How to make a plot

Ciaran O'Hare, USyd

Outline

1. Motivation and general advice
2. Designing a plot
3. Specific practical tips (matplotlib)
4. Examples of good/bad practice

Every plot in this presentation is reproducible via a Jupyter notebook.
Go to github.com/cajohare/HowToMakeAPlot

Apologies in advance, many slides are intentionally dense as I want this to be useful as a reference

Motivation: why bother?

- Figures are the most important part of a paper. As more and more papers appear on the arXiv every day, figures only grow in importance.
- Good plots grab attention and convey complex information quickly, helping you explain your science better and to more people. An attention to detail tells others that you ***care*** about being understood.
- Good figures help ***you***. If someone likes your plot, it is more likely they will cite it, use it in their talk and (hopefully) mention your name when doing so.
- Working hard on plots makes your science materially better. A figure you have spent more time working on is also a figure you have spent more time *looking* at. More likely to catch details, errors, room for improvement etc.

Getting started

- Spend *time* on your plots. Do not feel bad about spending upwards of days/weeks refining a plot that is going to convey the main result of a paper.

- An image will speak for itself, whether or not you intend it to. Make your plot in such a way that it is actually saying the thing you want it to say.

- Understand *what* the plot is saying by imagining what someone will think when they see it for the first time. If you can't imagine, just show it to someone and ask them what they think.

- Create a plot that is suitable for the setting in which it is presented. A good plot for a paper is not going to be a good plot for a talk or a poster. **Do not copy/paste plots from your paper into your slides.** (I am aware we all do this, but we shouldn't)

Plots are not scientific results

- Your plots are not scientific results in and of themselves, they are you ***communicating*** your scientific results. They are just as much propaganda as everything else you put in your paper, or say during a talk.

- With that in mind, do not just dump all of your data onto a plot and expect people to draw their own conclusions. Instead, consider what your plots actually are: **a visual description of a quantitative result.** Don't treat them as graphical databases, but rather as an image which is intended to convey a message.

*"Your paper is not what you did, it is what you **say** you did"*

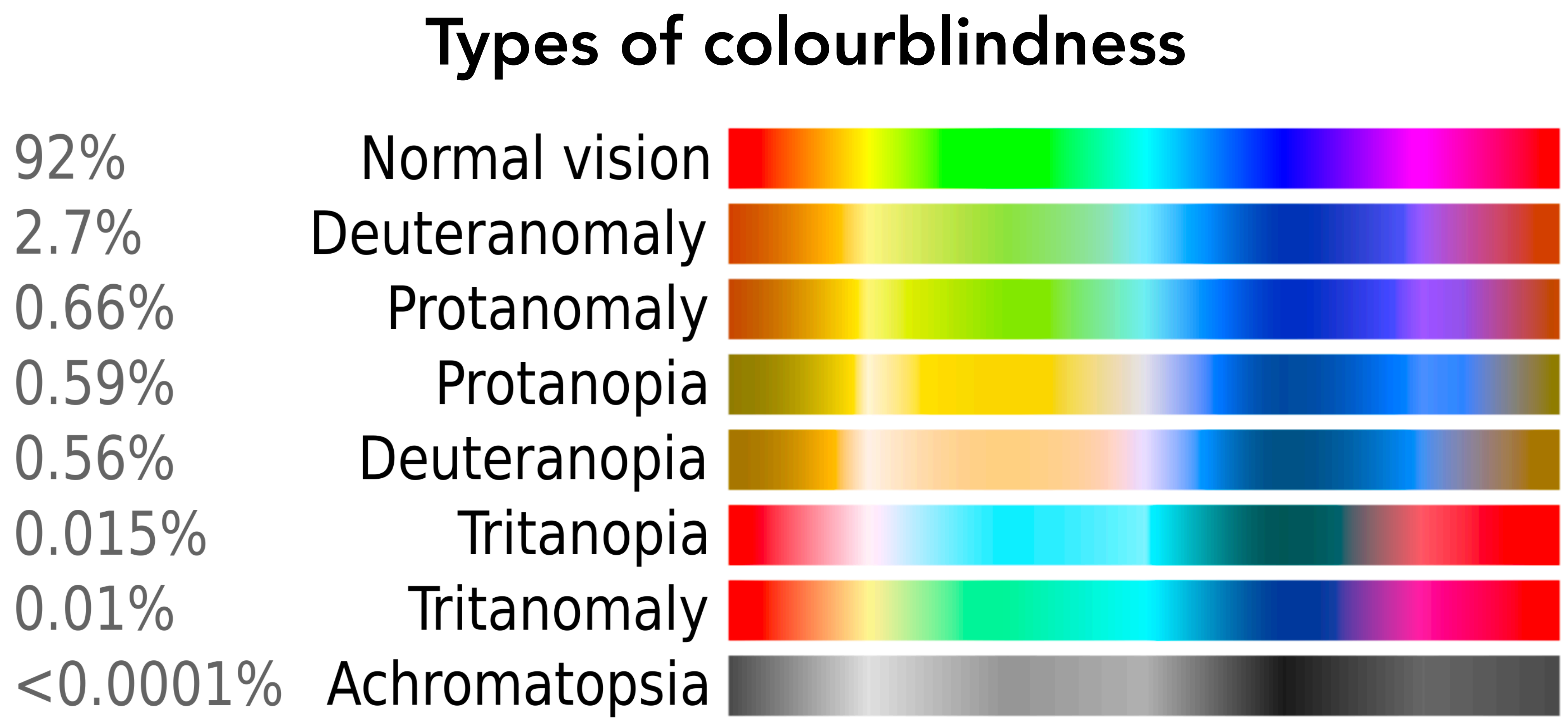
— A clever thing someone said to me once but I forgot who

