STAT 214 Spring 2025 Week 3

Zach Rewolinski

Heavily Borrowed from Anthony Ozerov's STAT 215A Materials.

🚨 Announcements 🚨

- Reading Assignment #1 is graded (see Gradescope)
- As requested, a README has been added to the `lab1` folder of the `stat-214-gsi` repo. It summarizes what I said in discussion last week regarding the lab.
- For those with questions about `git`, please see `git-supplementary-materials.pdf` in the `discussion/week3` folder of the repo. This also includes a bit of information on hidden files such as `.gitignore`.

Any Lab 1 questions?

Data Visualization

Different kinds of data visualizations:

Different kinds of data visualizations:

- Line plot
- Scatterplot
- Bar plot
- Boxplot
- Histogram
- Heatmap
- Actual map
- Correlation matrix
- Density plot (1d or 2d)
- Q-Q plot

https://matplotlib.org/stable/gallery/index.html

https://seaborn.pydata.org/examples/index.html

- Regression
- Spline fit
- Smoothing
- PCA
- Various dimension-reductions
- Clustering

What if none of the data is quantitative??

- Some of it probably is
- Key trick: you can make quantitative data from categorical data by computing proportions. e.g. "76% of observations in group A are ___, while only 40% of observations in group B are ___"
  - Barplots of proportions in different groups
  - Heatmap of proportions when split across 2 categorical variables (or 1 categorical 1 quantitative)
- Correlations of binary variables

Design aspects

What makes a good figure?

What makes a good figure?

A good figure should be:

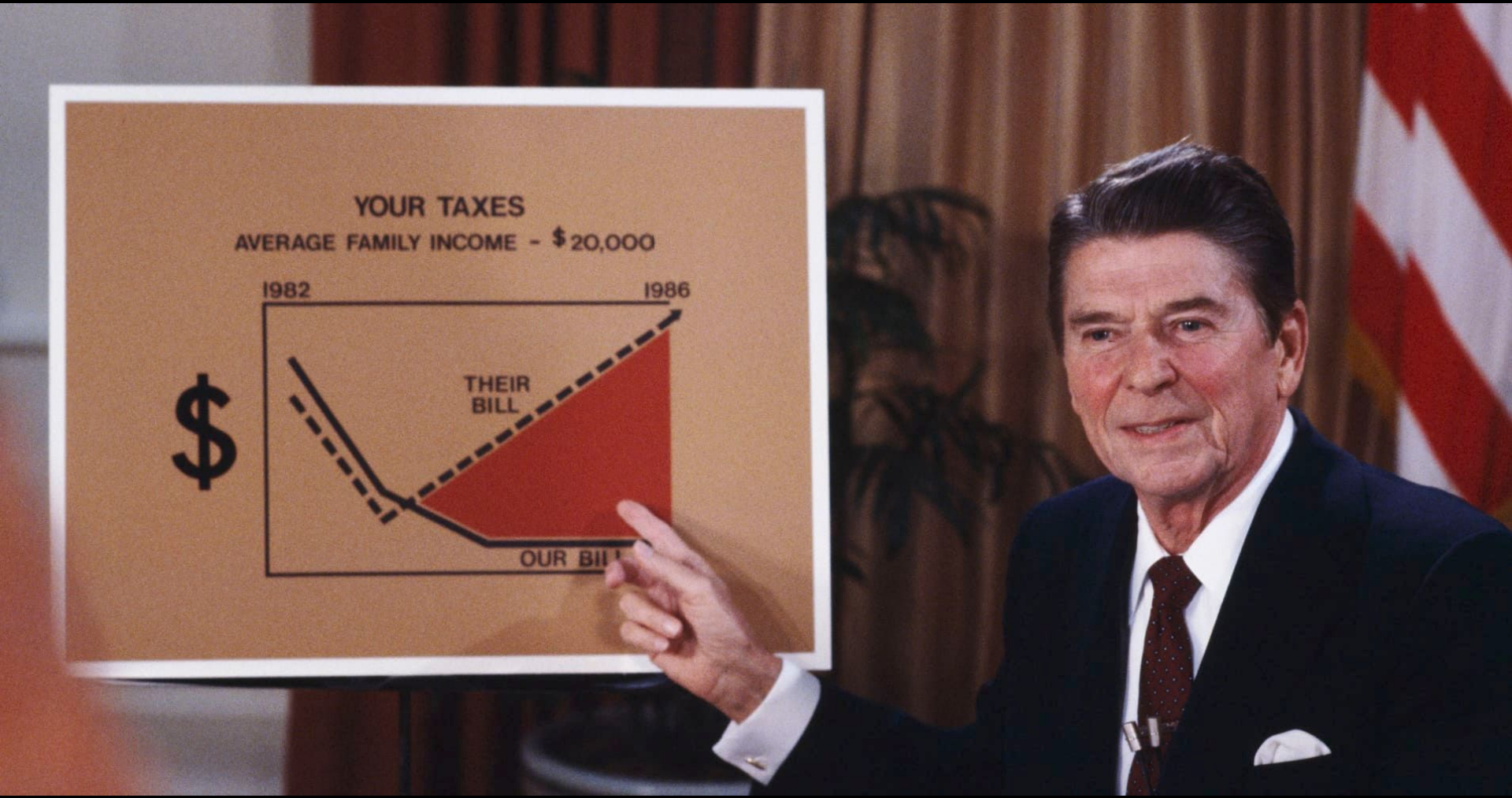
- Simple
- Informative
- Honest
- Efficient
- Aesthetic

