Github:AstroBrandt/DataVisTutorial

NASA/ESA/STScI

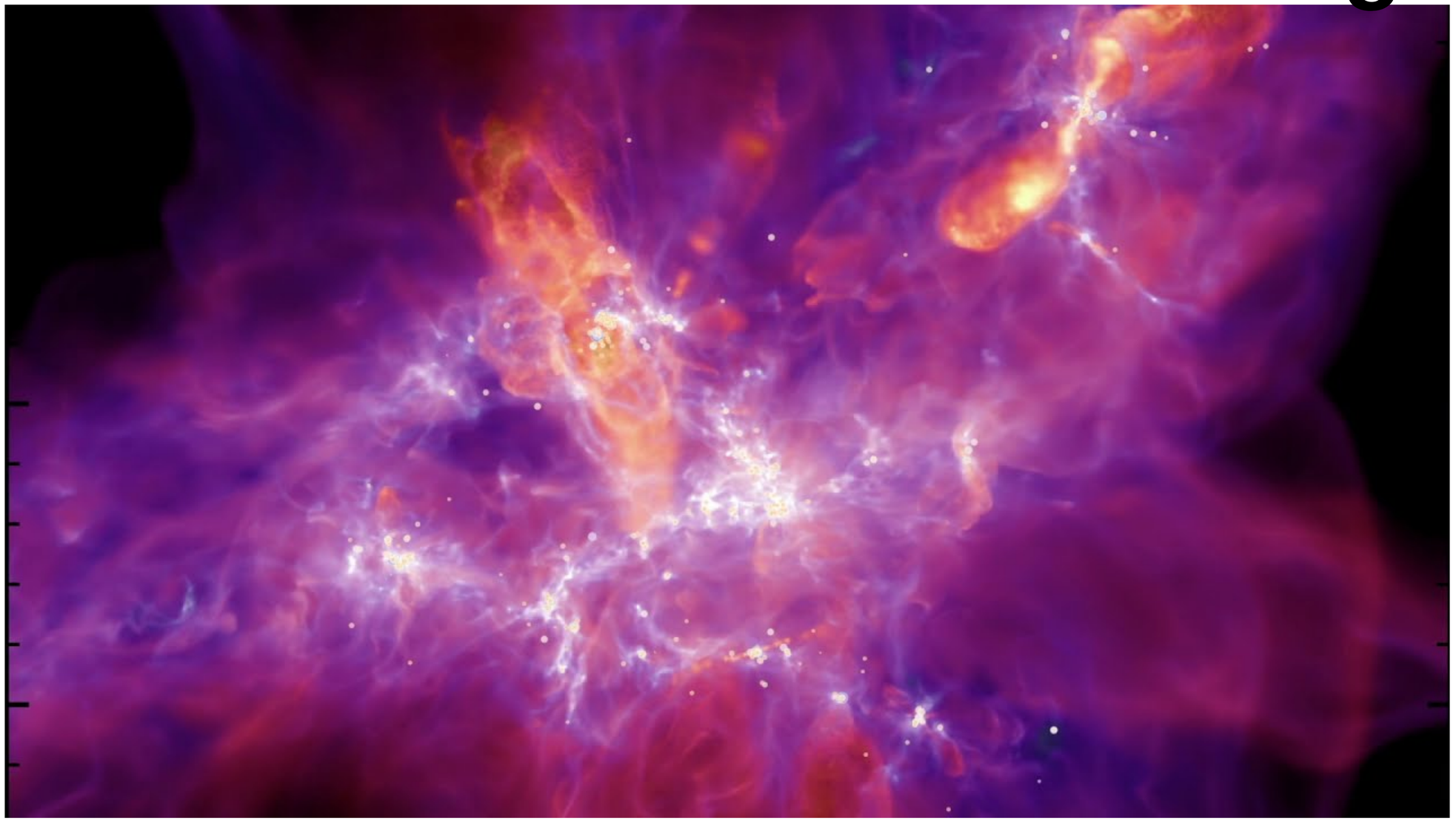
Data Visualization and Python Plotting Libraries

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
```

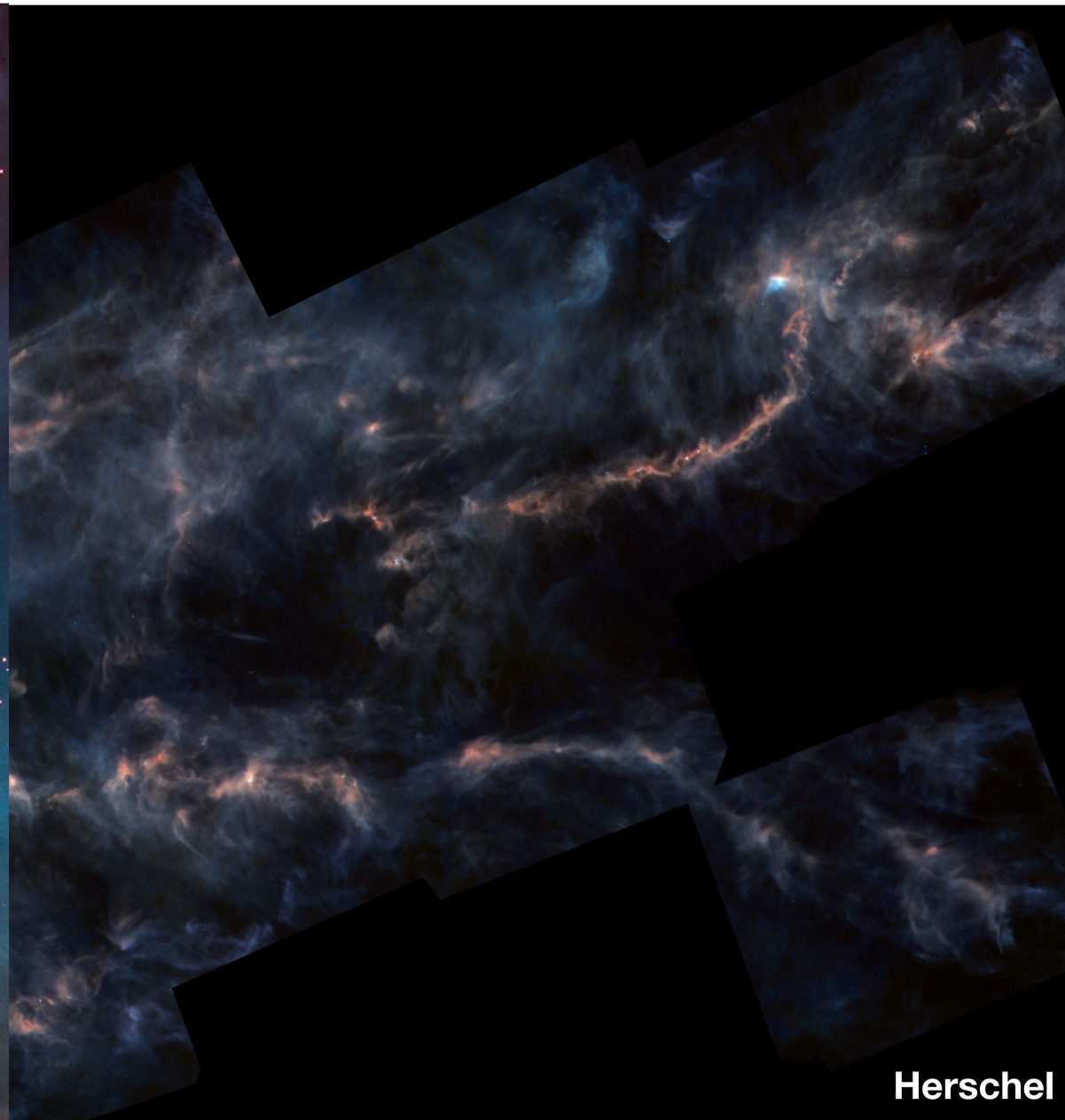
GitHub QR

STARFORGE volume renderings



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

Visualizations Matter



Visualizations Matter

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

Resources for Better Graphs

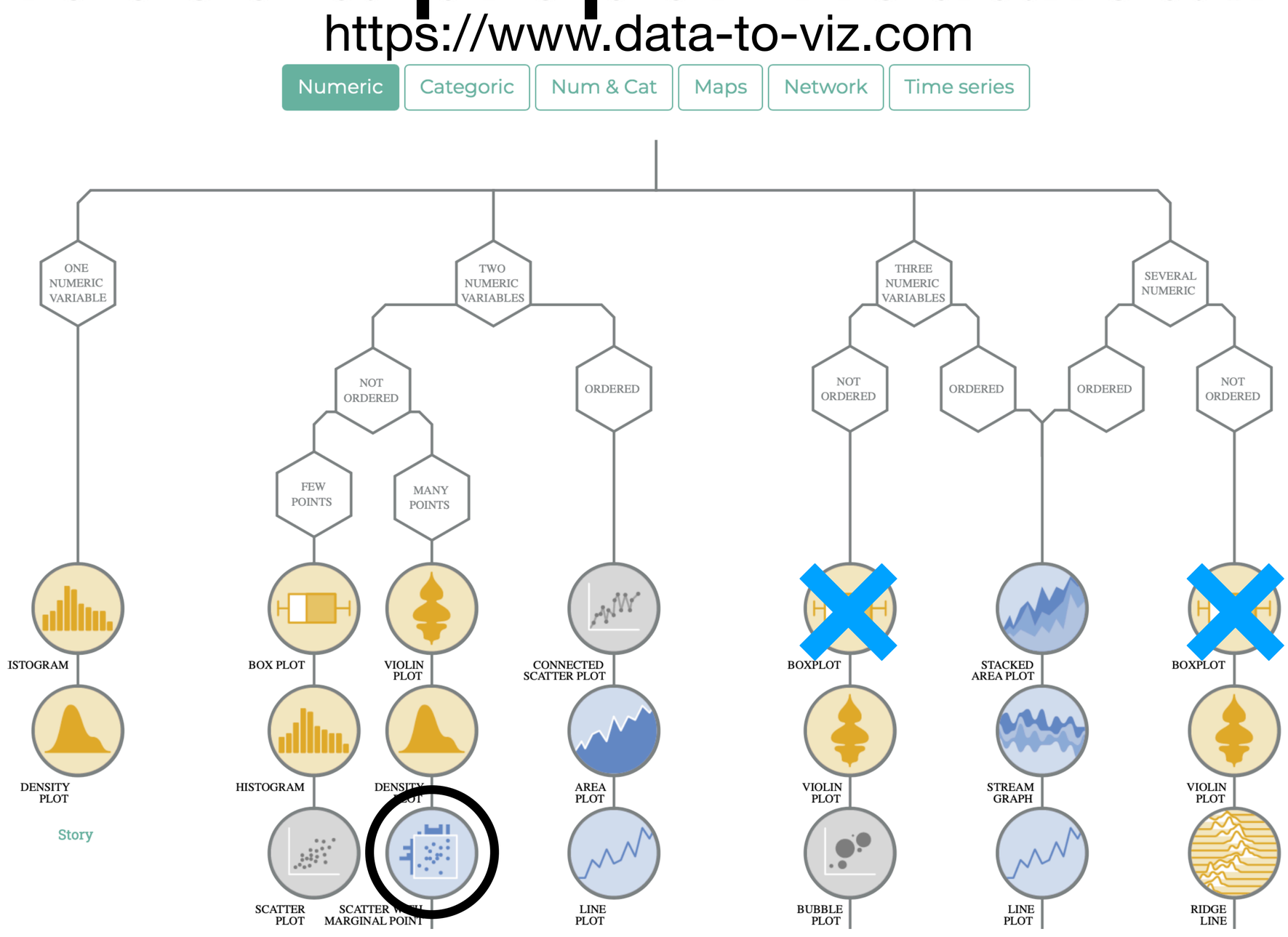
(Python) color map sources

- Matplotlib (includes colorbrewer colormaps).