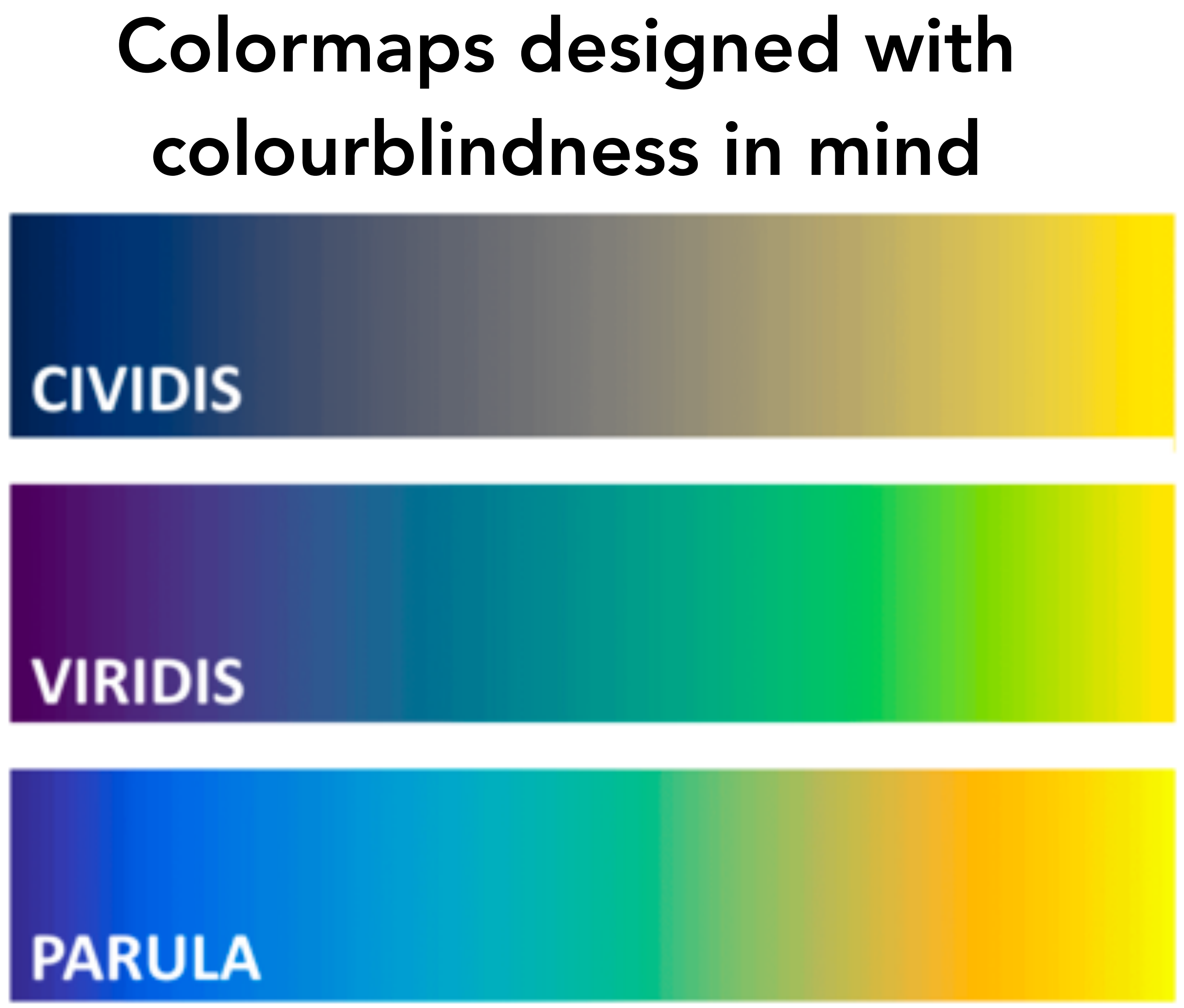
Designing plots

Non-negotiables

- **Careful with your use of colour.** ~4% of people have some form of colourblindness. There are [online tools](#) to apply filters to your plots to check if they're readable to those people. Best cautionary measure is to not have essential messages tied up in interpreting detailed colour information.
  (Or just avoid red and green next to each other)

- Be mindful about **file size**. If people can't load the paper pdf on their browser, it doesn't even matter what your plot looks like. Only in extreme cases will a plot be > 1 MB.





General technical tips

- **Cut your losses.** Python, Mathematica, MATLAB etc. are powerful, but can be very annoying when it comes to certain things. No shame in putting the final touches using something with a visual user interface like powerpoint, keynote, inkscape, illustrator, photoshop etc.

- Always make plots in a **vector graphics** format, unless there is no other option

- **Watch font sizes.** No font on a plot should appear smaller than the font of the main body of paper/slide. Ideally, it should be bigger, big enough to read when the pdf is zoomed out, or for a person at the back of the room.

Making good plots requires "soft" and "technical" skills

Soft

- An understanding of how humans perceive and process visual information
- An eye for detail
- Intuition for when something is unattractive or "off"

Technical

- Ability to translate an idea into code
- Knowing how to avoid graphical artefacts
- Knowledge of fonts, colour theory, pre-attentive visual attributes etc.