What should you consider when making a figure?

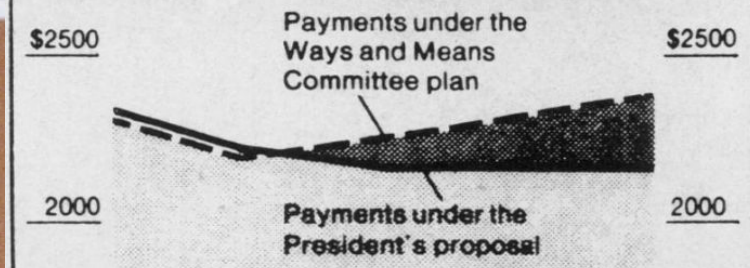
What should you consider when making a figure?

- Audience: Who is the figure intended for?
- Convention: How do people usually visualize similar things?
- Purpose: What should the figure accomplish? Is it useful?
- Medium: How will the viewer see the figure? How big will the figure be?

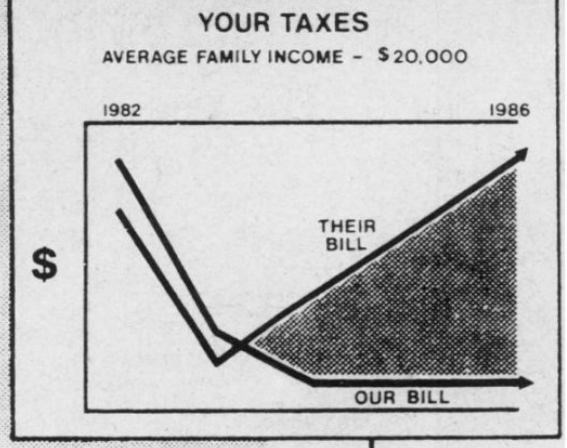

Live
The Oval Office


YOUR TAXES
AVERAGE FAMILY INCOME - $20,000
1982
1986
$
THEIR BILL
OUR BIL

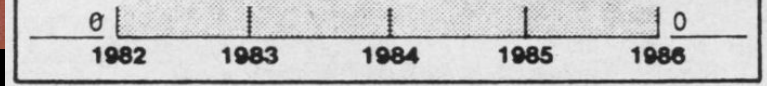
The Neutral View...



...And the President's



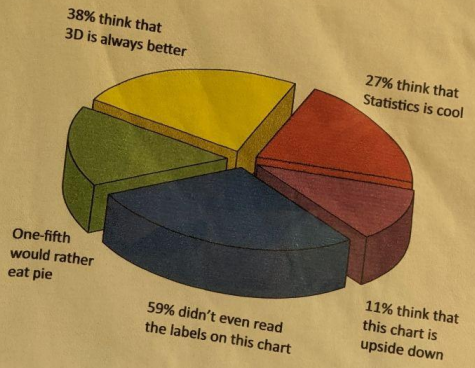
Taxes paid by a one-earner family with annual income of $20,000.



