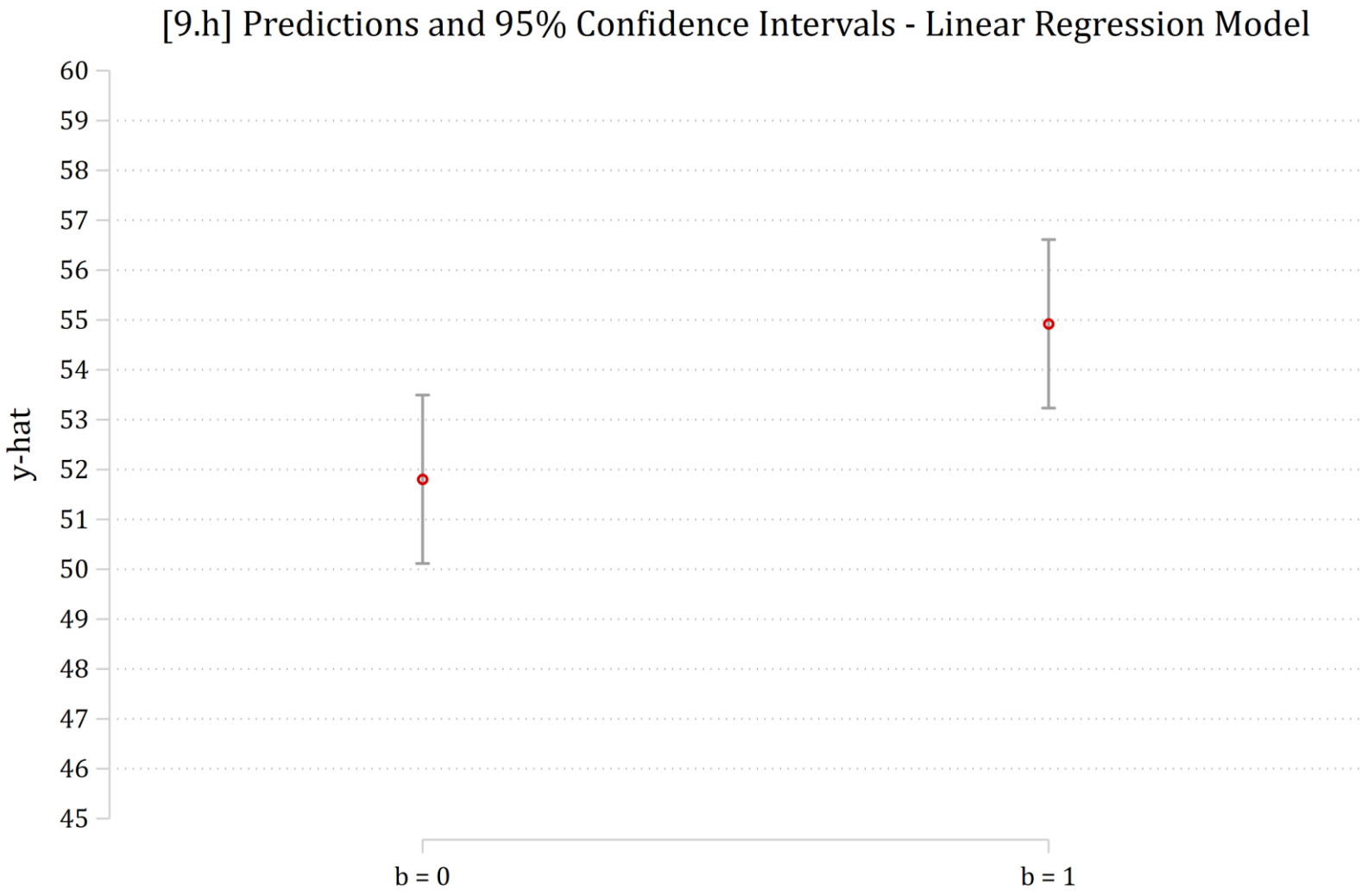


*Fun fact: these are all identical tests in this simple case…

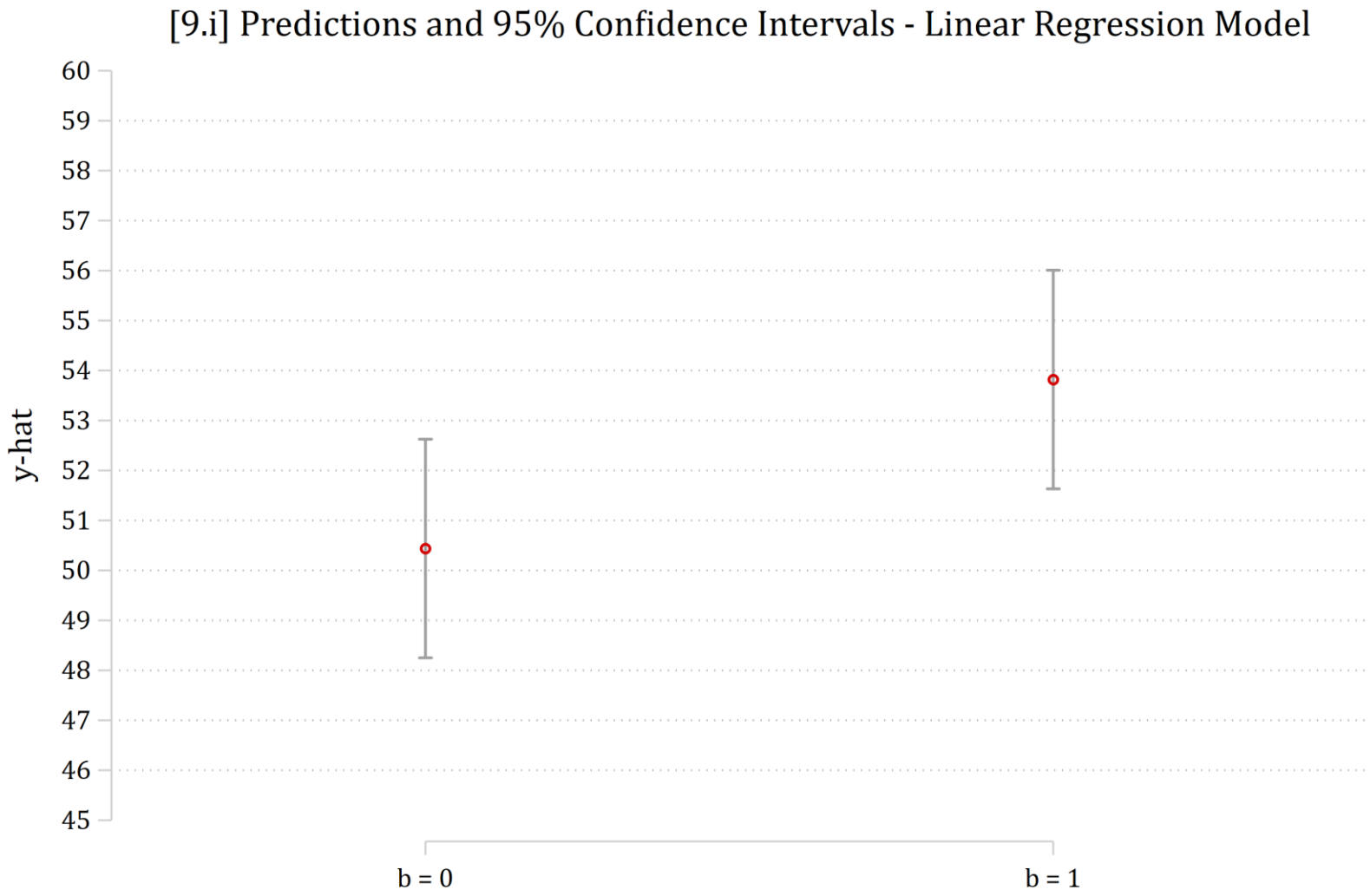
```
avg_predictions(lm(y30 ~ b, data = sim), variables = "b")
est restore pr_y30
coefplot, vertical ciop(recast(rcap)) …
```



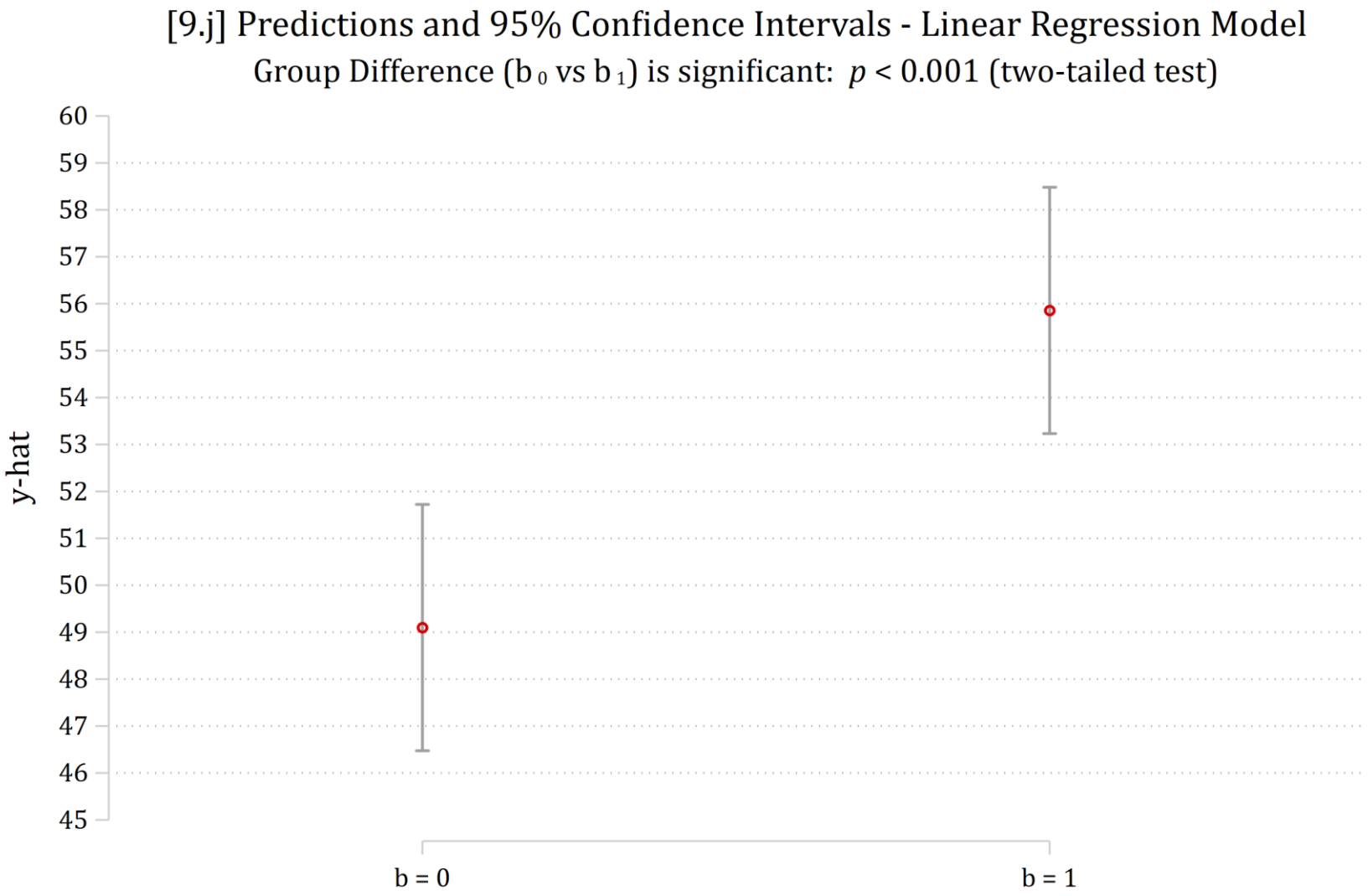
```
avg_predictions(lm(y20 ~ b, data = sim), variables = "b")
est restore pr_y20
coefplot, vertical ciop(recast(rcap)) …
```



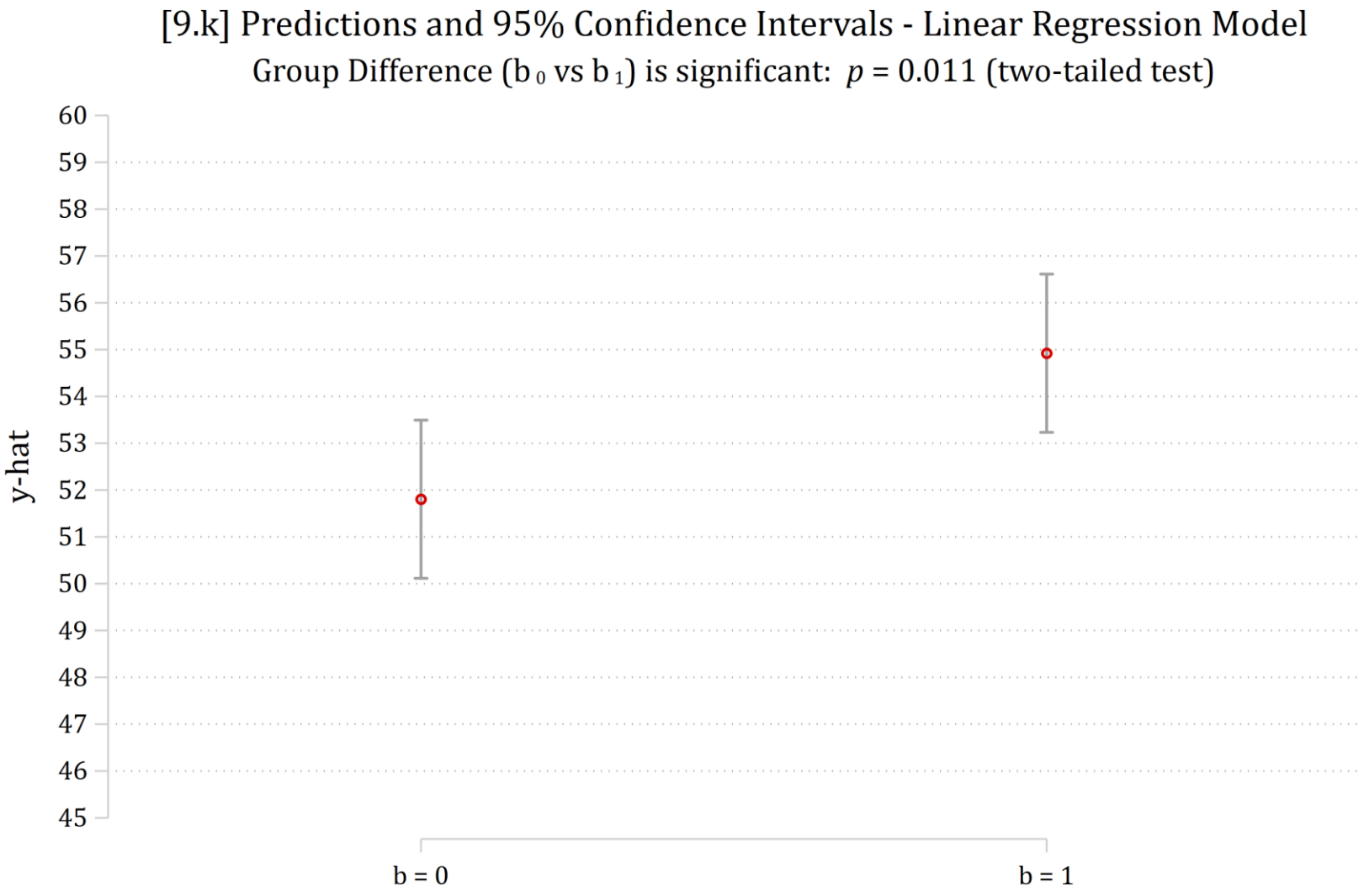
```
avg_predictions(lm(y25 ~ b, data = sim), variables = "b")
est restore pr_y25
coefplot, vertical ciop(recast(rcap)) …
```