Making good plots requires "soft" and "technical" skills

Soft

- An understanding of how humans perceive and process visual information
- An eye for detail
- Intuition for when something is unattractive or "off"

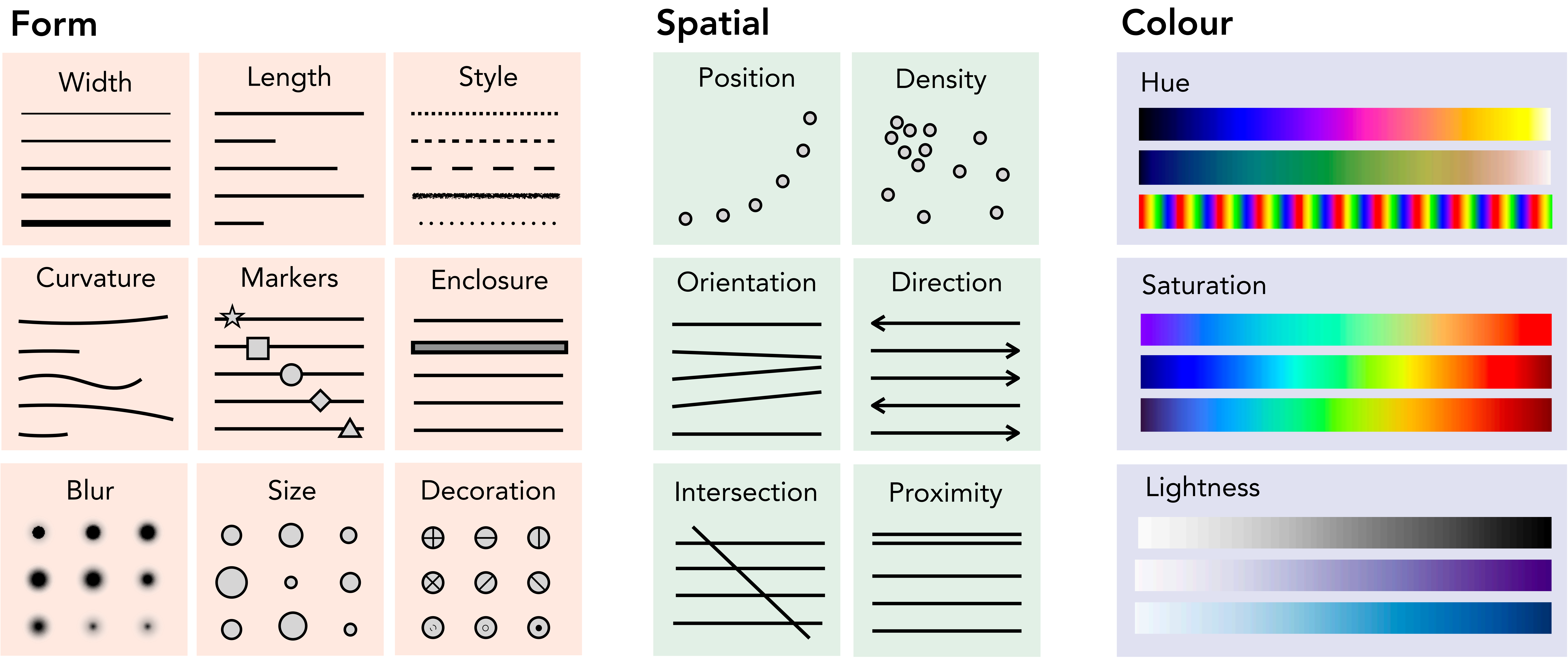
Technical

- Ability to translate an idea into code
- Knowing how to avoid graphical artefacts
- Knowledge of fonts, colour theory, pre-attentive visual attributes etc.

Disclaimer: I am not qualified to lecture you on either, but I can give you tips from experience. You are also welcome to disagree with me on anything that I say.

Pre-attentive visual attributes

Attributes your spatial memory subconsciously processes in the first few milliseconds when looking at an image. This is the "*Standard Model*" of information design. You can combine any and all of them to convey both quantitative and qualitative information.