  - Viscm tool to make colormaps and export them. All viscm colormaps are perceptually uniform (or diverging) and are generally printer and color blind safe)
- Cmocean: used viscm to make many colormaps, mostly for oceanographic uses, but still very good general ones)
- Cmasher: Also used viscm. Many alternative colormaps **and** tools for manipulating and sampling colormaps!
- Colorcet - uses a different approach (Kovesi 2015), all perceptually uniform but not constrained by printer/color blindness

Get support from InfraVis!
https://infravis.se/

Choose a proper visualisation

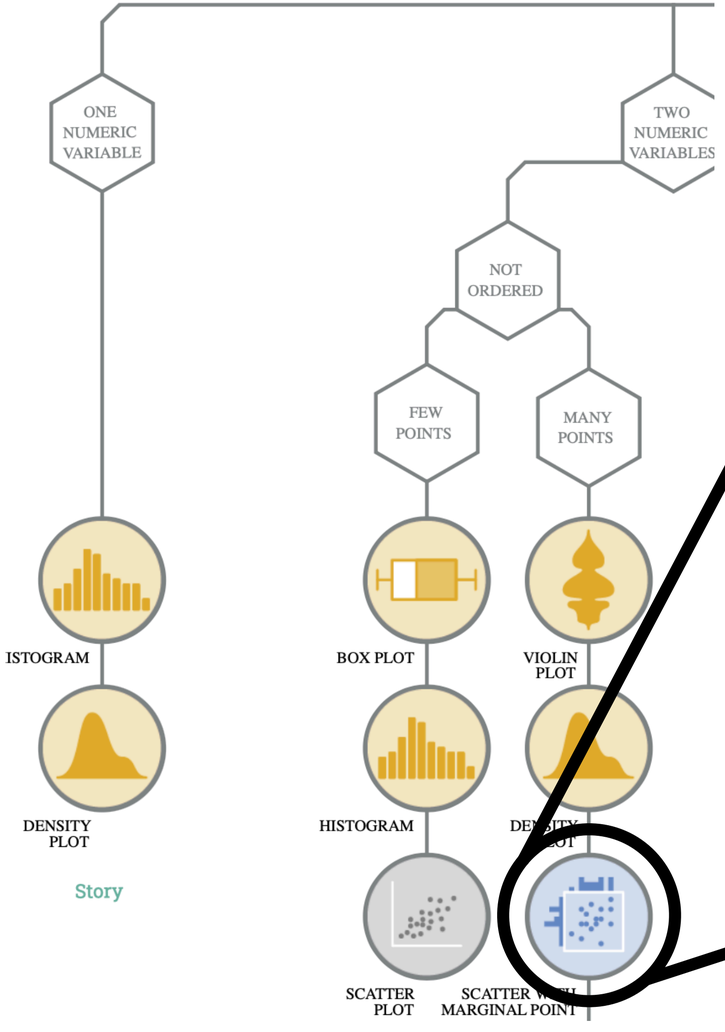
https://www.data-to-viz.com



Choose a proper visualisation

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

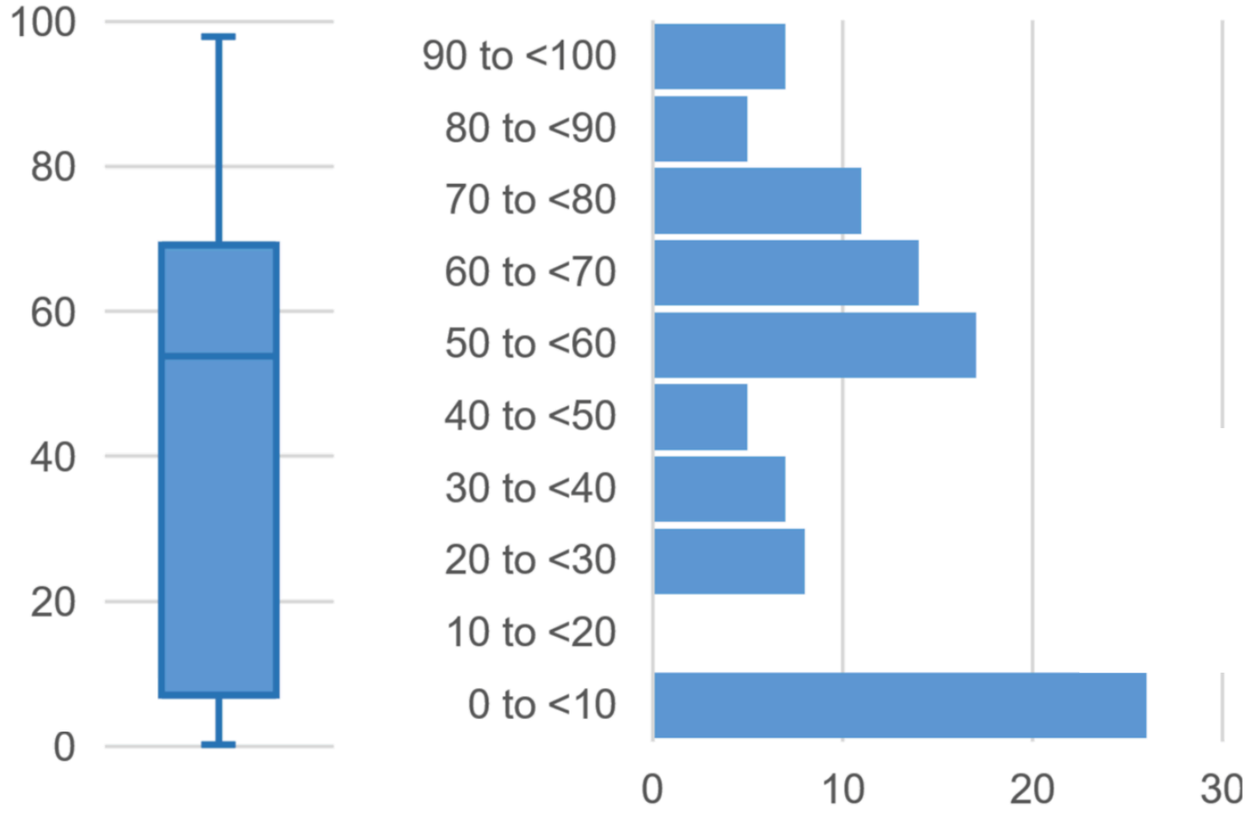
Numeric | Categoric



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



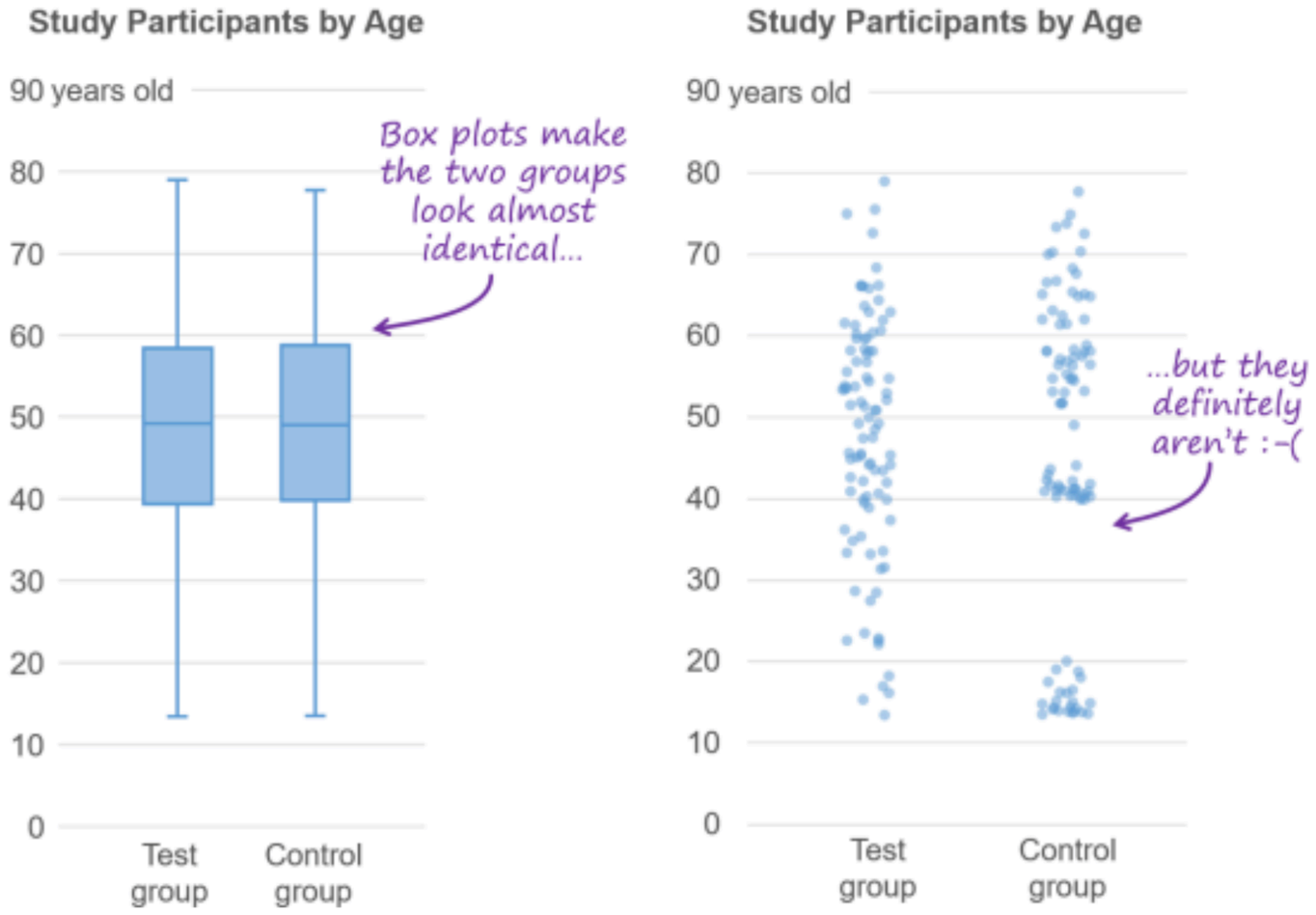
Box plots can hide useful information. In the left, a significant amount of the distribution is hidden in the “small” separation between box and whisker. Thus, it can hide the real distribution.

Replace with:

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

Source: Nightingale, Nick Desbarats Nov 2021

Useful aside:
Why box plots are bad



Source: Nightingale, Nick Desbarats Nov 2021

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:

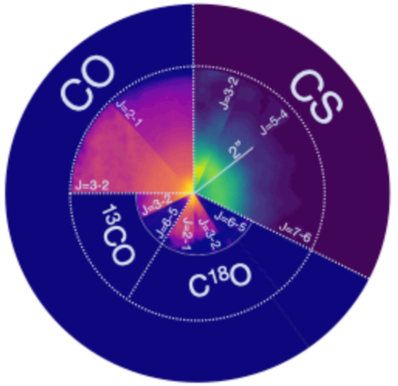
1. Violin plot
2. Shaded/area differenced boxes in vertical bins
3. Jittered strip plots

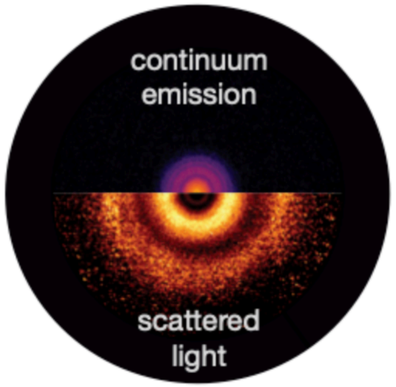
Guidelines for Clear Graphs

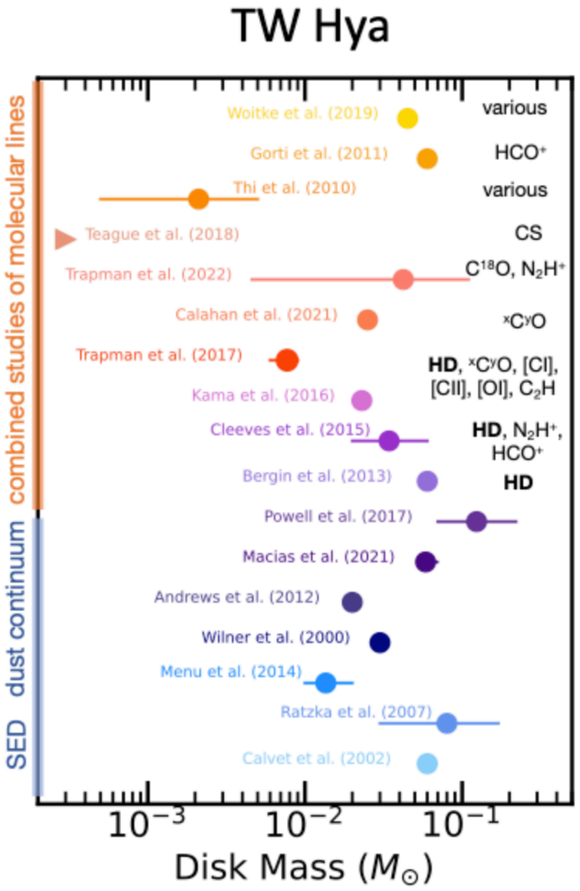
1. First step should always be ask: “What is the main takeaway for the figure?”
2. All text should be readily readable on the page.
    1. Aim for the axis labels to be the same size as the in-page text
3. If using categorical data, or data from multiple sources, reduce the number of point types to as 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!