Test...
ice intervals
und...
keley
of Statist

PORT COMPANY

Berkeley Statistics

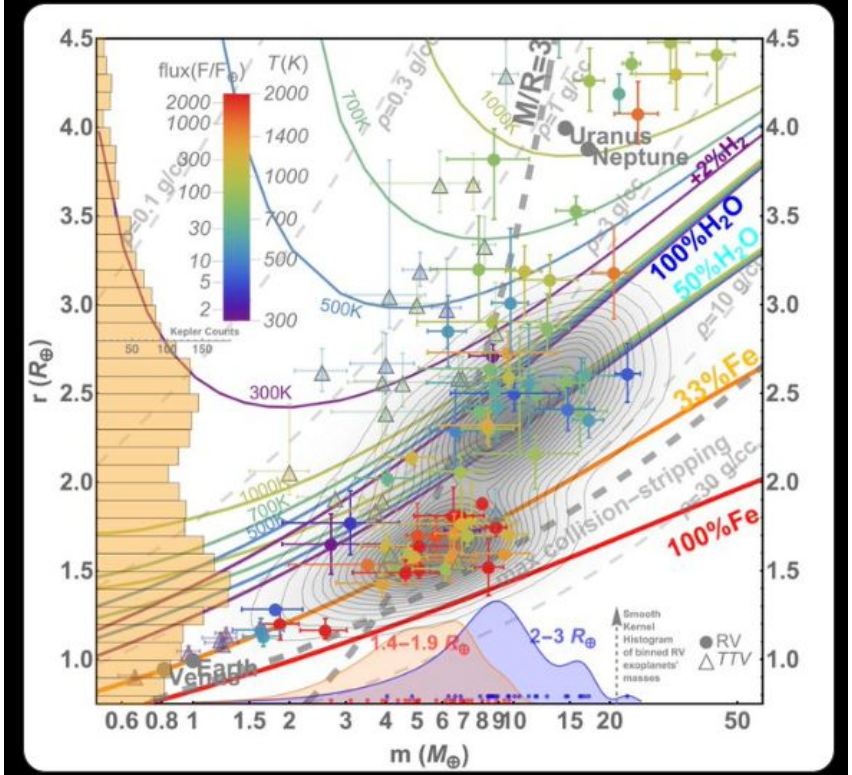


Saving the world from 3D pie charts

Elisa Granato
@Prokaryota