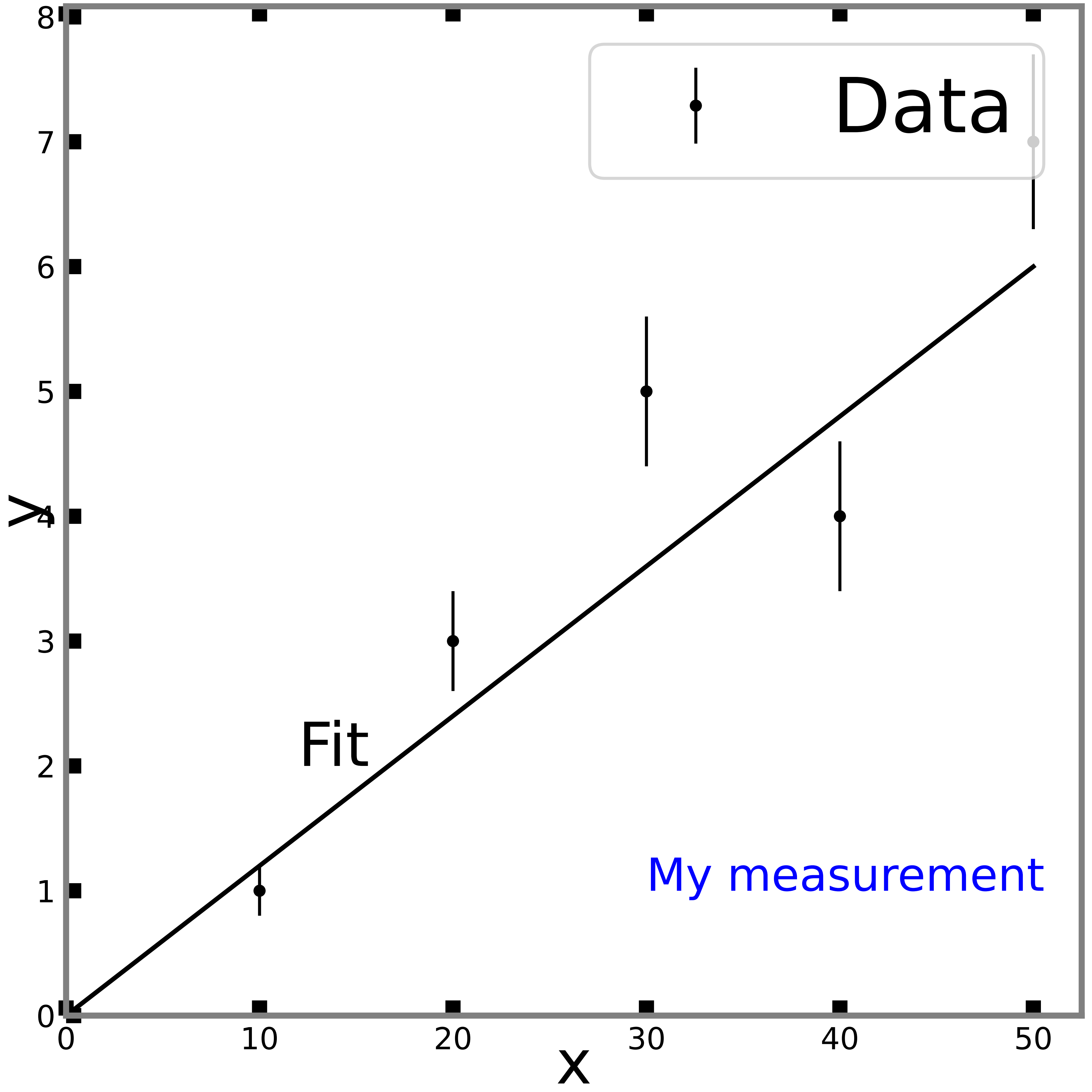
Using pre-attentive attributes

- Every plot is built out of pre-attentive visual attributes automatically. Your task is to **control** which ones you use and to remove ones which appear by accident. Make sure every element is purposeful and supports the main message.

- The reason most bad plots ***look*** bad is because of sloppy use of these attributes, i.e. **visual clutter**. Individually these may seem like small things, but our brains are tuned to spot anything out of place.

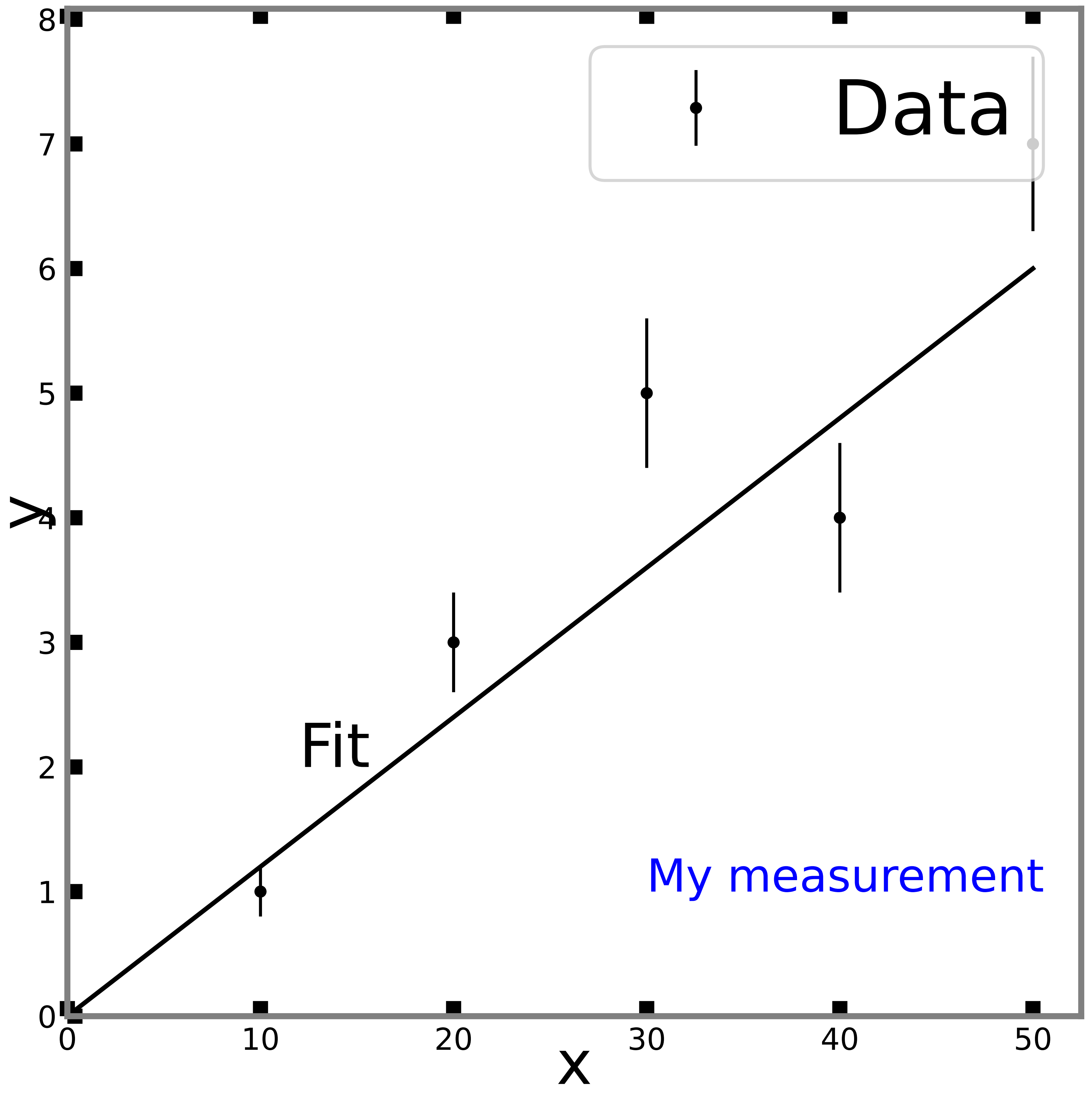
- Even seemingly trivial things like asymmetries, misalignment, or overlapping elements can stack up and degrade the intelligibility of a plot.

Does this plot look okay to you?