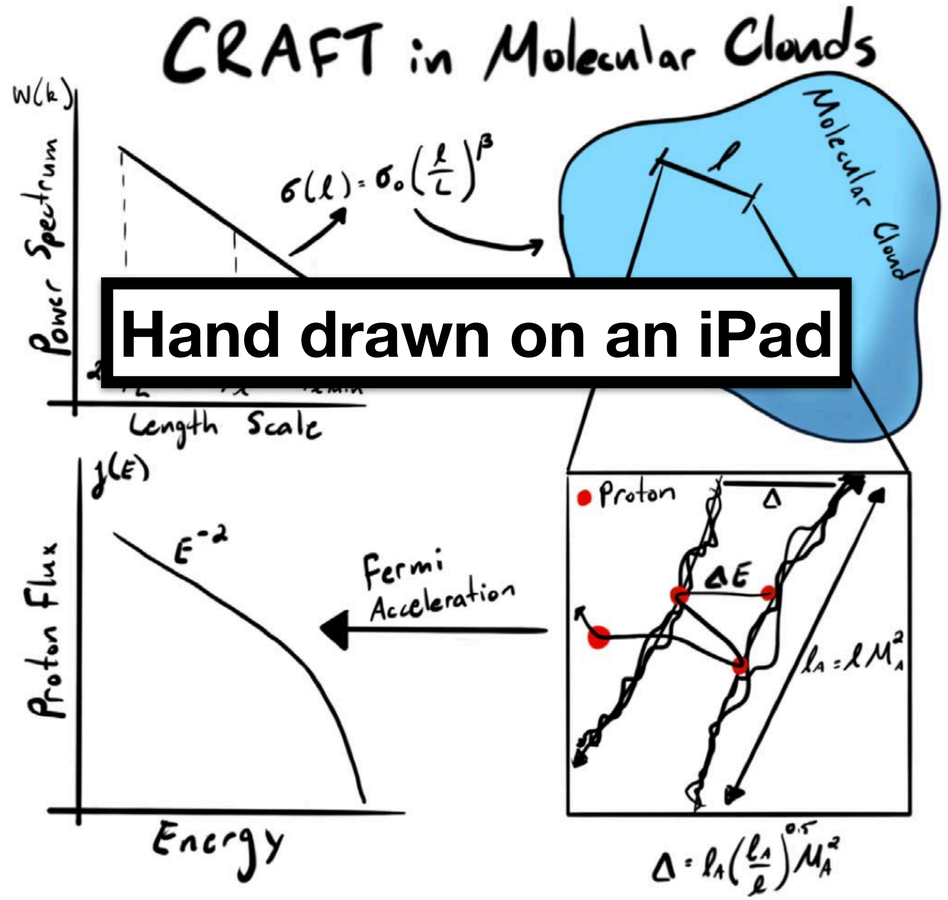
Hand drawn or vector schematics (can even make in keynote or ppt!) can be very useful for conveying the major point of your paper. Don’t be afraid to spend considerable time for a good summary figure for a paper!







Miotello+2022



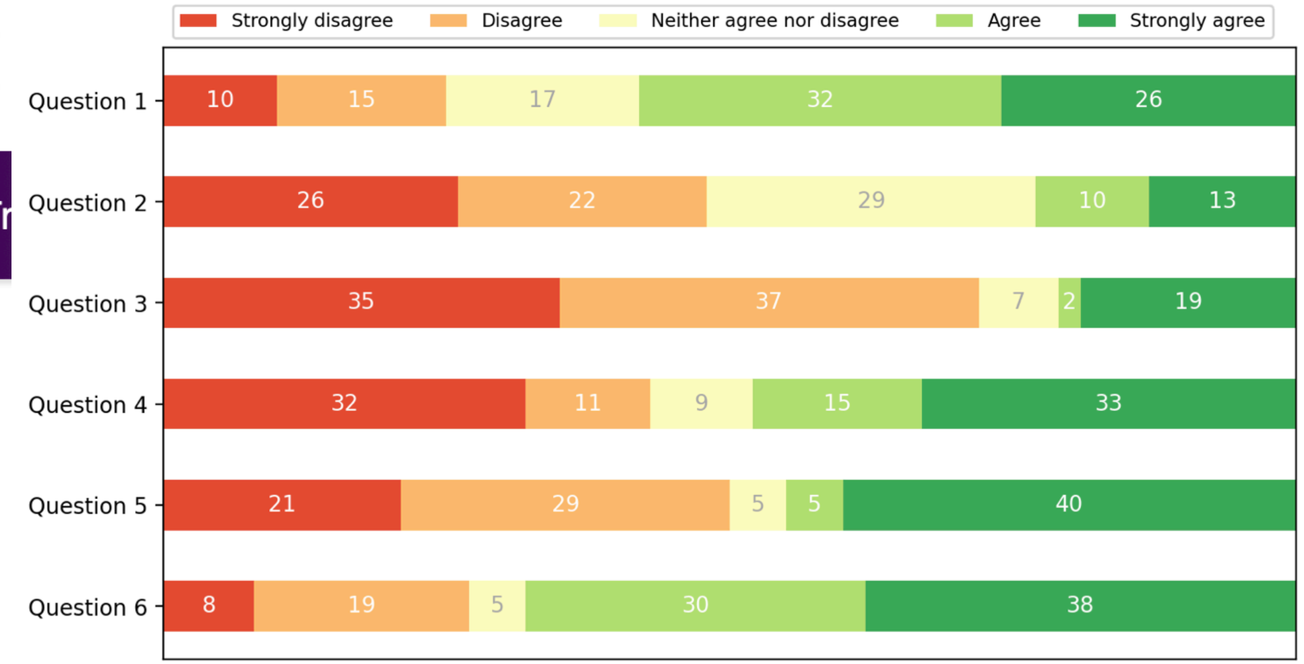
Gaches+2022

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
-
-



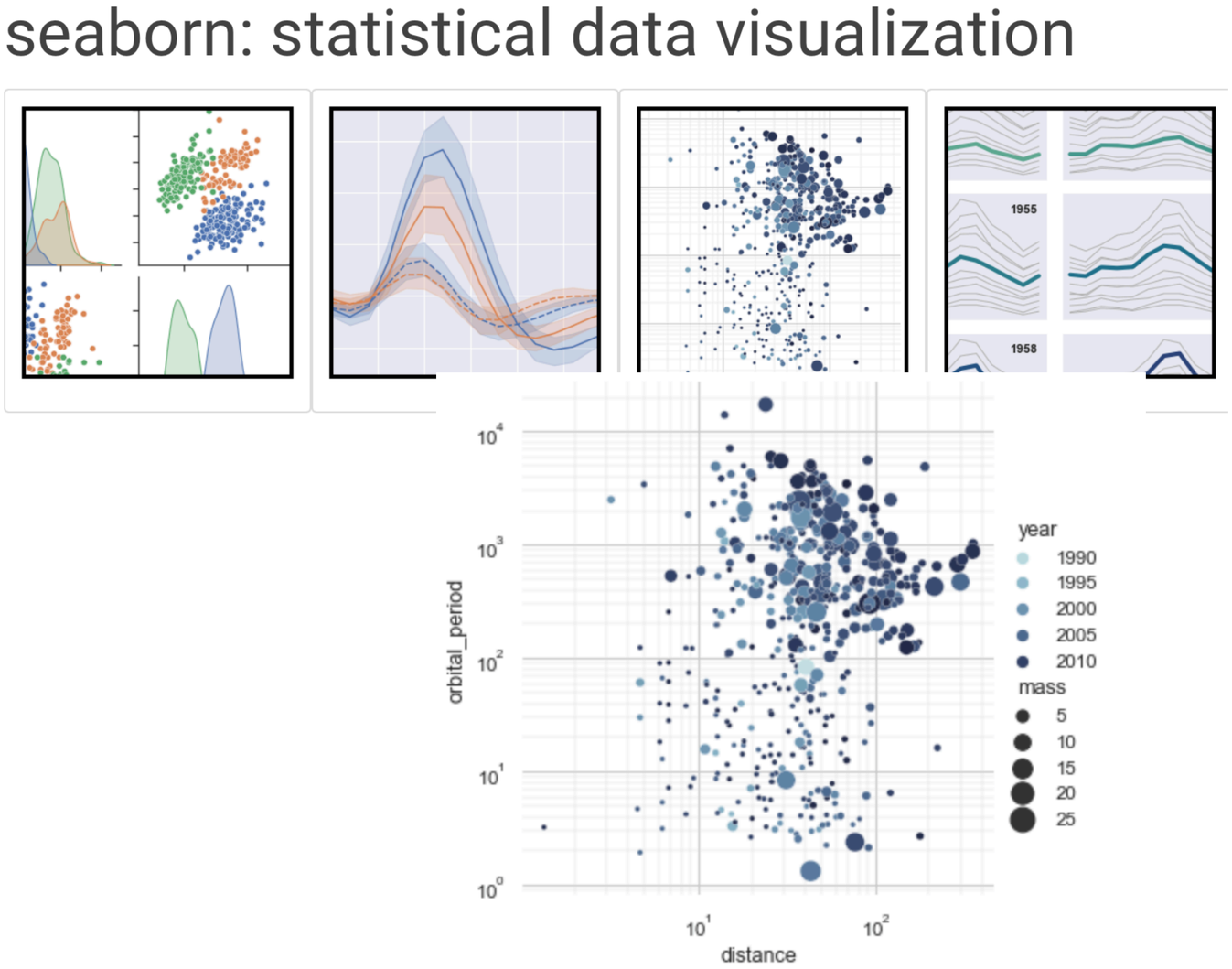
Benefits:

- 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



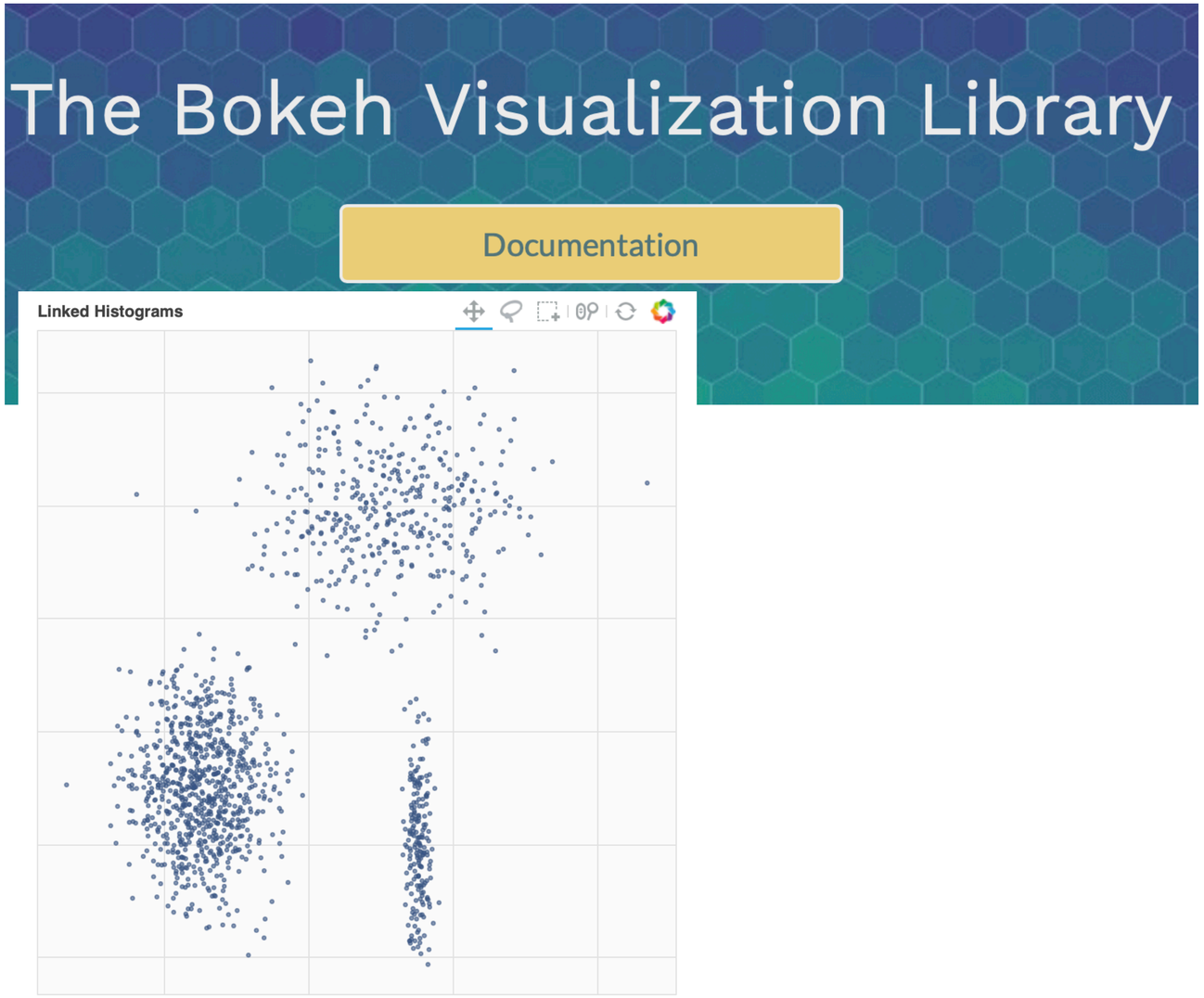
Benefits:

- 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



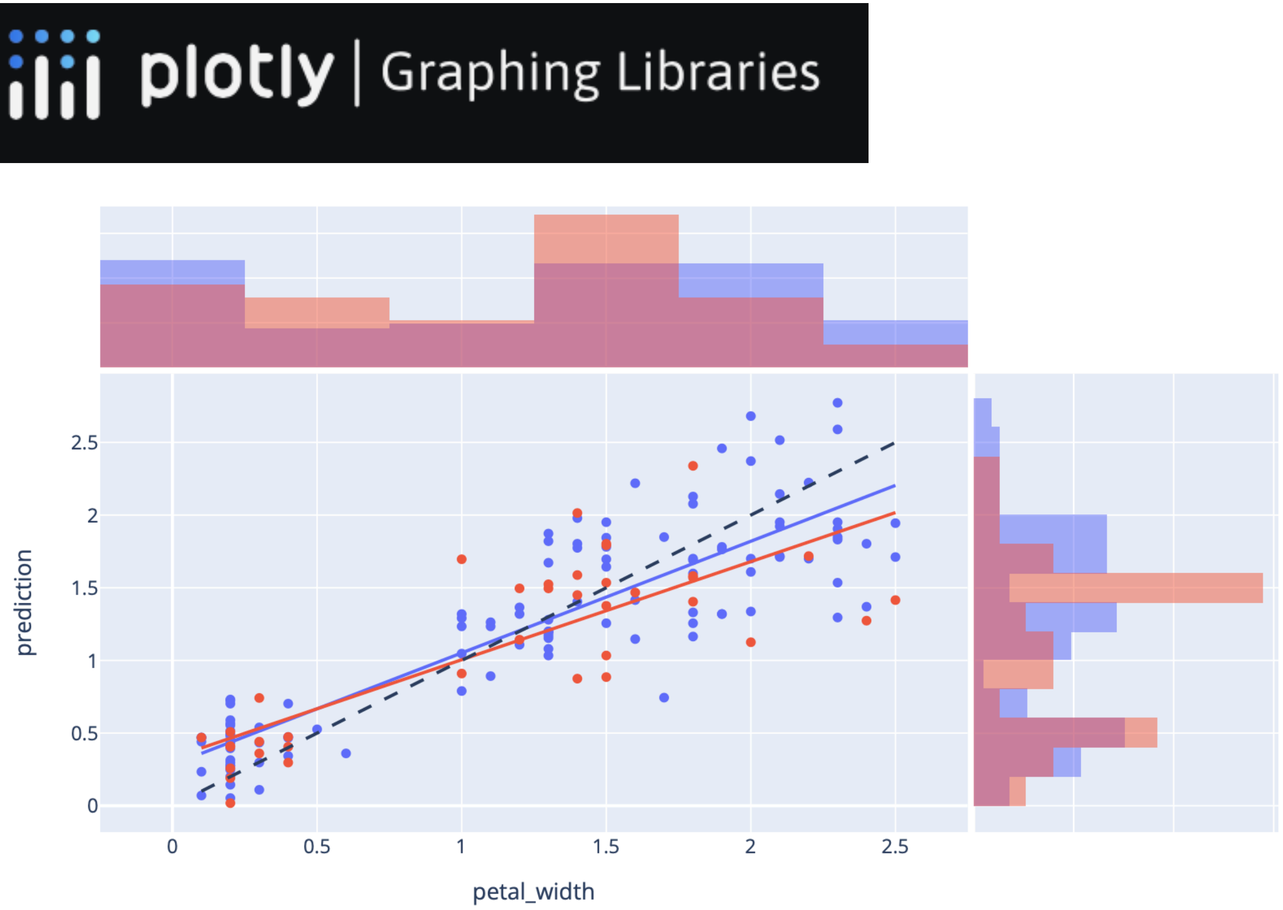
Benefits:

- 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

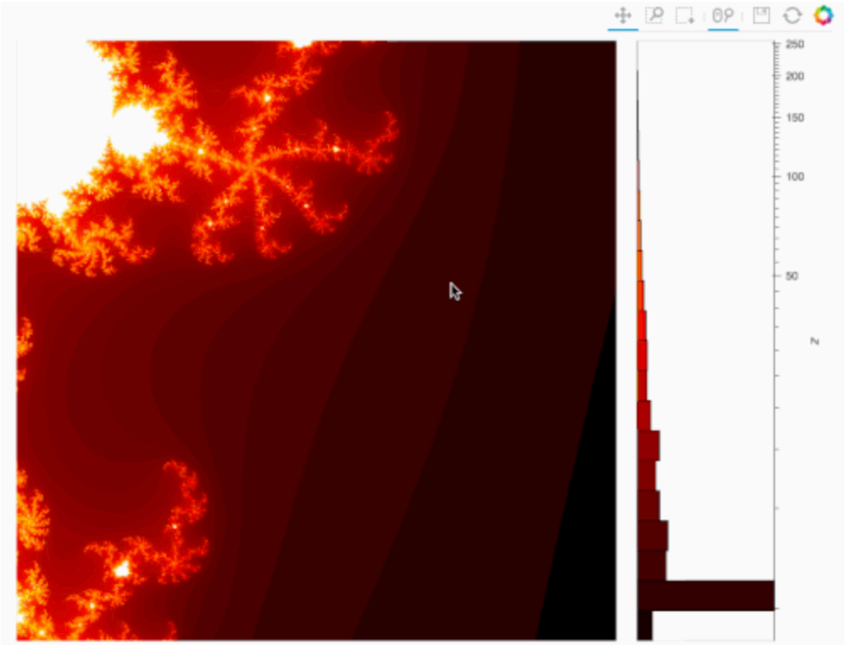
Cons:

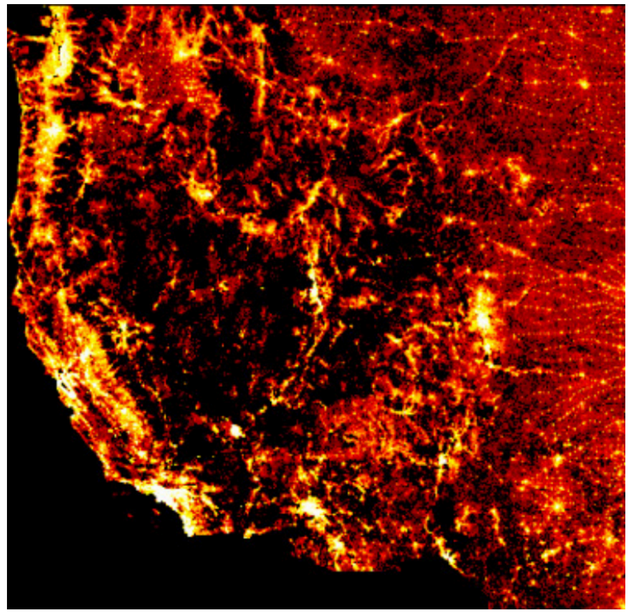
- Not as commonly used: not as much support available
- Not as useful for making publication pdfs. Journals are only just starting to accept interactive figures

Python packages

Holoviews+Datashader

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

These are all bundled under holoviz

Panel hvPlot HoloViews GeoViews Datashader Lumen Param

Colorcet

Python packages
(Honorable mentions)

Many of these are tailored for dash boards, or interactive plots using dataframes, or for web purposes