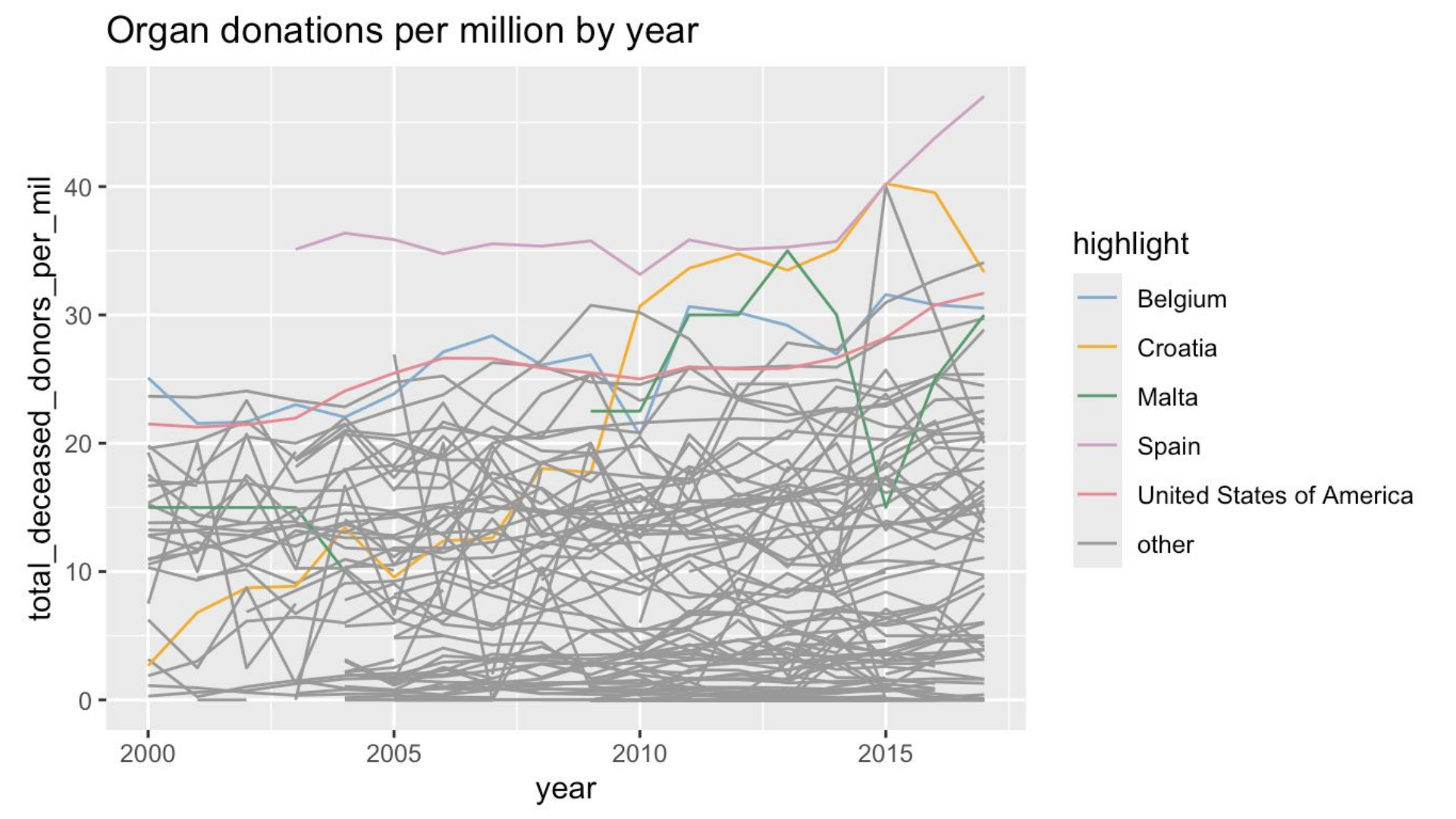
when you're only allowed 4 figures in the main text

Перевести пост



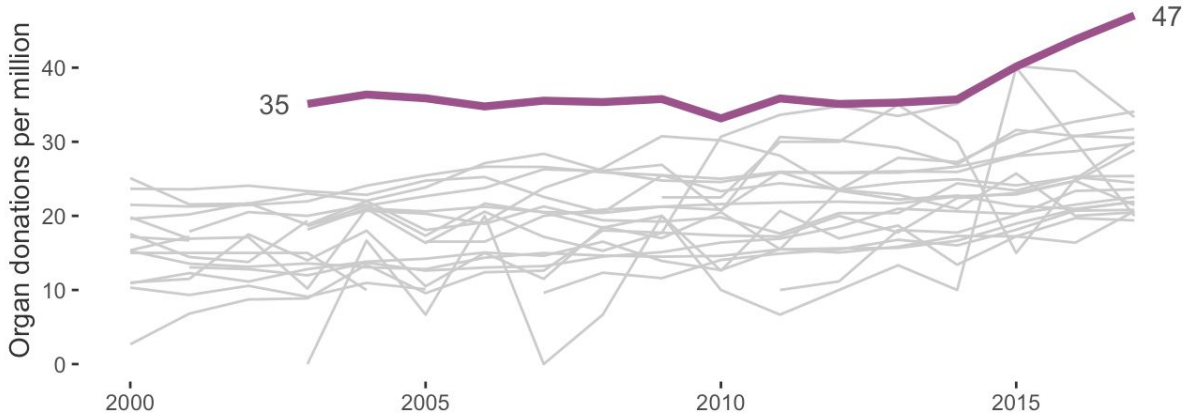
7:51 AM · 21 нояб. 2022 г.