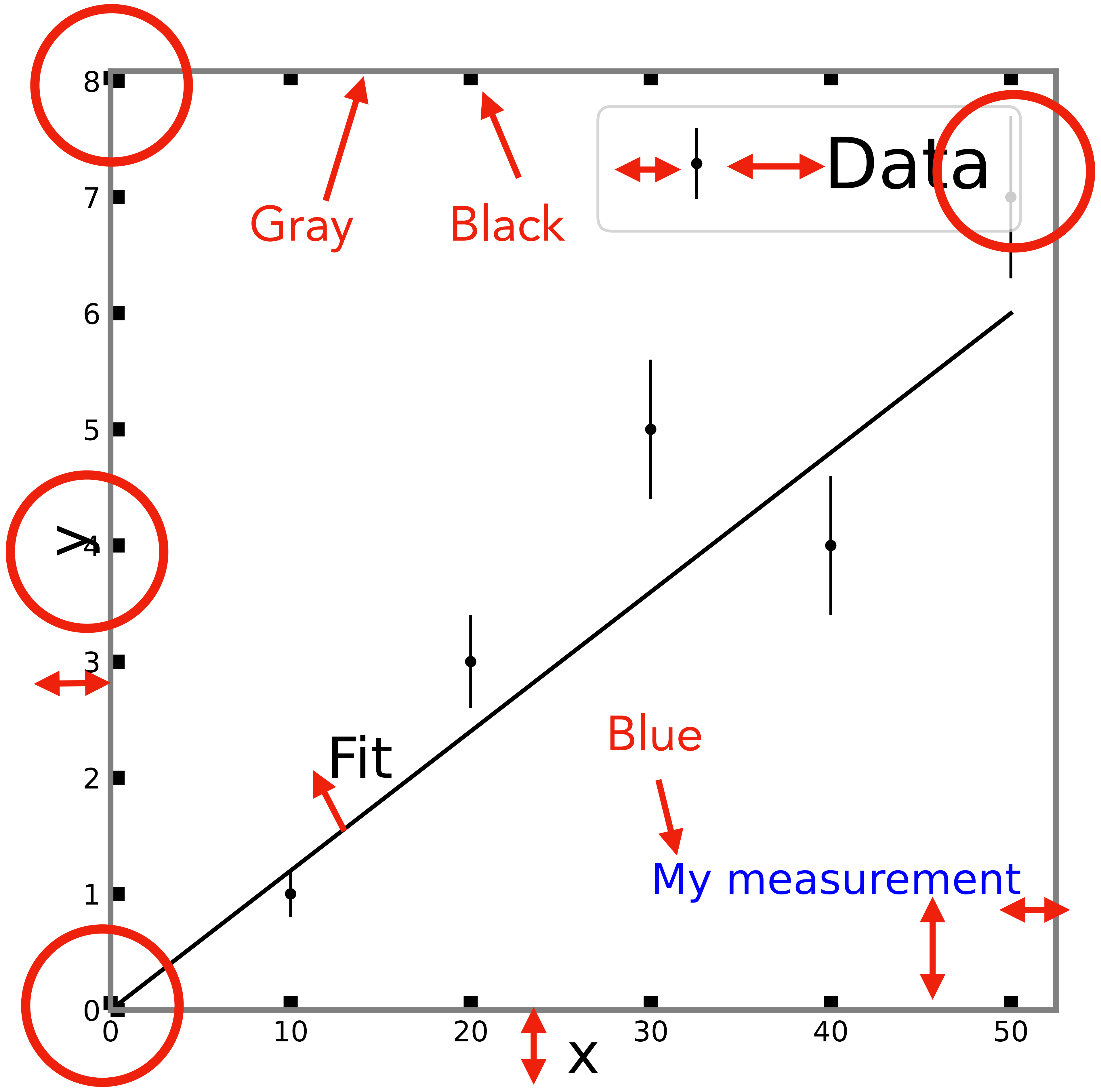
Using pre-attentive attributes

You can instantly improve the visual just by removing visual elements that were in the plot unintentionally