1. Vega - uses json to make interactive plots with html or svg
2. Altair/Vega-altair - python based, uses the Vega
3. Ggplot2 - in R, based on a high-level framework for visualising dataframes
4. Plotnine - implementation of the ggplot2 framework but in python
5. d3.js and chart.js: These are javascript based, can make a variety of charts for web purposes

Dashboard/web deployments

There are **many** options, but I'll highlight two

Dash plotly

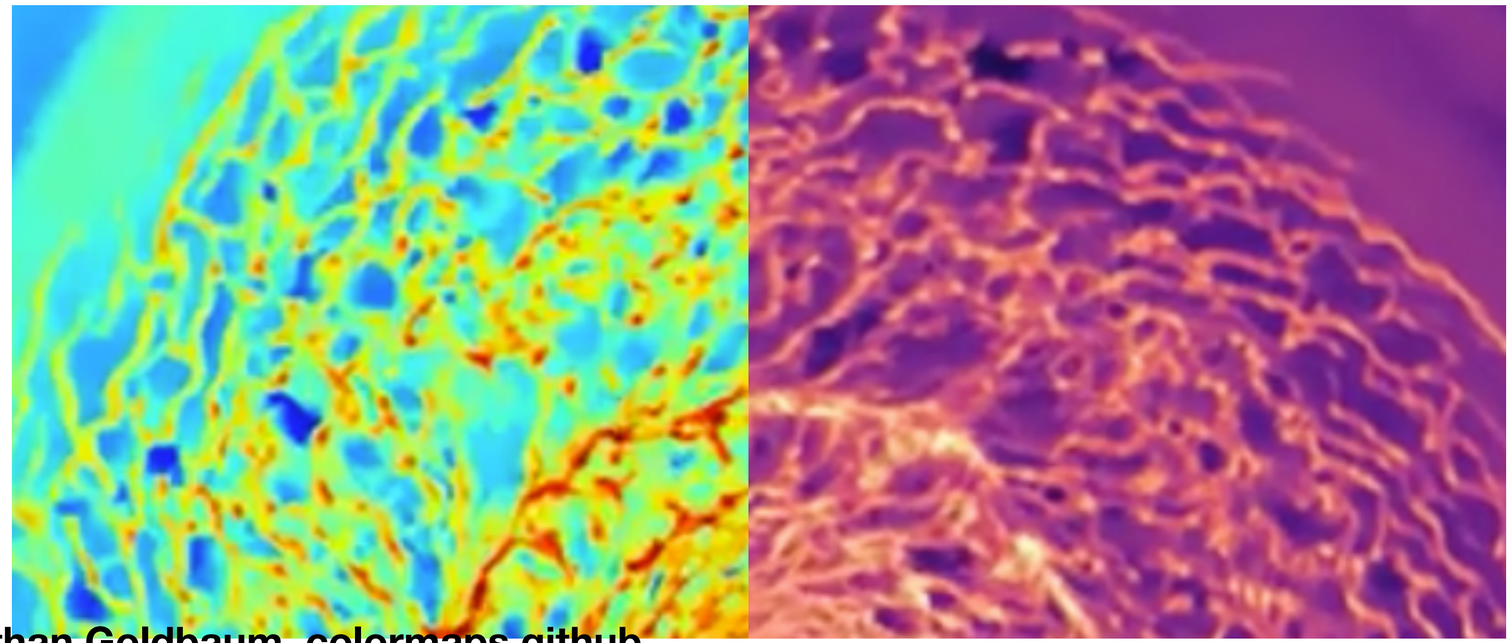
- **Pros:**
  - It is available in multiple languages (notably python, R and Julia)
  - Builds off of plotly, so if you're familiar with plotly there is less of a learning curve
  - Large community base for extensions
  - Can be run in Jupyter notebooks
- **Cons:**
  - It has a steeper learning curve. Lots of customisation options, but this also means there is more to learn and less "all-in-one" calls
  - No free included deployment option **from plotly**. Have to deploy somewhere like Heroku

Streamlit

- **Pros:**
  - It is purely pythonic with lots of "all-in-one" calls. Very simple to get started. It is very easy, too, to convert python scripts into streamlit apps
  - A large, and growing, community making widgets for streamlit
  - Can integrate html to some degree.