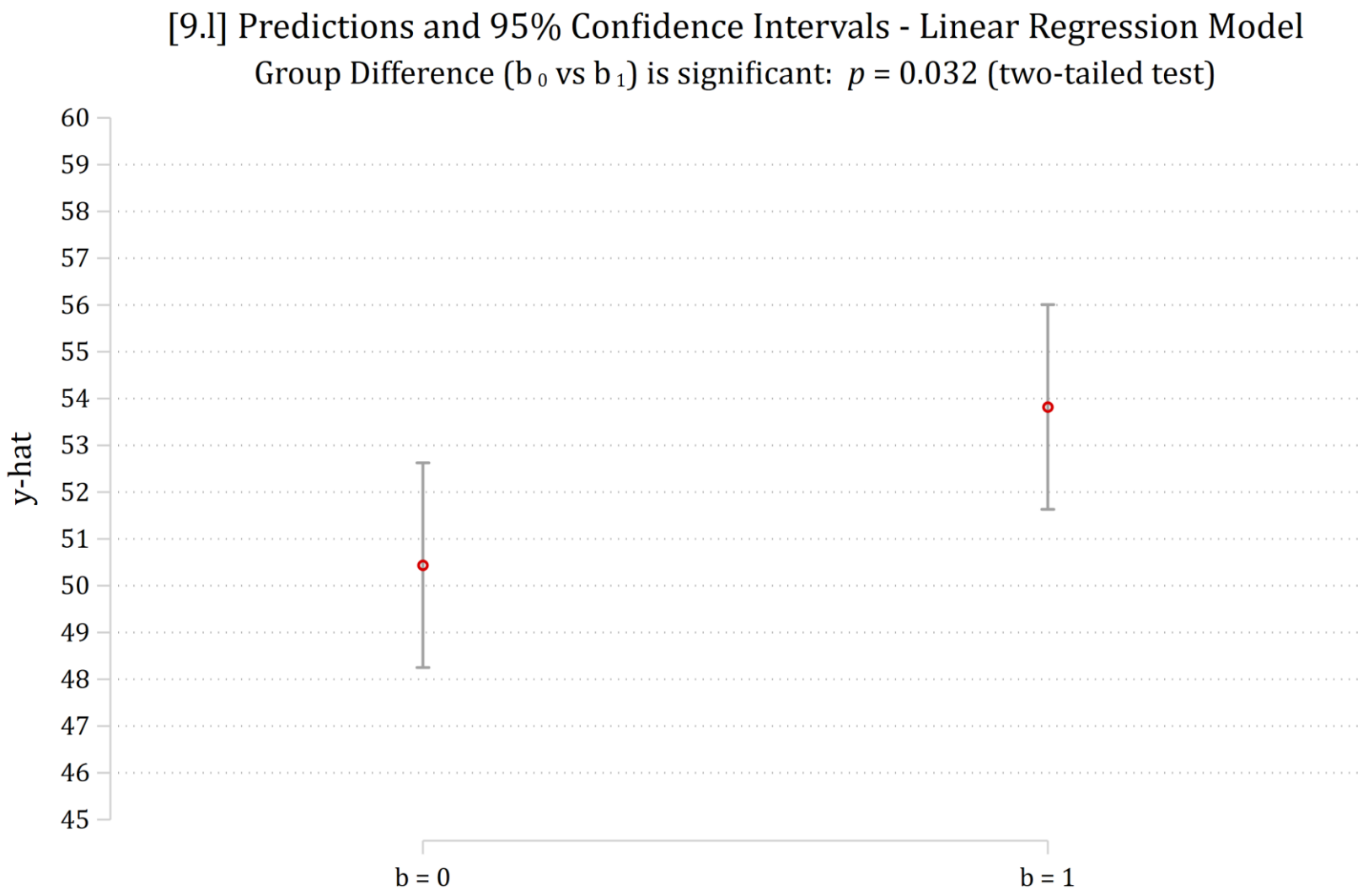
```
avg_comparisons(lm(y30 ~ b, data = sim), variables = "b")
lincom 2._at - 1._at
```



```
avg_comparisons(lm(y20 ~ b, data = sim), variables = "b")
lincom 2._at - 1._at
```



```
avg_comparisons(lm(y25 ~ b, data = sim), variables = "b")
lincom 2._at - 1._at
```



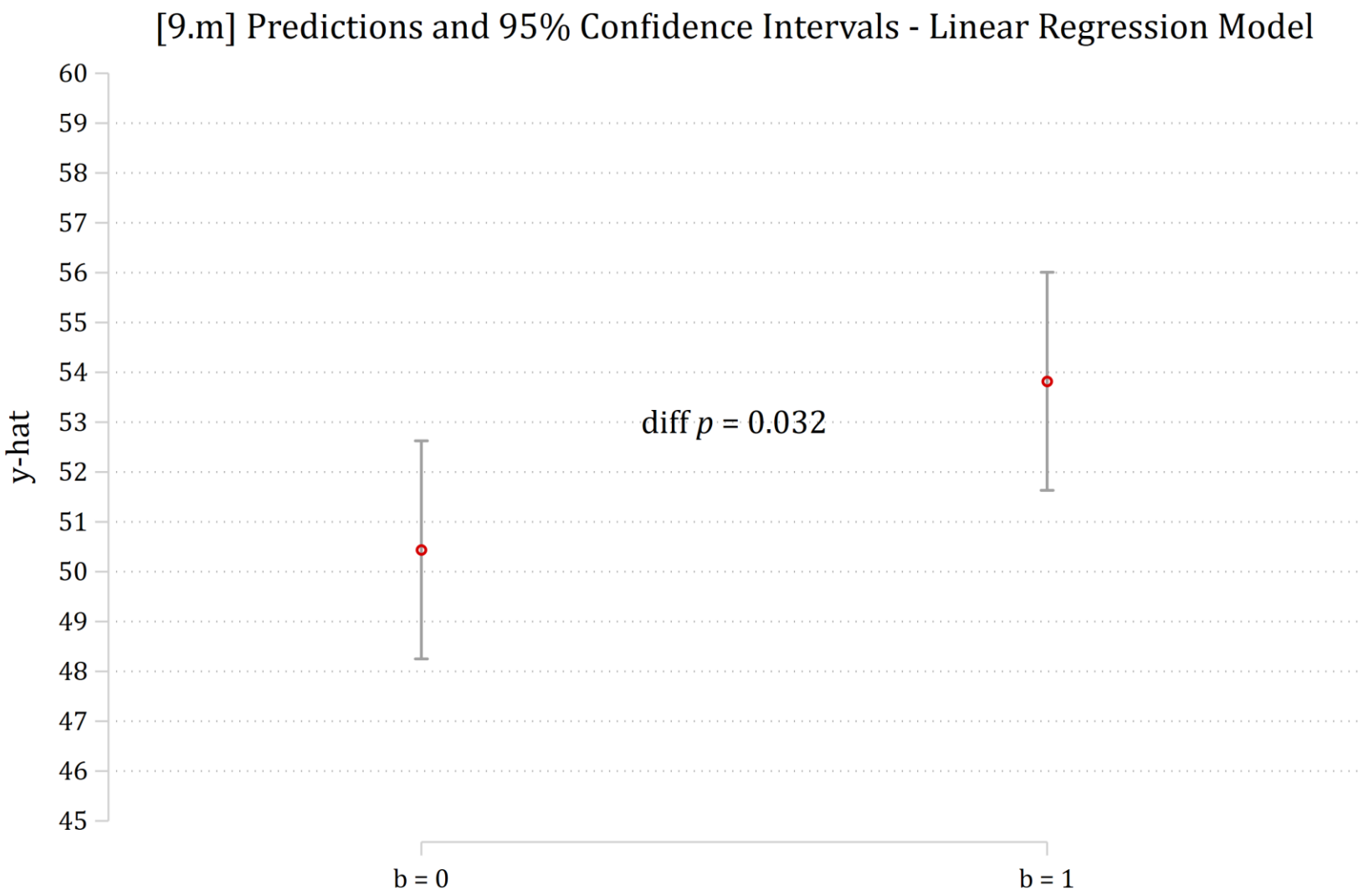
Advice on confidence intervals and standard errors

1. Most people—including PhD researchers—do not understand what overlapping confidence intervals indicate about statistical significance
    a. You need to recognize this when making graphics to present your findings

2. Importantly, when confidence intervals do not overlap (e.g., on the 1st and 2nd plots above) this indicates that the group difference (i.e., men vs women) is significant
    a. However, when the confidence interval does overlap this *may or may not* indicate the group difference is non-significant*: you must test for this directly. As Belia et al. (2005) gently put it:

> Little is known about researchers' understanding of confidence intervals (CIs) and standard error (*SE*) bars. Authors of journal articles in psychology, behavioral neuroscience, and medicine were invited to visit a Web site where they adjusted a figure until they judged 2 means, with error bars, to be just statistically significantly different ($p < .05$). Results from 473 respondents suggest that many leading researchers have severe misconceptions about how error bars relate to statistical significance, do not adequately distinguish CIs and *SE* bars, and do not appreciate the importance of whether the 2 means are independent or come from a repeated measures design. Better guidelines for researchers and less ambiguous graphical con-

```
ci_plot("y25", "9.m") + annotate("text", x = 1.5, y = 53, label = "diff p = 0.032")
coefplot, … text(53 1.5 "diff {it:p} = 0.032")
```



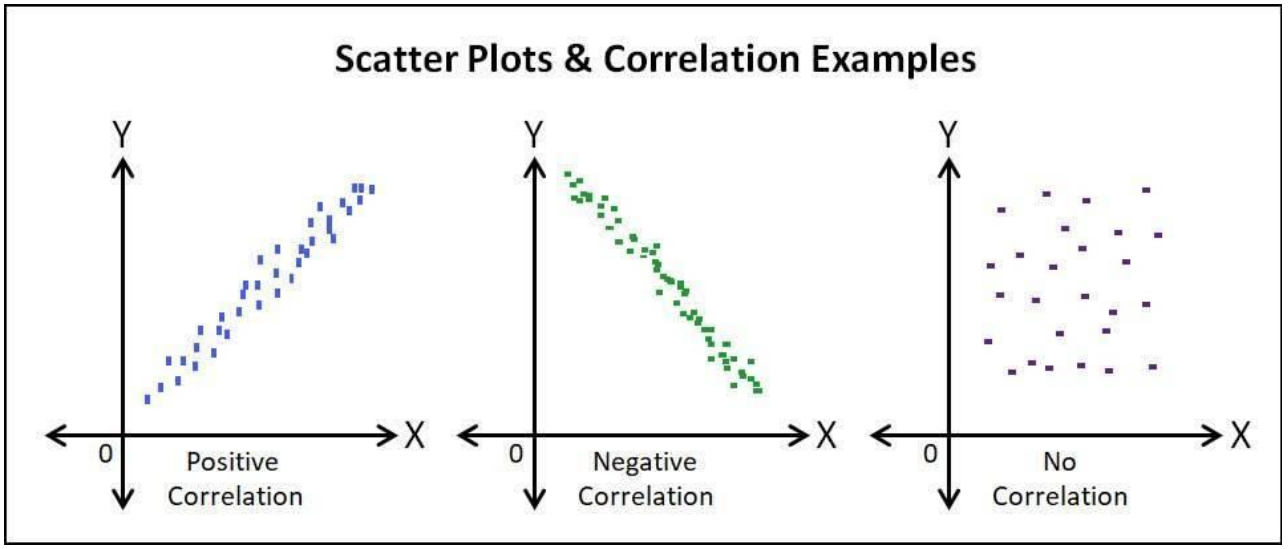
Principles of good data visualization

1. Look at your data
2. Use graph types people read accurately
3. Maximize the data-ink ratio, within reason (maybe)
4. Show uncertainty
5. Make groups easy to tell apart
6. Choose axes that fit the data
7. Design for your audience
8. Use graphs, not tables, for model results
9. Show what the model predicts
10. **Don’t assume linearity**

