

Data Visualization and Python Plotting Libraries

Follow along

```
conda create -n scivis python=3.9
conda activate scivis
conda install numpy pandas matplotlib seaborn bokeh
conda install -c plotly plotly=5.7.0
conda install -c conda-forge cmocean
conda install "jupyterlab>=3" "ipywidgets>=7.6"
conda install -c conda-forge -c plotly jupyter-dash
```

GitHub QR



STARFORGE volume renderings

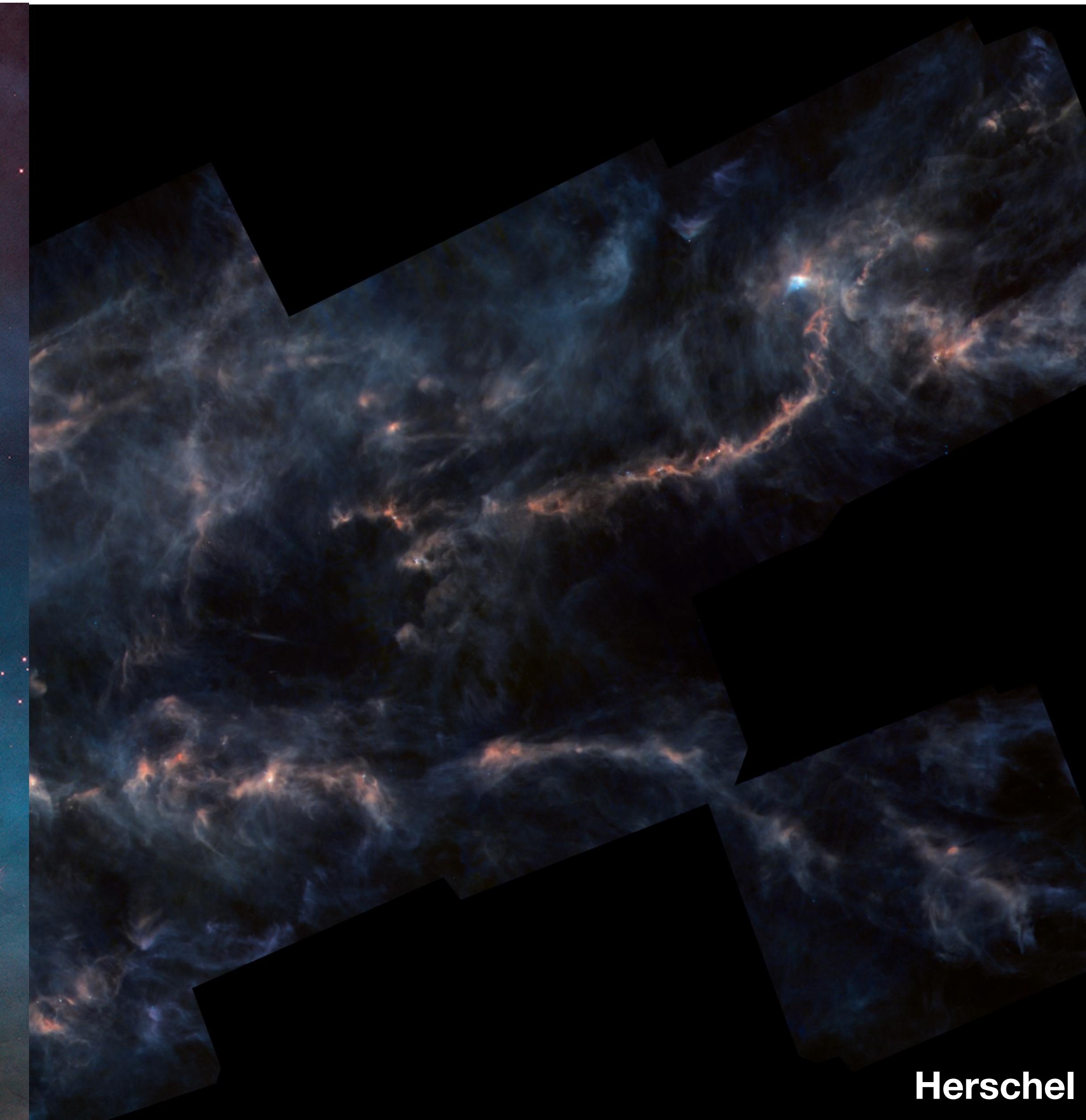


<https://www.starforge.space/movies.html>

Visualizations Matter



NASA/ESA/STScI/AURA



Herschel

Resources for Better Graphs

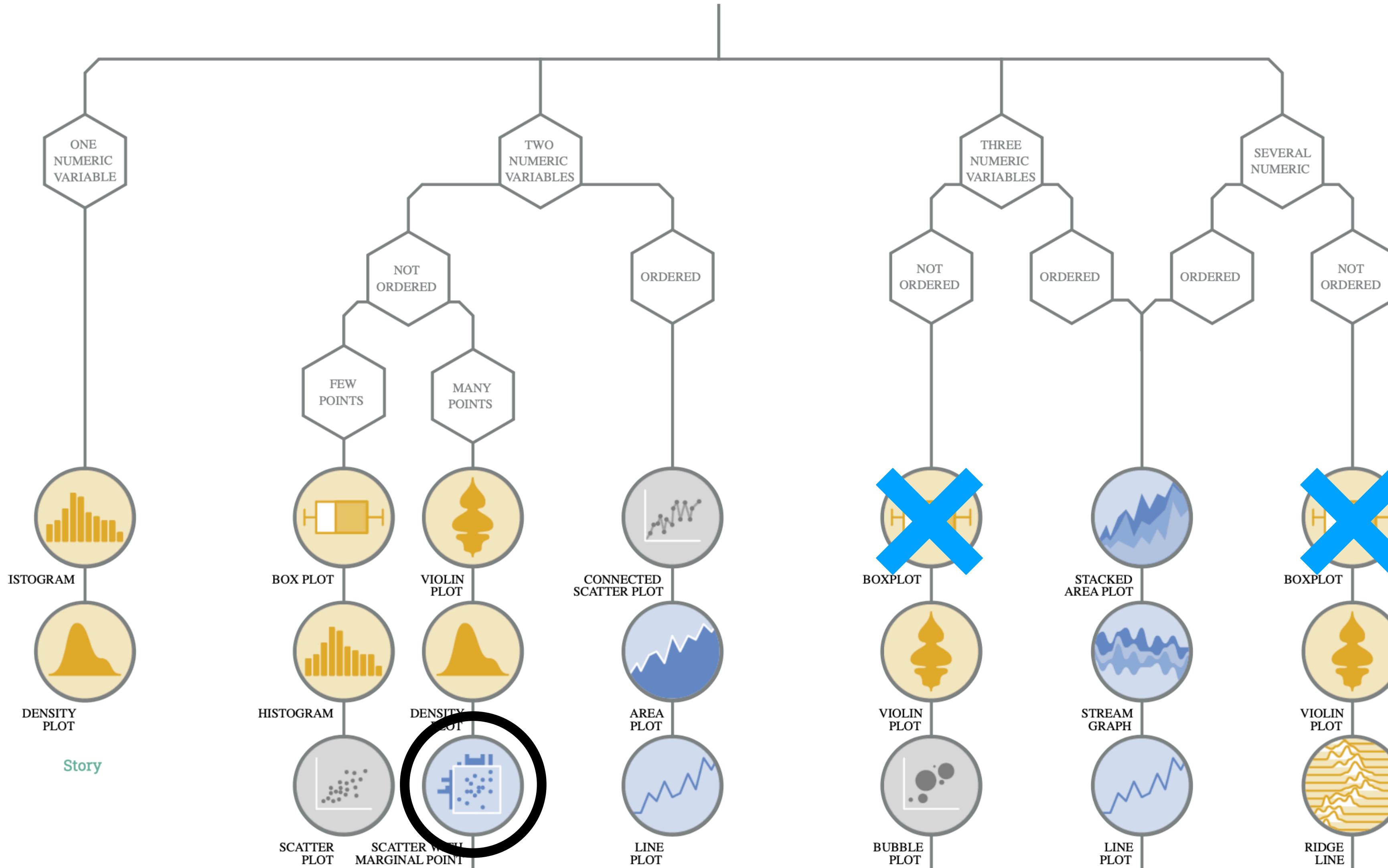
Color resources

- Colorbrewer: <https://colorbrewer2.org/> . Provides premade lists of hex color strings for a wide variety of purposes. Many colormaps already built into python packages
- MyColor: <https://mycolor.space> . A simple website where you provide one color and it provides a large list of 2 - 4 color spaces. Useful for plotting categorical data.
- Hclwizard: <https://hclwizard.org> . A palette creator following certain color schemes
- Adobe colors: <https://color.adobe.com/create/color-wheel> . Palette creator from adobe

Choose a proper visualisation

<https://www.data-to-viz.com>

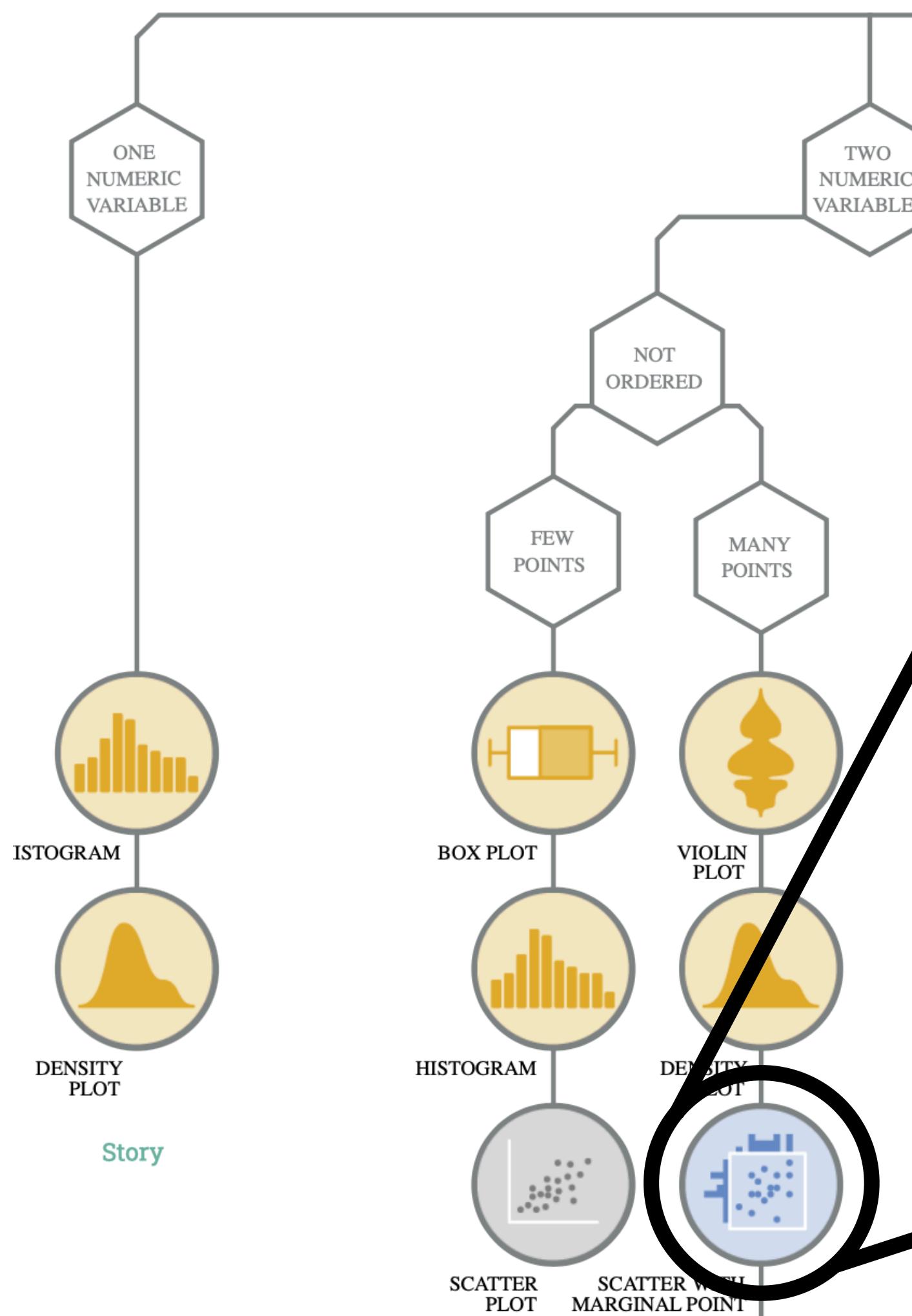
Numeric Categoric Num & Cat Maps Network Time series



Choose a proper visualisation

<https://www.data-to-viz.com>

Numeric Categoric

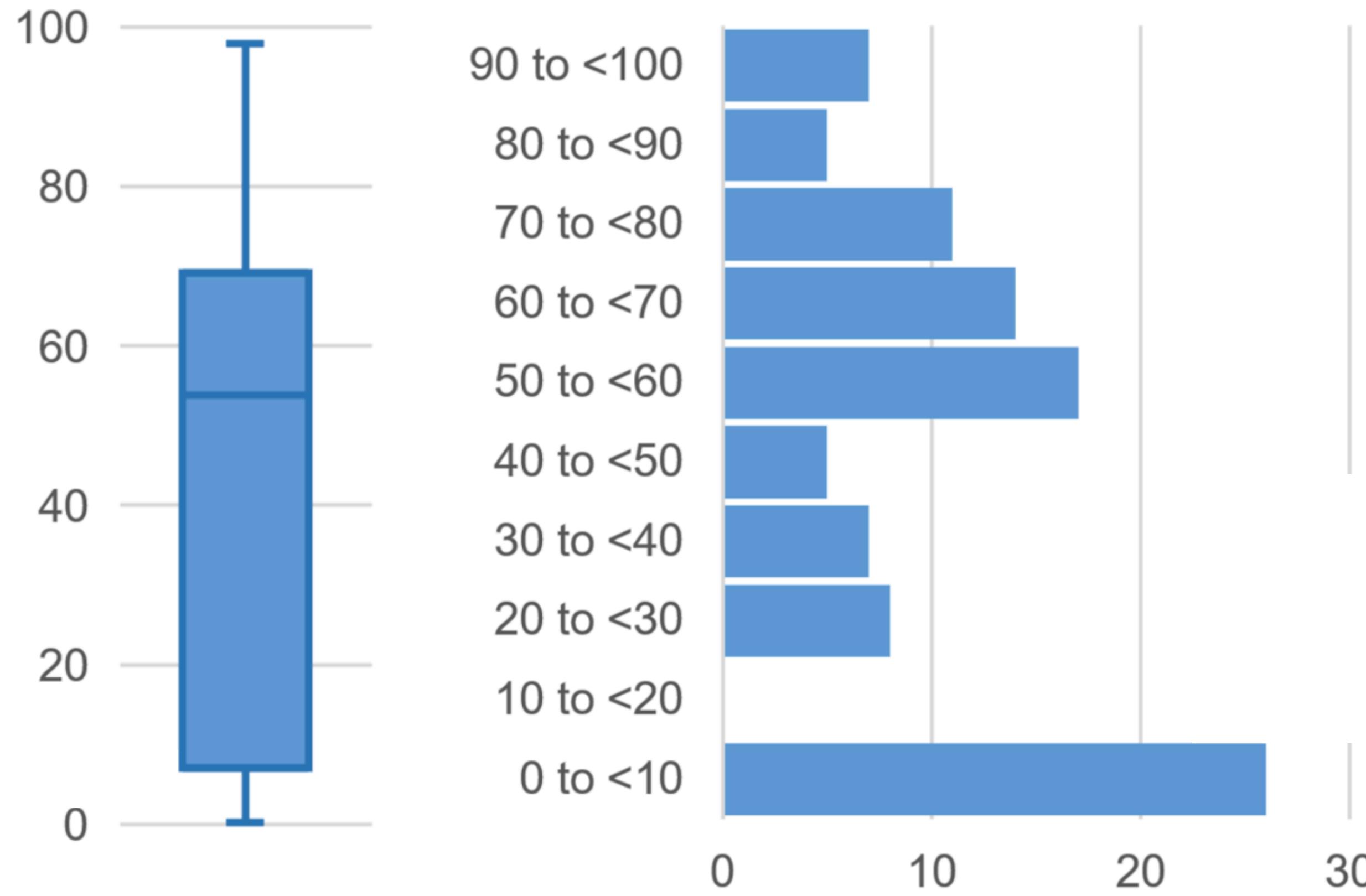


Scatterplot



A [scatter plot](#) displays the relationship between 2 numeric variables. Each data point is represented as a circle. Several tools allow to build one in python, this section provides code samples for [Seaborn](#), [Matplotlib](#) and [Plotly](#) for interactive versions. Note that this [online course](#) has a chapter dedicated to scatterplots.