  - Can host for free on their site (although limited). This enables easy embedding in iframes
- **Cons:**
  - Much less customisation options
  - Better deployment must be done over other services like AWS (but has documentation)

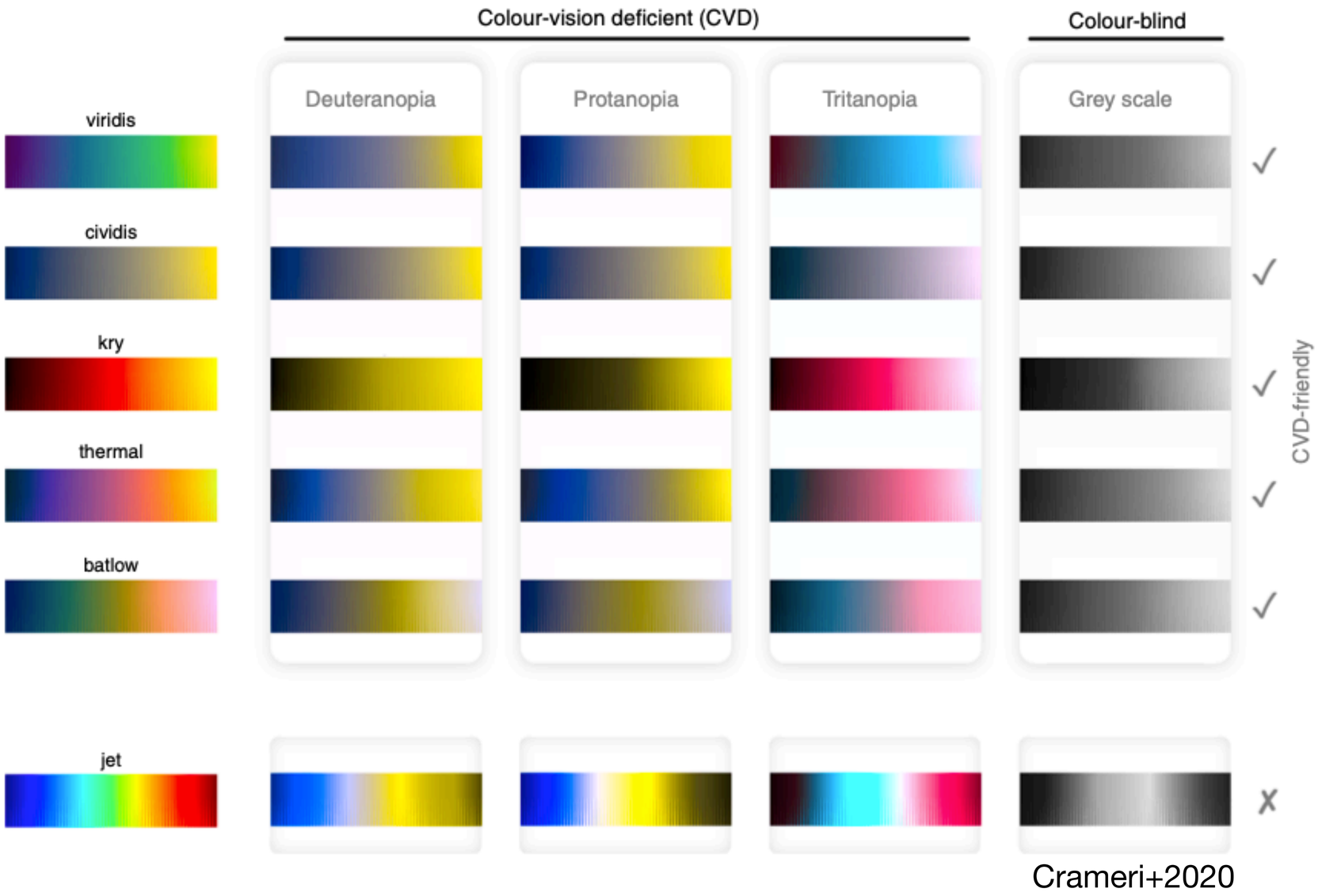
And now…a python interlude….
GitHub clone:
https://github.com/AstroBrandt/DataVisTutorial

Choosing the best colors for you and your audience



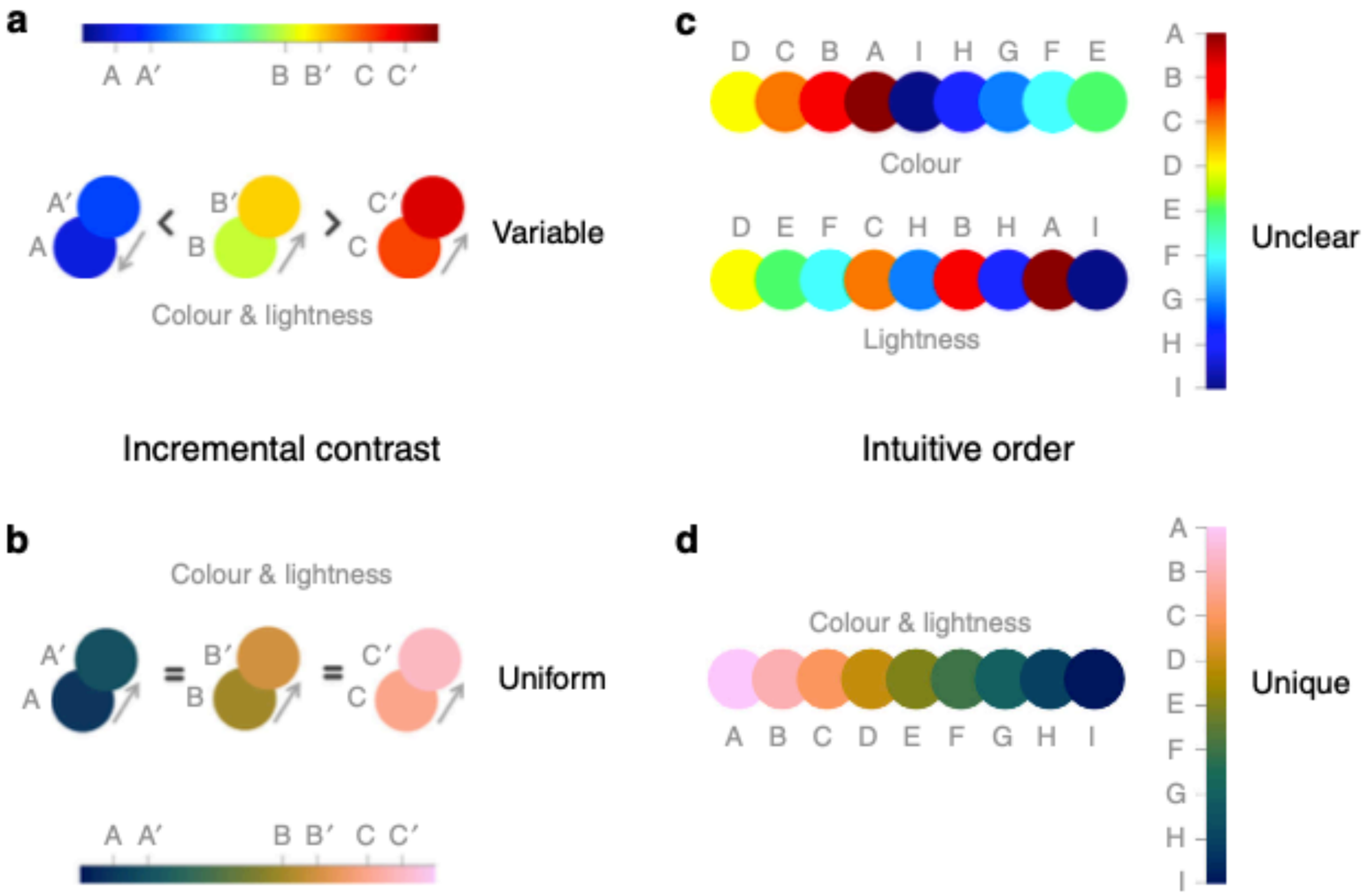
Nathan Goldbaum, colormaps github

Colorblind-proof your plots!