523 Репосты **76** цитат **7 107** отметок «Нравится» **186** закладок


Organ donations per million by year
total_deceased_donors_per_mil
0
10
20
30
40
2000
2005
2010
2015
year
highlight
Belgium
Croatia
Malta
Spain
United States of America
other

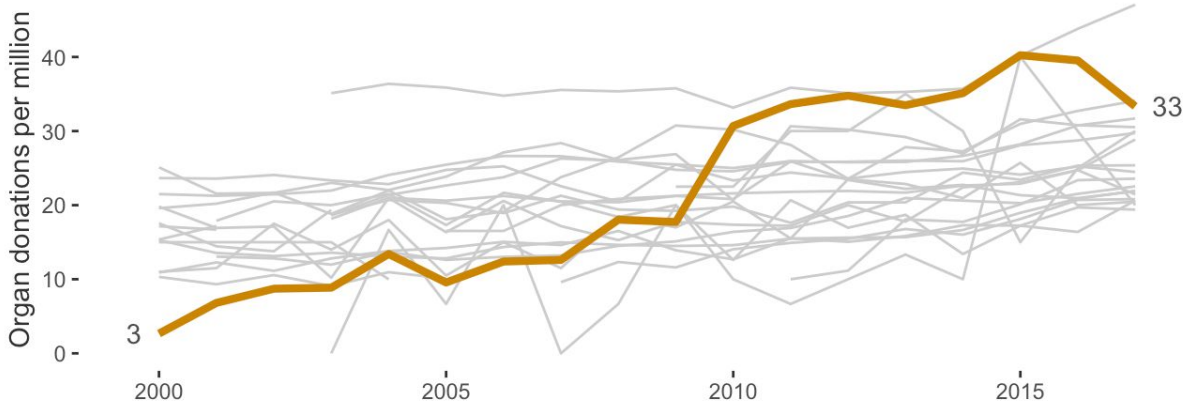
(a) **Spain** is the world leader in organ donations

Organ donations per million (for the top 20 countries in 2017)



(b) **Croatia** has dramatically increased their organ donation rates

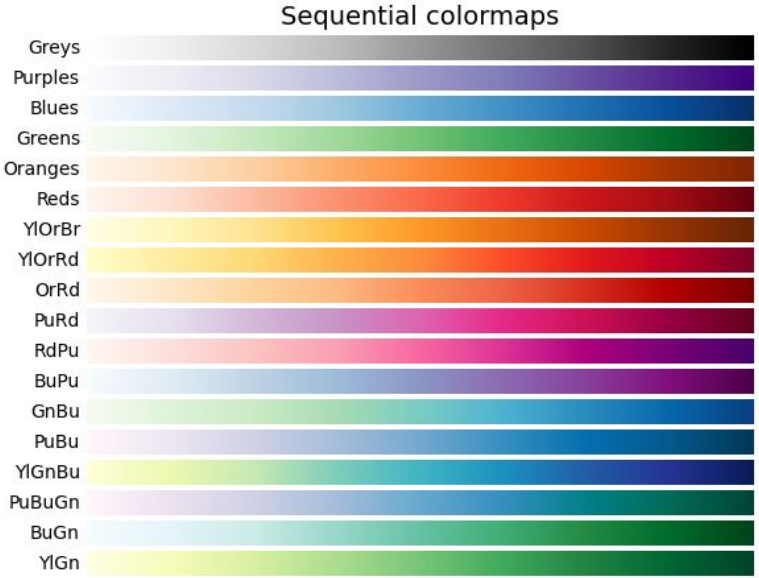
Organ donations per million (for the top 20 countries in 2017)