Useful aside: Why box plots are bad

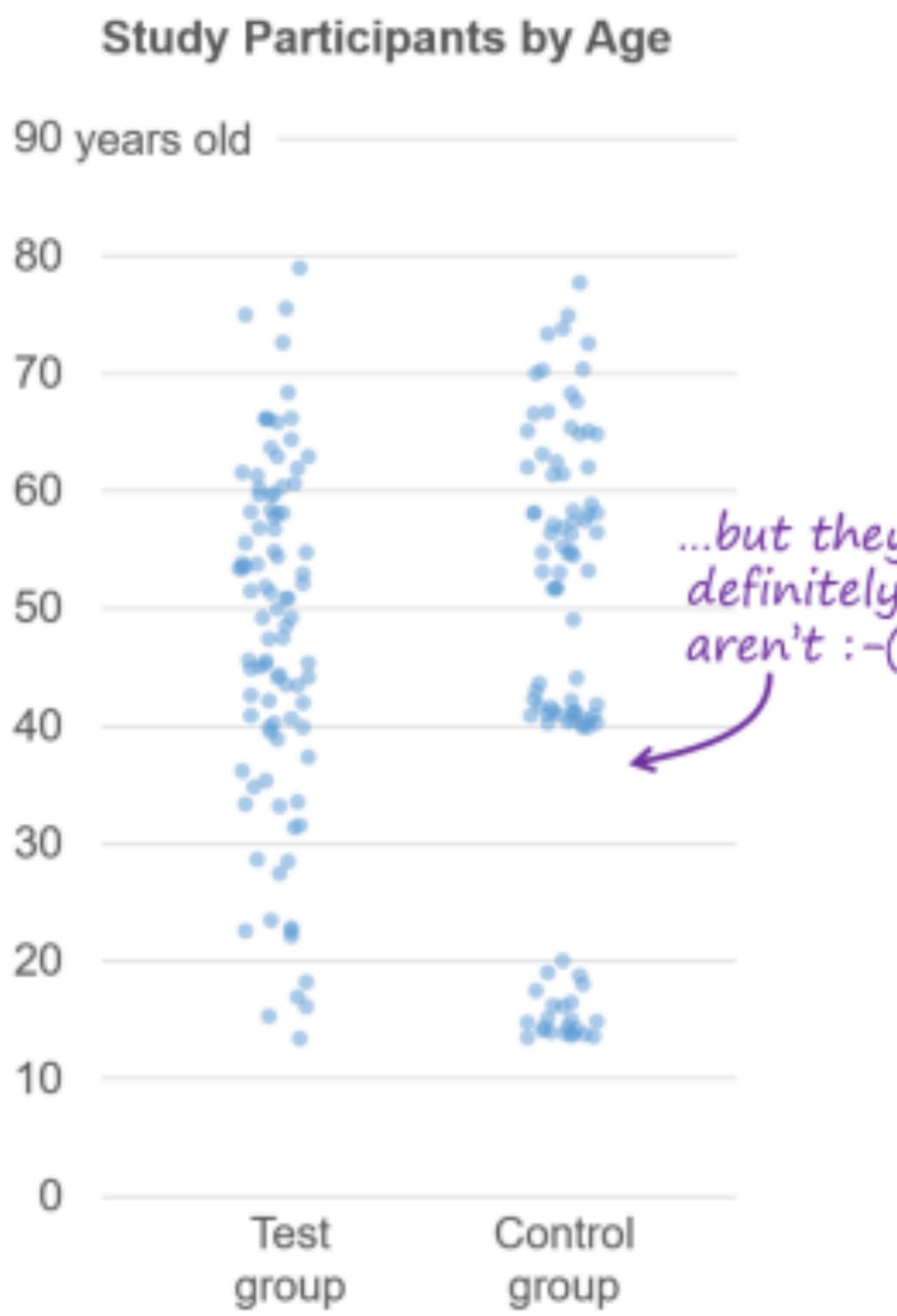
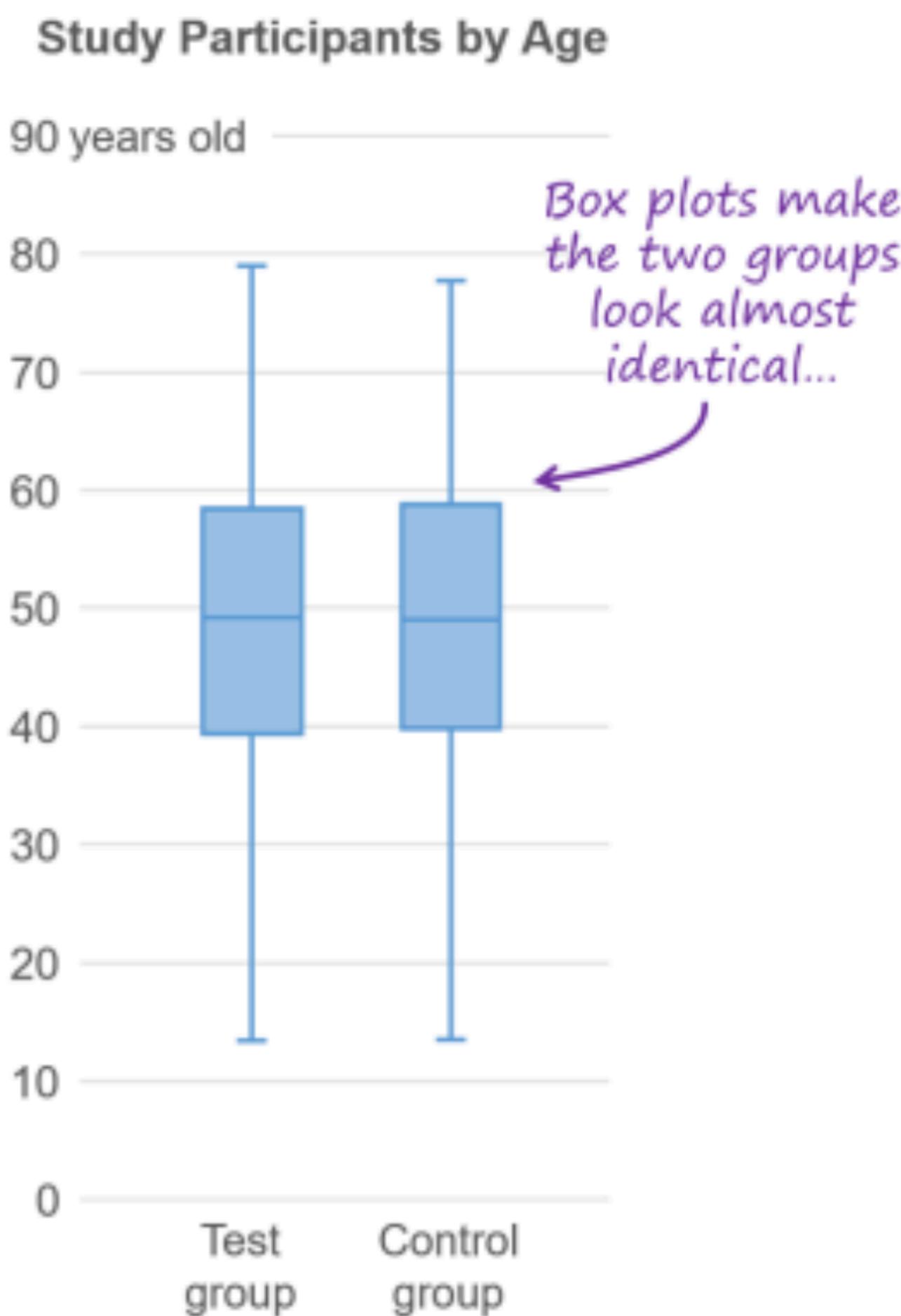


Box plots can hide useful information. In the left, a significant amount of the distribution is hidden in the “small” separation between box and whisker. Thus, it can hide the real distribution.

Replace with:

1. Violin plot
2. Shaded/area differenced boxes in vertical bins
3. Jittered strip plots

Useful aside: Why box plots are bad



Box plots can hide useful information. In the left, a significant amount of the distribution is hidden in the “small” separation between box and whisker. Thus, it can hide the real distribution.

Replace with:

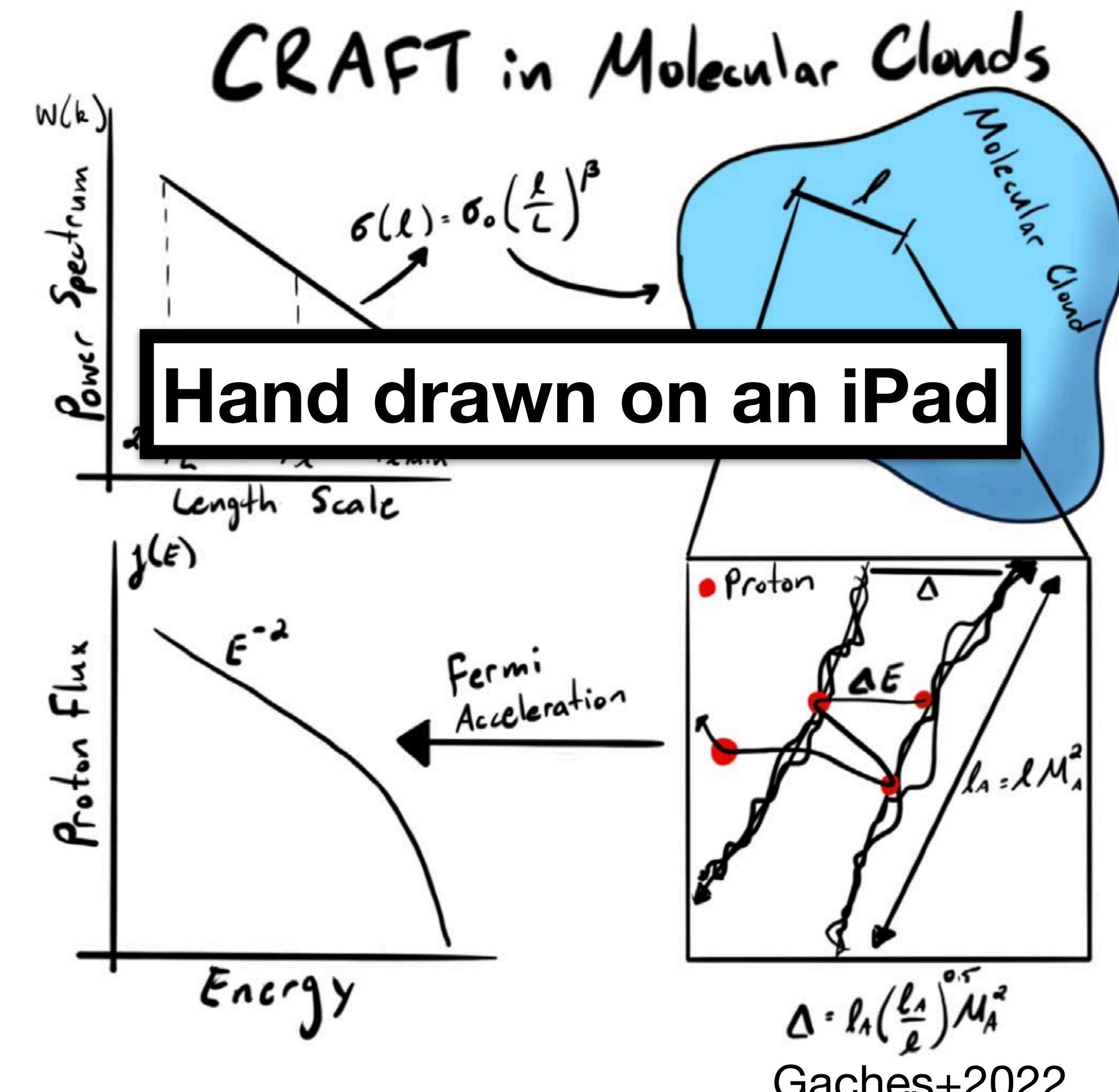
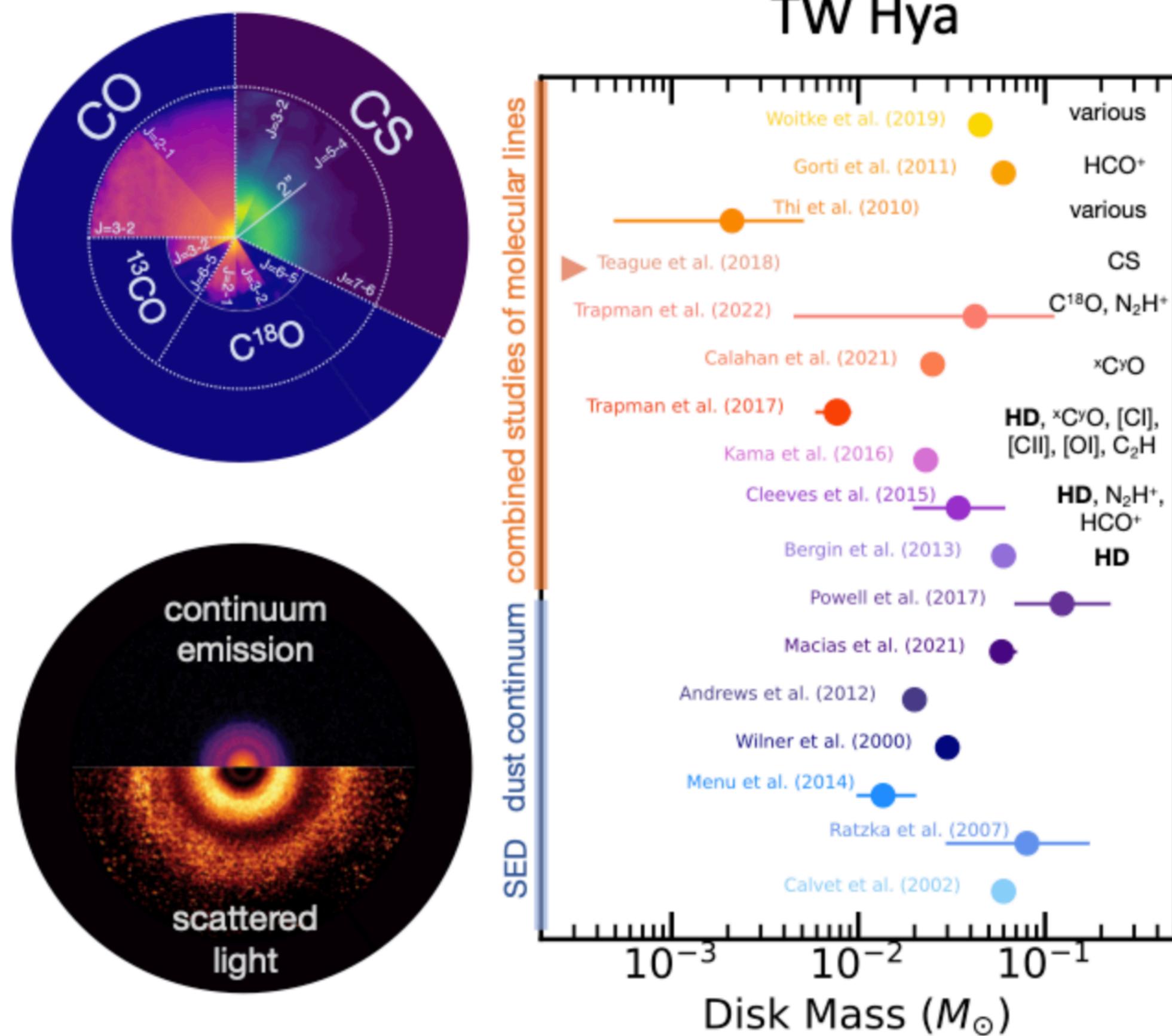
1. Violin plot
2. Shaded/area differenced boxes in vertical bins
3. Jittered strip plots

Guidelines for Clear Graphs

1. First step should always be ask: “What is the main takeaway for the figure?”
2. All text should be readily readable on the page.
 1. Aim for the axis labels to be the same size as the in-page text
3. If using categorical data, or data from multiple sources, reduce the number of point types to as few as possible. Consider grouping sources, and use captions for citations.
4. Use an appropriate color scheme.
5. Share it with others, and ask them what they think the figure shows! Sometimes what you think really shows a point clearly is very confusing to an outsider.

Don't be afraid to use schematics!

Hand drawn or vector schematics (can even make in keynote or ppt!) can be very useful for conveying the major point of your paper. Don't be afraid to spend considerable time for a good summary figure for a paper!