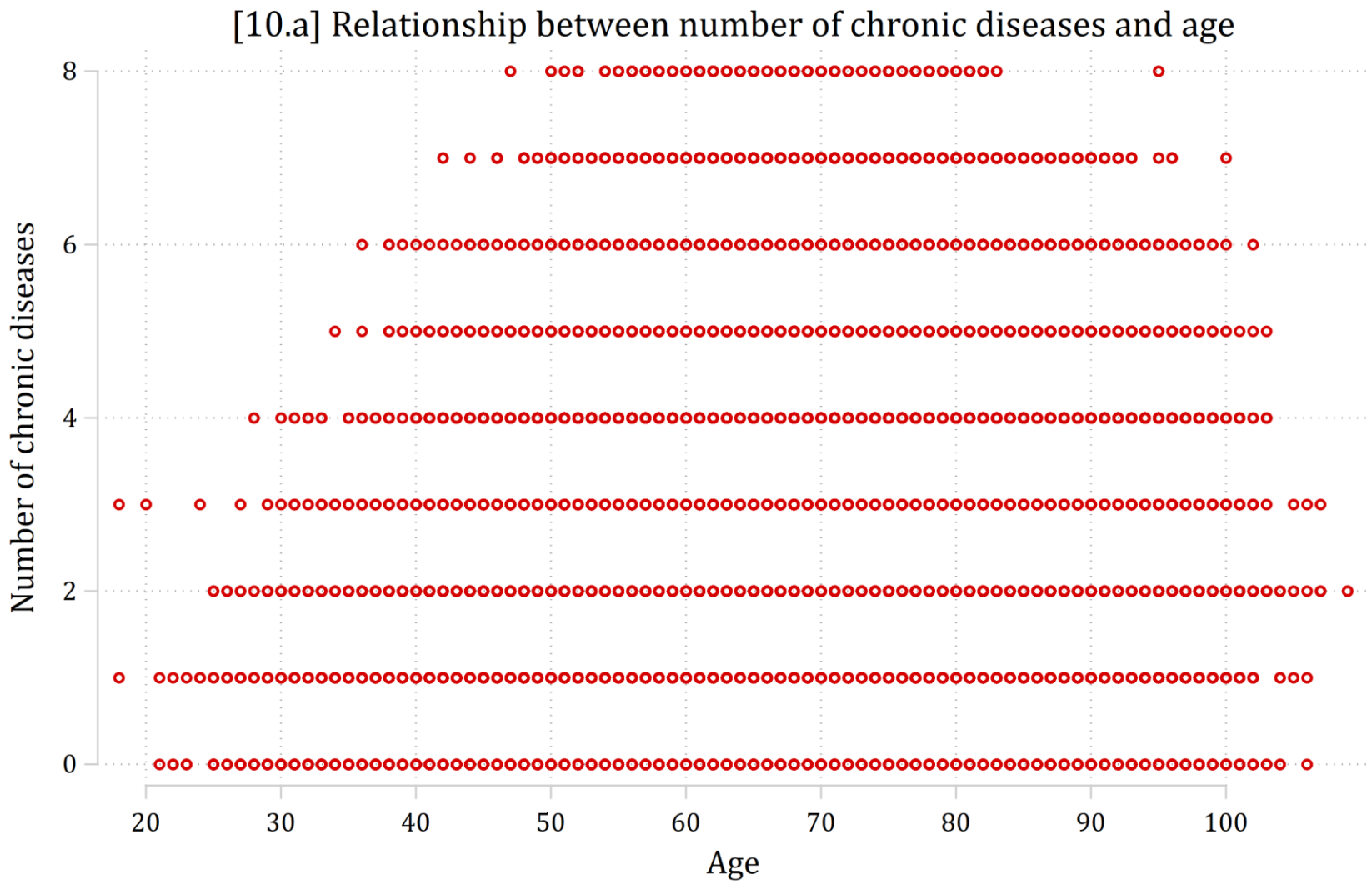
Scatterplots of two continuous variables

1. Scatterplots are among the most common and intuitive data visualization techniques

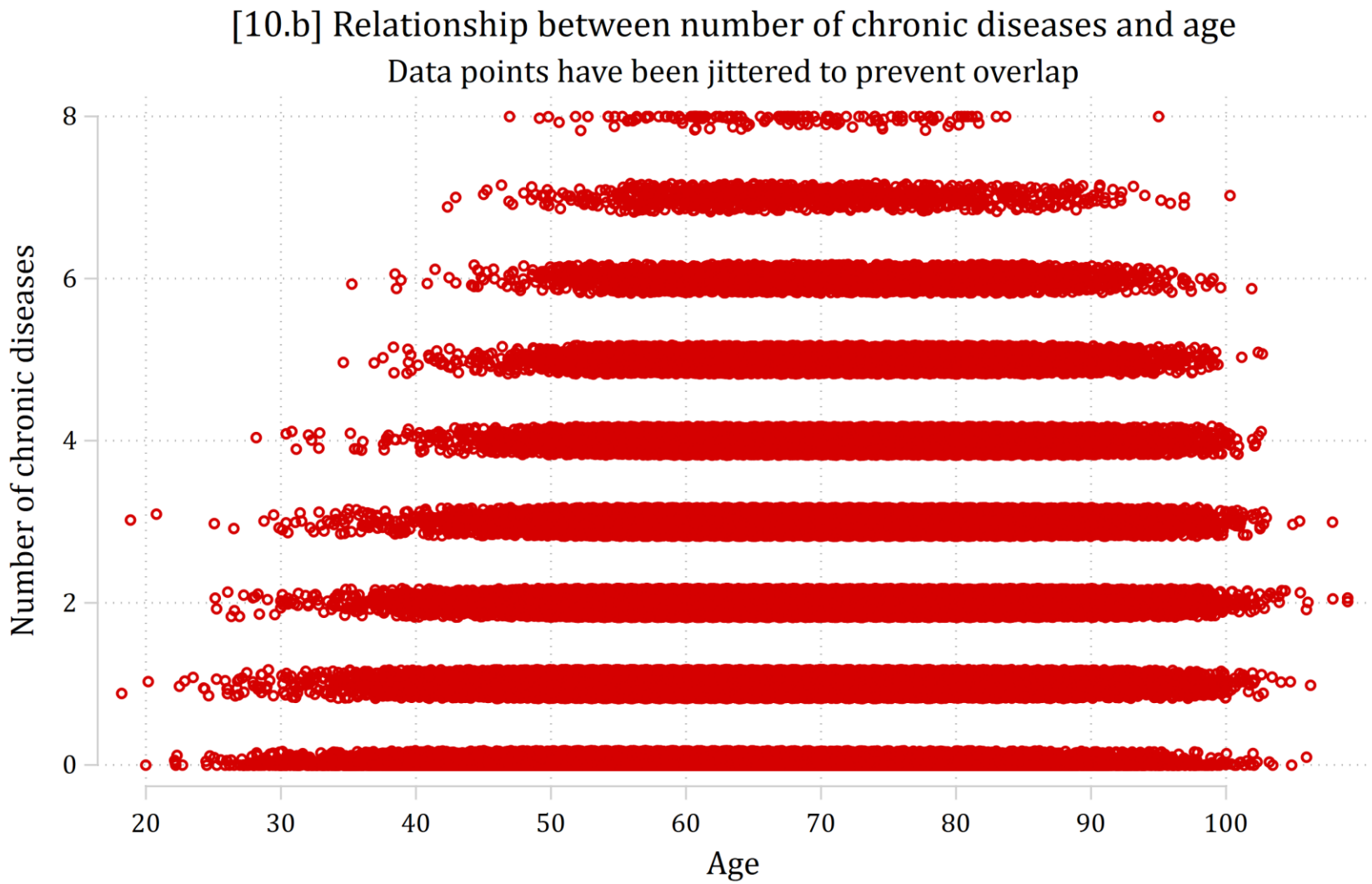
2. Plot each individual observation by placing one variable on the *x*-axis and another on the *y*-axis, and placing a dot where each observation falls
   a. The dependent variable (DV) or outcome is the *y*-variable
   b. The dependent variable (IV) or cause is the *x*-variable



```
ggplot(hrs, aes(x = age, y = dzcount)) + geom_point( )
twoway scatter dzcount age …
```



```
ggplot(hrs, aes(x = age, y = dzcount)) + geom_jitter(height = .25, width = .25)
twoway scatter dzcount age, jitter(3) …
```



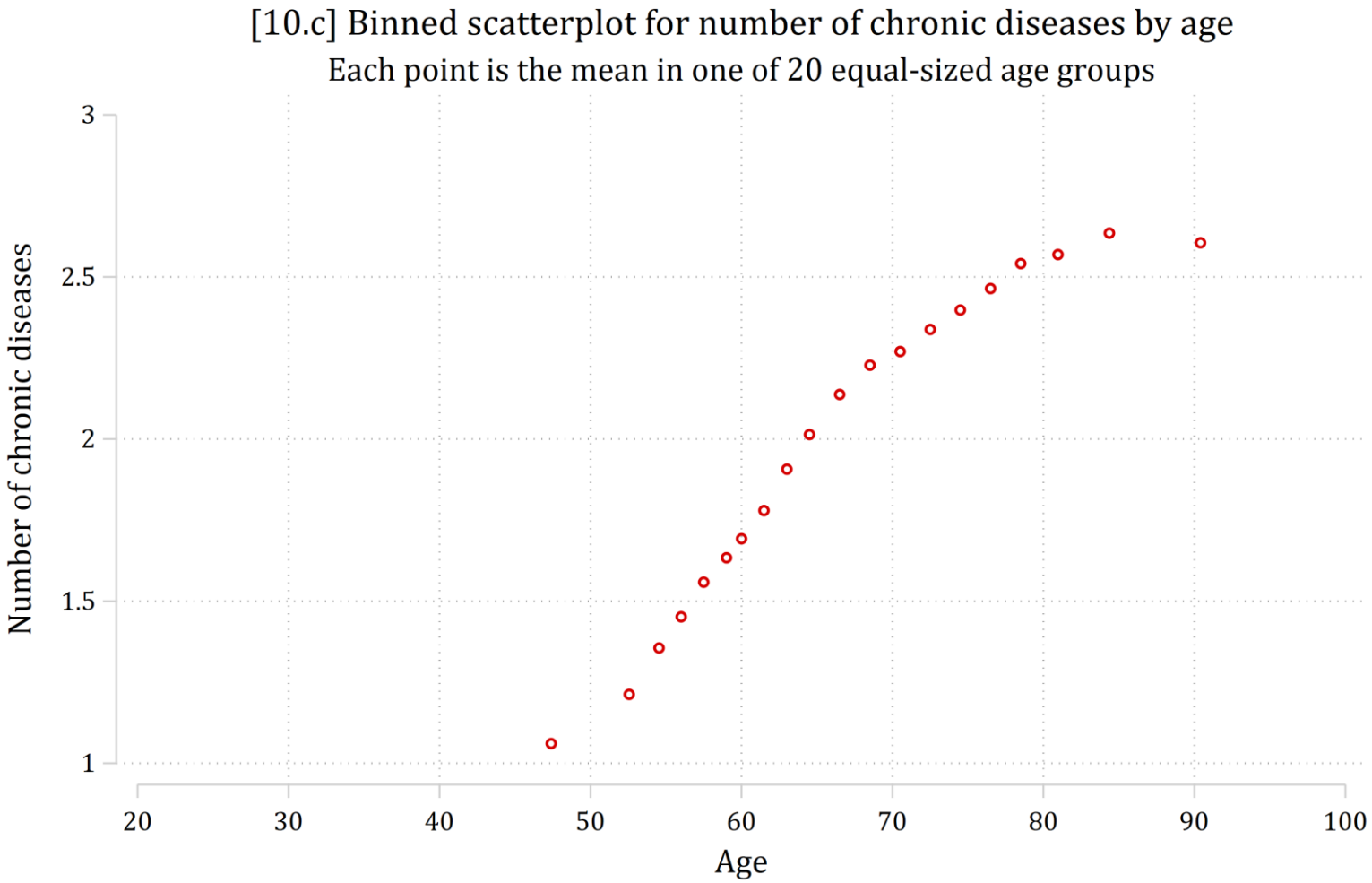
Binned scatterplots

1. Scatterplots and scatterplot smoothing methods like lowess are less useful when you have lots of data
    a. Graphs become impossible to read (but could plot random subsample)

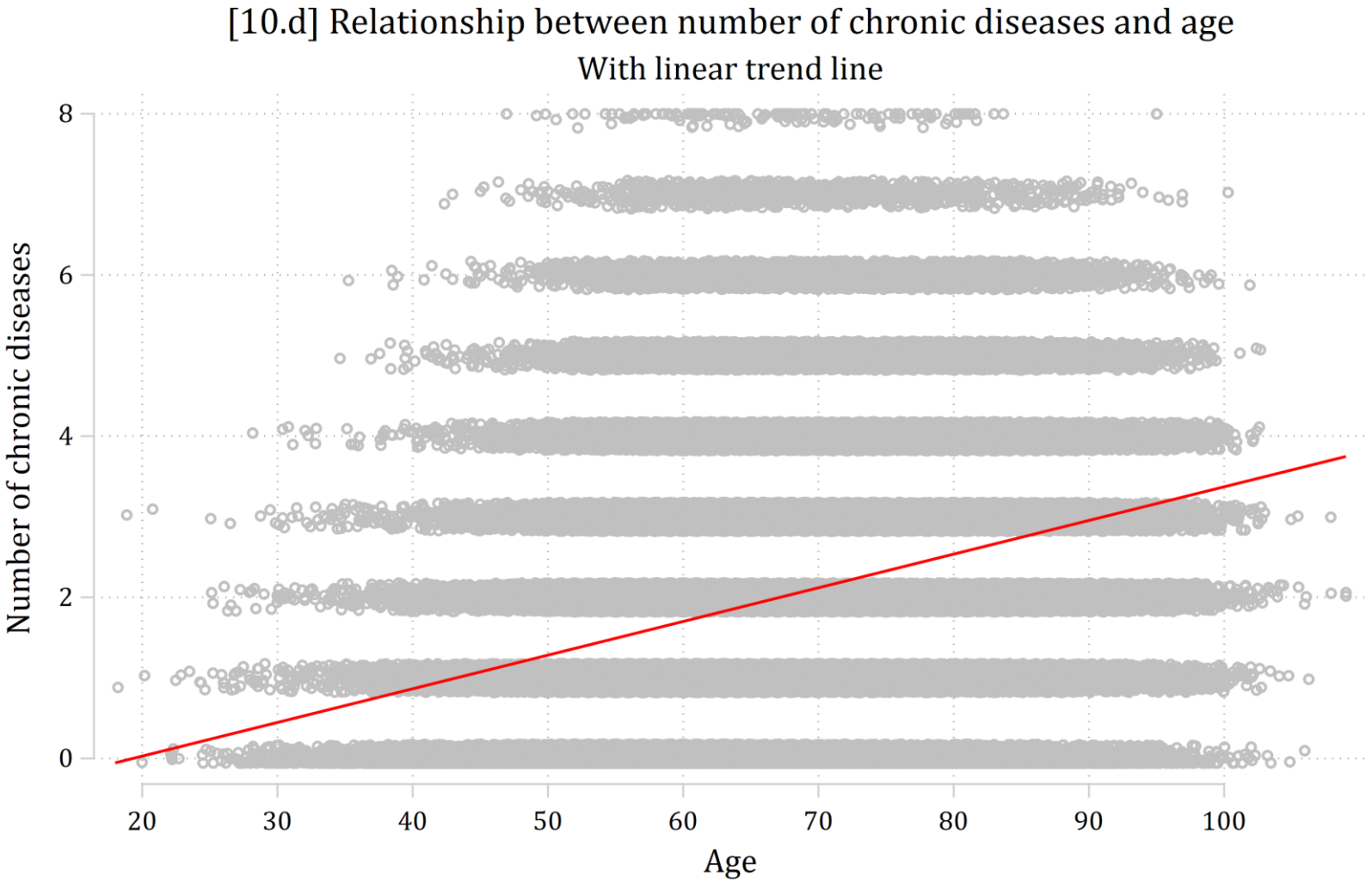
2. A binned scatterplot is a useful tool for these and other cases (Stepner 2014)
    a. "bin" the *x* variable into groups (e.g., 20 age groups) and then calculate the mean of the *y* variable for each bin

```
hrs_bins <- hrs %>% mutate(age_bin = ntile(age, 20)) %>% …
ggplot(hrs_bins, aes(x = age, y = dzcount)) + geom_point( )
```