Color

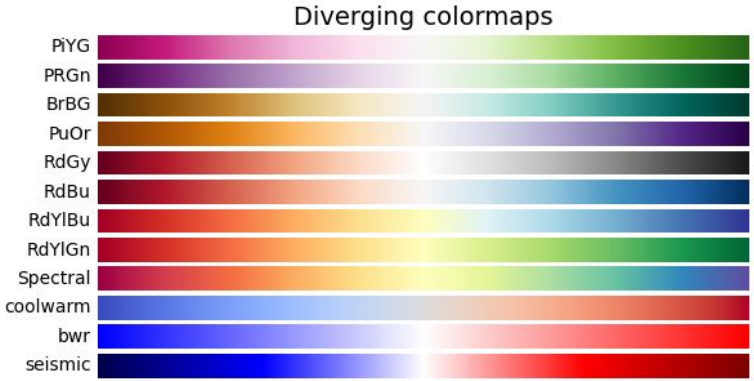
Continuous color schemes

✓

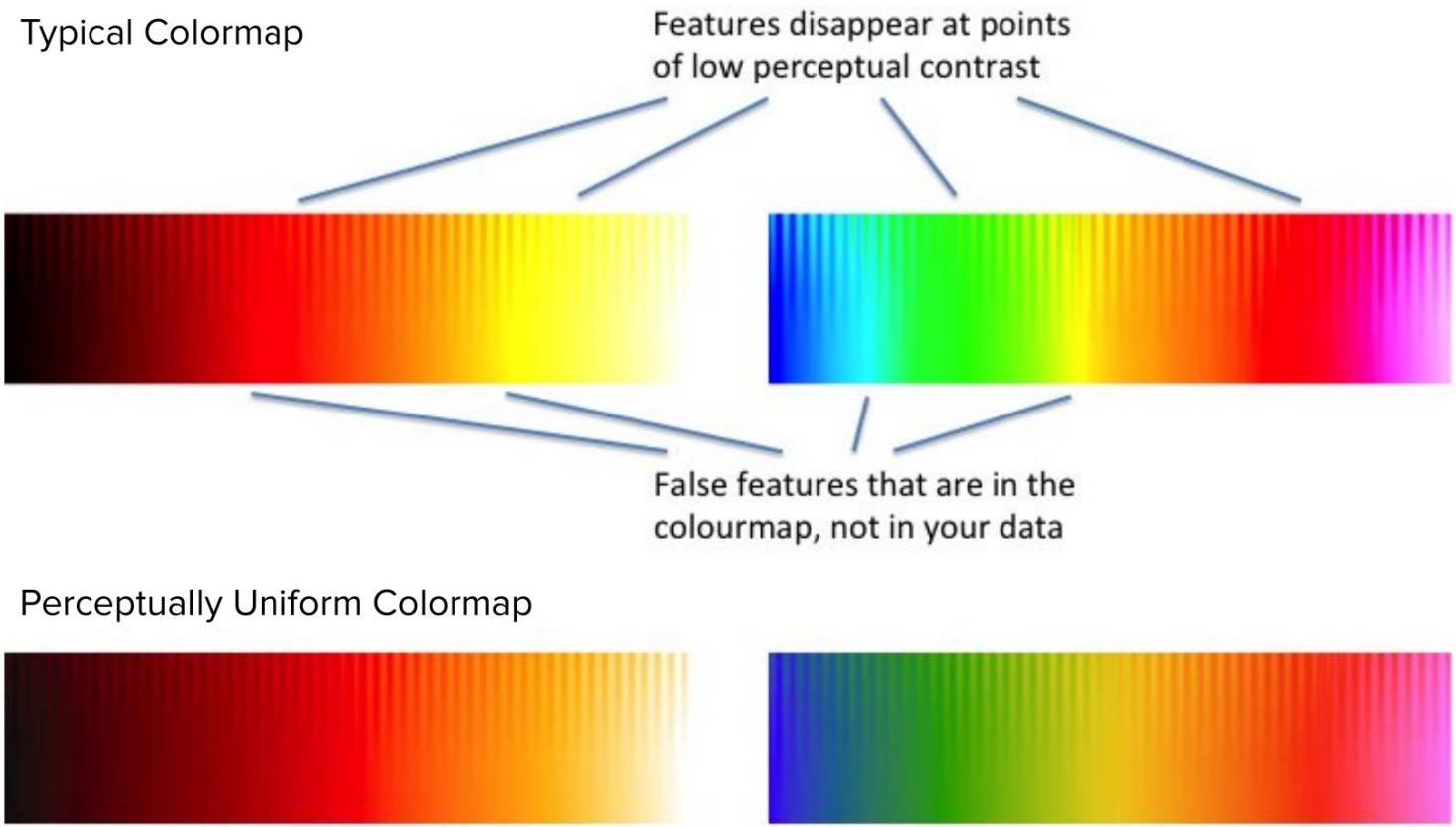


Not as good

✓(only if there is something special about the middle value)