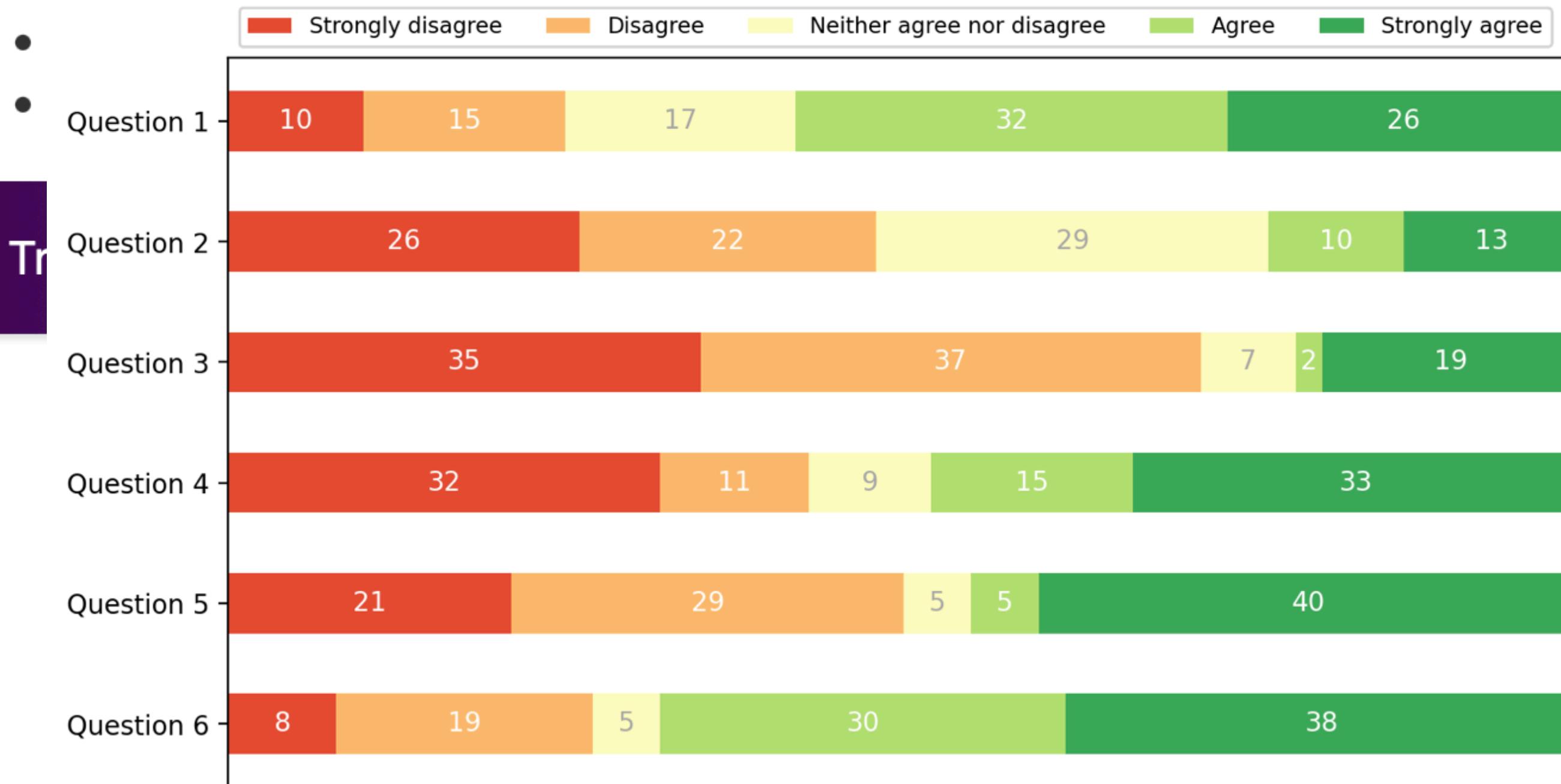
Python packages

Matplotlib/Pyplot

Matplotlib: Visualization with Python

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible.

- Create [publication quality plots](#).
- Make [interactive figures](#) that can zoom, pan, update.
- Customize [visual style and layout](#).
- Export to [many file formats](#)



Benefits:

- Very easy to use
- Commonly used - can easily find user solutions for anything
- Low level: you can highly customise almost every aspect of the plot

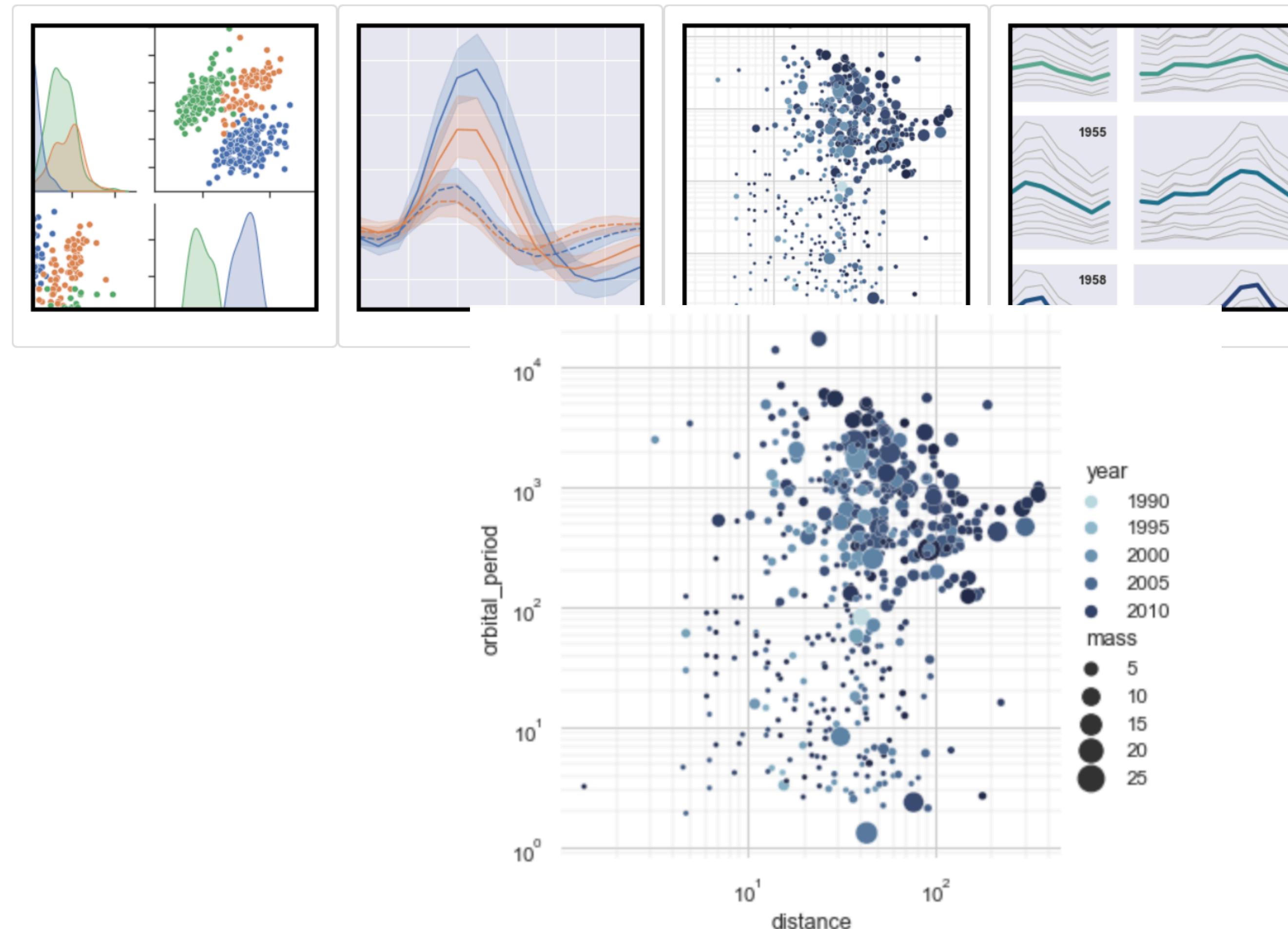
Cons:

- Very low level: to build complex plots takes a lot of time!
- Hard to make interactive
- Does not natively interface with pandas
- Bad for large data sets

Python packages

Seaborn

seaborn: statistical data visualization



Benefits:

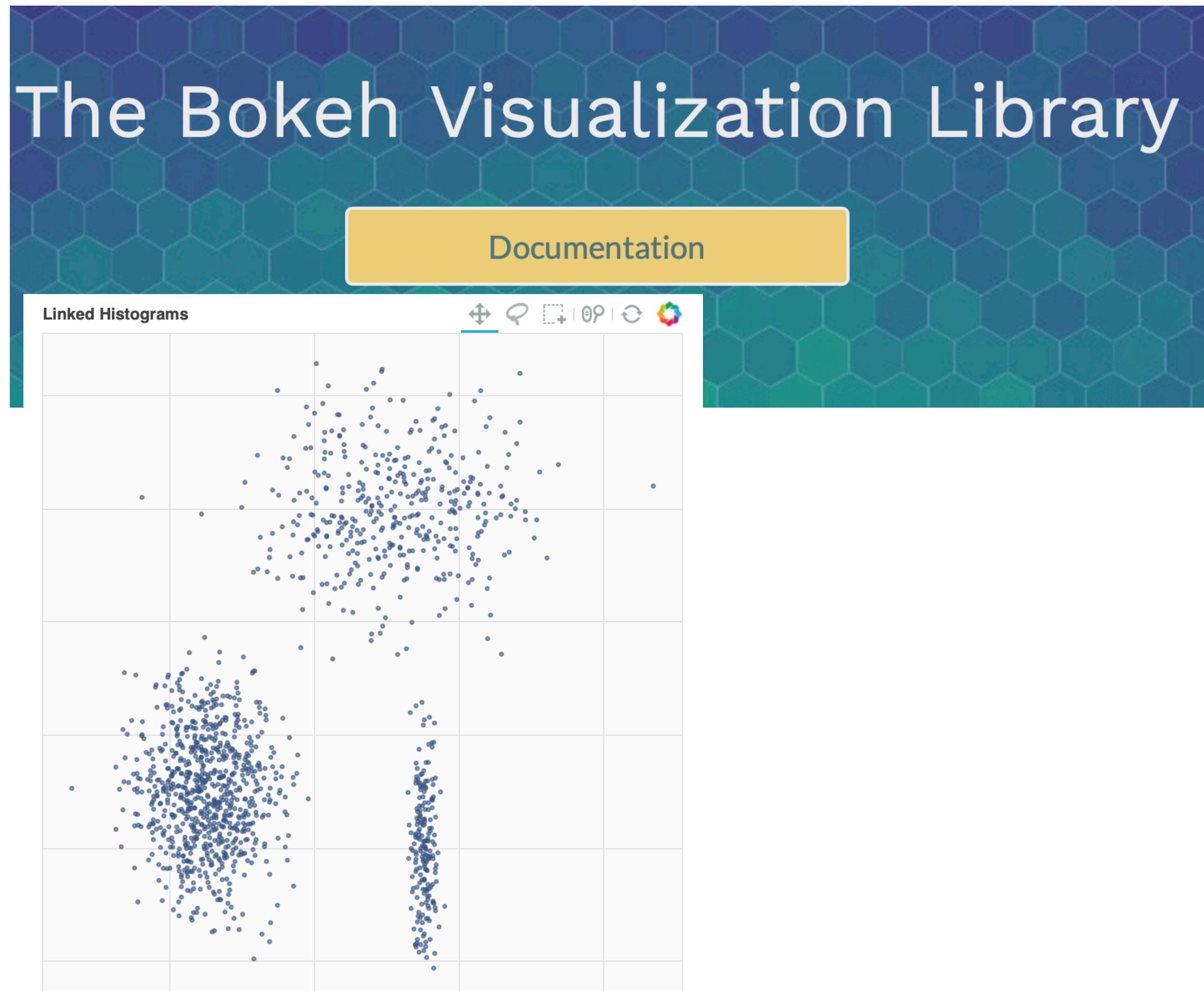
- Built on top of matplotlib: advanced users can directly access plots to customise
- Very high level: can build complex plots and even do regression in one line!
- Interfaces with pandas directly

Cons:

- Customising plots can be tedious
- Can not easily built subplots
- Not interactive

Python packages

Bokeh



Benefits:

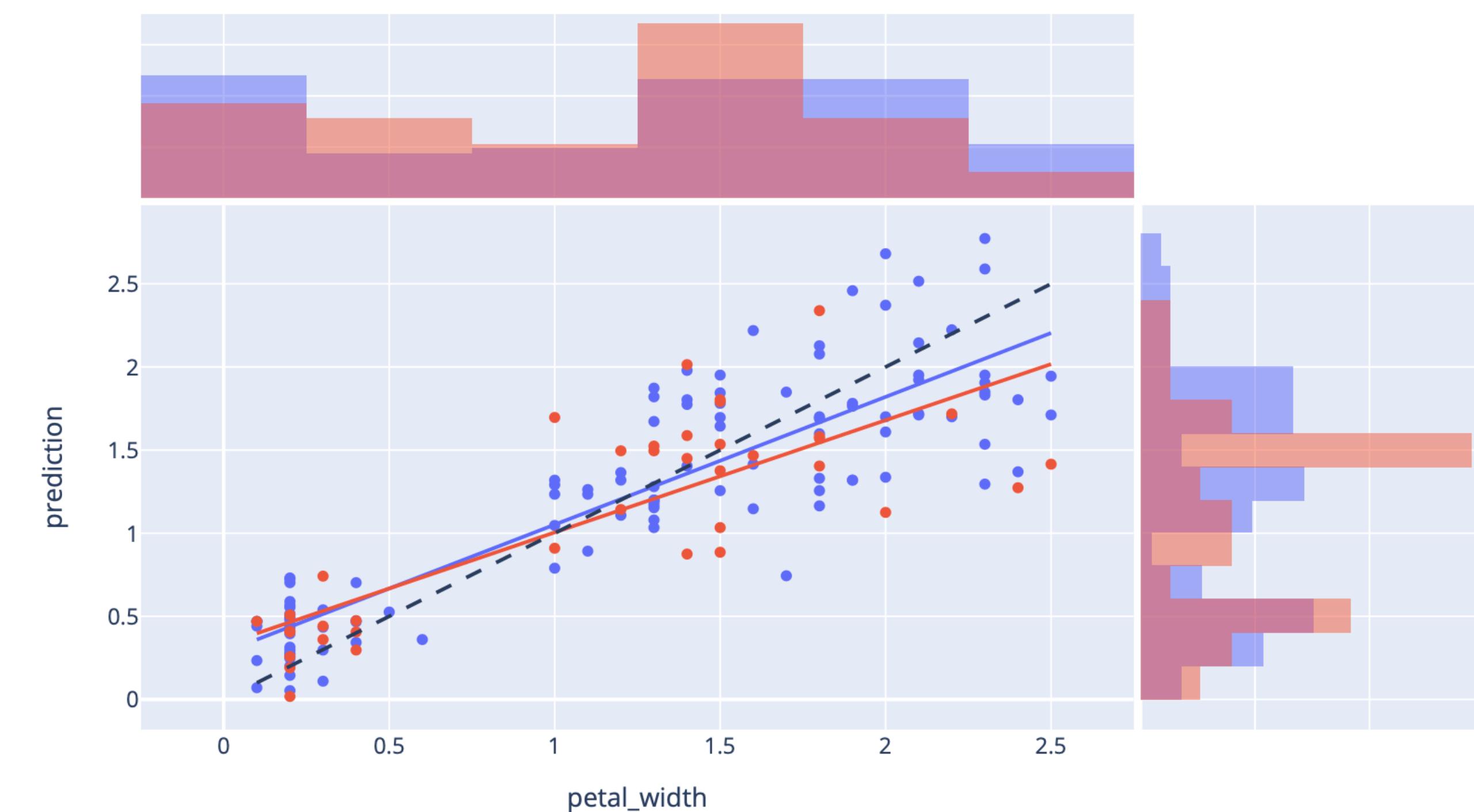
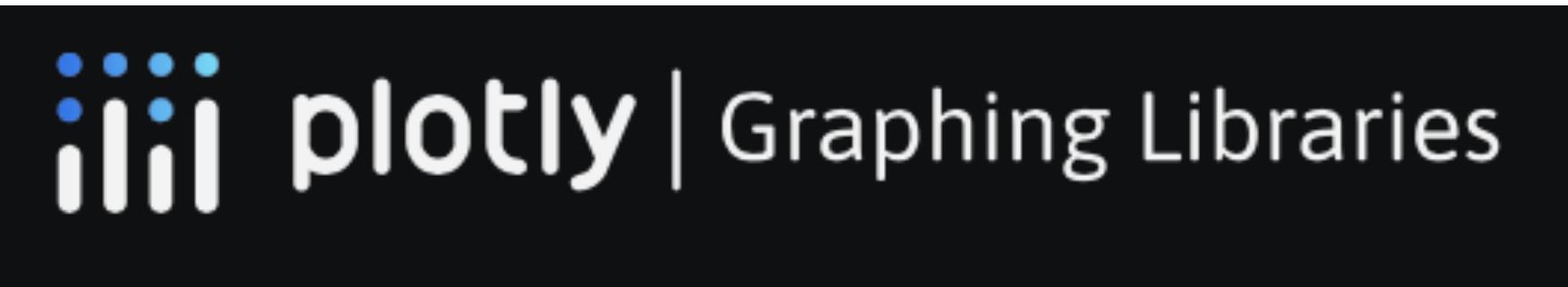
- Built on javascript: interfaces easily into Jupyter notebooks
- Interactive
- Interfaces directly with pandas
- Can export interactive plots
- Can built visualisation boards