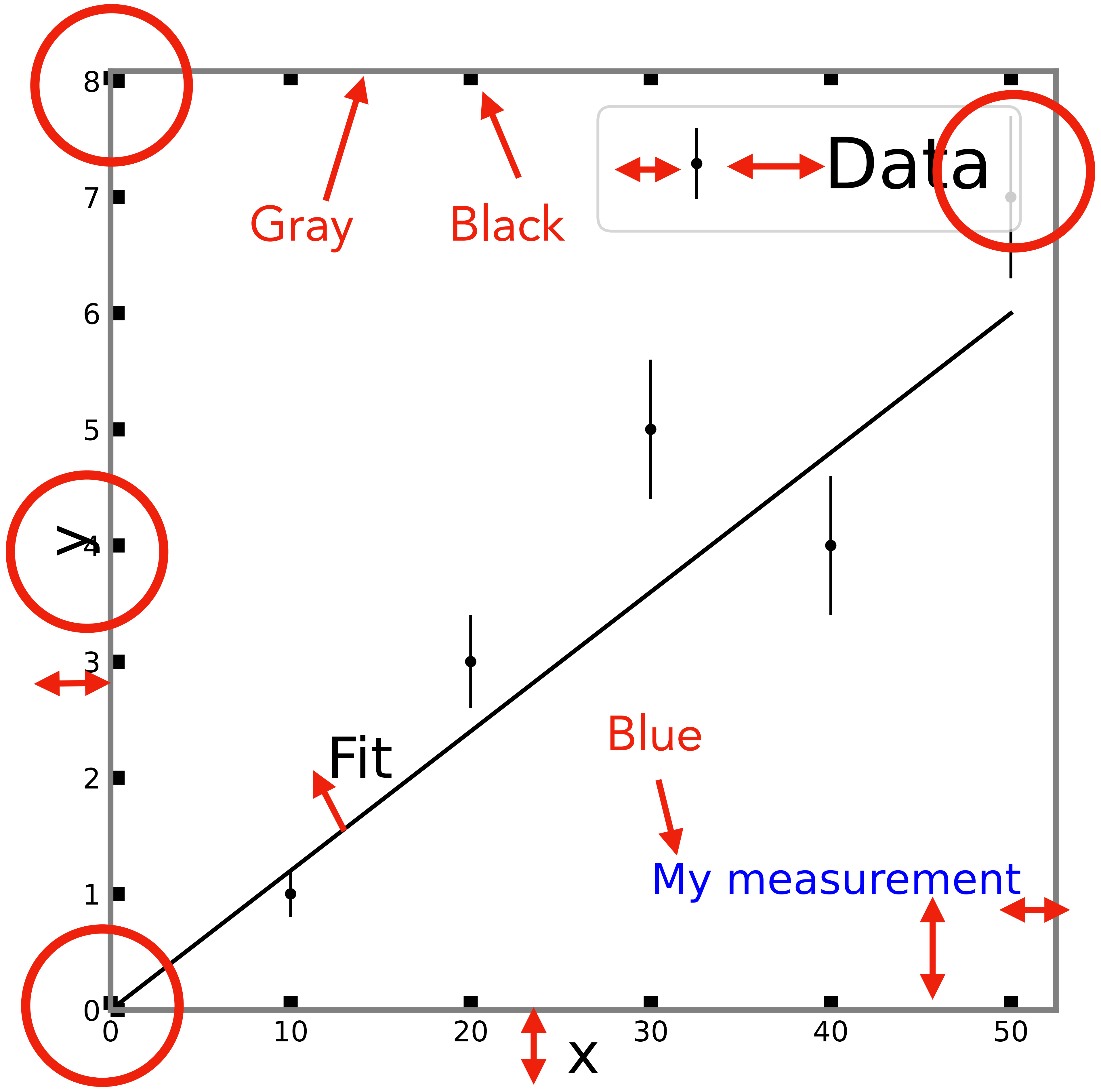
Using pre-attentive attributes

You can instantly improve the visual just by removing visual elements that were in the plot unintentionally

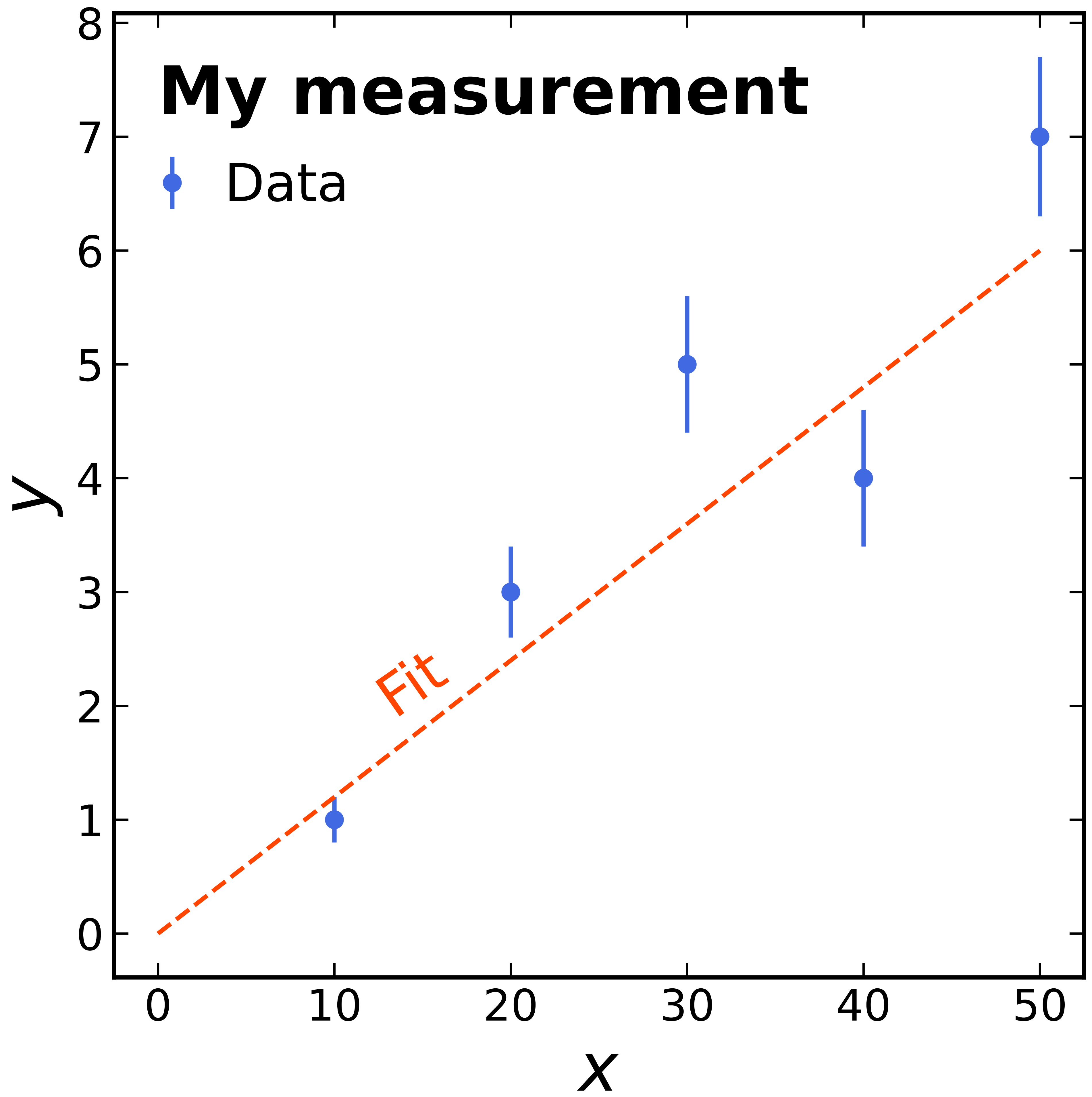


Using pre-attentive attributes

You can instantly improve the visual just by removing visual elements that were in the plot unintentionally