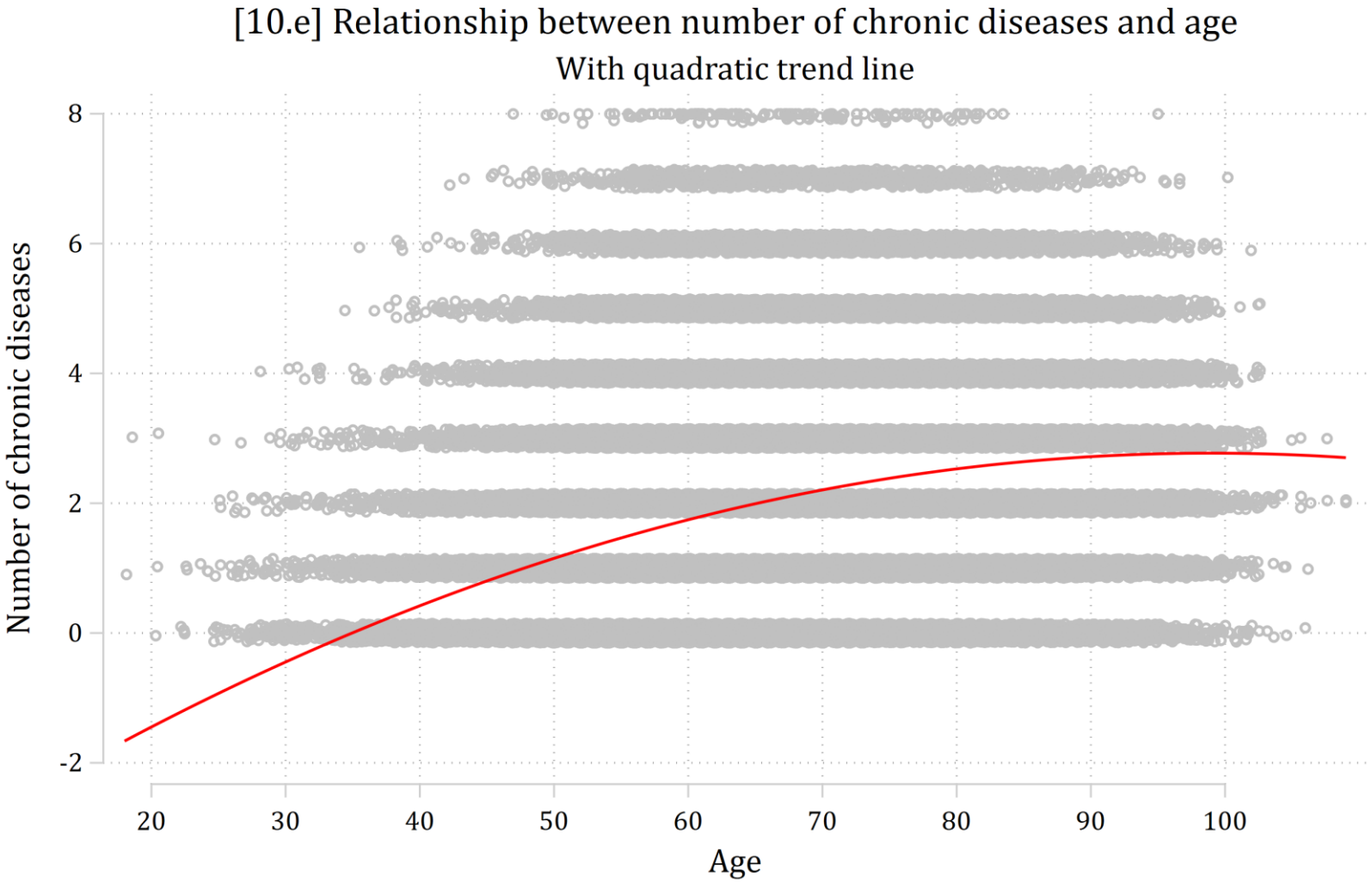
```
binscatter dzcount age, linetype(none) …
```



```
… + geom_jitter( ) + geom_smooth(method = "lm")
twoway (scatter dzcount age, jitter(3)) (lfit dzcount age) …
```



```
… + geom_jitter( ) + geom_smooth(method = "lm", formula = y ~ poly(x, 2))
twoway (scatter dzcount age, jitter(2)) (qfit dzcount age) …
```



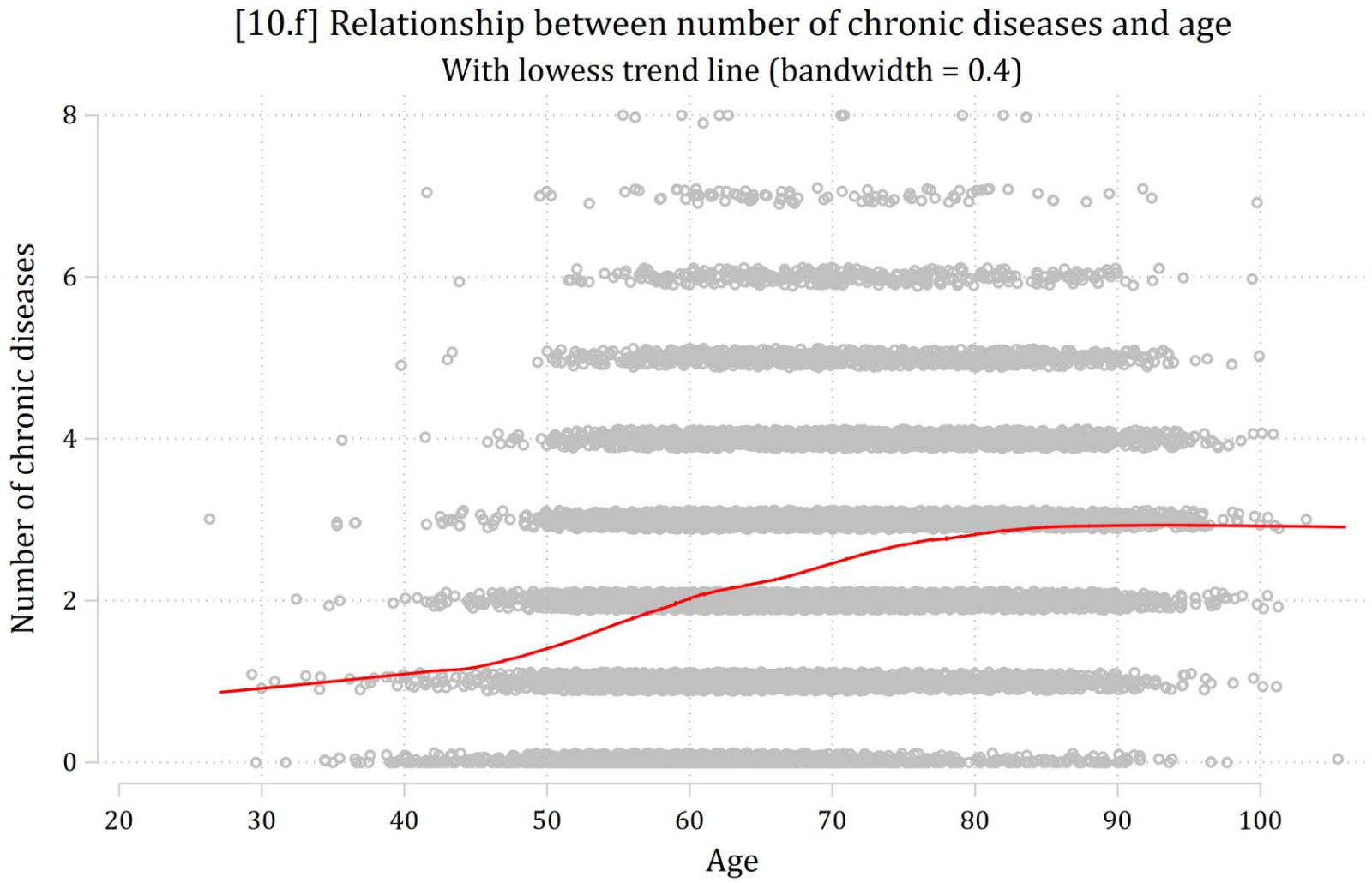
LOWESS: locally weighted scatterplot smoothing

1. The fitted lines shown above are parametric techniques

2. LOWESS performs "local" regressions on a subset of your data
    a. Primarily an exploratory technique—you can plot the smoothed regression line but you do not get parameter estimates/significance tests
    b. It can be a helpful way to quickly assess nonlinearity and often suggests possible parametric modeling solutions for the nonlinearity (e.g., need both *age* and age^2 in model)

3. Can specify a bandwidth in `span( )` option to determine how much to smooth (0 to 1 as a proportion of your data)
    a. E.g., A bandwidth of 0.4 means you want the 40% of the data closest to a given point used to calculate the line at that point
    a. Smaller bandwidths will let you find more nonlinearities but are more susceptible to outliers and other quirks of your data
        i. I find 0.3 or 0.4 good starting places

```
… + geom_jitter( ) + geom_smooth(method = "loess", span = .4)
twoway (scatter dzcount age, jitter(2)) (lowess dzcount age, bwidth(.4)) …
```



Published example

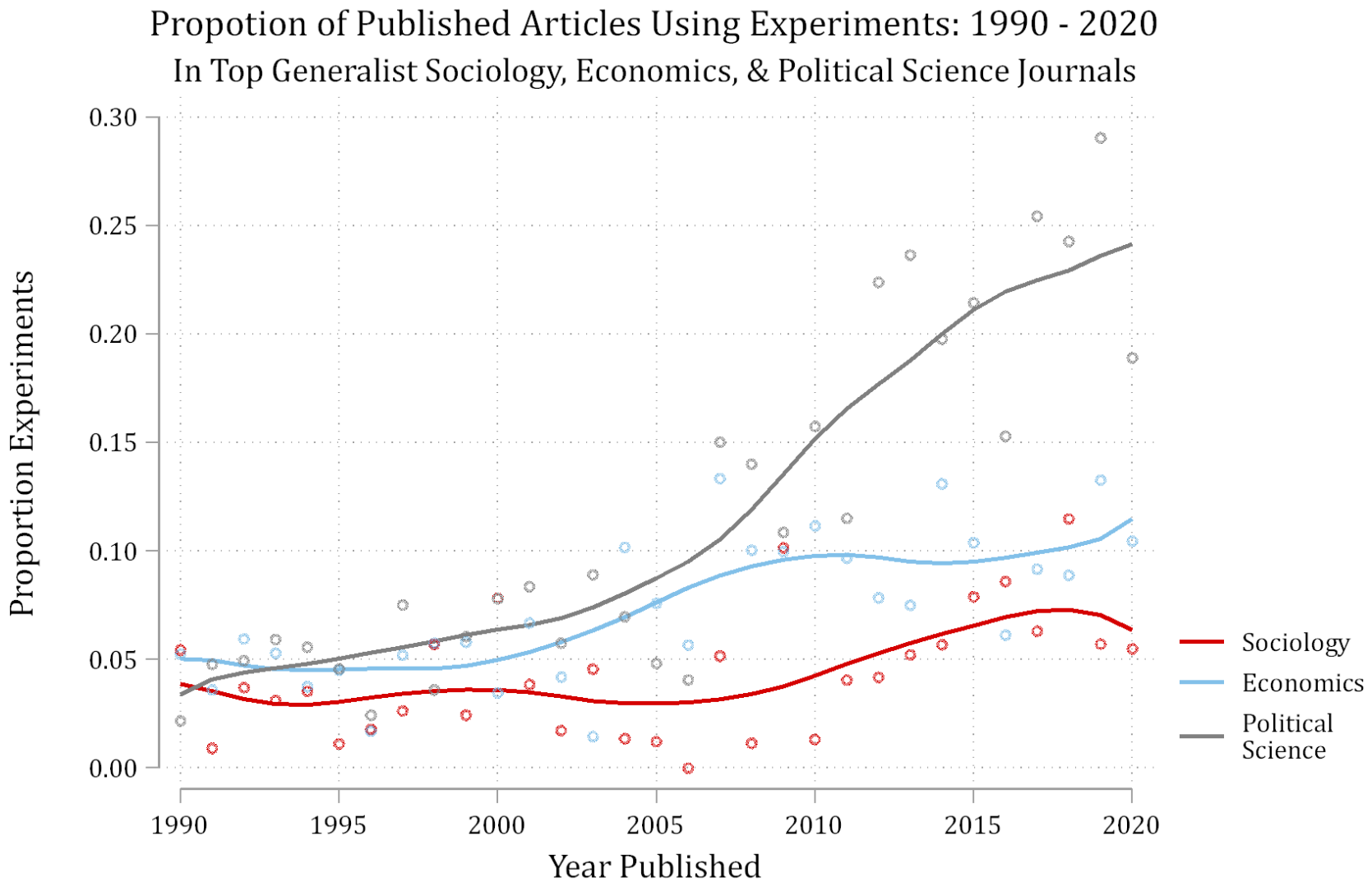
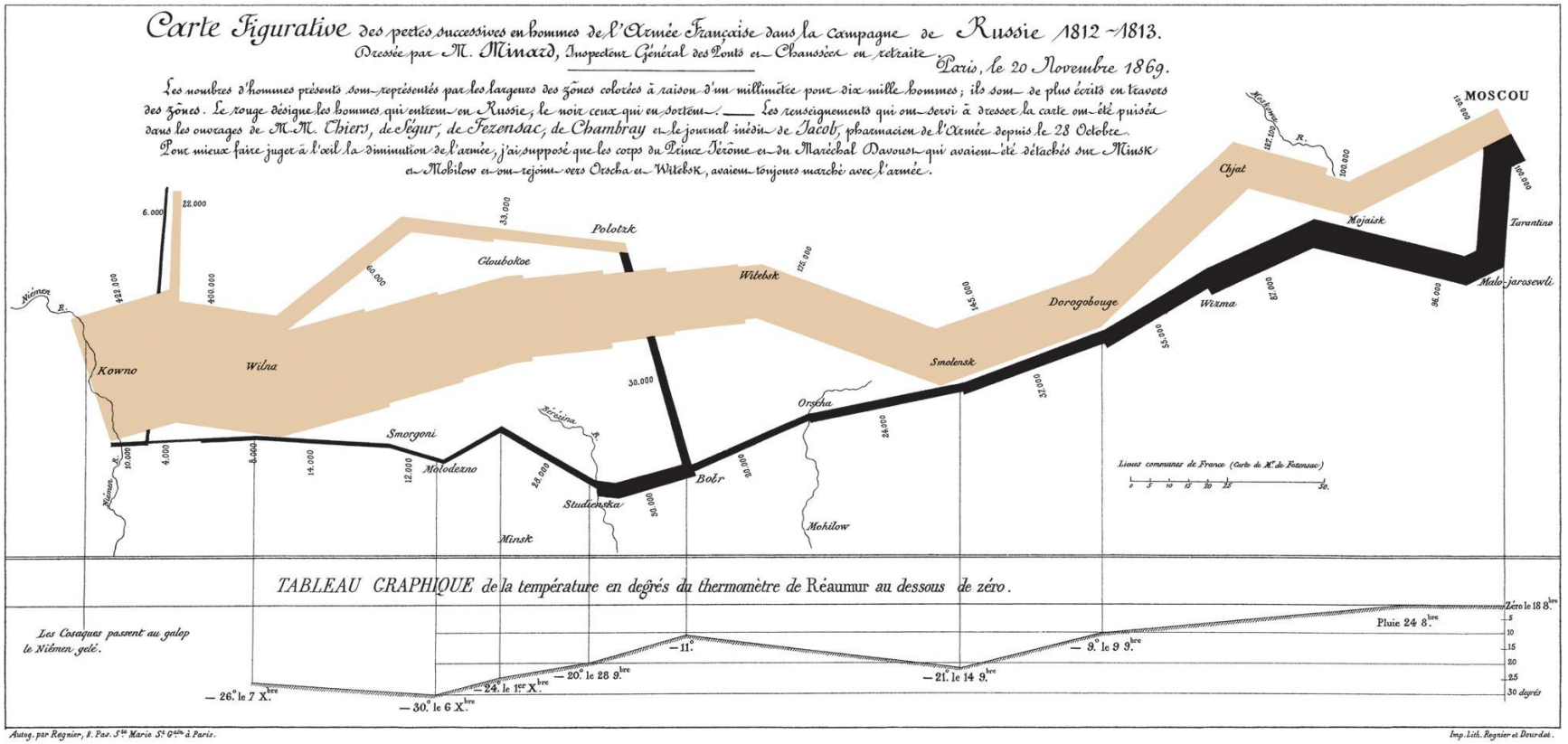