Cons:

- Not as commonly used: not as much support available
- Have to built plots from “scratch” (e.g. each point type is it’s own function)

Python packages

Plotly/python



Benefits:

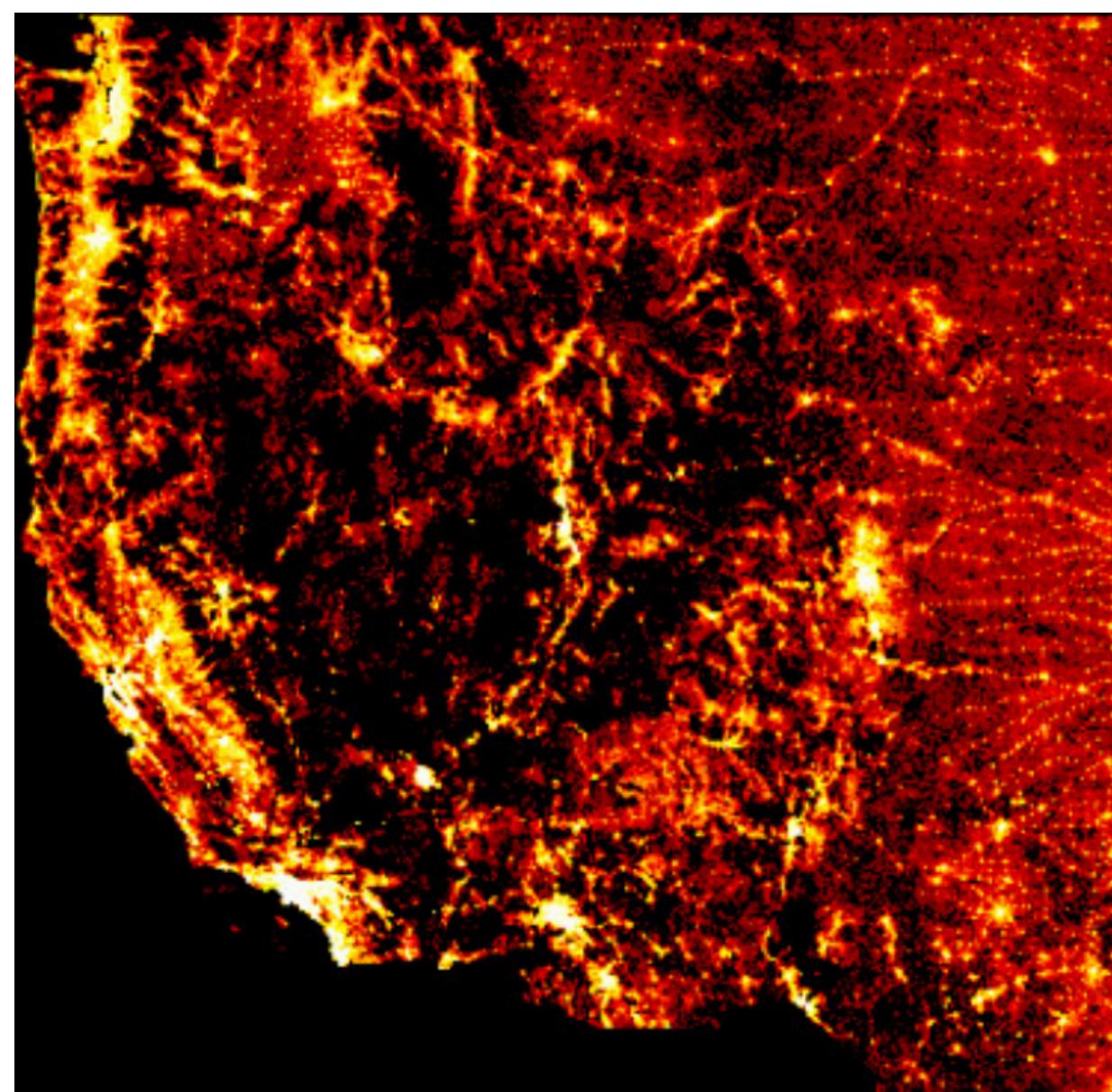
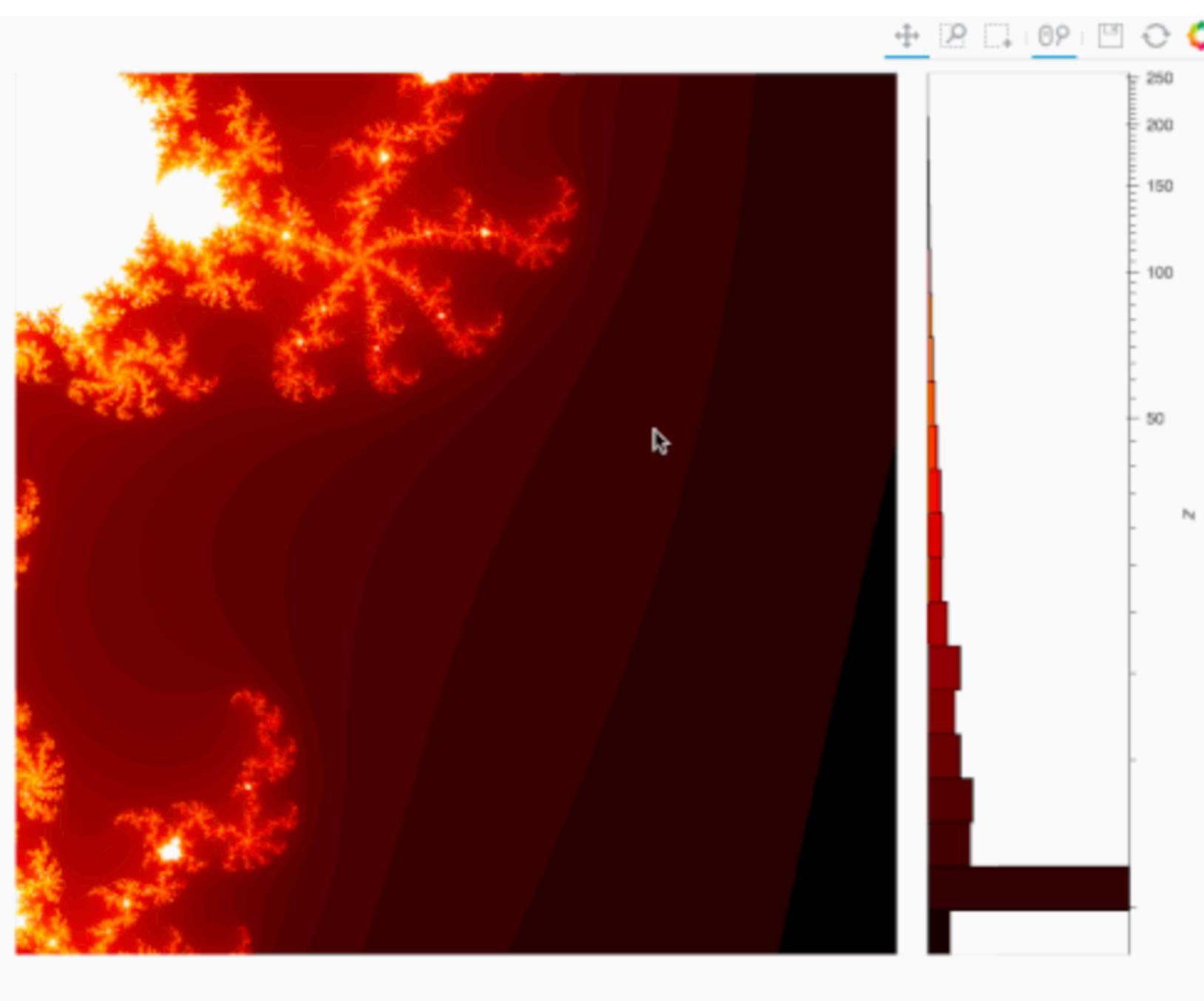
- Interactive
- Interfaces directly with pandas
- Can easily build visualisations
- Have **many** plots built in natively
- Comes with dash to built web-based apps

Cons:

- Not as commonly used: not as much support available
- Not as useful for making publication pdfs. Journals are only just starting to accept interactive figures

Python packages

Holoviews+Datashader



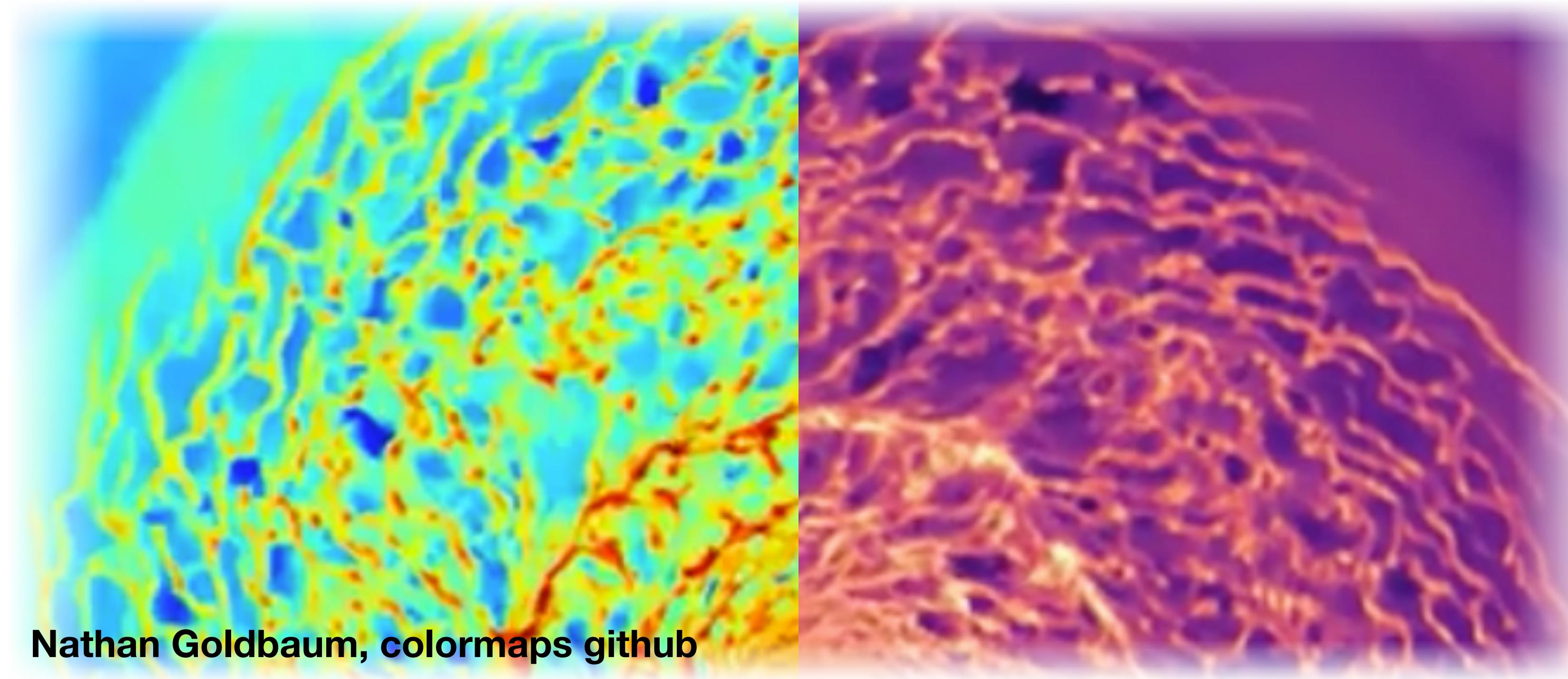
Benefits:

- Interactive
- Can deal with very large data sets
- Can swap between different backend libraries
- Build up plots through “algebra”: define the data, then easily swap between plots

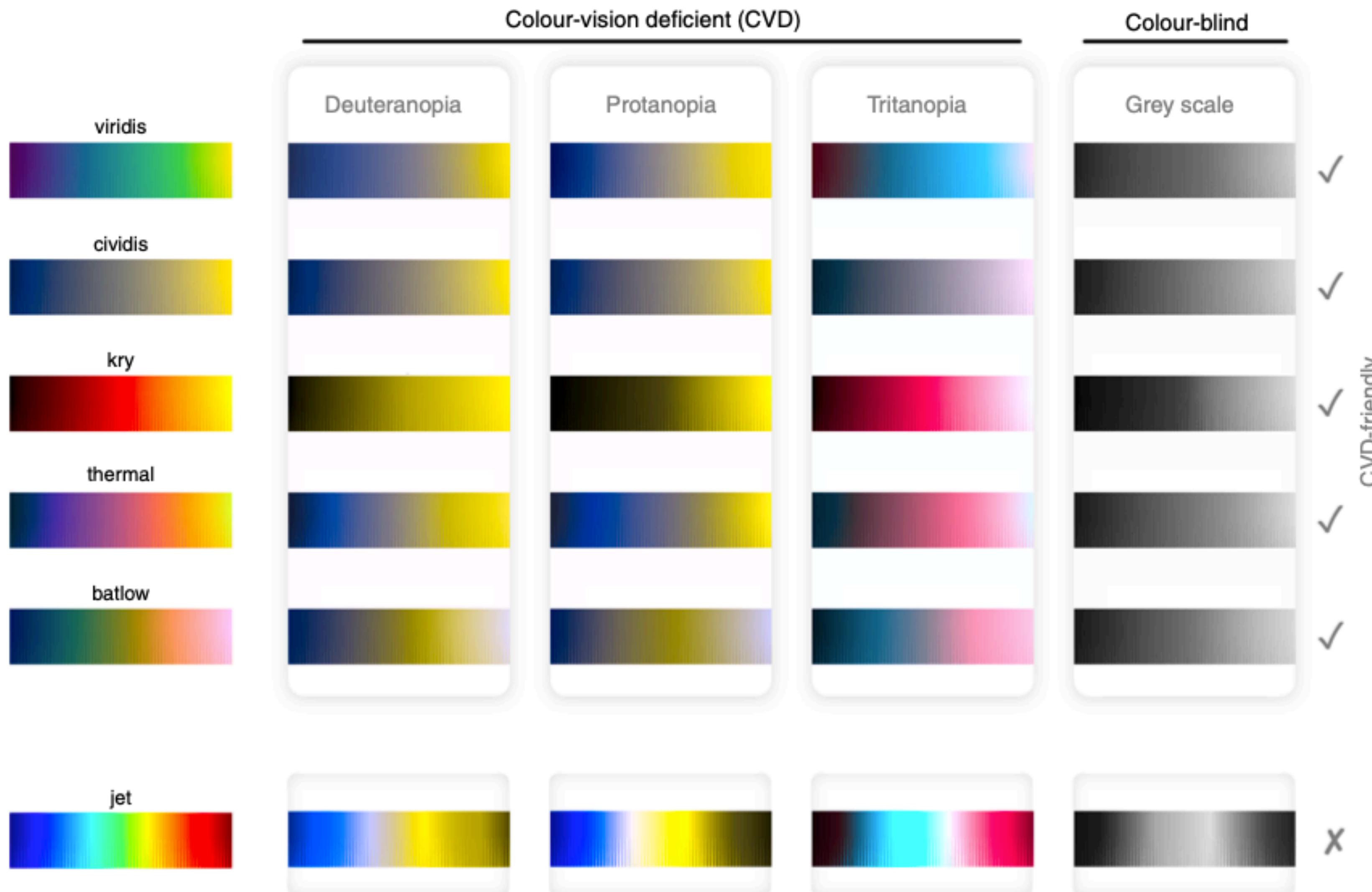
Cons:

- Not always particularly intuitive or easy to get the publication plots you want