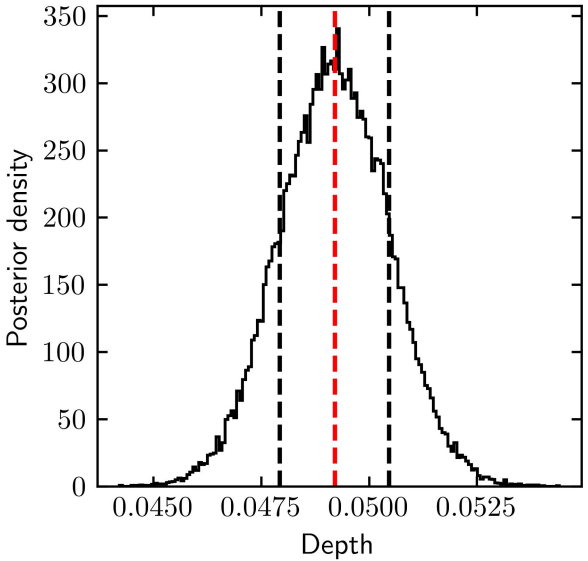


Color choice can lead to misleading visualizations



Other color tips

- It is hard to read a plot with more than about 6 colors. Be careful using discrete color schemes for categorical data
- Do not solely rely on color. Can distinguish plot elements using:
  - Different line styles (dashed, dotted, etc)
  - Different markers (circles, squares, triangles, etc)
  - etc.
- **If you don't need color, don't use it.**
- Using a single, bright color in a plot can highlight a key feature.
- Be intentional. Always use colors as a deliberate choice, not as a default. Are there colors which naturally represent what you are plotting? (e.g. green for vegetation)



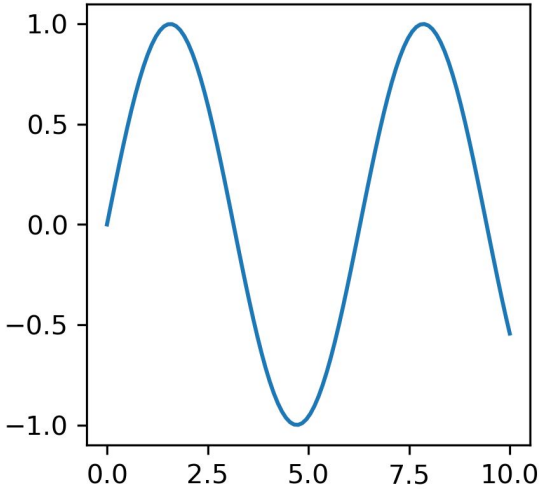
Tips

- A few information-rich figs >> many information-poor figs
- Think carefully about the color palette
- Avoid large amounts of whitespace—put a legend there or rescale
- The perfect figure needs no caption, or only a very short one
- Choose a nice font and use it consistently
- **Never use the default settings of a plotting library.** Default matplotlib or ggplot are extremely obvious and are a disappointment to see.
- Always export figures as vector graphics (pdf or svg) instead of rasters (png, jpg).
- Once the figure is finalized, it can be good to annotate it in external software.
- Use a legend if it helps understand the plot. Your caption shouldn't be a legend.
- Use transparency and sizing to avoid "overplotting" (points or lines covering each other)

Data Visualization

(in Python)

matplotlib




```
import matplotlib.pyplot as plt

x = np.linspace(0, 10, 100)

y = np.sin(x)

plt.plot(x, y)
```

Creates a plot object which lives somewhere in the Python environment

```
plt.title('my plot')
```

Works with the plot object

```
plt.show()
```

Displays the plot object and deletes it

For anything more complicated, refer to documentation and its examples, or look it up, or comment what you want then use copilot, or ask your favorite LLM, or ask me.