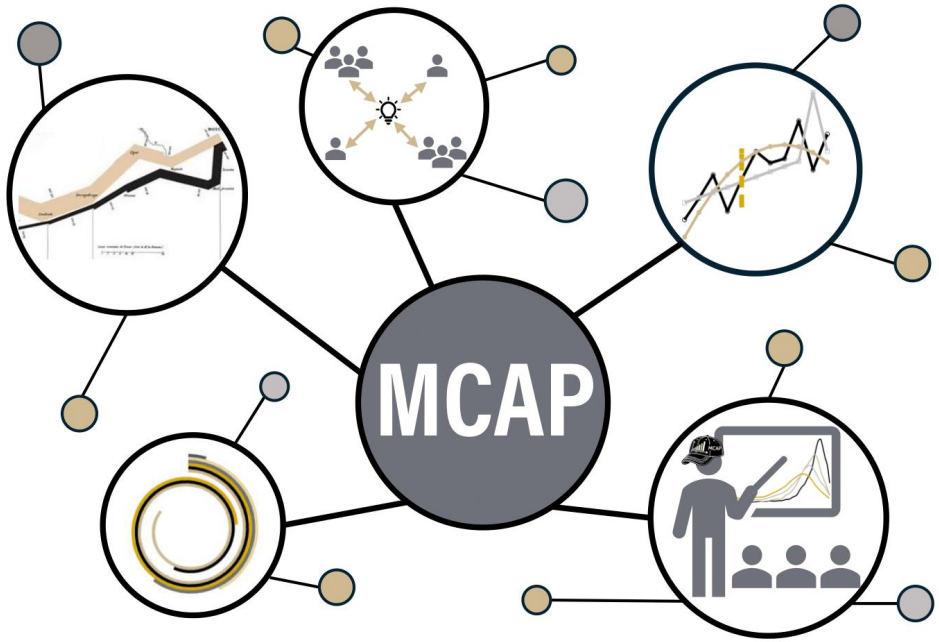
*Figure 1 from Mize and Manago 2022

Historical example

Napoleon's march to, and retreat from, Moscow in the War of 1812

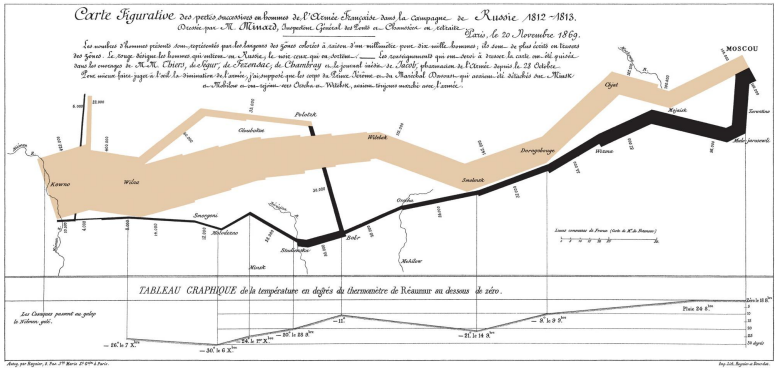


*Read more about the graphic here



THE METHODOLOGY CENTER

PURDUE UNIVERSITY



*Read about The Methodology Center graphic

Principles of good data visualization

1. **Look at your data**
2. **Use graph types people read accurately**
3. **Maximize the data-ink ratio, within reason (maybe)**
4. **Show uncertainty**
5. **Make groups easy to tell apart**
6. **Choose axes that fit the data**
7. **Design for your audience**
8. **Use graphs, not tables, for model results**
9. **Show what the model predicts**
10. **Don’t assume linearity**

Principles of good data visualization

1. **Look at your data**
    a. Problems and/or quirks with the data are easily ascertained
    b. Modeling strategies often become clear only after examining the data

2. **Use graph types people read accurately**
    a. Angles and areas are difficult to judge accurately; a common scale is necessary to compare across groups
    b. Bar charts and dot plots are among the most effective tools for comparing amounts across groups

3. **Maximize the data-ink ratio, within reason (maybe)**
    a. All aspects of the graph should aid in presenting the data
    b. My less strict rule: all color should convey some information

4. **Show uncertainty**
    a. Confidence intervals and standard errors should be included for model estimates / predictions
    b. Be clear in your graph and/or text whether CIs or SEs are being plotted

Principles of good data visualization

5. **Make groups easy to tell apart**
    a. A legend should clearly delineate between the groups or data should be labeled directly
    b. The colors and/or symbols on the graph should clearly delineate between different lines/groups

6. **Choose axes that fit the data**
    a. The “ticks” and gaps on the axes should be clear and logical
    b. The y-axis (and usually the x-axis as well) should cover at least the 10th to the 90th percentile of observed values

7. **Design for your audience**
    a. All graphs should have a descriptive, informative title and substantive axis labels
    b. Talks: the font in the graph should be large enough to be easily read from afar

Principles of good data visualization

8. **Use graphs, not tables, for model results**
    a. Talks: keep your audience awake (and aid more rapid interpretation): use graphics whenever possible
    b. Coefficient plots are very effective for replacing regression tables

9. **Show what the model predicts**
    a. The most effective way to understand a model is almost always to examine what the model predicts
    b. Do not connect predictions for nominal IVs with lines

10. **Don't assume linearity**
    a. Examine the functional form of the relationship between your predictor and your outcome before you fit a model


THE 10 COMMANDMENTS OF DATA VISUALIZATION
I. LOOK AT YOUR DATA
II. USE GRAPH TYPES PEOPLE READ ACCURATELY
III. MAXIMIZE THE DATA-INK RATIO, WITHIN REASON (MAYBE)
IV. SHOW UNCERTAINTY
V. MAKE GROUPS EASY TO TELL APART
VI. CHOOSE AXES THAT FIT THE DATA
VII. DESIGN FOR YOUR AUDIENCE
VIII. USE GRAPHS, NOT TABLES, FOR MODEL RESULTS
IX. SHOW WHAT THE MODEL PREDICTS
X. DON'T ASSUME LINEARITY

Conclusion

1. Visualizations are often the most effective way to convey complex statistical results to both statistically savvy **and** non-statistically savvy audiences
    a. Visualizations increase the audience for your work
    b. Visualizations also help you understand your own data and results

2. Be creative; data visualization gives you the chance to bring some art into the science of data analysis
    a. Talks and research papers can (sometimes) be boring [sad but true!]
        i. Visualizations offer a way to grab the attention of your audience



Go forth and make pretty graphs

1. The magical thing about data visualizations is that they are simultaneously:
    a. Easier to understand than tables
    b. And show a broader range of your data/findings
        i. E.g., Predictions across the entire range of the data
        ii. E.g., The entire magnitude of a variable's effect

2. So, go forth and visualize your data and models!
    a. Using pretty graphs and clean plots, of course 😊



References & Recommendations

General recommendations (books)

Wilke, Claus O. 2019. *Fundamentals of Data Visualization*. O'Reilly Media. (read online for free)

- Excellent overview with practical advice
- Software agnostic

Healy, Kieran. 2018. *Data Visualization: A Practical Introduction*. Prince University Press. (read online for free)

- Helpful for applied researchers as the book covers model visualization (few others do). Focused on R and ggplot.
- I recommend the preface and Chapter 1 to everyone—even if you never plan to use R

Schwabish, Jonathan. 2021. *Better Data Visualizations.* Columbia University Press.

- Excellent general overview
- Software agnostic

General recommendations (articles)

Overview of data visualization

Franconeri, S. L., Padilla, L. M., Shah, P., Zacks, J. M., & Hullman, J. (2021). The science of visual data communication: What works. *Psychological Science in the public interest*, *22*(3), 110-161.

Overview of making predictions and calculating marginal effects in R

Rohrer, J. M., & Arel-Bundock, V. (2026). Models as prediction machines: how to convert confusing coefficients into clear quantities. *Advances in Methods and Practices in Psychological Science*, *9*(2).

Substantive articles with many visualizations

Mize, T. D., & Kincaid, R. (2025). Role-Accumulation and Mental Health across the Life Course. *American Sociological Review*, 00031224241313394.

Mize, T. D. (2025). Competence perceptions at the intersection of gender and race-ethnicity. *Social Psychology Quarterly*, *88*(2), 271-284.

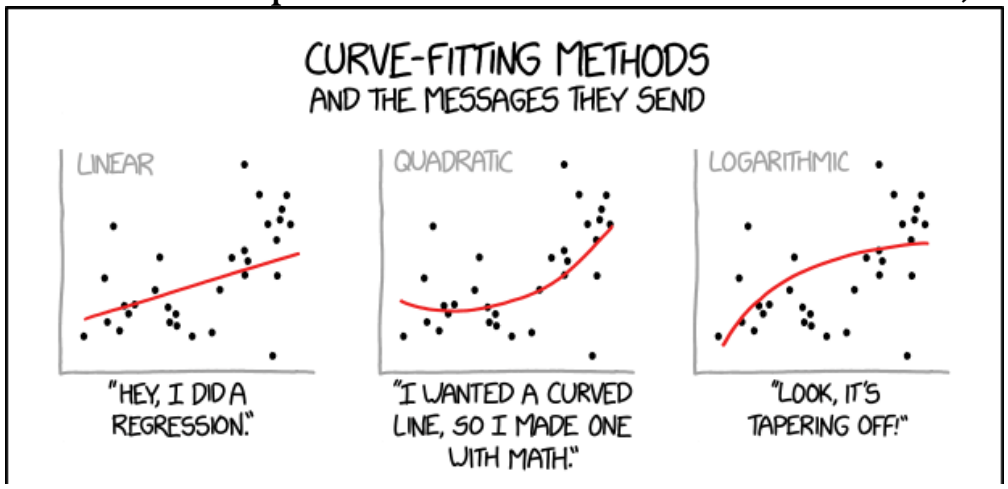
Appendix

Assessing Nonlinearities & Model Specification Using Predictions

1. Before you fit a model, it is important to examine the functional form of the relationship between your predictor and your outcome (don't assume linearity!)

2. Above, we covered some bivariate methods (consider only *x* and *y*)
    a. However, most of our models include multiple predictors / control variables