Choosing the best **colors** for you and your audience

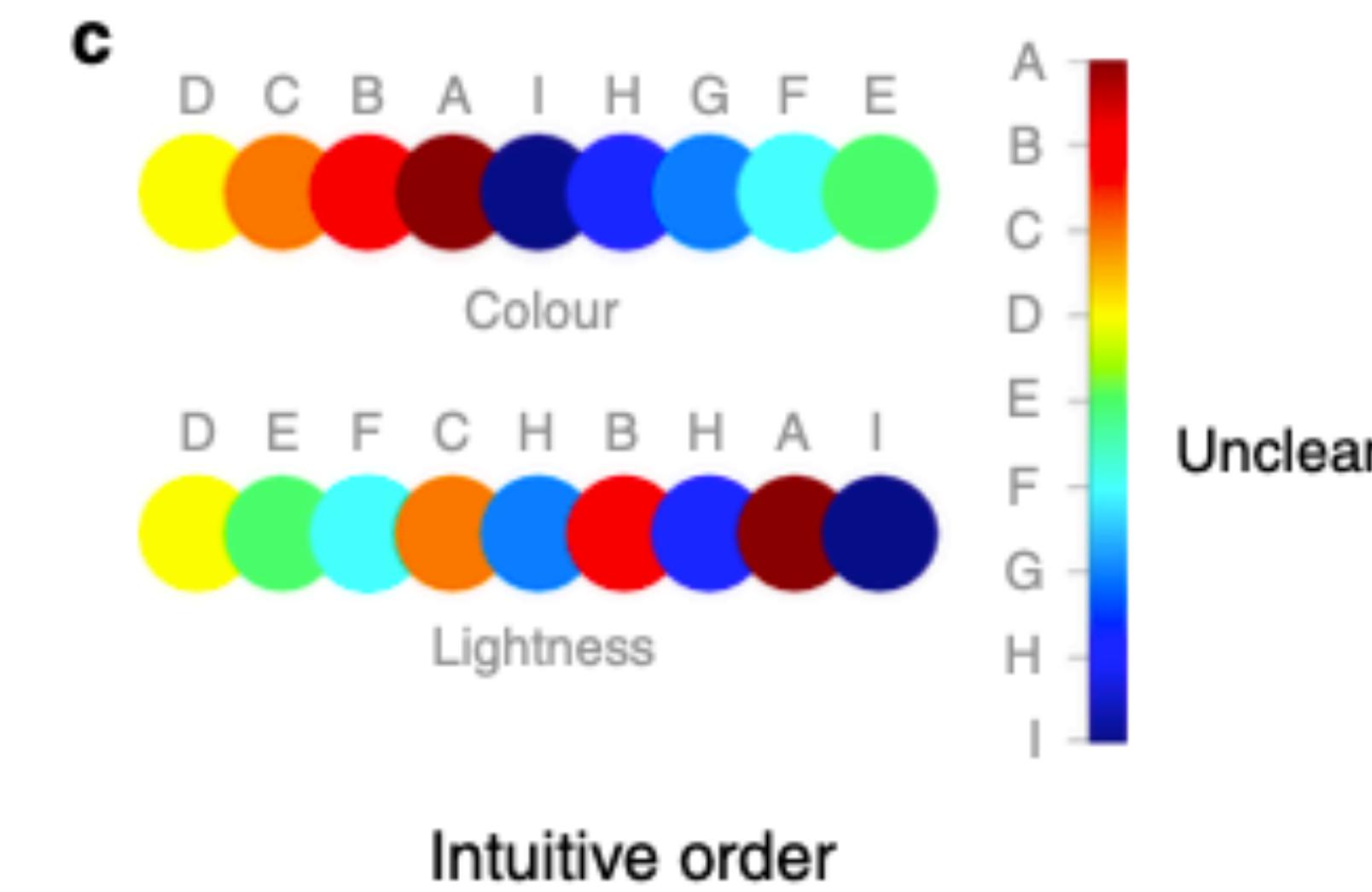
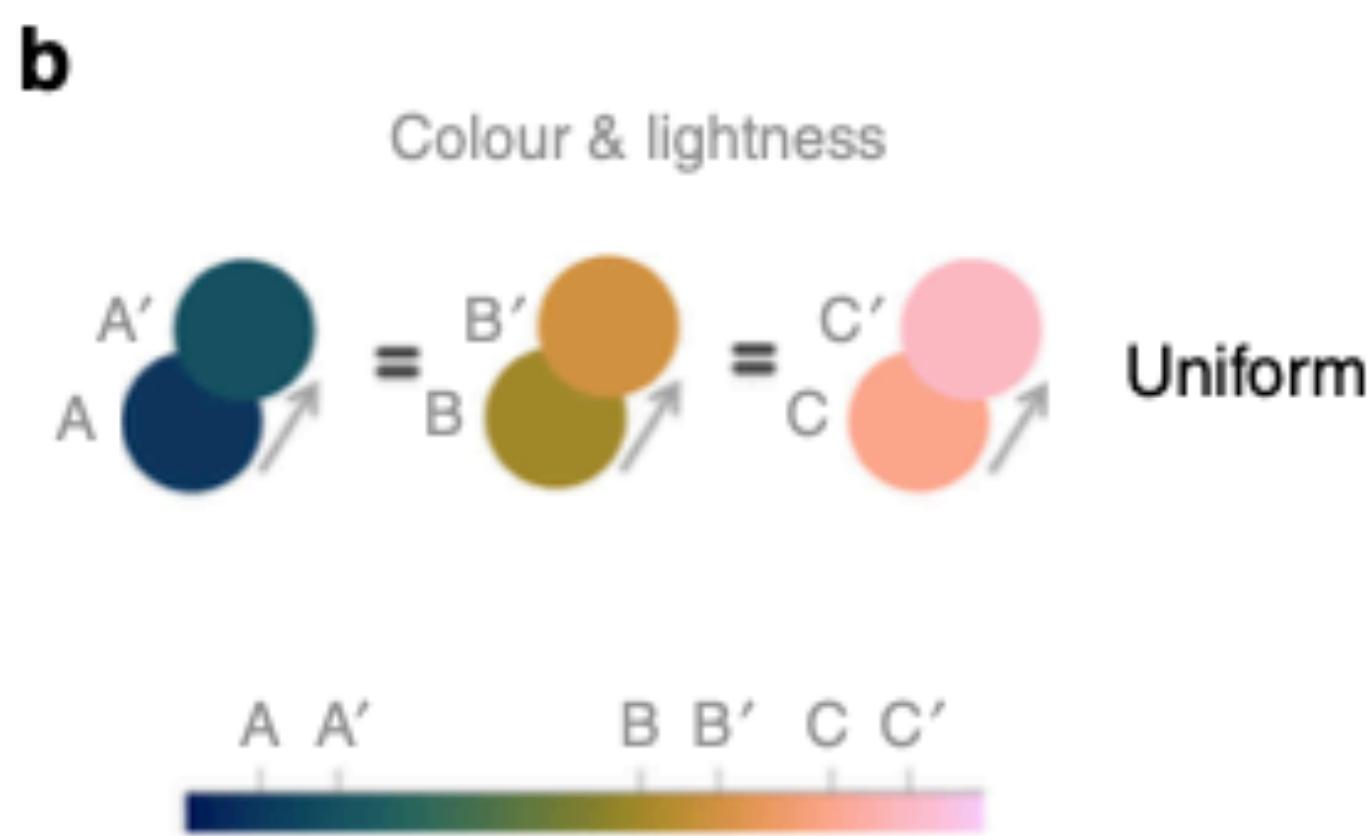
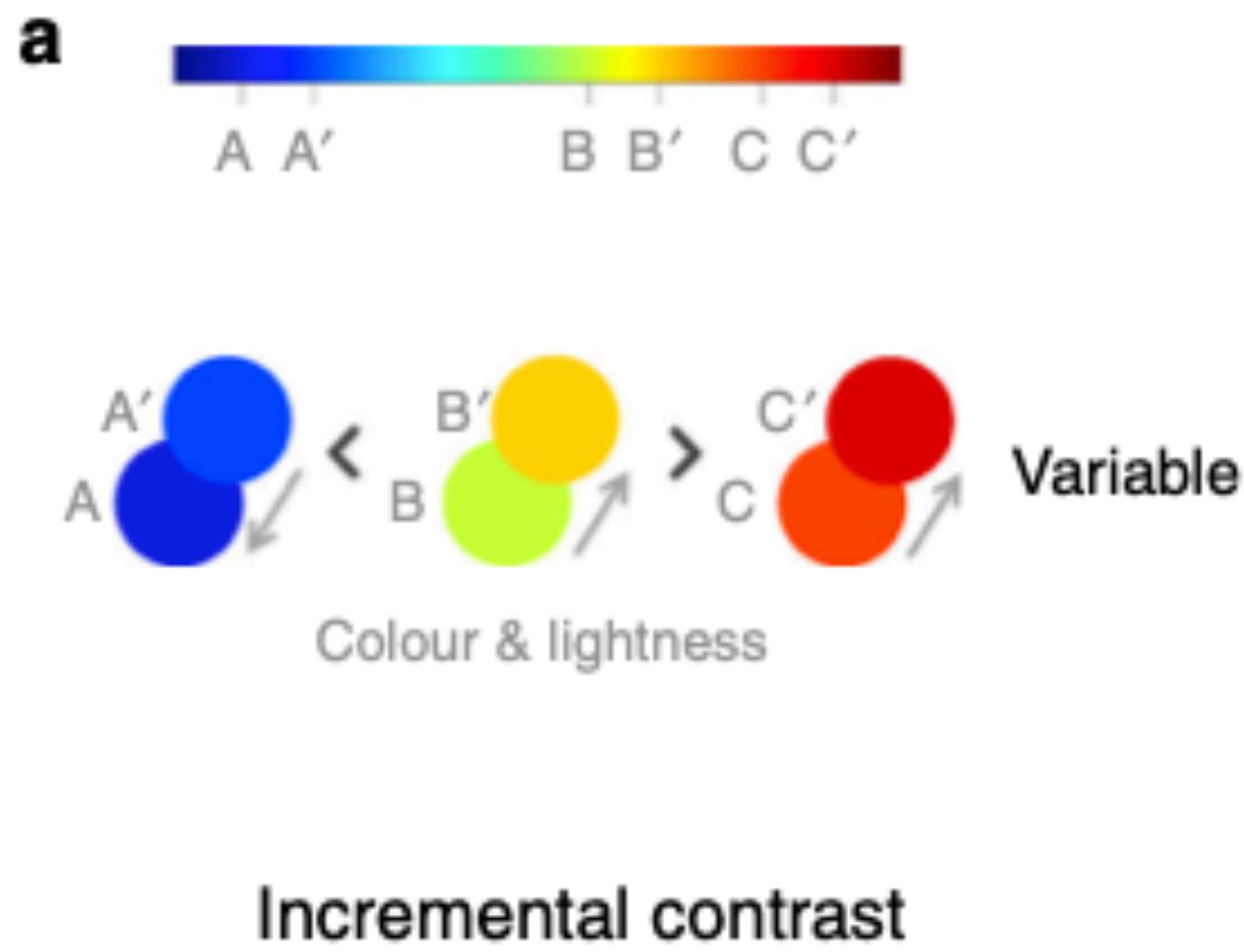


Colorblind-proof your plots!



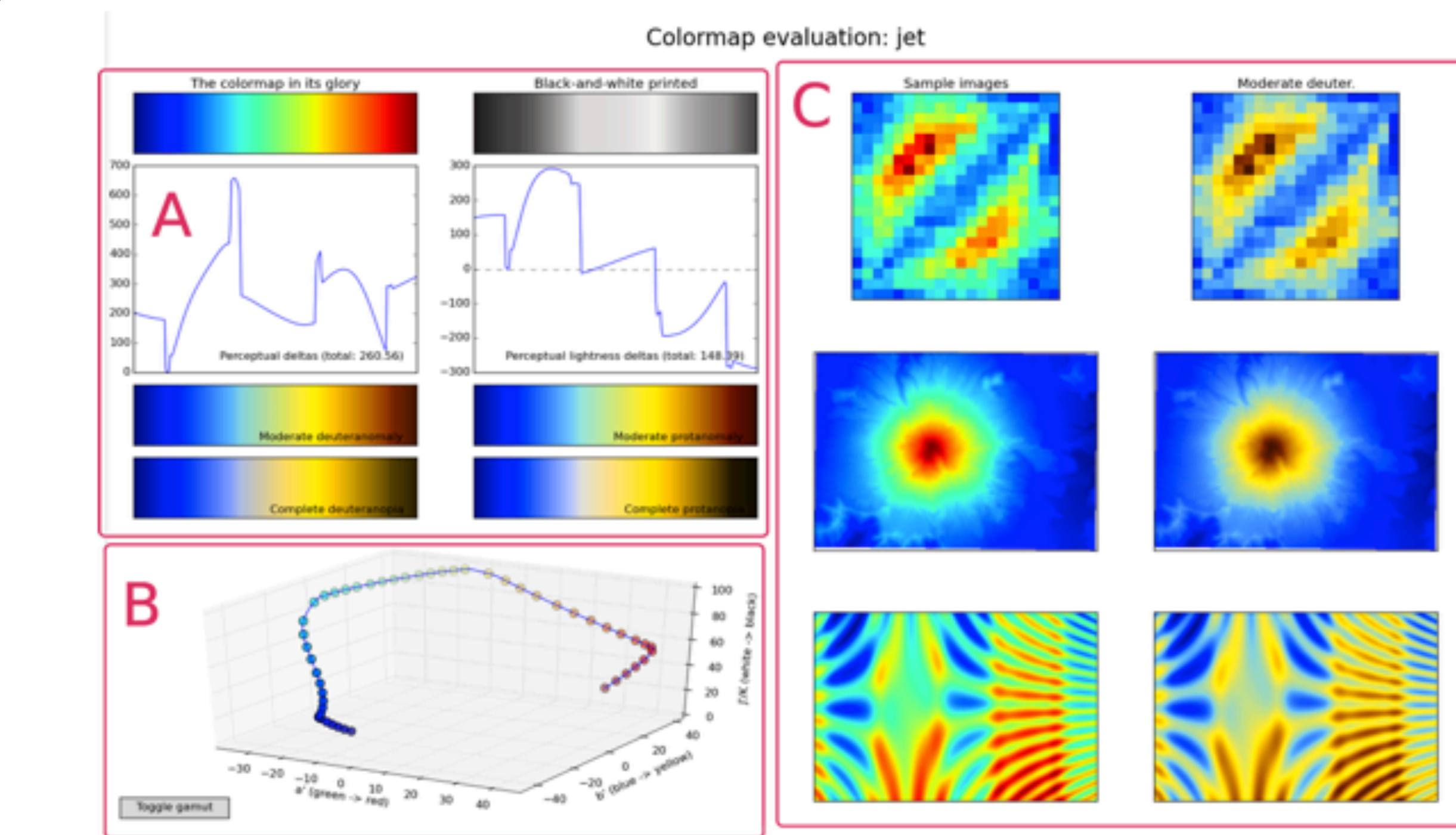
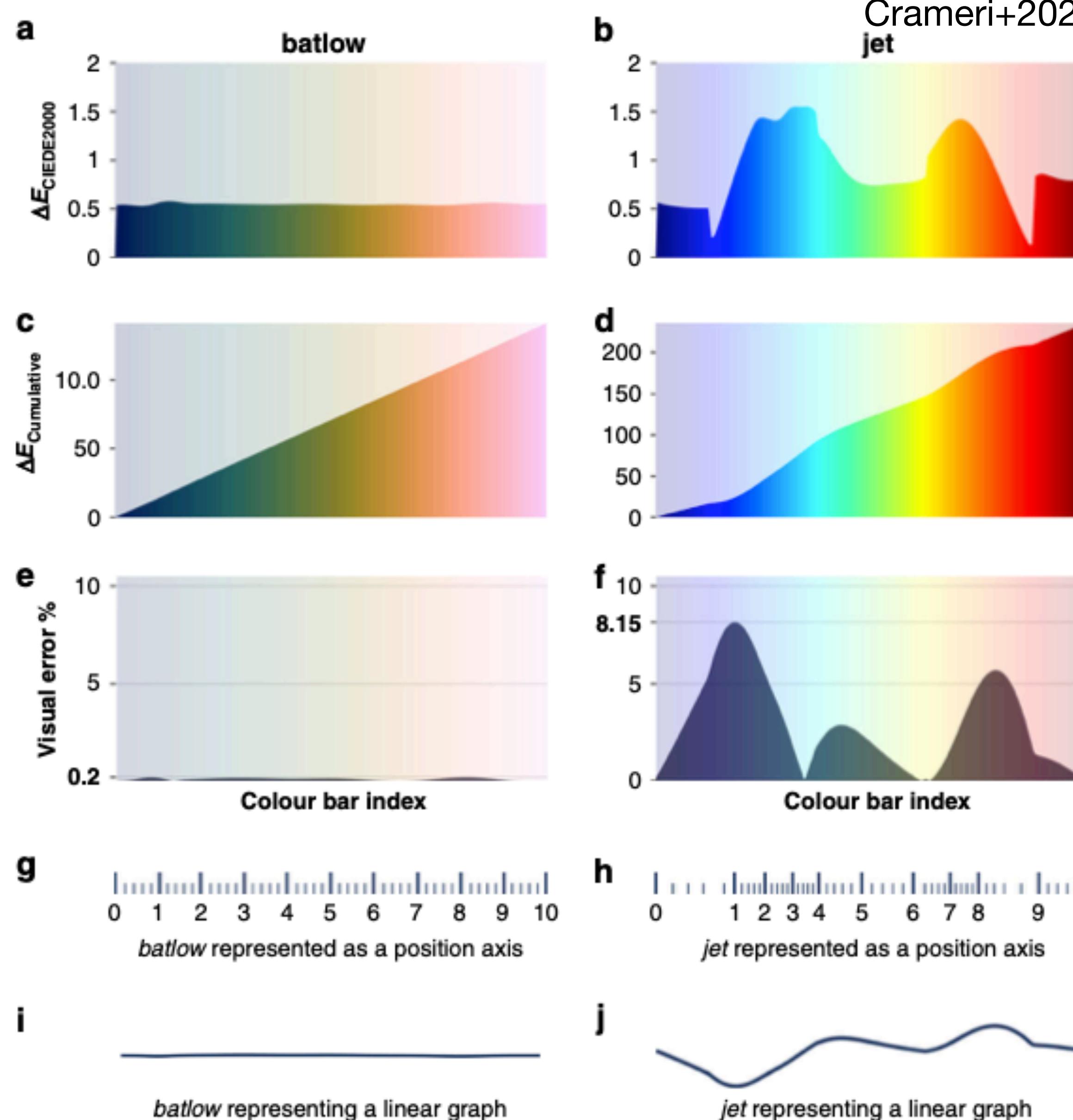
Cramer+2020

What is perceptual uniformity?