Crameri+2020

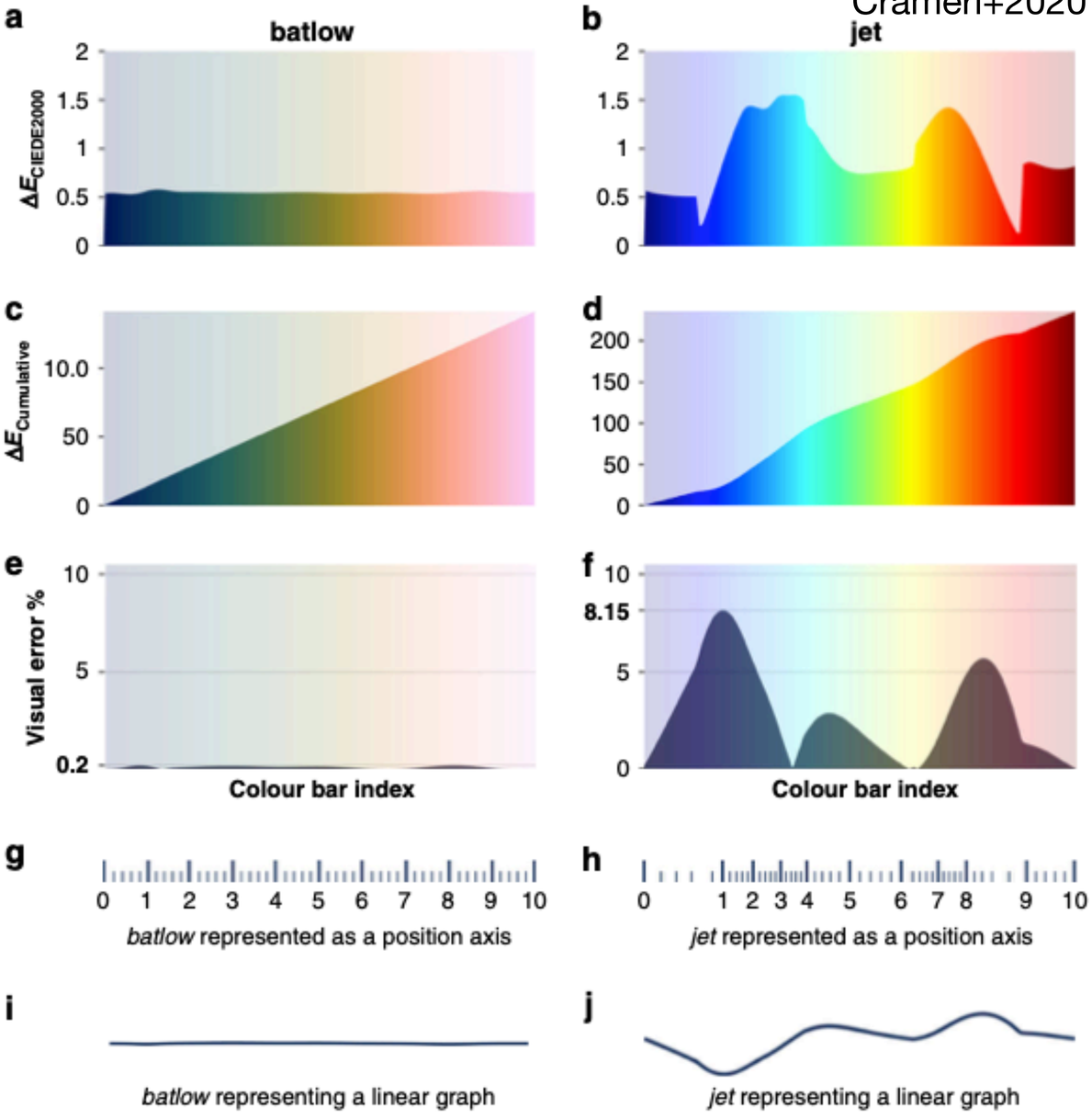
What is perceptual uniformity?

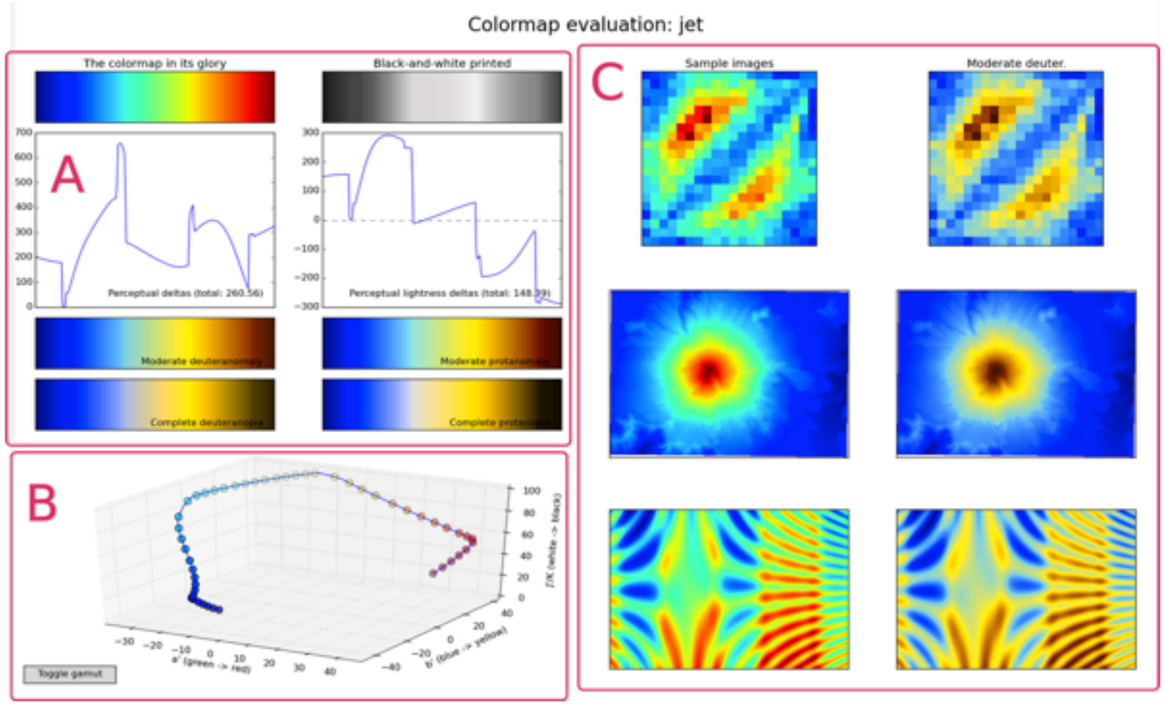


Crameri+2020

Checking your colormaps: Perceptual uniformity

Crameri+2020





Above: using viscm

How to:

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

Can make you OWN colormaps using:

python -m vidcm edit

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

What's wrong with the rainbow? An interdisciplinary review of empirical evidence for and against the rainbow color scheme in visualizations

Gołębiowska & Çöltekin 2022

Visualization Viewpoints

Editor:
Theresa-Marie Rhyne

Rainbow Color Map (Still) Considered Harmful

Borland & Taylor 2007

Data visualization: the end of the rainbow

Rogowitz & Treinish 1998

Rainbow color map distorts and misleads research in hydrology – guidance for better visualizations and science communication

Stoelzle & Stein 2021

Michael Stoelzle and Lina Stein