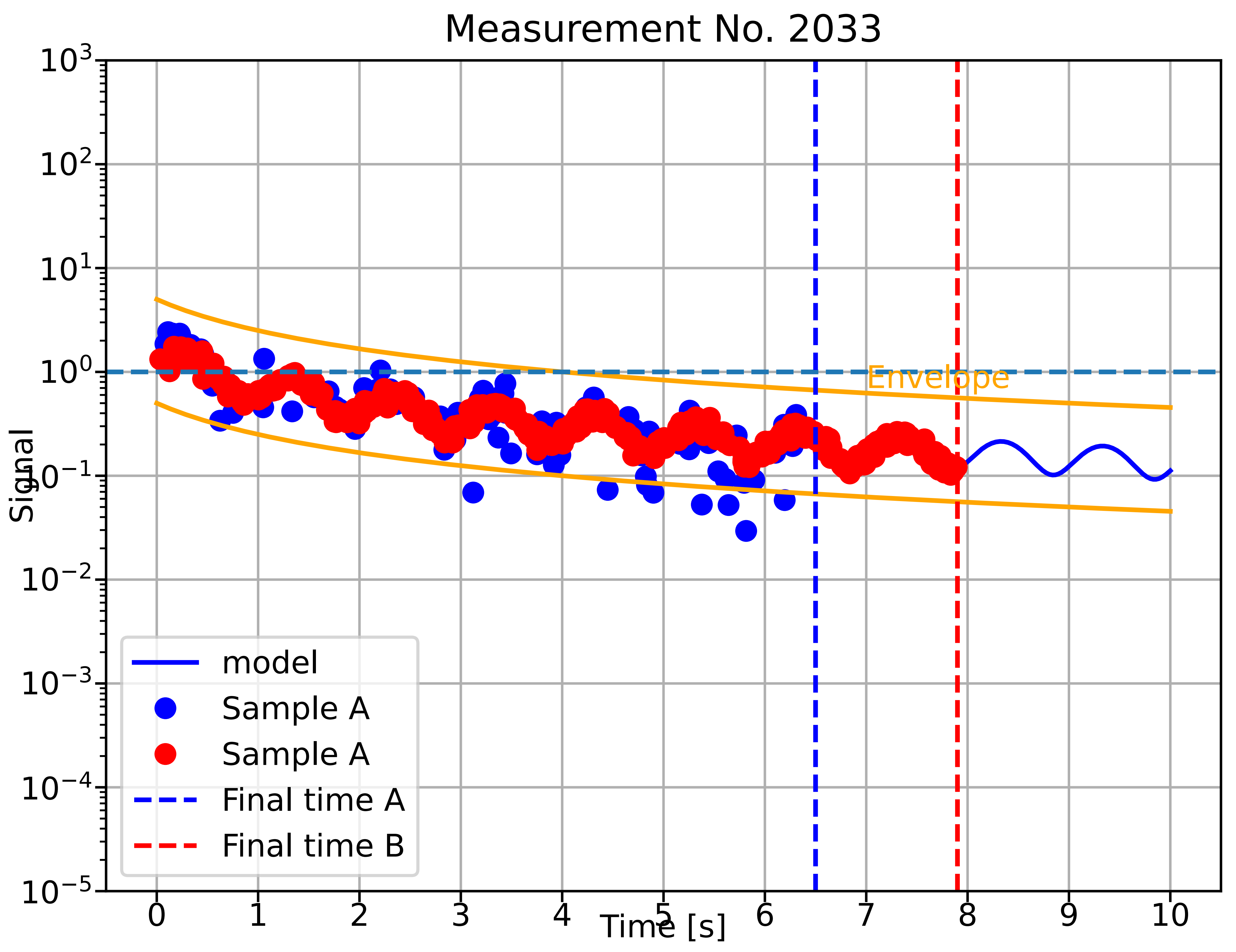
Clutter.ipynb

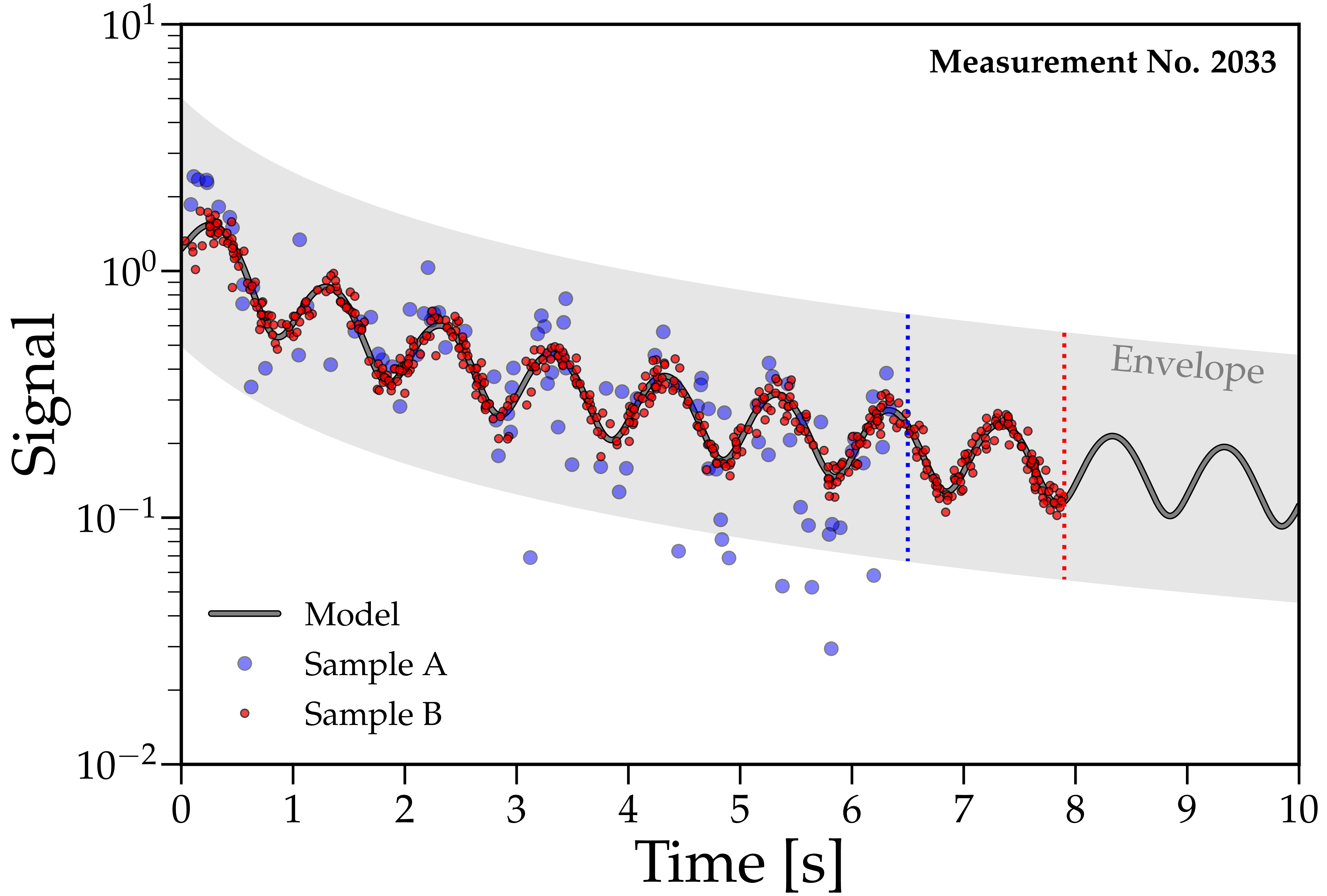


Improving visual clarity with better design

Avoid any element that might confuse or confound interpretability. Make