Cramer+2020

Checking your colormaps: Perceptual uniformity



Above: using viscm

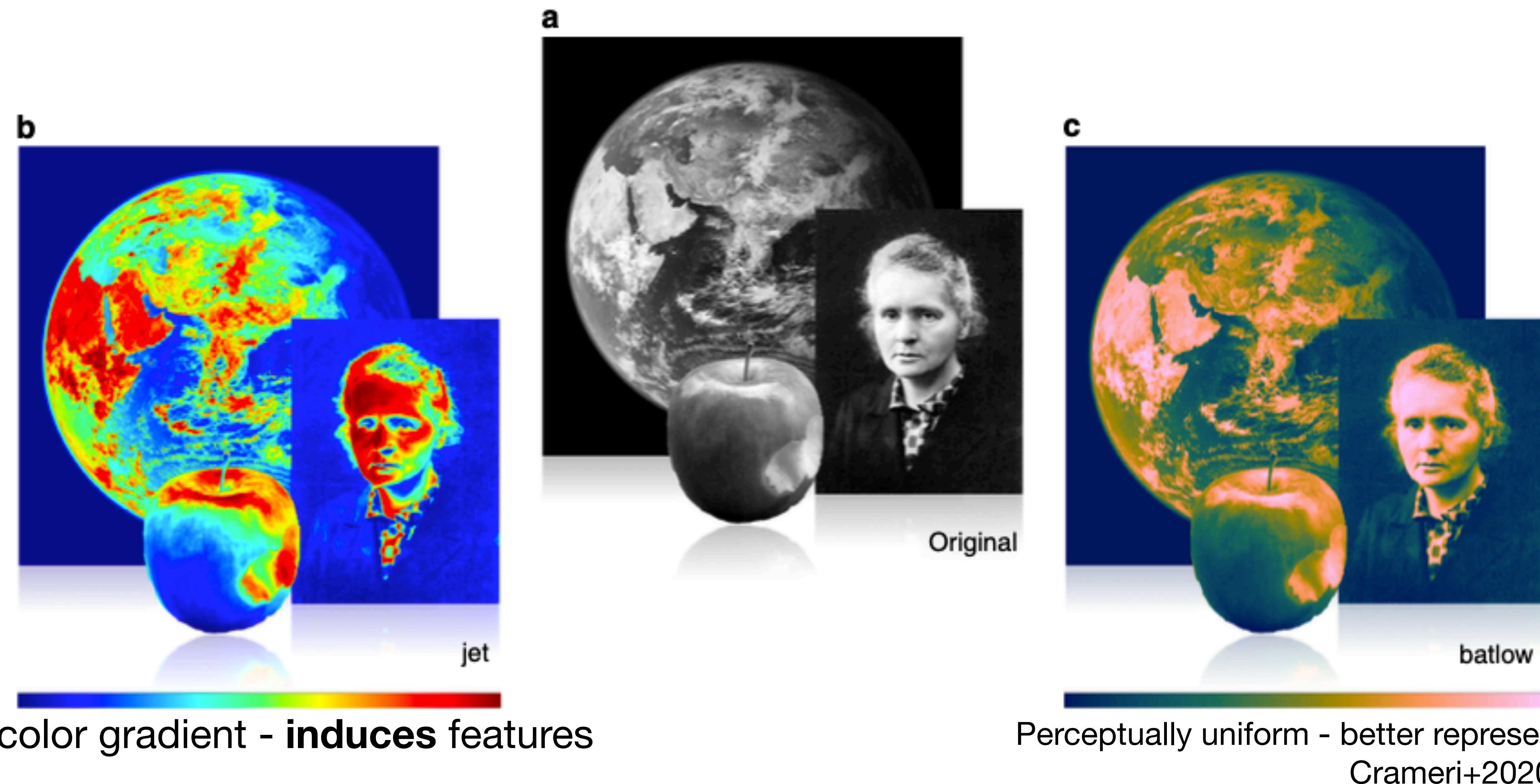
How to:

1. pip install viscm
2. python -m viscm view jet

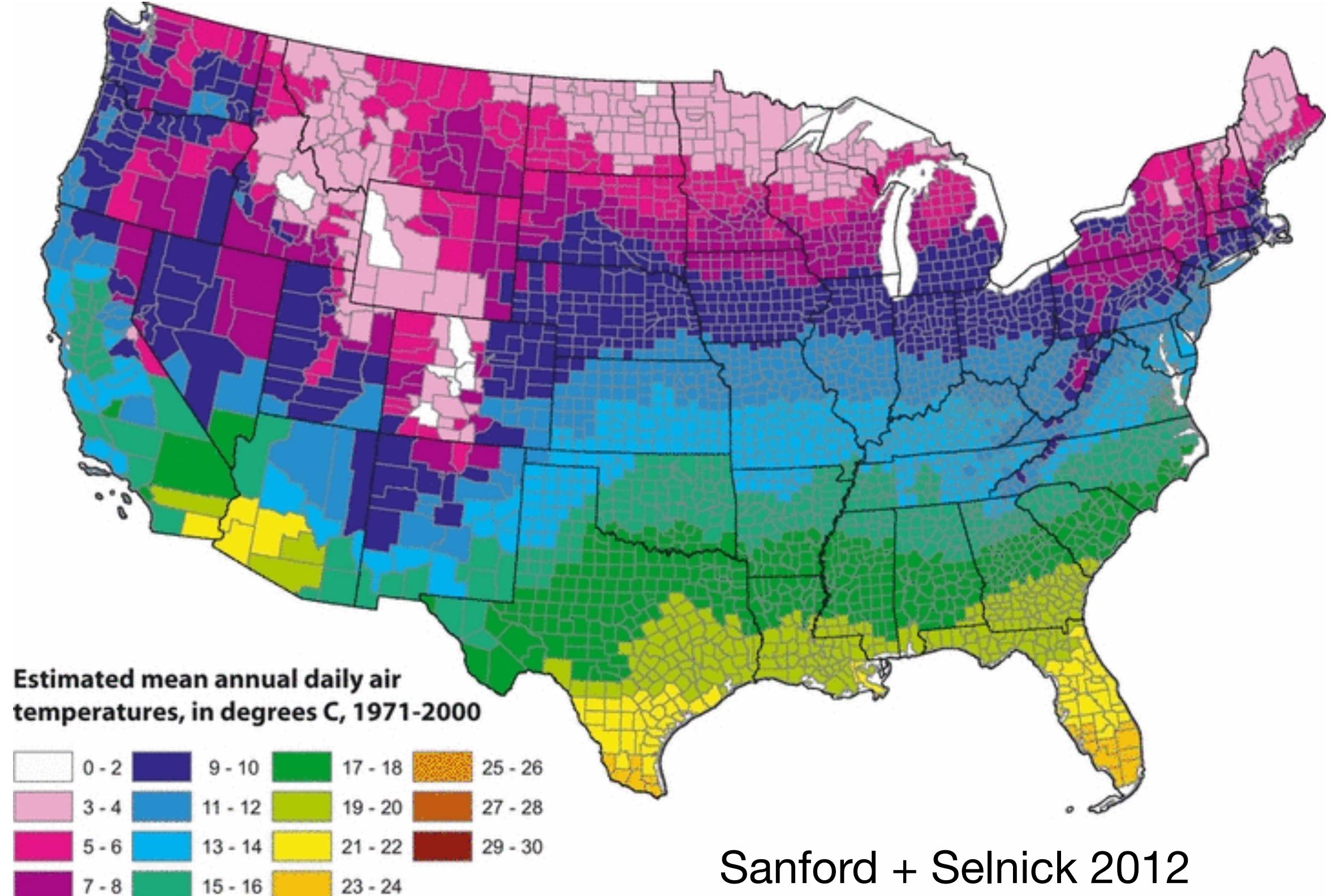
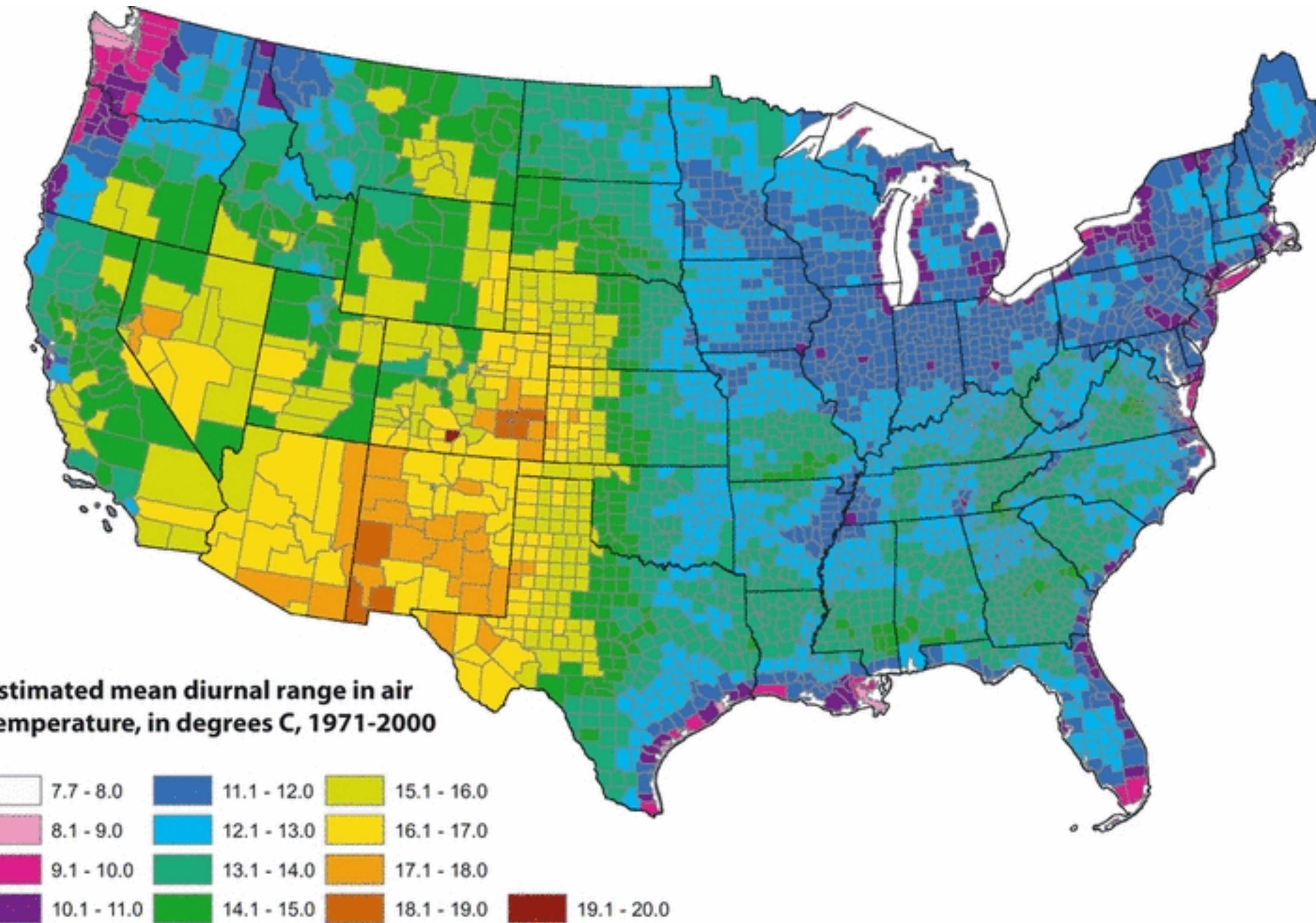
Can make your OWN colormaps using:
python -m vidcm edit