Spectral Schemes: Controversial Color Use on Maps

Cynthia A. Brewer

scientific visualization. Recent research, however, has shown that spectral schemes are preferred and are interpreted accurately when used as multi-hue renditions of diverging schemes. Both spectral and diverging schemes can emphasize a critical point within a data range with light colors and

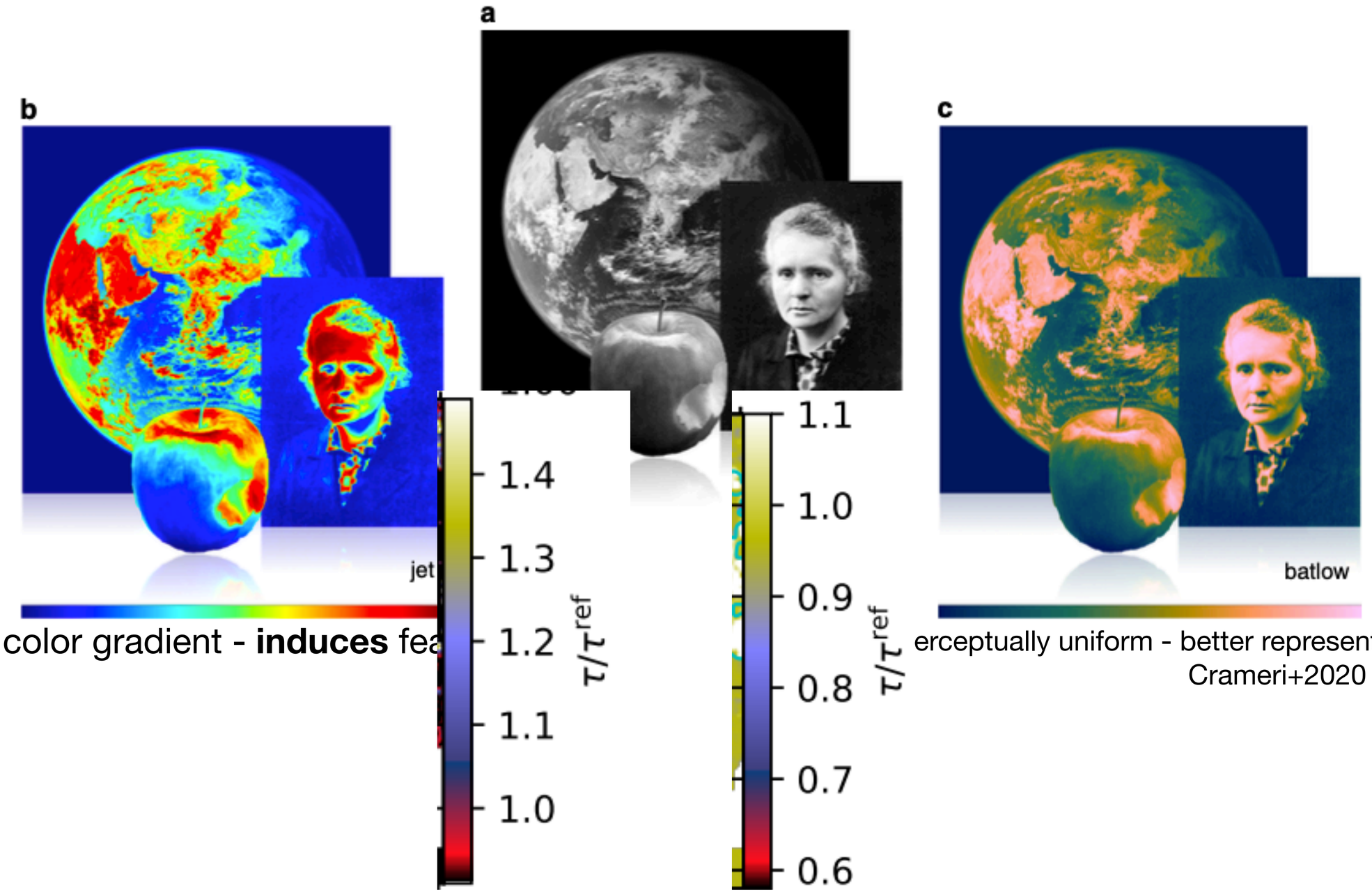
Brewer 1997 (of Colorbrewer)

Why We Use Bad Color Maps and What You Can Do About It

Stoelzle & Stein 2021

Kenneth Moreland; Sandia National Laboratories; Albuquerque, New Mexico, USA

Feature bias from colormap selection

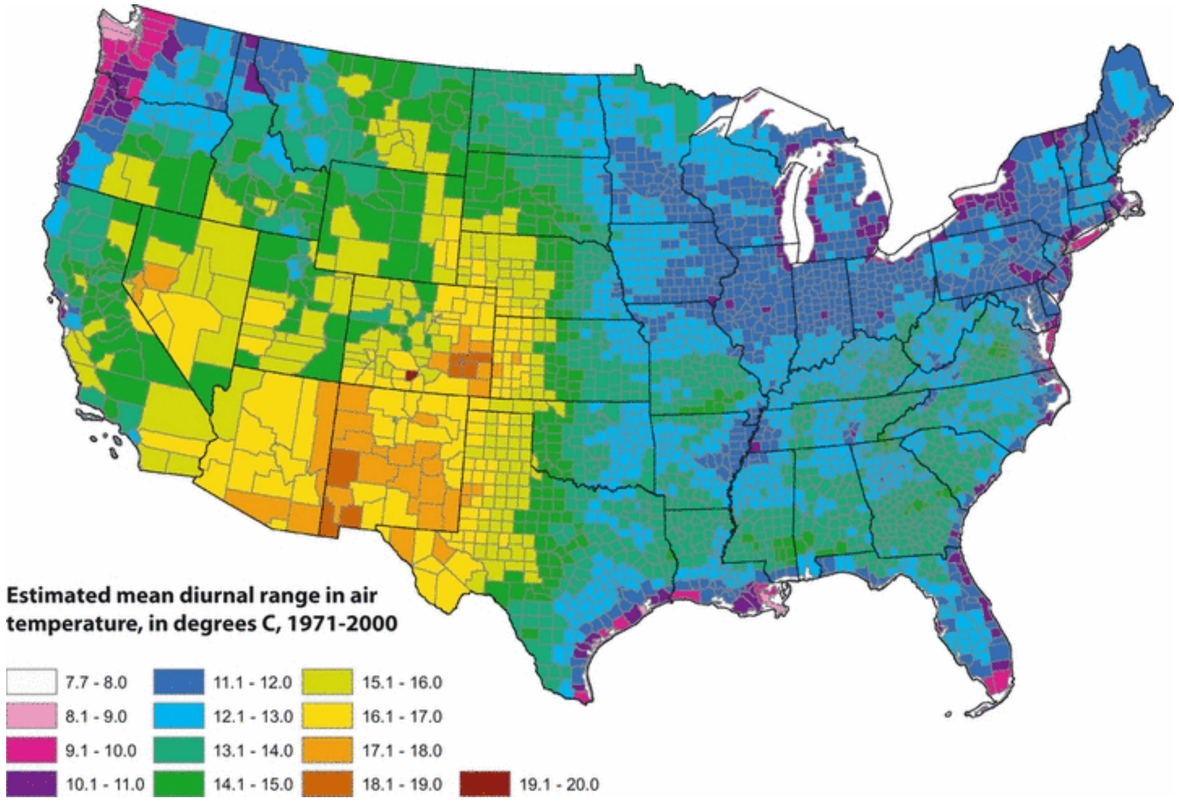


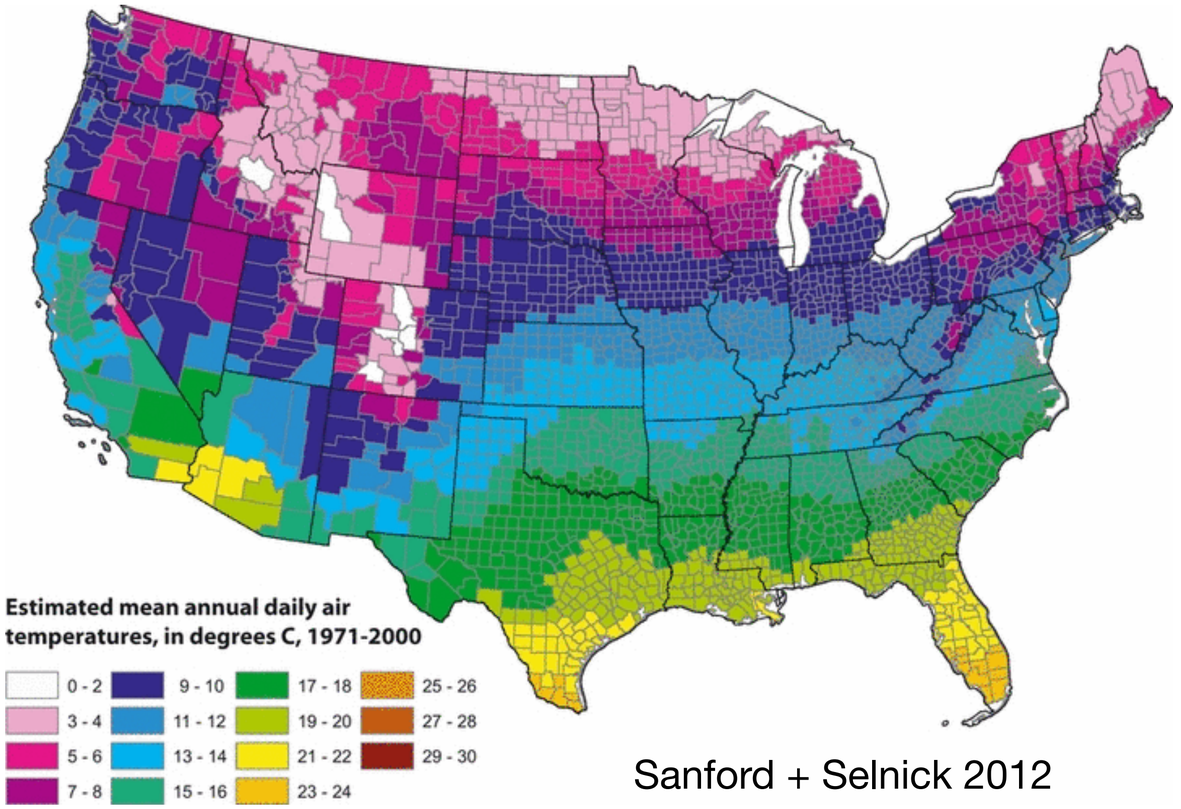
Big color gradient - **induces** fea

erceptually uniform - better represents

Crameri+2020

Feature bias from colormap selection



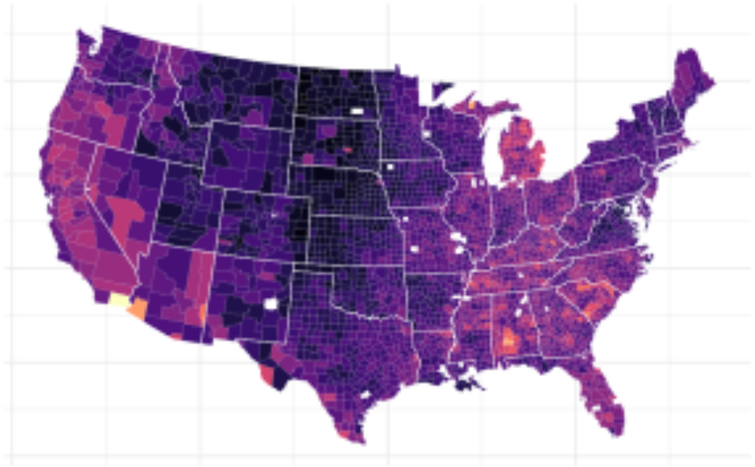


Sanford + Selnick 2012

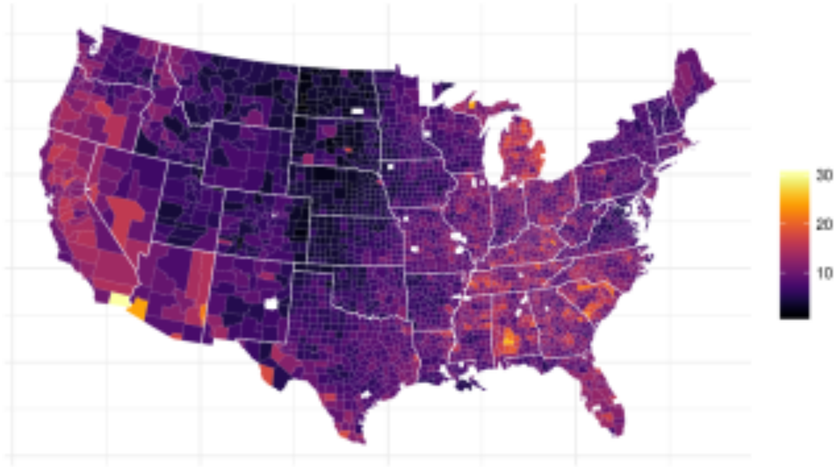
Feature bias from colormap selection