3. Ordinal predictors can be specified as continuous or nominal, with tradeoffs



Treating continuous predictors as nominal

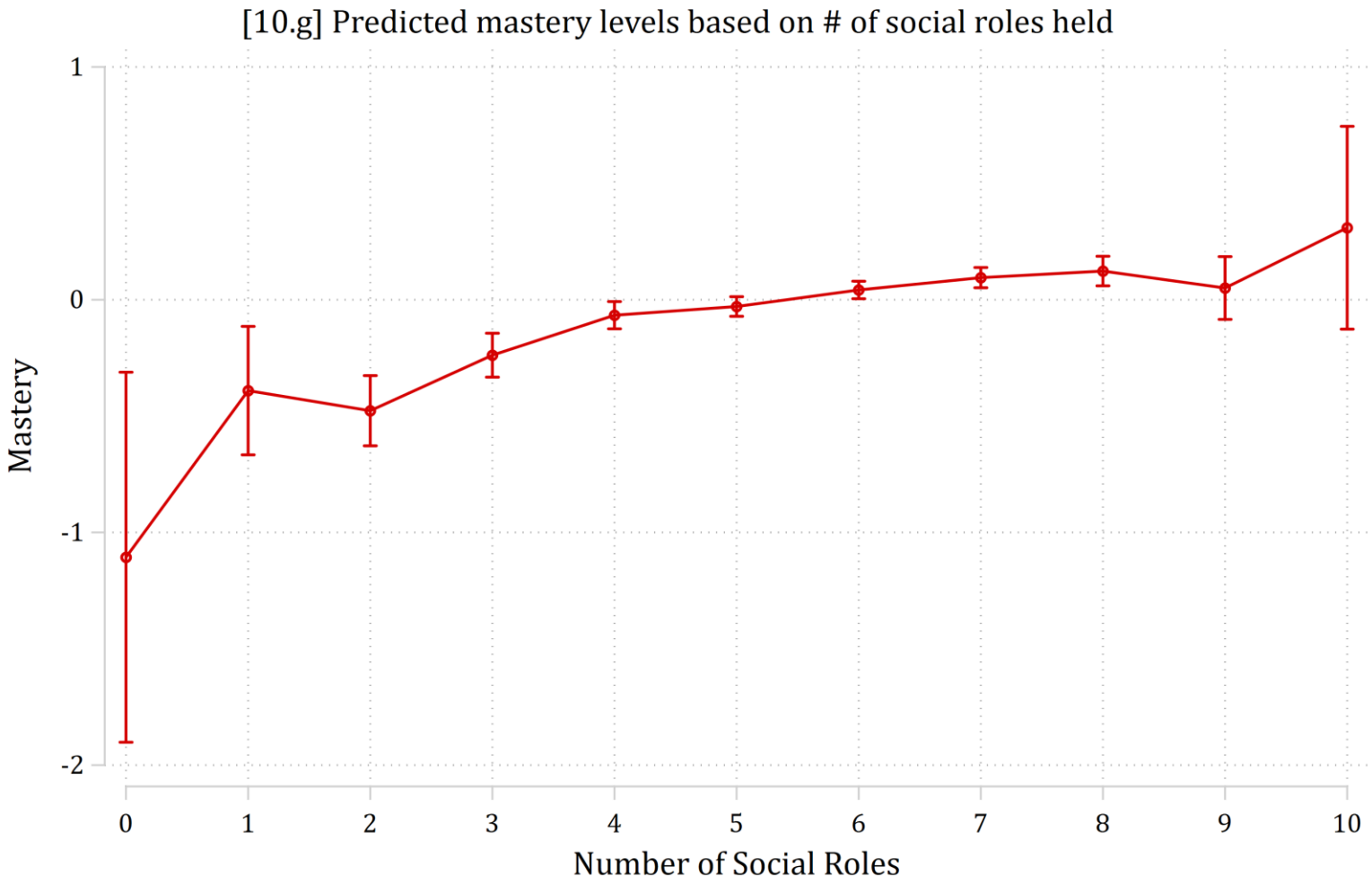
1. An elegantly simple but remarkably flexible way to examine the functional form of a continuous or ordinal IV is to treat it as a nominal variable in your regression model
    a. E.g., For a count variable of # of *chronic diseases* (0 to 8) omit 0 as reference category; add coefficients for 1, 2, 3, 4…8 diseases
    b. For IVs with lots of values, you can "coarsen" or "bin" them first
        i. E.g., Create age categories of 20 – 30, 31 – 40, 41 – 50, etc.
            1. Use `cut( )` function in R

2. Can use this technique for almost any regression model and almost any outcome
    a. Can also incorporate control variables
    b. Can incorporate fixed and/or random effects
    c. Etc.

3. For this technique, you want to go straight to graphs rather than worrying about the coefficient estimates

```
plot_predictions(reg_roles_nom, condition = "role")
```

```
reg masterySS i.role c.age c.income i.race c.exercisSS
margins, at(role=(0(1)10)) saving(prs_nom)
```



Comparing nominal and continuous specifications

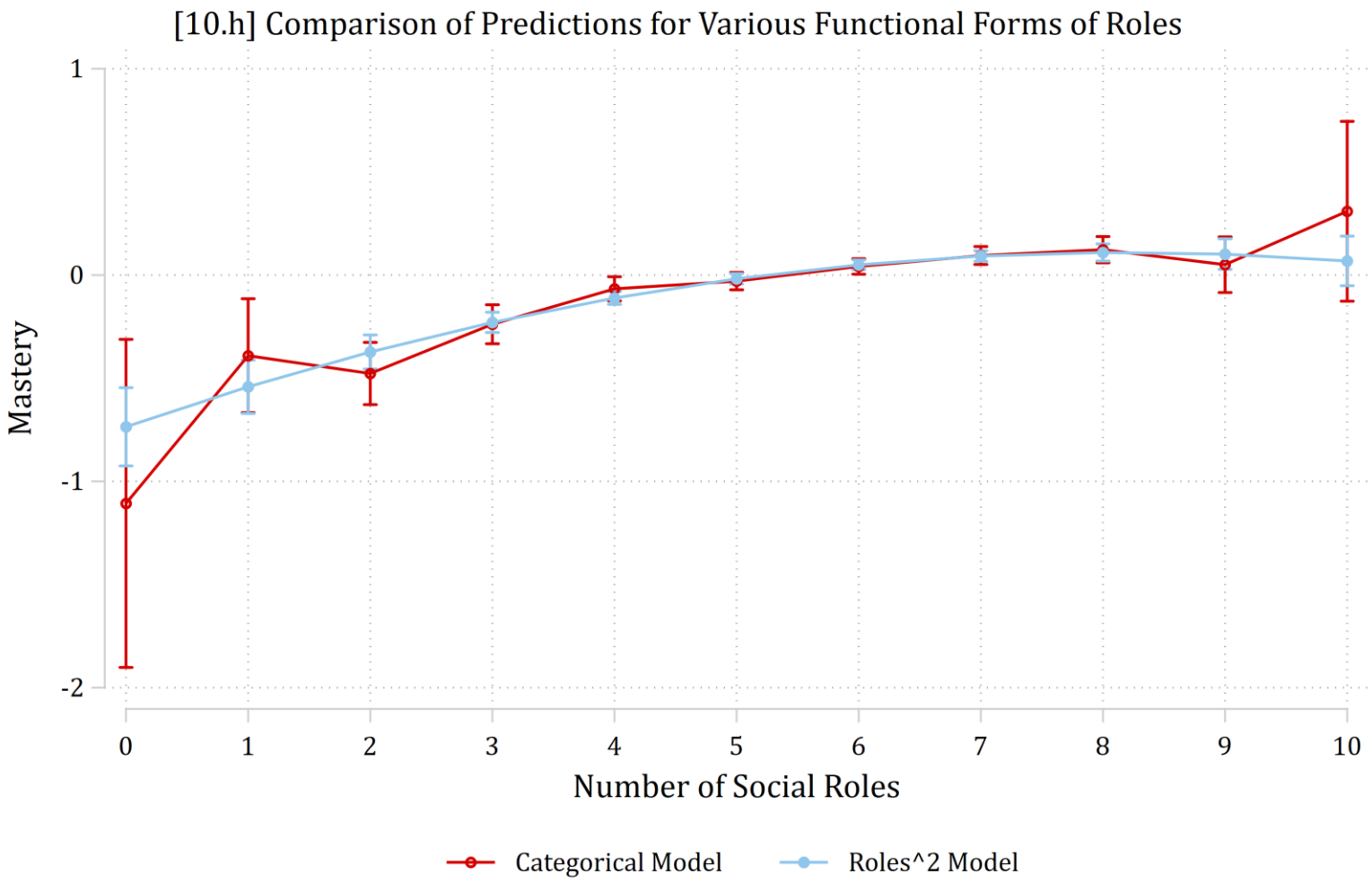
1. The nominal approach is mostly meant as descriptive
    a. It took 10 coefficient estimates for the predictions on the prior page

2. Compare nominal specification to a nonlinear continuous specification
    a. E.g., Polynomials like *roles* and $roles^2$

3. Overlay predictions from the two specifications
    a. Stata: **`combomarginsplot`**; R: **`ggplot`** (see code files)

```
plot_predictions(reg_roles_sq, condition = "role") +
  geom_pointrange(data = pred_roles_nom, …)
```

```
combomarginsplot prs_nom prs_sq, labels("Categorical Model" "Roles^2 Model") …
```



Specifying ordinal predictors

1. Independent variables can be entered into a regression model one of two ways:
    a. Continuous or nominal (n.b., binary is nominal but with only 2 categories)

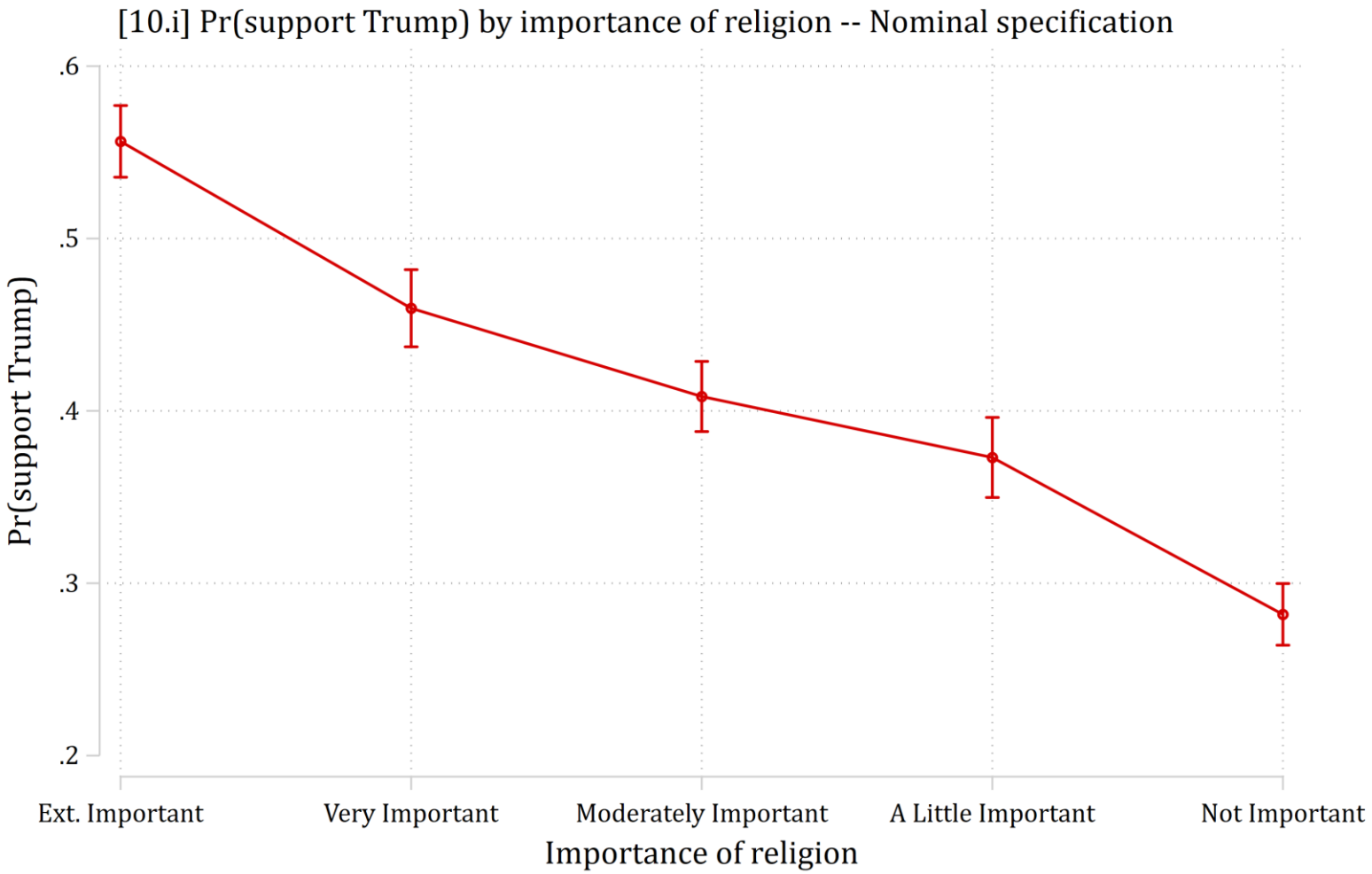
2. For an ordinal predictor, you must choose (ordinal is not an option)
    a. Continuous is simpler
        i. E.g., For a 5-category ordinal IV only need to estimate 1 coefficient
    b. Nominal is more complicated but always more accurate*
        i. If entering as nominal, need 4 coefficients (representing 4 categories and one omitted reference category)
        ii. 4 coefficients will always provide a somewhat better fit* than 1

3. Which should you choose? Try both:
    a. If inferences are very similar, choose the simpler specification (continuous)
    b. If inferences diverge, choose the more complicated but more accurate specification (nominal)

*You can get a significance test of this. But R^2 will always be the same or higher in nominal specification (even if not significantly different). An information criterion like AIC or BIC is more helpful

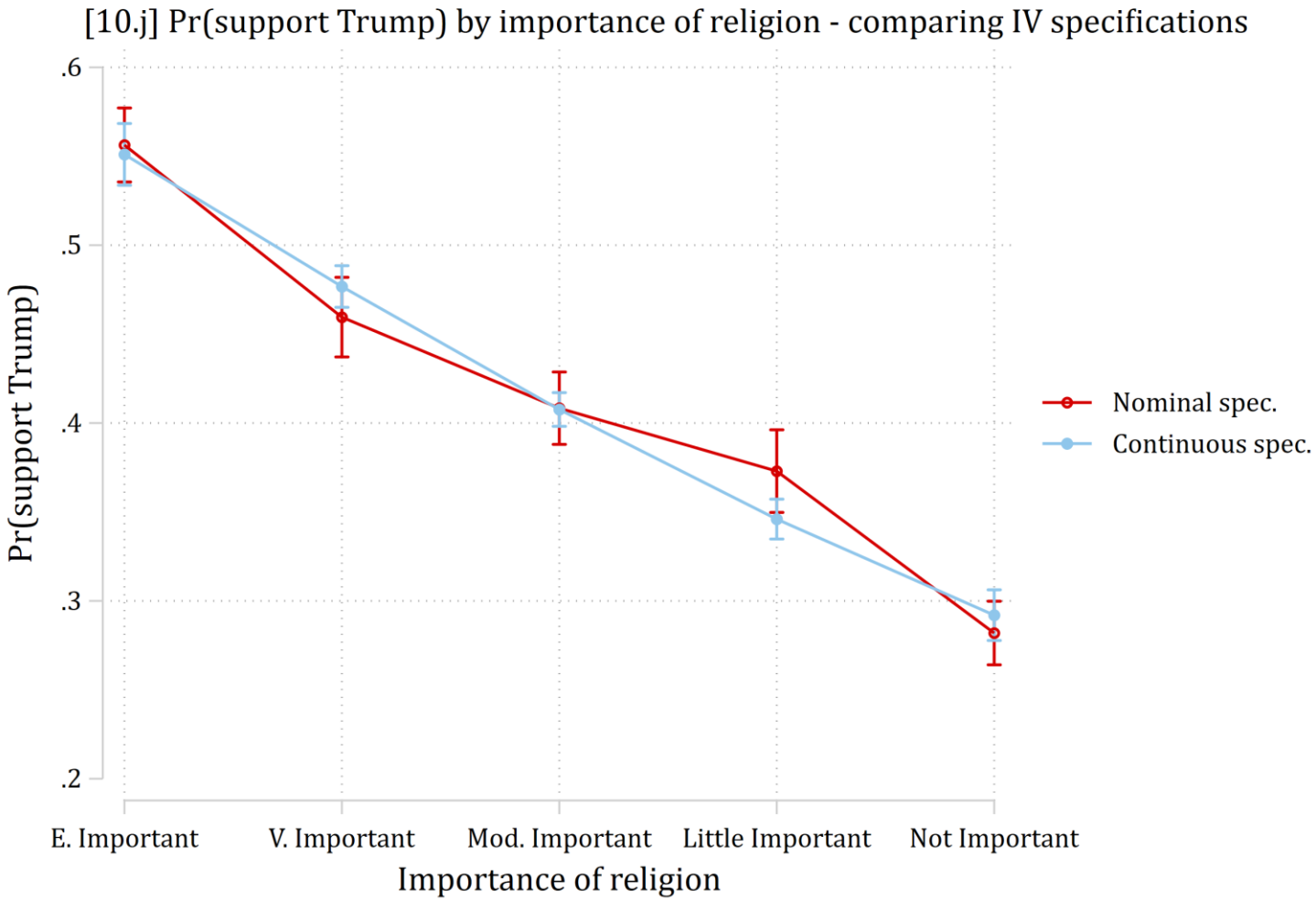
```
plot_predictions(logit_rel_nom, condition = "relimport")
```

```
logit vote_trump i.educ i.relimport age hhincome …
margins, at(relimport=(1(1)5)) saving(pr_rel_nom)
```



```
plot_predictions(logit_rel_con, condition = "relimportC") +
  geom_pointrange(data = pred_rel_nom, …)
```

```
combomarginsplot pr_rel_nom pr_rel_con, labels("Nominal spec." "Continuous spec.") …
```