<https://bids.github.io/colormap/>

Feature bias from colormap selection



Feature bias from colormap selection

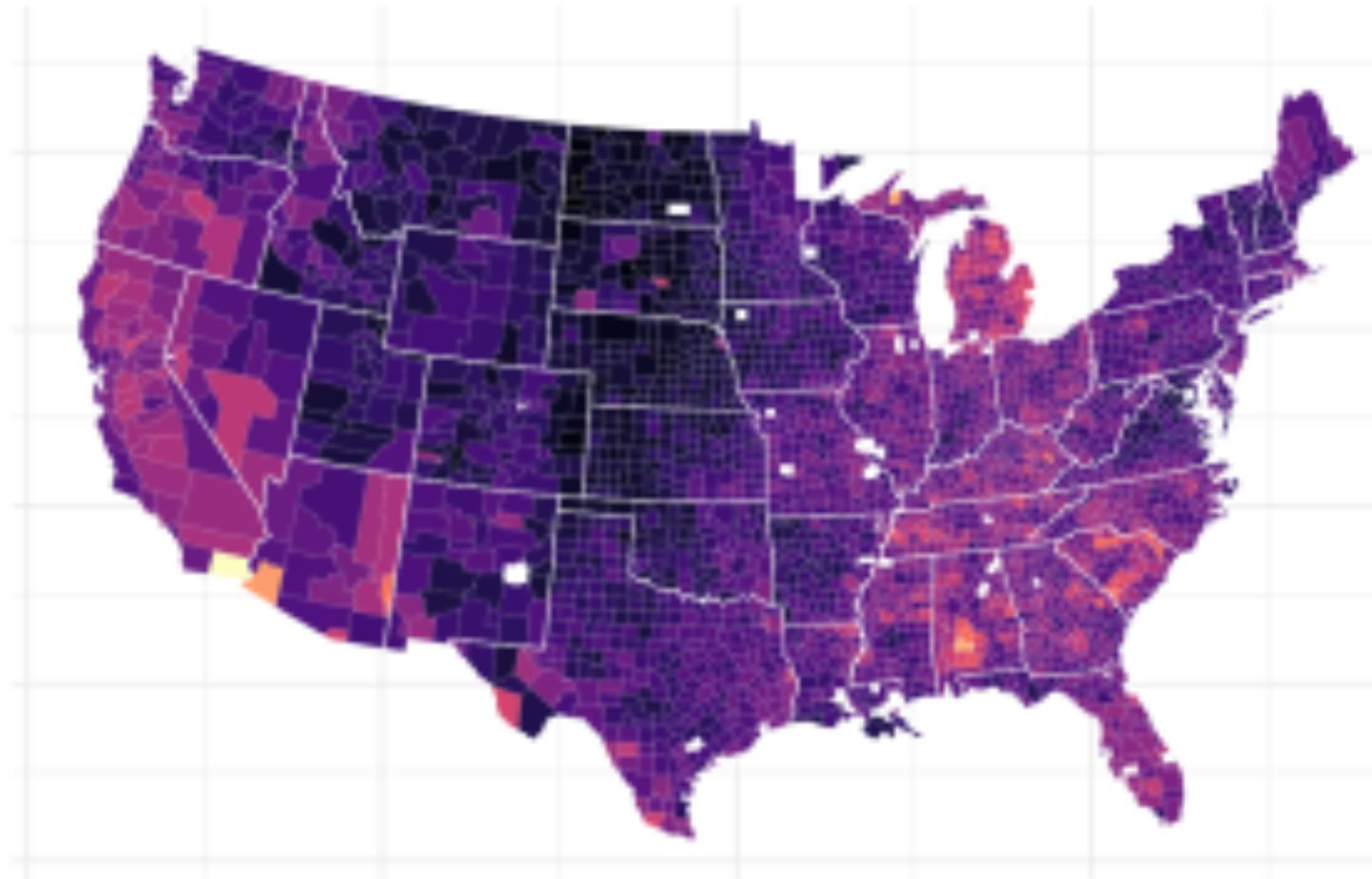


Sanford + Selnick 2012

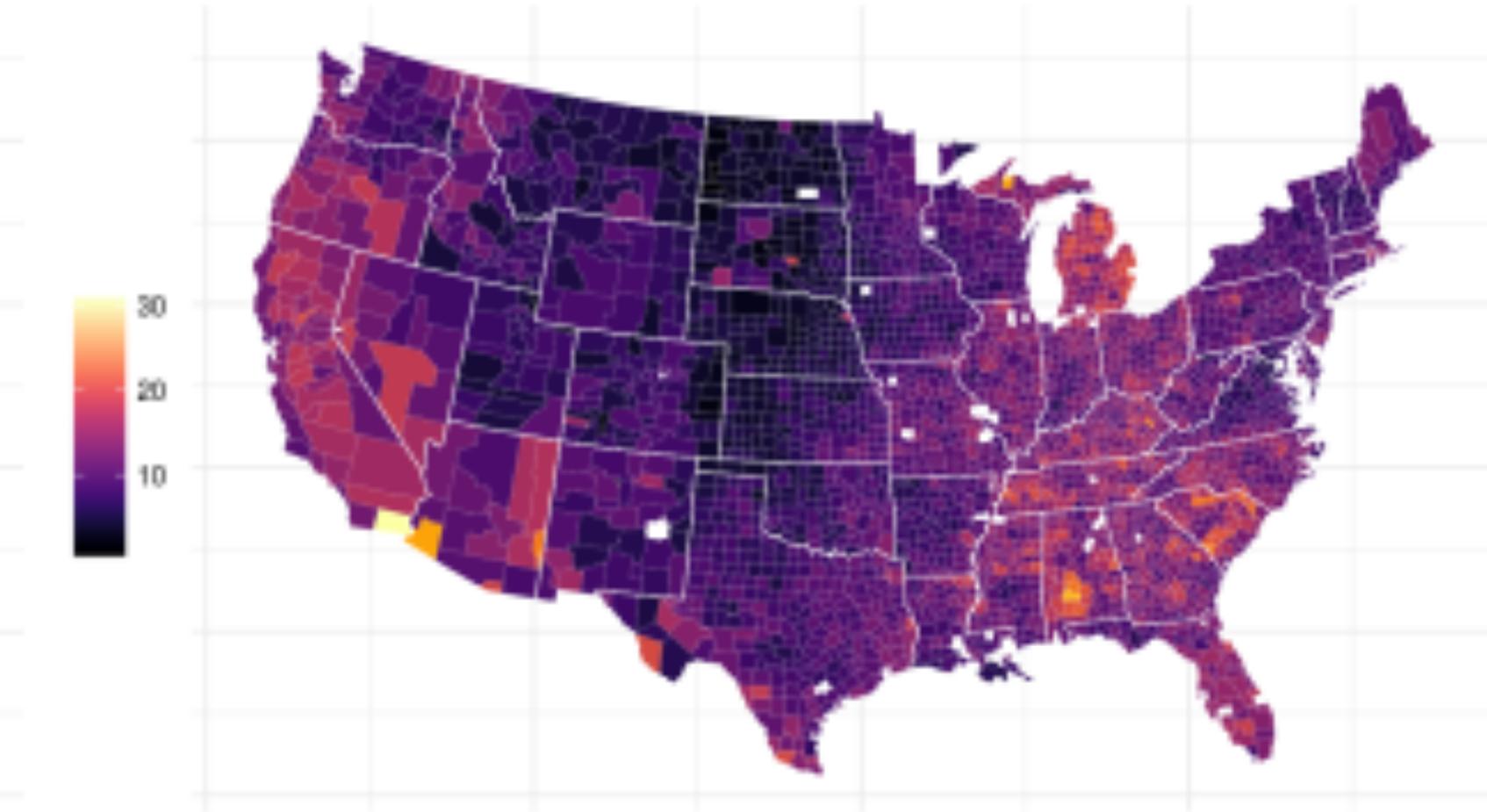
Feature bias from colormap selection

US unemployment rate by county

option A aka 'magma'

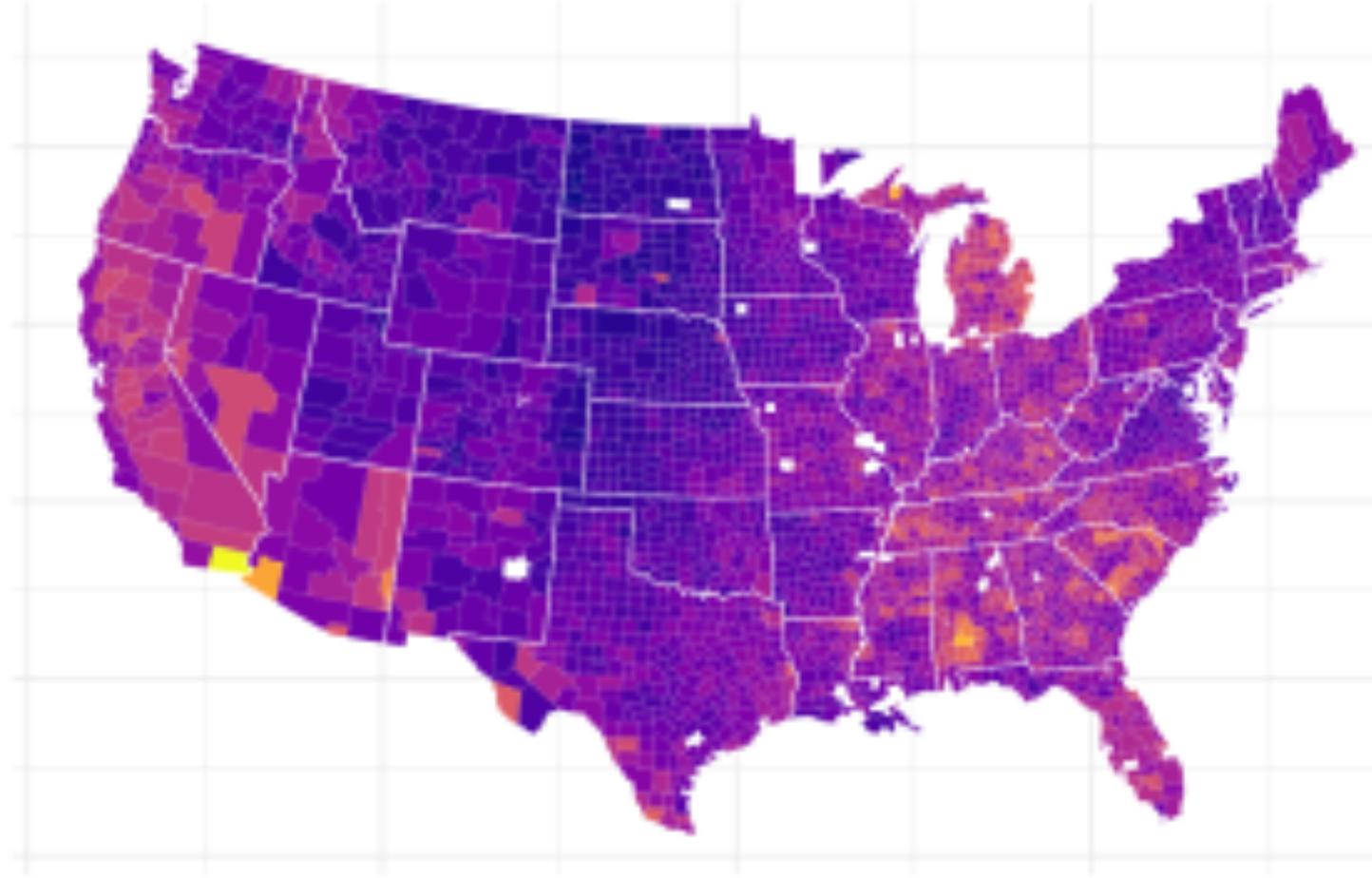


option B aka 'inferno'

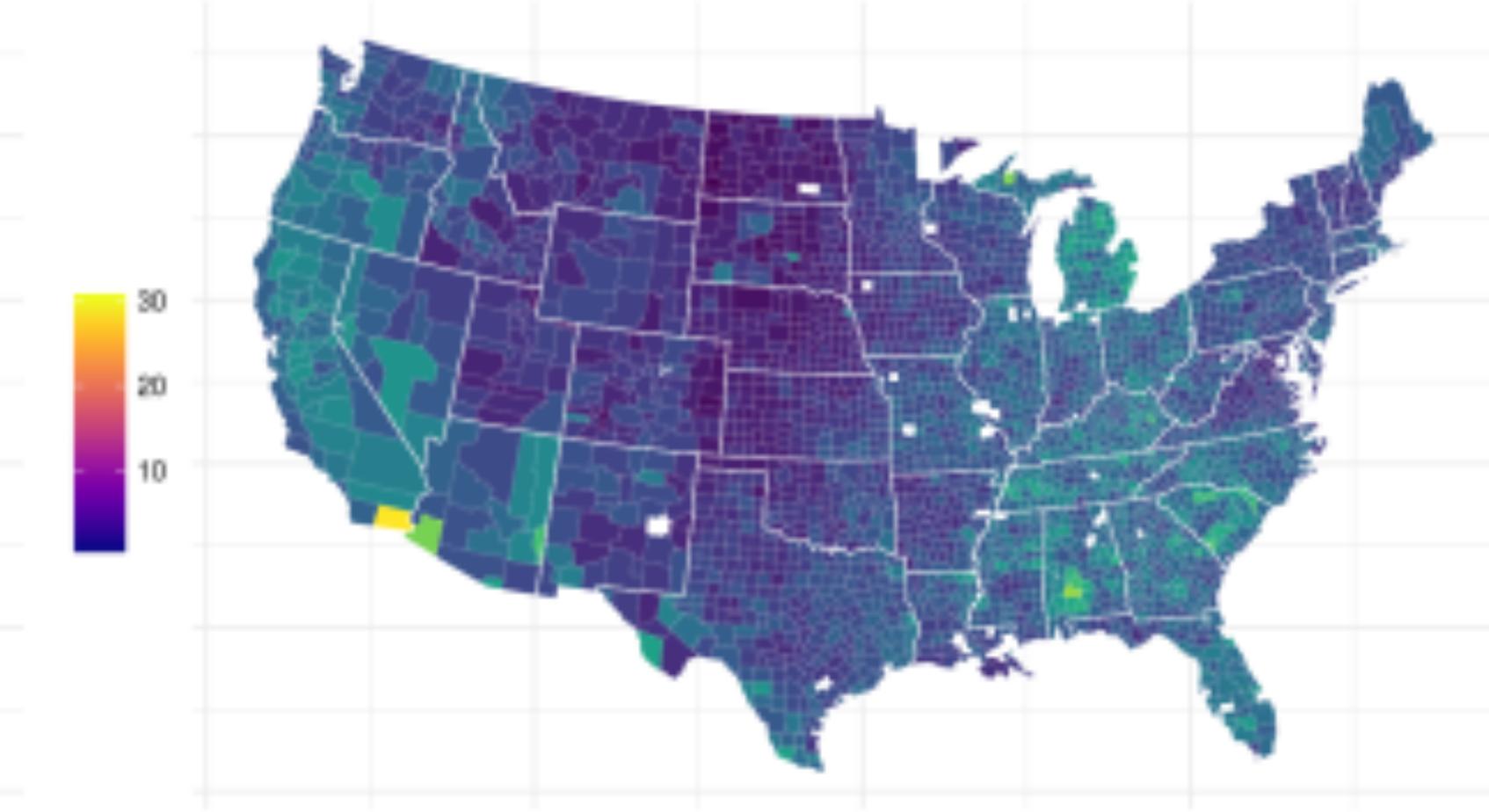


<https://betterfigures.org/2015/06/23/picking-a-colour-scale-for-scientific-graphics/>

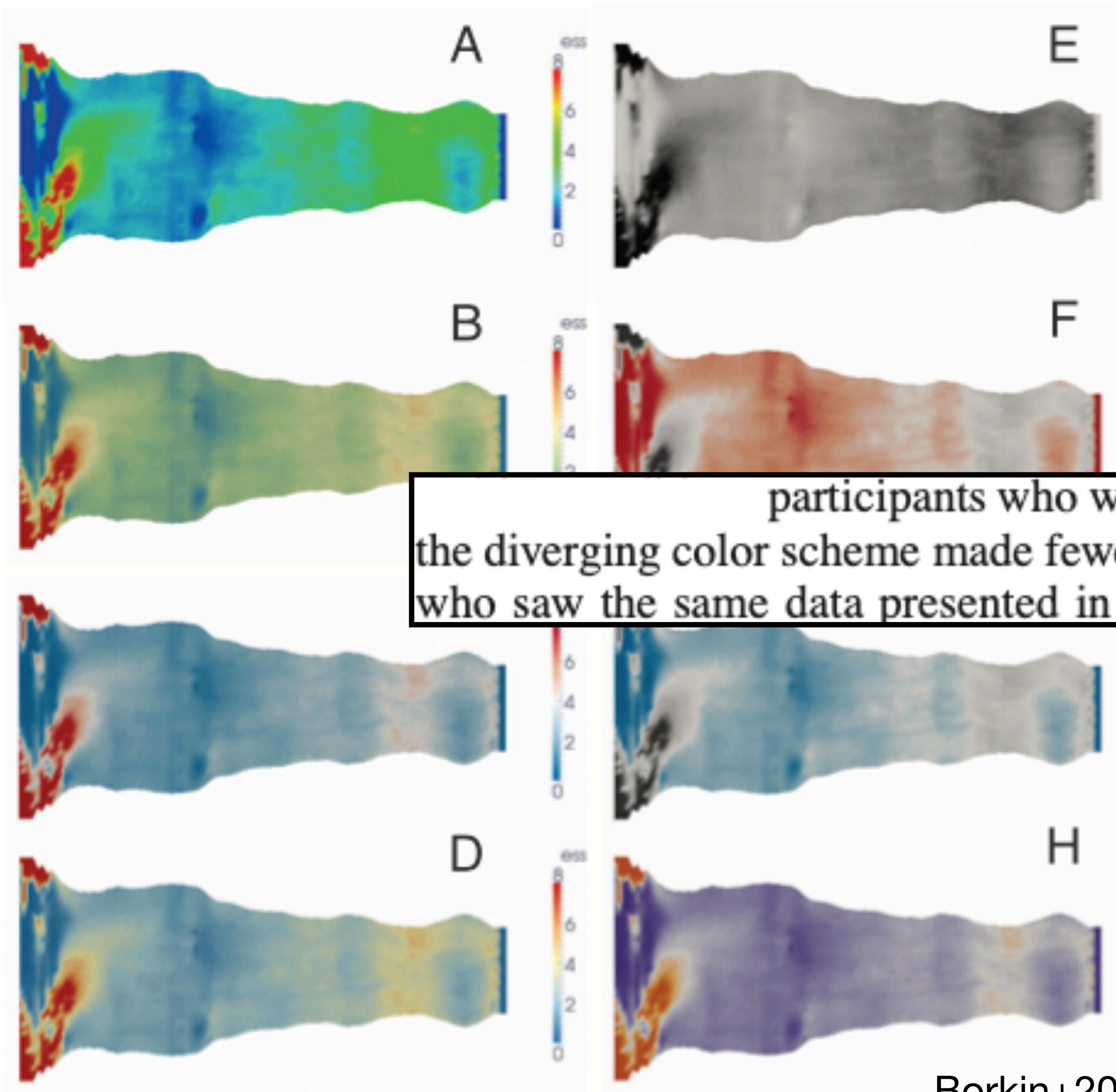
option C aka 'plasma'



option D aka 'viridis'