US unemployment rate by county

option A aka 'magma'

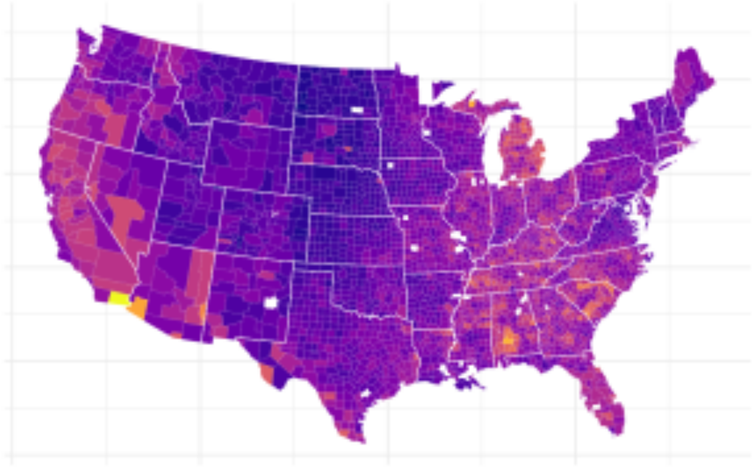


option B aka 'inferno'

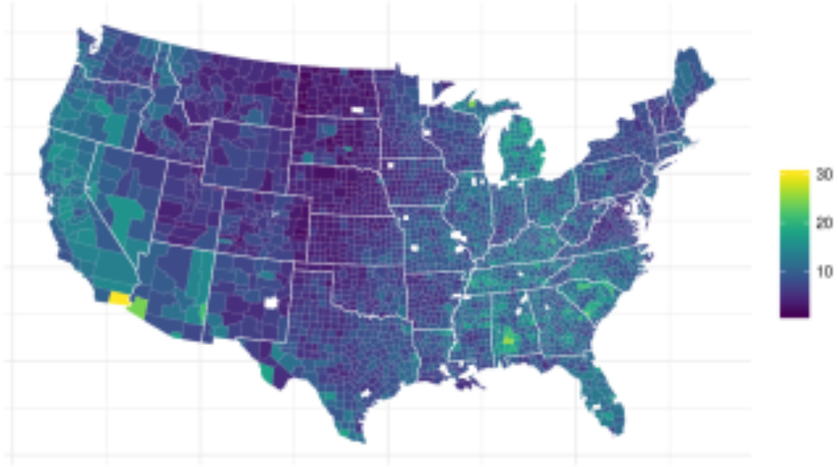


https://betterfigures.org/2015/06/23/picking-a-colour-scale-for-scientific-graphics/

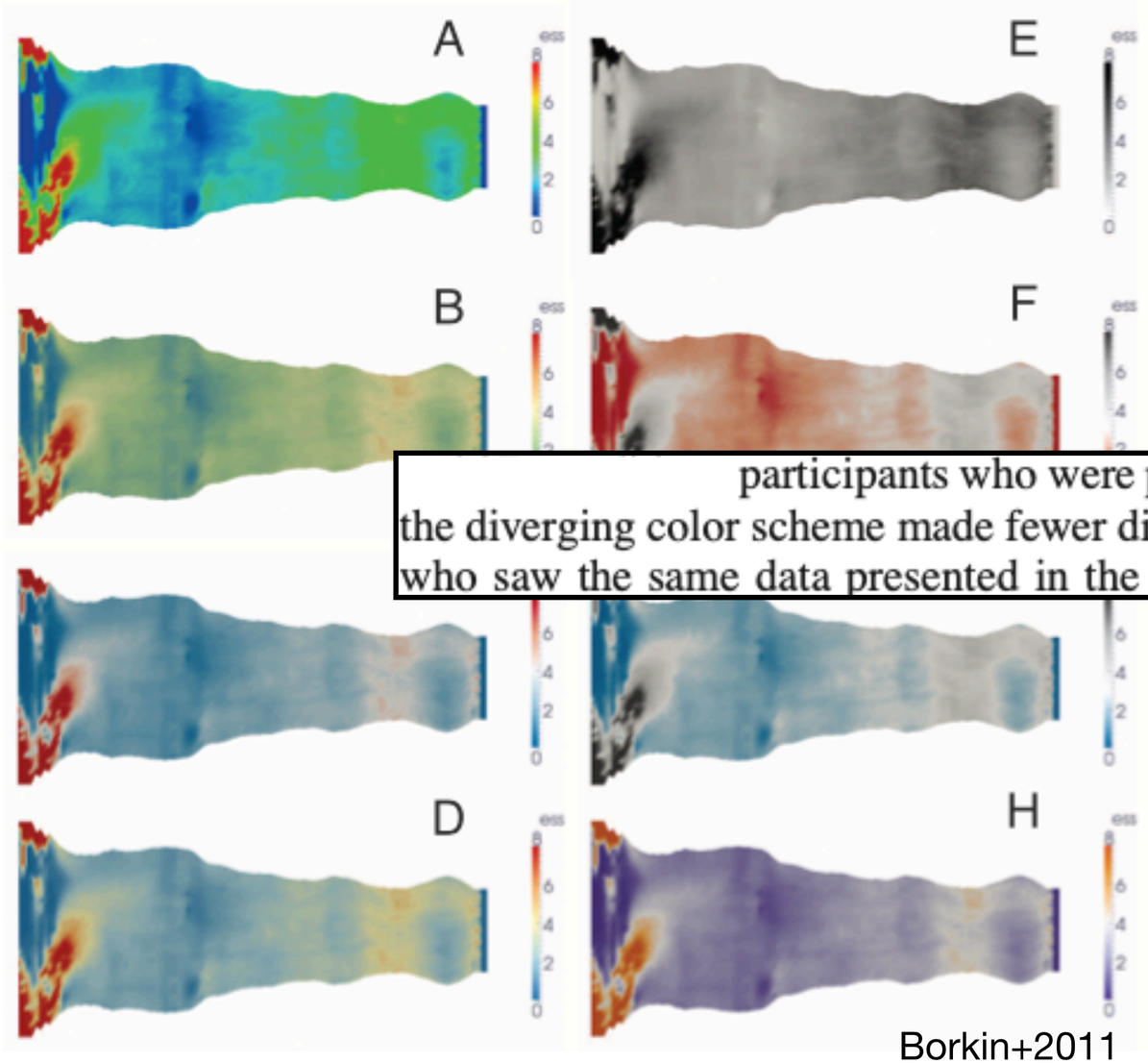
option C aka 'plasma'



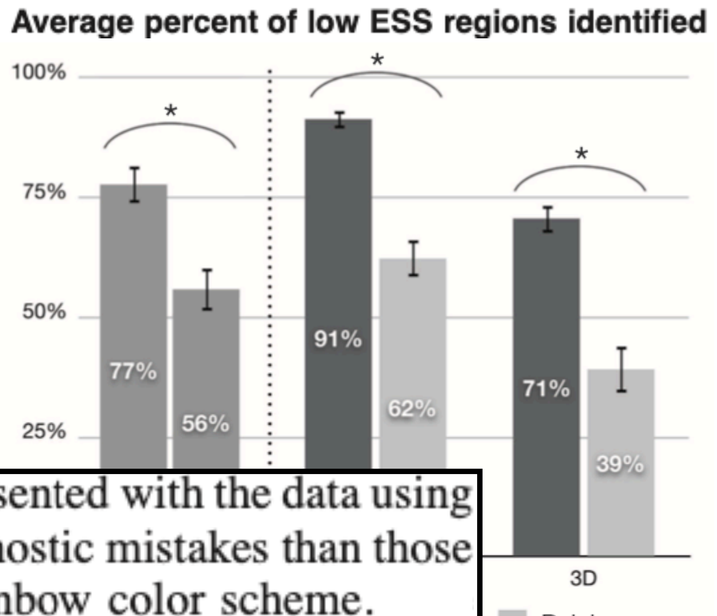
option D aka 'viridis'



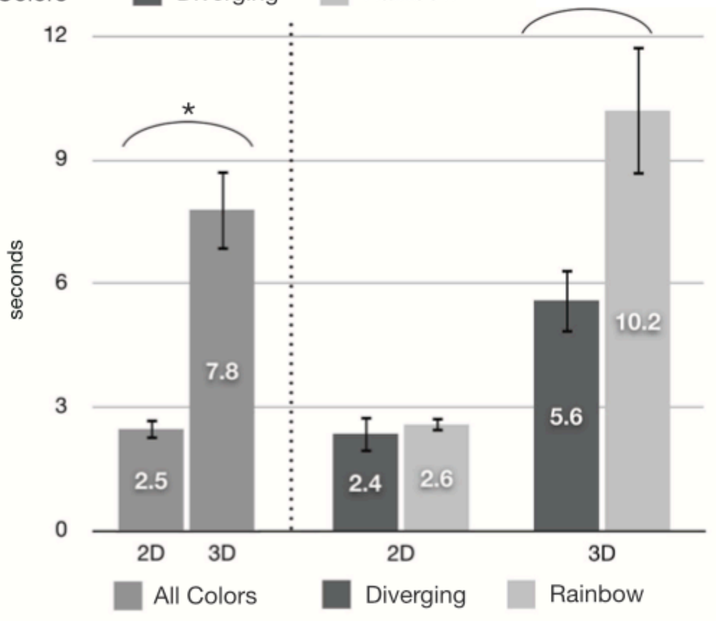
Colormap selection: a matter of life or death!



Borkin+2011

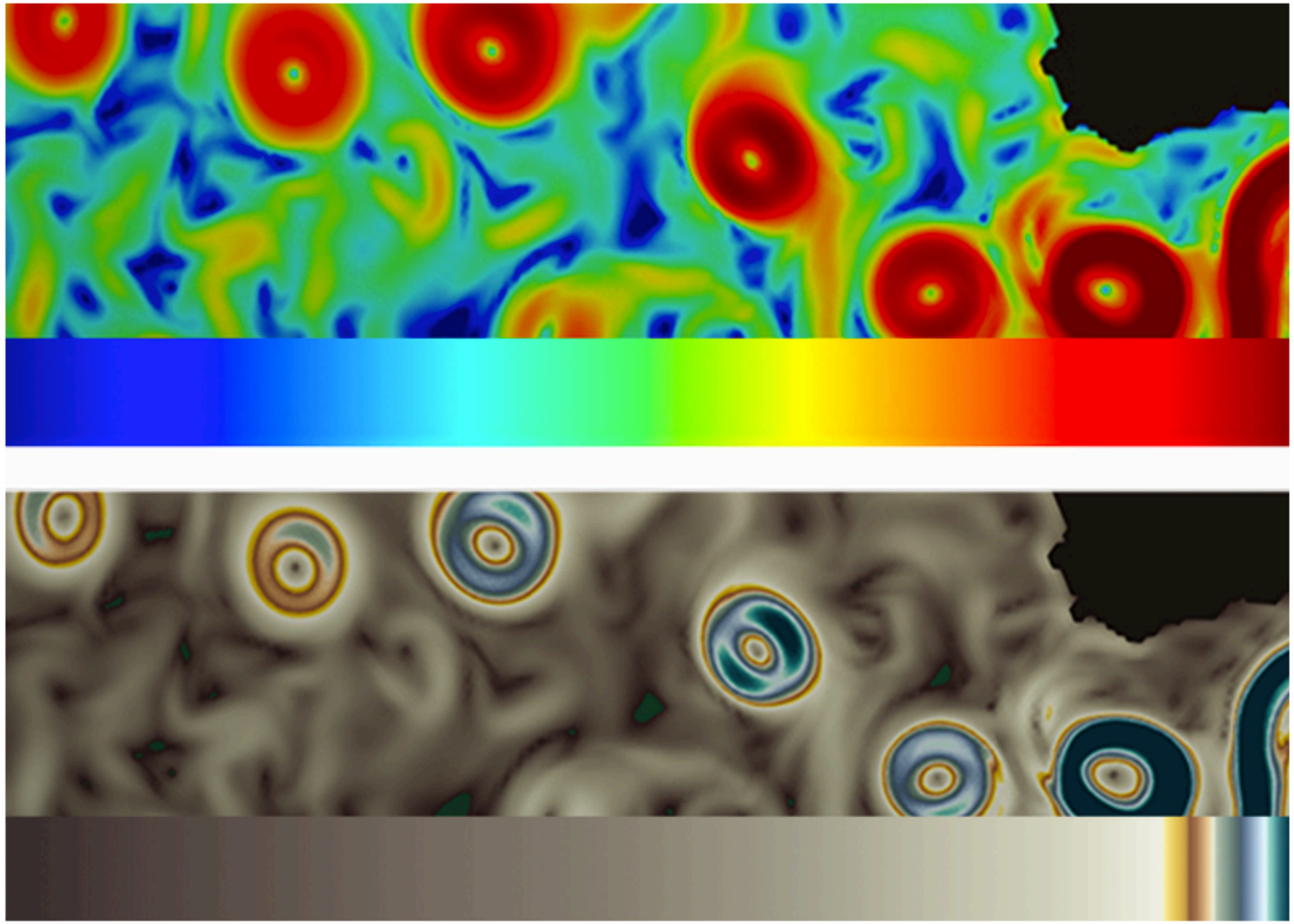


participants who were presented with the data using the diverging color scheme made fewer diagnostic mistakes than those who saw the same data presented in the rainbow color scheme.



Colormaps can help highlight **wanted** features!

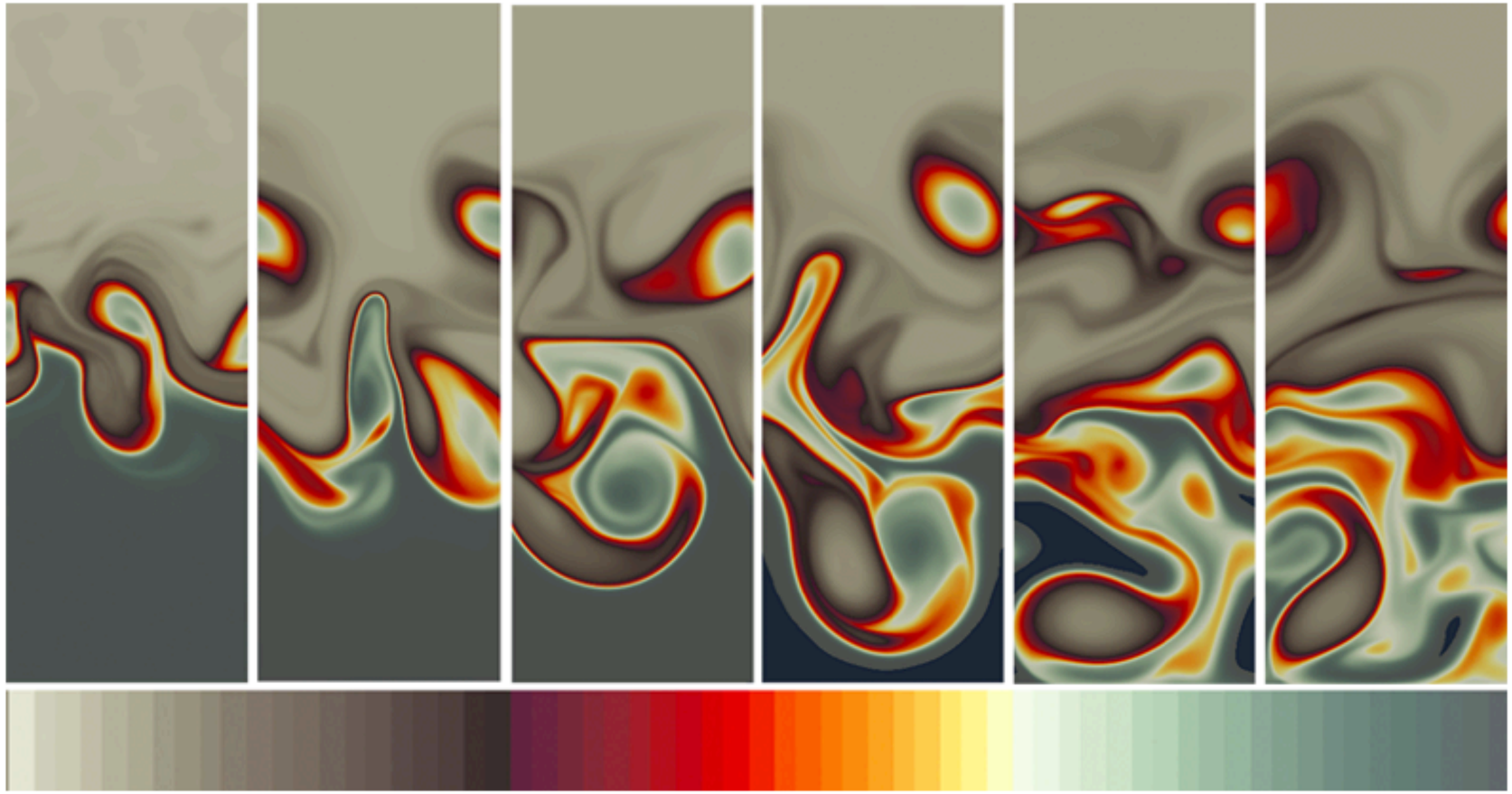
Example: want to highlight **eddies** in hydro flow



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…**more** python!

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/