Choosing confidence intervals or standard errors

1. In my opinion, plots designed to show group differences (e.g., bar charts) are sometimes more effective with standard error bars than confidence intervals
    a. Standard errors are intuitive
        i. Regression tables are normative and most often include standard errors as measures of uncertainty
    b. Standard errors = 68% confidence intervals
        i. 68% confidence intervals = ± 1 SE
        ii. 95% confidence intervals = ± 1.96 SEs

2. Confidence intervals are normative for line charts and coefficients plots

3. Unless the measures of uncertainty (CI or SE) greatly hinder the interpretability of the plot, it is always best to include these on plots of sample data (i.e., data that does not include the entire population)

4. Be clear in your graph and/or text whether CIs or SEs are being plotted
    a. Audiences will assume they are 95% CIs unless you clearly state otherwise

Use logical colors

1. The colors you chose should not fight against logic or the audience's inclinations
    a. E.g., When you think of Democrats: what color do you think of? For Republicans? For the Green Party?
        i. But depends on your context and audience!

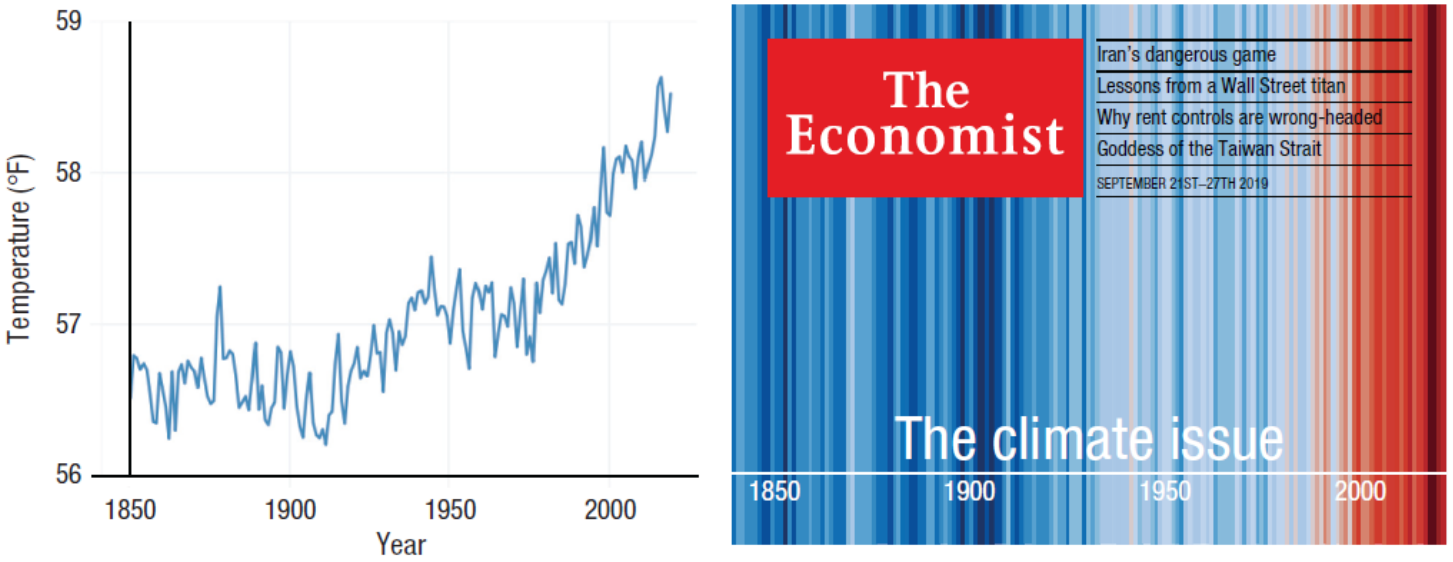


Fig. 2. Example of a visualization design using color intensity to focus the viewer on big-picture statistics. The line graph at the left encodes global climate data precisely. The cover of a September 2019 issue of *The Economist*, shown on the right, used color intensity to depict the same data with far less precision for each value, leading the viewer to focus on the bigger picture.

*From Franconeri 2021. The colors correspond to the values of the temperature; the blue (for colder temps) and red (for hotter temps) are logical choices

Graphics themes in R and color palettes

1. Graphics themes (`ggplot2`'s term) are sets of defaults for graphics
    a. These can dramatically change the "look and feel" of a graph
    b. See an overview of what themes are and different options here

2. Base R's default graphics theme leaves much to be desired
    a. `ggplot2`'s default theme (`theme_gray`) is much better, but still imperfect
    b. The good news is there are good additional options built into `ggplot2`

3. You can change a theme on a single graph by specifying it as an element on a `ggplot2` graph. E.g.,

```
fig1 + theme_gray( )
```

The `cleanplots` R package

1. `cleanplots` is now also available as an R package to automate the best practices I've covered in these slides and use `cleanplots` colors
    a. First, load or install the `remotes` packages

```
install_github("tdmize/cleanplots")
```

- `scale_color_cleanplots()` — Applies the `cleanplots` colors to a plot's color aesthetic; the theme and everything else are left unchanged
- `scale_fill_cleanplots(palette = "bars")` — Applies the softer `cleanplots` colors to a plot's fill aesthetic, designed for ink-heavy elements like bars, boxplots, and areas
- `scale_shape_cleanplots()` — Applies the `cleanplots` marker symbols: hollow shapes for the dark colors, solid shapes for the light colors, so groups are distinguishable even without color.
- `scale_linetype_cleanplots()` — Applies the `cleanplots` line patterns (solid, solid, dashed, dashed, shortdash, shortdash, longdash, longdash), keeping line graphs readable in black & white and for colorblind readers.

The `cleanplots` R package

- `theme_cleanplots()` — Applies the `cleanplots` look to a single plot: white background, dotted gridlines, gray axis lines, frameless legend at the right, and outlined facet strips.

- `cleanplots_defaults()` — One call applies the full `cleanplots` setup for the session: theme, automatic colors (main palette for color, soft bar palette for fill), larger markers, thicker lines, and customized fit lines.

- `cleanplots_save()` — Saves a figure at a fixed, publication-ready size (7 × 5 inches, 300 dpi) so plots look the same regardless of your plot window. Format follows the file extension (.png, .pdf, .tiff, ...).

Stata graphics and the `cleanplots` graphics schemes

`cleanplots` implements a host of features that draw on best data visualization practices (some of which are borrowed from the excellent black and white graphics scheme **`plainplots`** from Bischof (2017) ; these are also repeated in the excellent **`white_tableau`** by Naqvi). This allows you to make publication quality Stata graphics with very little effort. With **`cleanplots`**, many of the defaults you would change before via code are already changed for you

For example:

a. Default colors are easier to distinguish from each other and are more aesthetically pleasing
    i. Default colors allow graphs to be printed in color or black and white
b. Marker shapes and line features (e.g. solid, dashed, etc.) better distinguish lines and ensure lines will be distinguishable if printing in black and white
c. All axis markers are horizontal – thus easier to read
d. A non-invasive light grid is on all plots
    i. Both vertical and horizontal gridlines
e. The background of the plot is white
f. Legends are placed to right of the graph
g. And many more!

Presenting multiple model results

1. Coefficient plots are very effective when the goal is to compare effects across models and/or groups

2. E.g., If multiple models are fit for different groups, useful to plot coefficients from each model for easy comparison of effects
    a. Confidence intervals are very important here

Incorporating distributional information

1. Distributional information about an IV can be incorporated into plots of predictions via overlaid histograms, kernel density plots, or rug plots
    a. Especially useful when a distribution deviates notably from uniform or normal

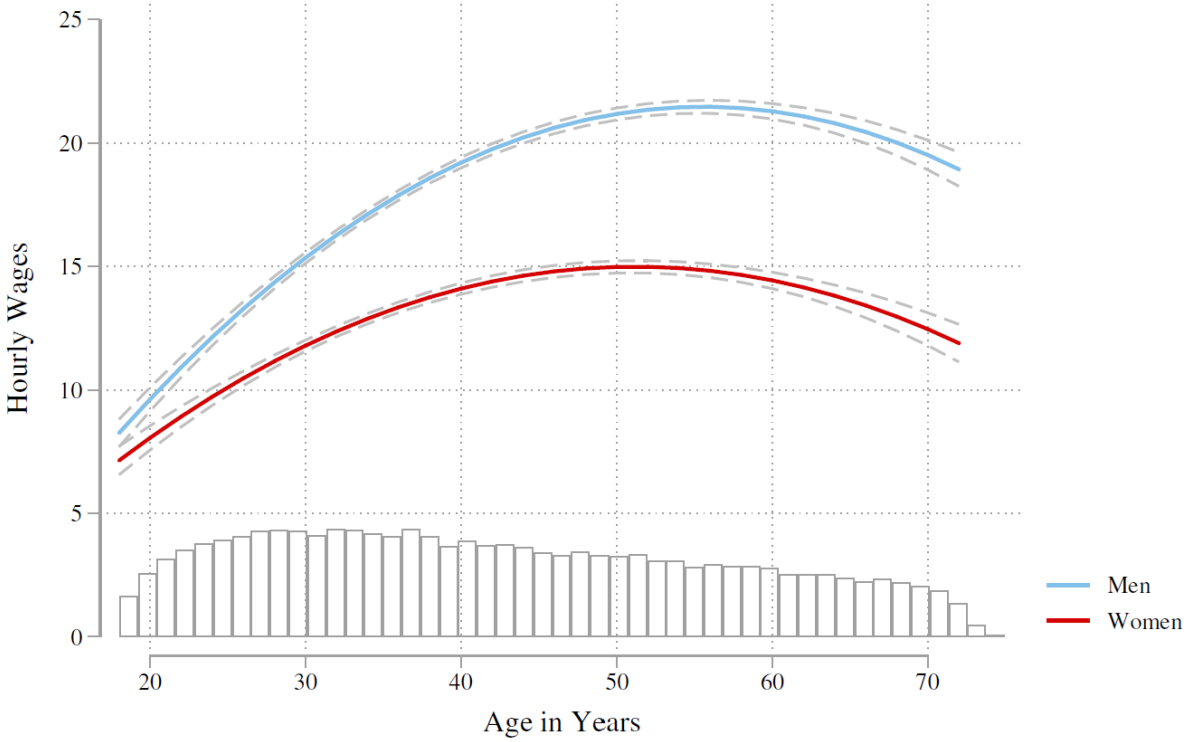
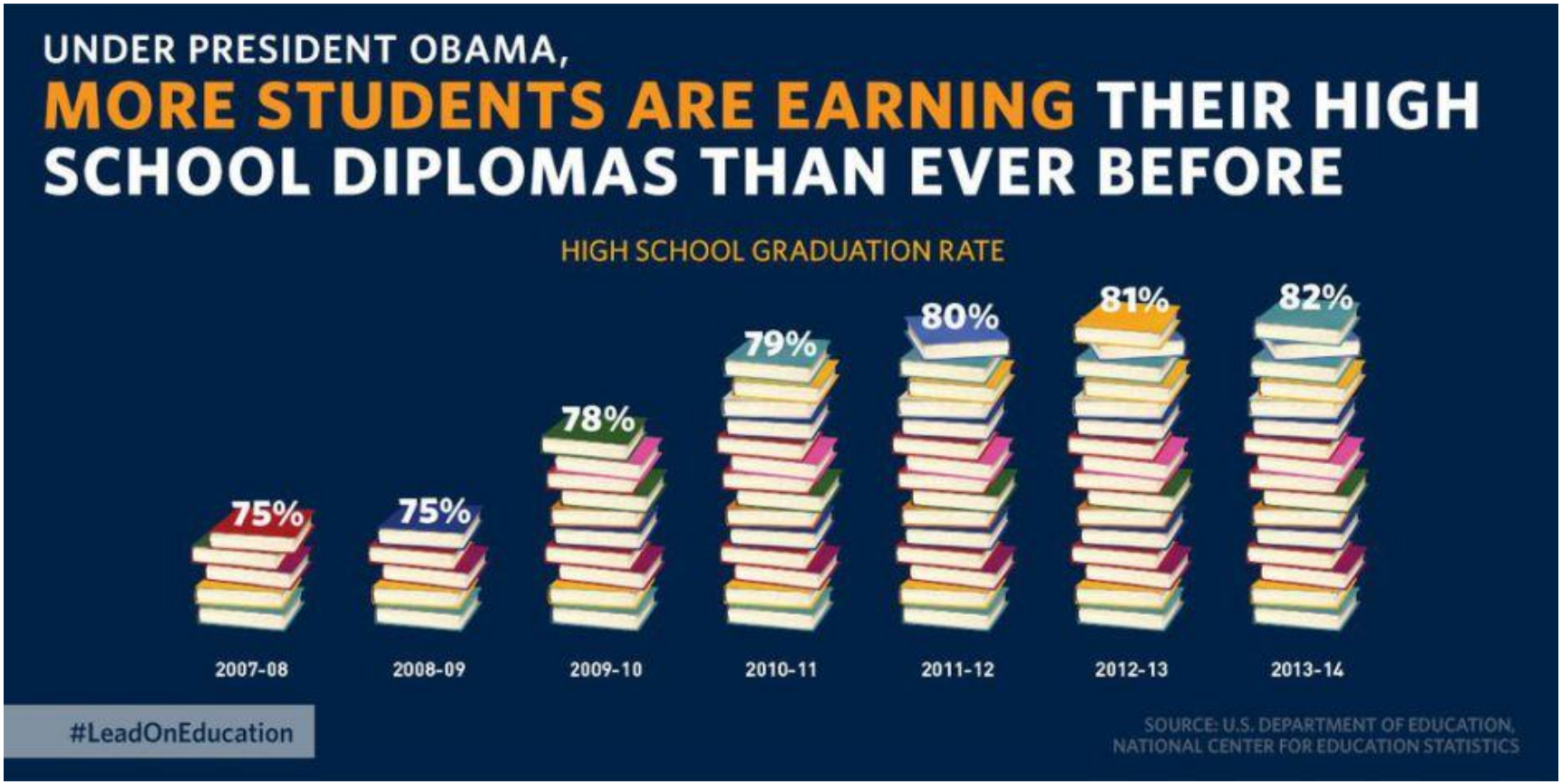