Good `matplotlib` defaults

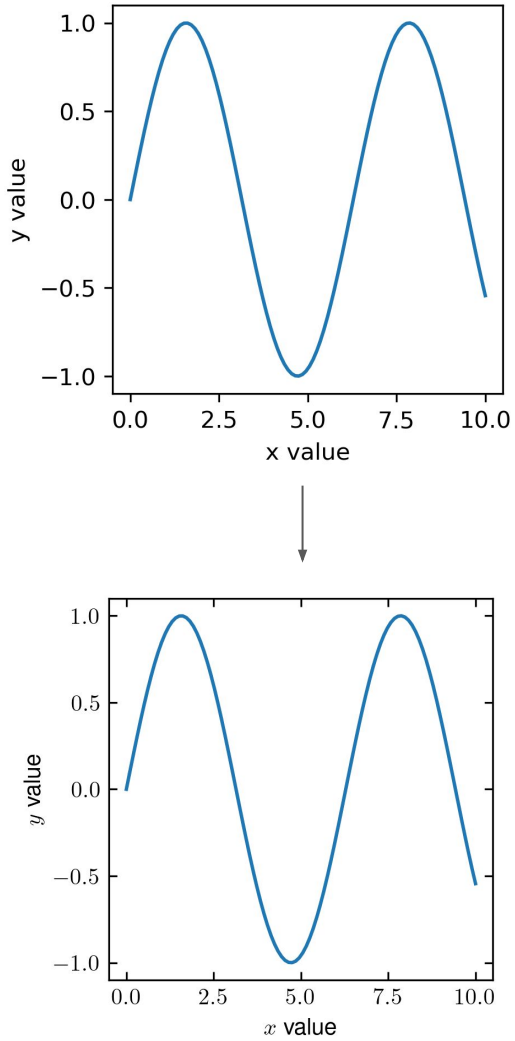
Add these lines before your plots to make them look more official:

Use TeX for rendering:

```
plt.rcParams['text.usetex'] = True
plt.rcParams['font.family'] =  'Helvetica'
```

Make ticks face inwards and appear on the top and bottom of plots:

```
plt.rcParams['xtick.direction'] = 'in'
plt.rcParams['ytick.direction'] = 'in'
plt.rcParams['xtick.top'] = True
plt.rcParams['ytick.right'] = True
```



Saving plots

```
plt.savefig('plotname.pdf')

plt.savefig('plotname.pgf')

plt.savefig('plotname.svg')
```

Run one of these **before** `plt.show()`

Don't use .png, .jpg, or any other raster formats! Use vector formats wherever possible. This will make the plot and text not blurry when you zoom in, and when you import into LaTeX the text in the plots will show up as real, selectable text.

(exception: if your plot has a very large number of points/lines (thousands), consider a .png)

`seaborn`

I am not an expert on seaborn. But it can do many painful matplotlib things in a painless way! It is especially good when you want to visualize >3 variables together.

It is built on top of matplotlib. I see it as just a more efficient way to make matplotlib graphics.

See stat-215-a-gsi/disc/week3/dataviz.ipynb (or .html) for examples

Other tools

Feel free to use whatever. For example:

- Plotly (for interactive or 3D plots)
- Cartopy (for maps)
- Seaborn
- Altair (??)
- Plotnine (for ggplot-like interface)
- ???
- R

I don't care what you use!
Just make sure the figures are simple, informative, honest, efficient, and aesthetic.

For R, feel free to explore the data visualization materials in the 2023 stat-215-a GSI repo:
https://github.com/cz-ye/stat-215-a-gsi

Data Visualization in context

Questions to ask yourself

- What visualizations will help me:
  - Understand more about the data
  - Understand more about the real world
  - Figure out how to solve the domain problem
- If I change how I visualize this relationship, does my interpretation change? **Stability.**
  - Changing scales, color schemes, and the type of visualization
- If I make a large number of visualizations, and only focus on the one with the most interesting result, could that result be due to noise?
- (reality check) Does the result in the figure correspond with:
  - My domain knowledge?
  - Other data which exist? (easy mode: what do published papers say about the result, or similar results?). If they agree with me, then my result has some **predictive** power.