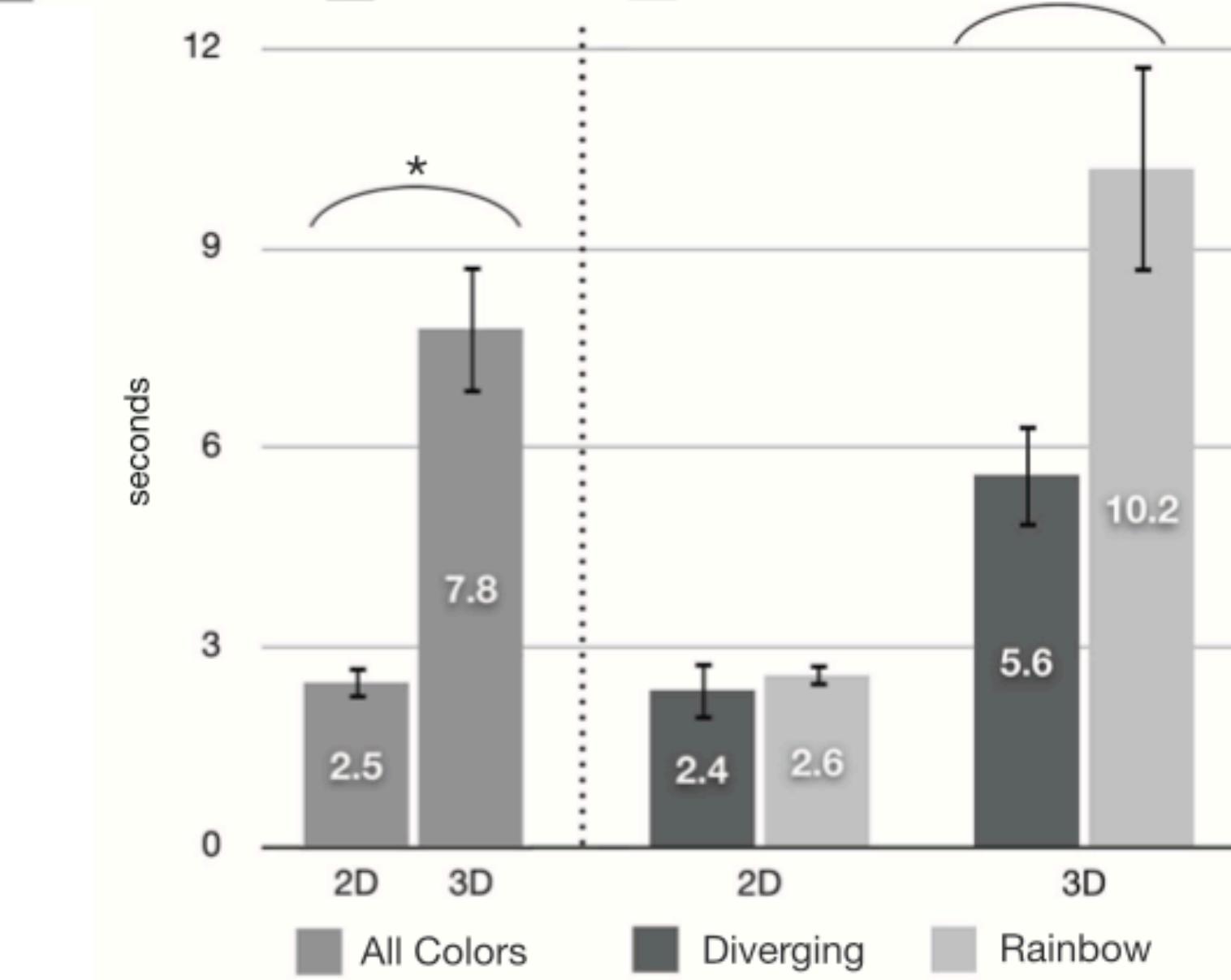
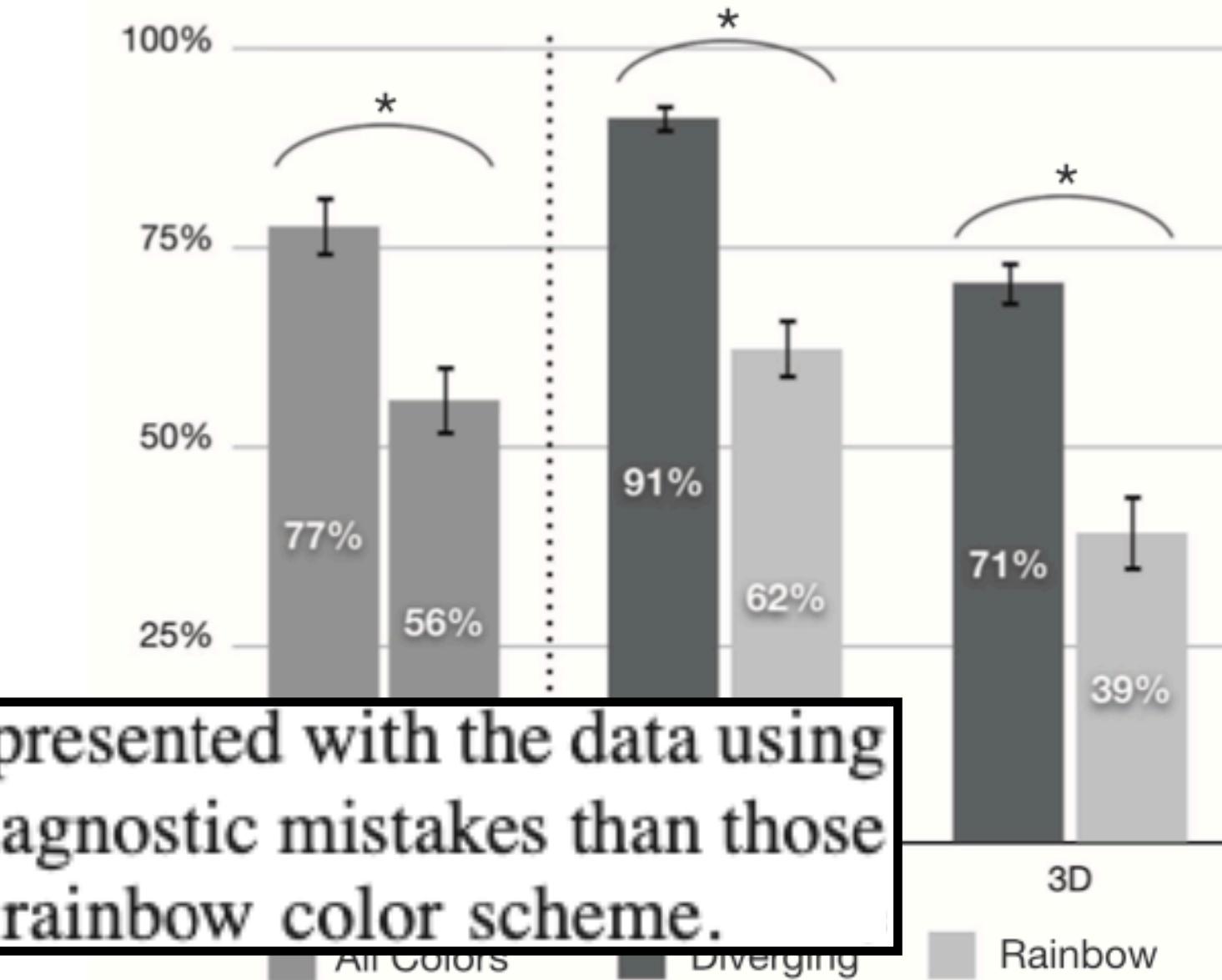
Colormap selection: a matter of life or death!



Borkin+2011

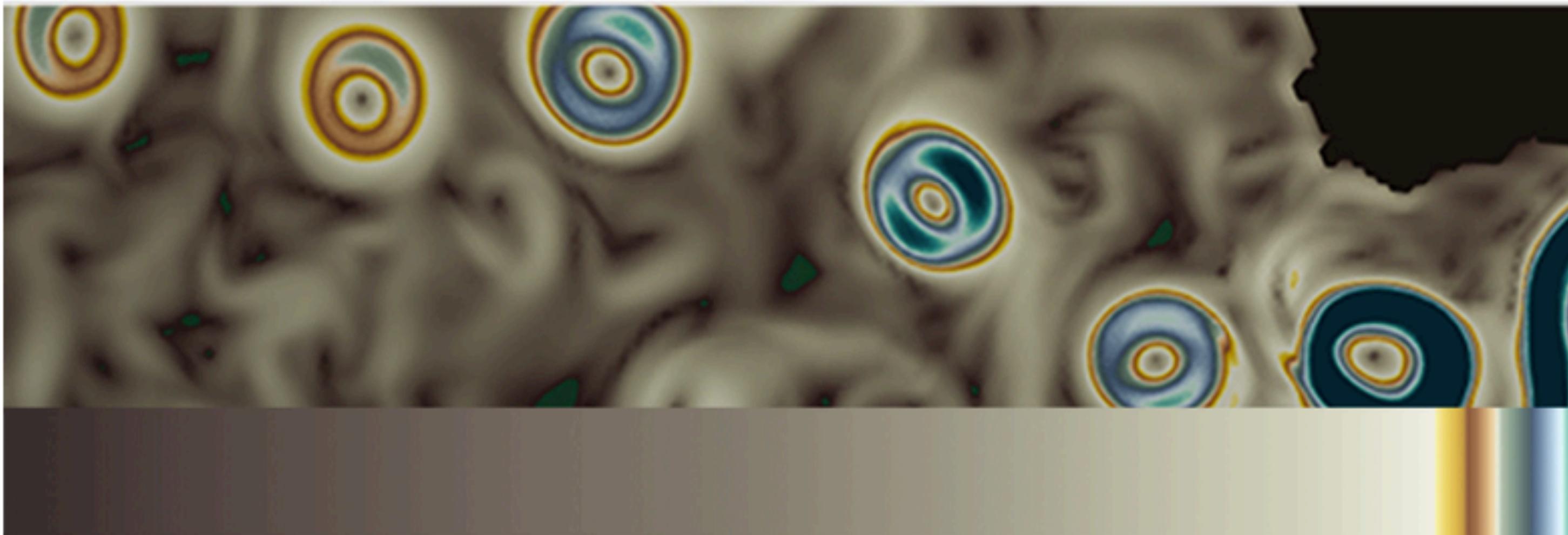
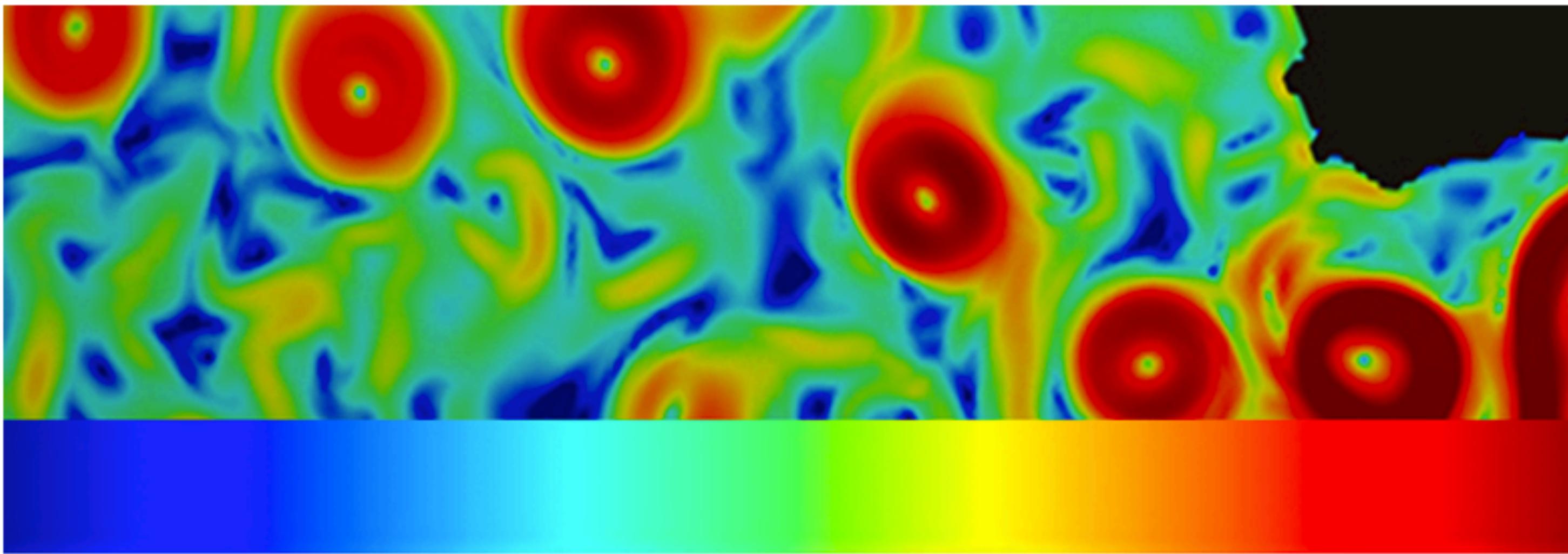
participants who were presented with the data using the diverging color scheme made fewer diagnostic mistakes than those who saw the same data presented in the rainbow color scheme.

Average percent of low ESS regions identified



Colormaps can help highlight **wanted** features!

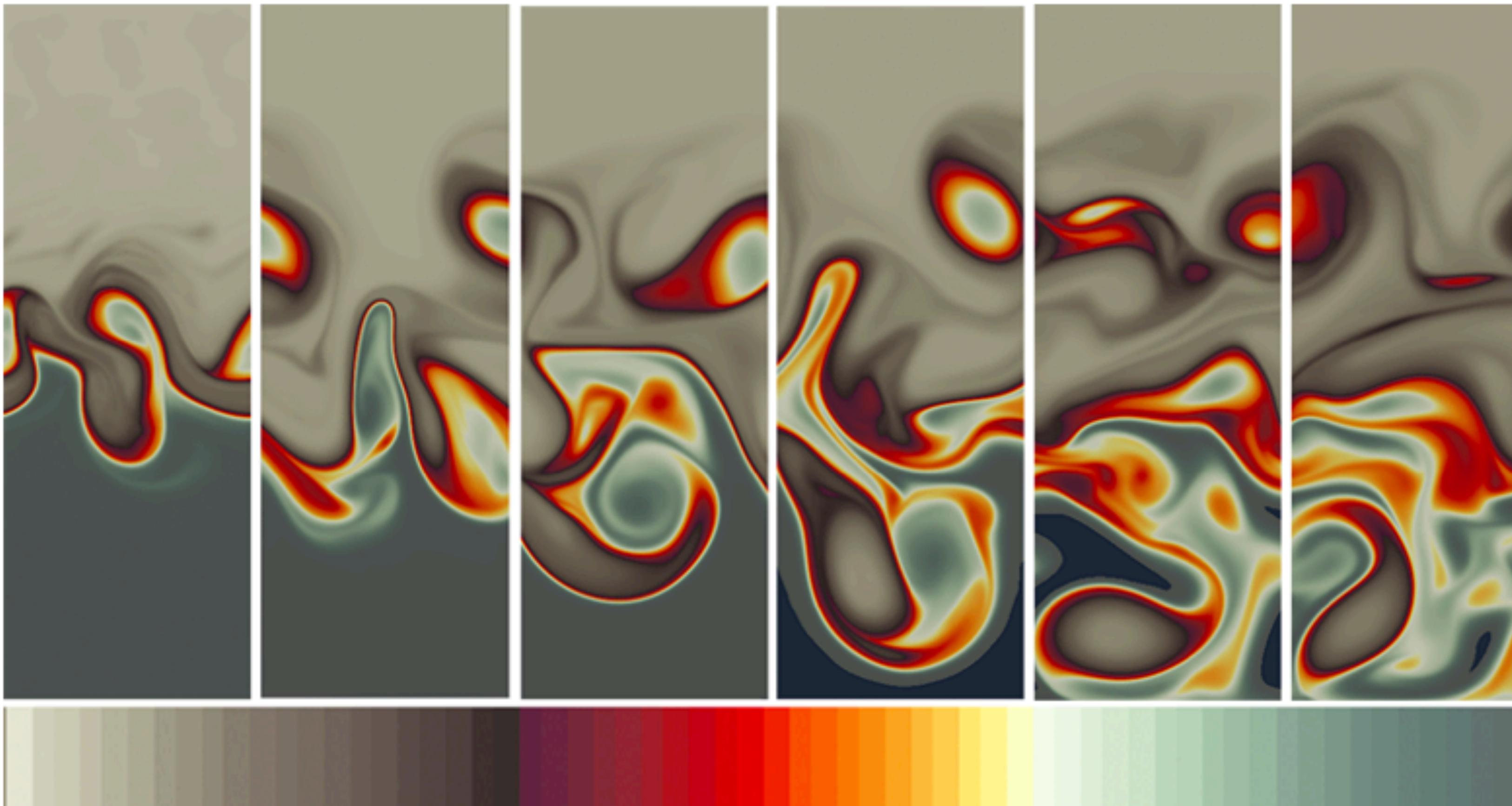
Example: want to highlight **eddies** in hydro flow



EOS. Created by Francesca Samsel w/ data from M. Petersen, LANL.

Colormaps can help highlight **wanted** features!

You can break from perceptual uniformity with the
intention to highlight features



And now...a python interlude....

GitHub clone:

<https://github.com/AstroBrandt/DataVisTutorial>

Resources

- F. Crameri+2020, *The Misuse of Colour in Science Communication*
 - <https://www.nature.com/articles/s41467-020-19160-7>
- BetterFigures
 - <https://betterfigures.org/2015/06/23/picking-a-colour-scale-for-scientific-graphics/>
- Borland & Taylor 2007
 - http://people.renci.org/~borland/pdfs/RainbowColorMap_VisViewpoints.pdf
- Eos. Zeller & Rogers, 2020
 - <https://eos.org/features/visualizing-science-how-color-determines-what-we-see>
- Matplotlib colormaps (old, before viridis was default), viscm
 - <https://bids.github.io/colormap/>
- Colorbrewer2:
 - <https://colorbrewer2.org/>