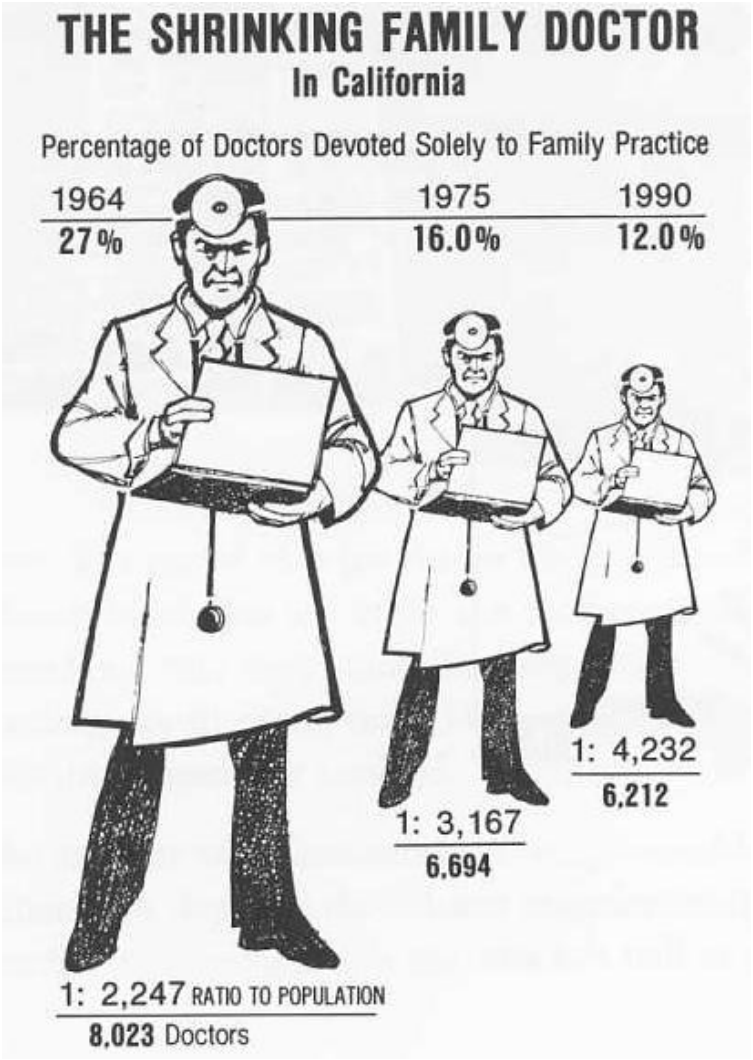
Figure 1 from Mize (2019)

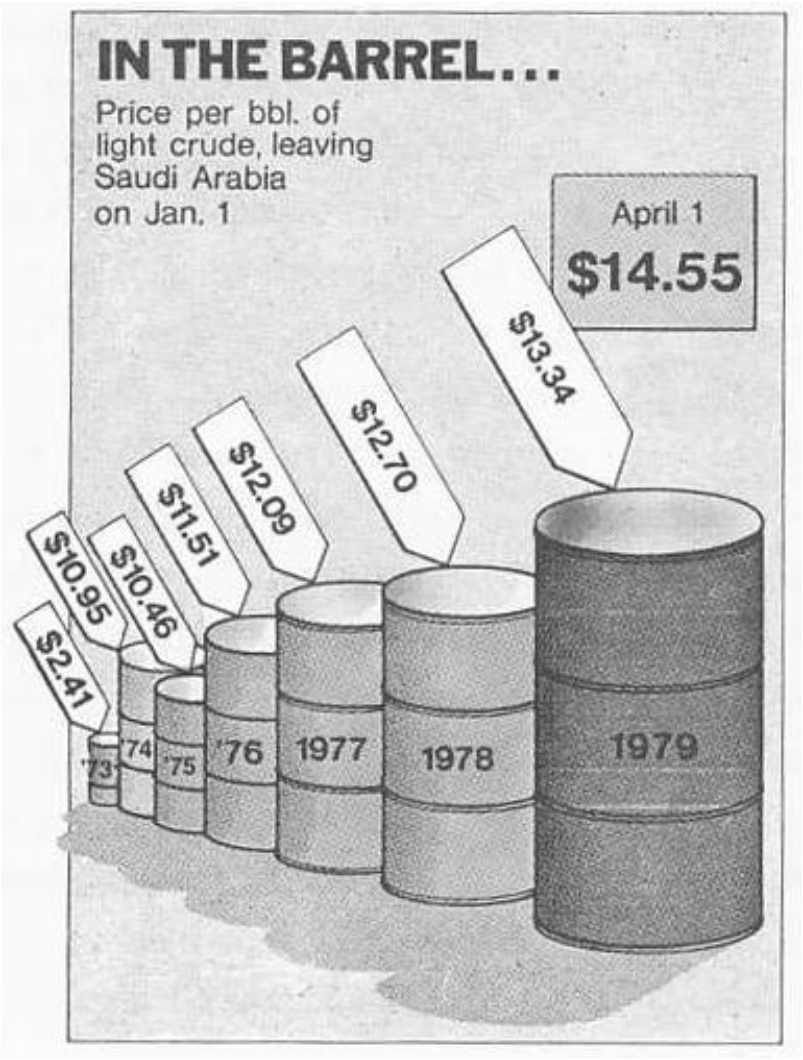
Figure 1: Predicted hourly wages based on age and gender: curvilinear effect of age and interaction effect between age and gender.

Misleading example: area ≠ rate



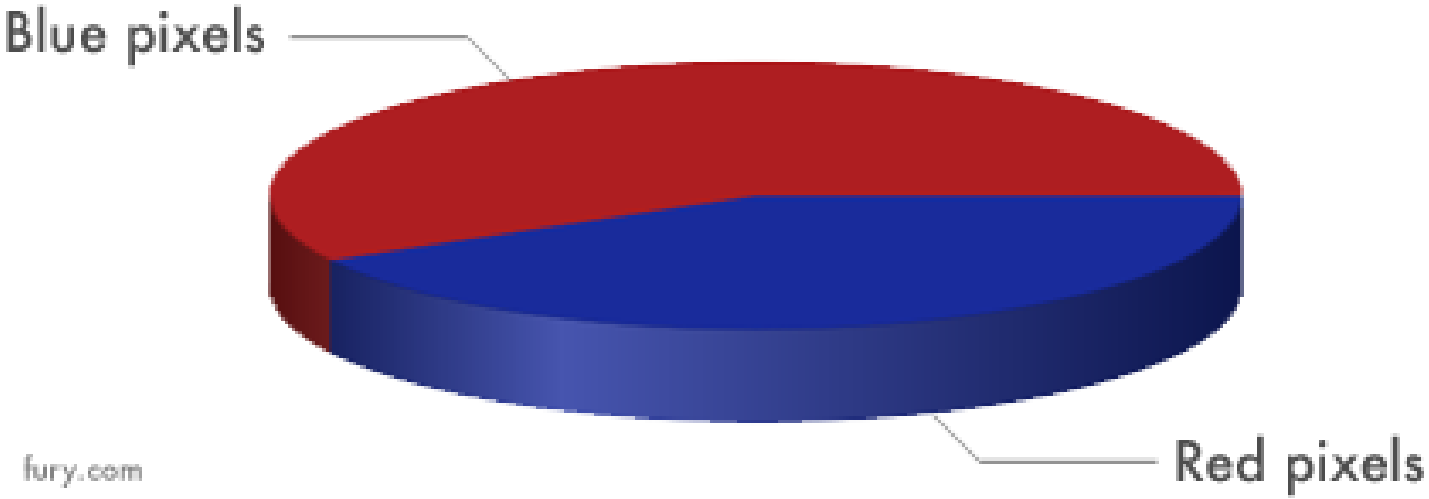
Misleading examples: area ≠ rate or price

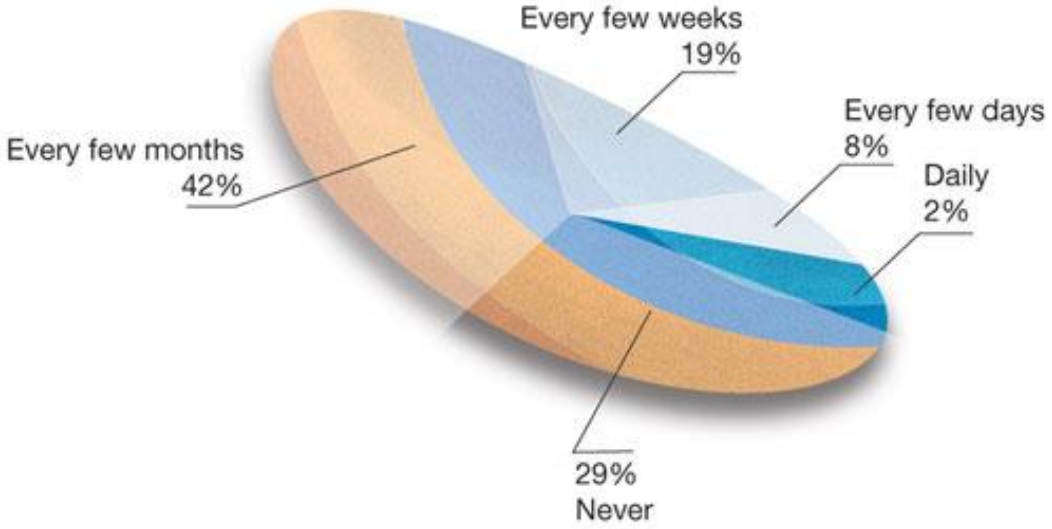




*Examples from Tufte (1983)

3-D Graphics: Just say no





General advice: 3D graphics

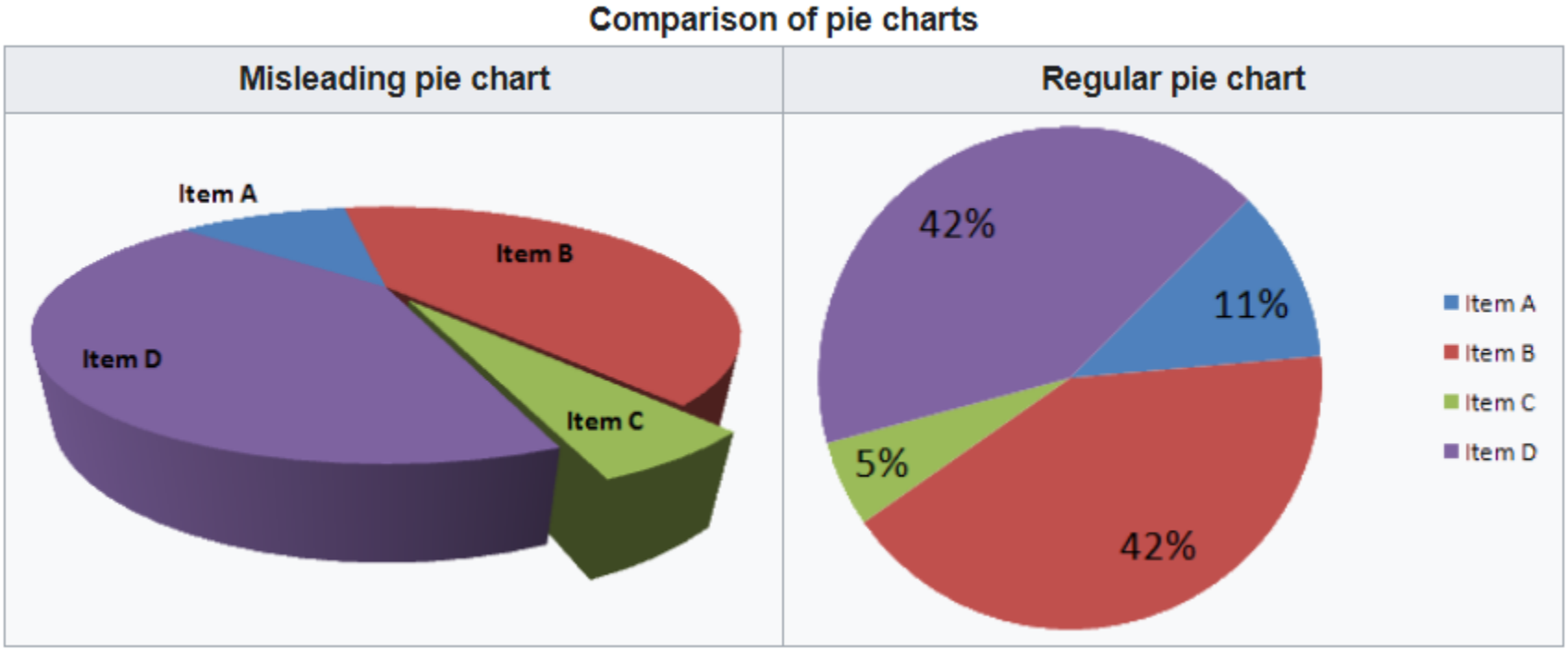
1. 3D-graphics usually make it more difficult to read graphs rather than easier
    a. It is very difficult to read the information encoded on the third (z) axis
    b. Adding a third axis makes it more difficult to read the first two (x and y) axes!

2. Without the ability to manipulate a 3D graphic—the choice of orientation greatly affects the perception of the plot (e.g. the length of lines)

3. For all of these reasons, I recommend avoiding 3D-graphics

4. Exceptions: 3-D graphics can sometimes be useful pedagogically (when you have ample time to explain them)
    a. Can also work with "interactive" graphics where the user can spin the graph to look at it from multiple angles

3D graphs can be misleading



1. Here, the 3-D nature of the graph is not only unnecessary—it also leads to a misleading graph where pie slices in the foreground look larger than those in the back

*Image from Wikipedia's "misleading graphs" collection