sure all use of pre-attentive attributes is **intentional**. Guide the reader's eye around the plot so that they see your message as quickly as possible.

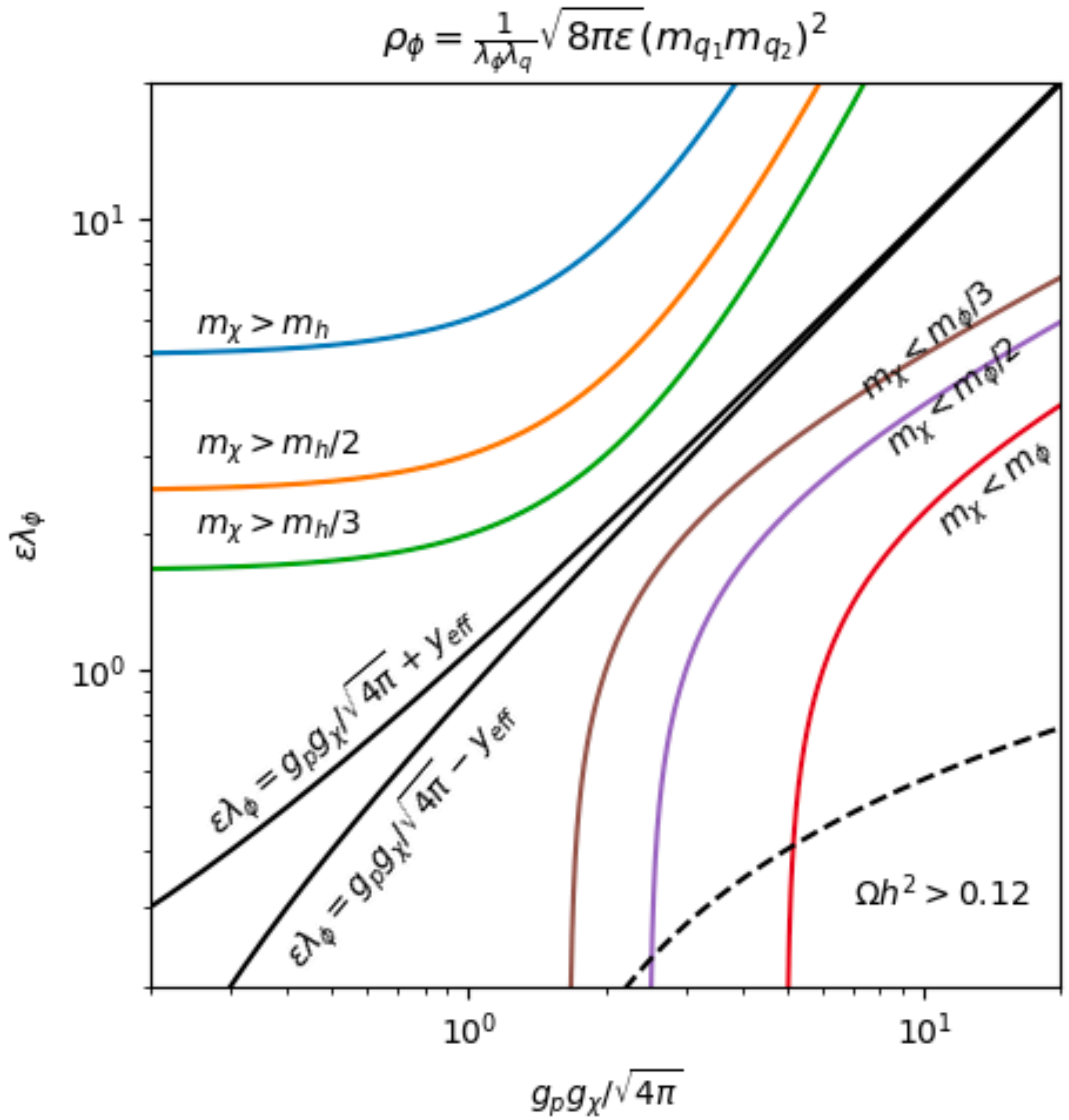
VisualClarity.ipynb





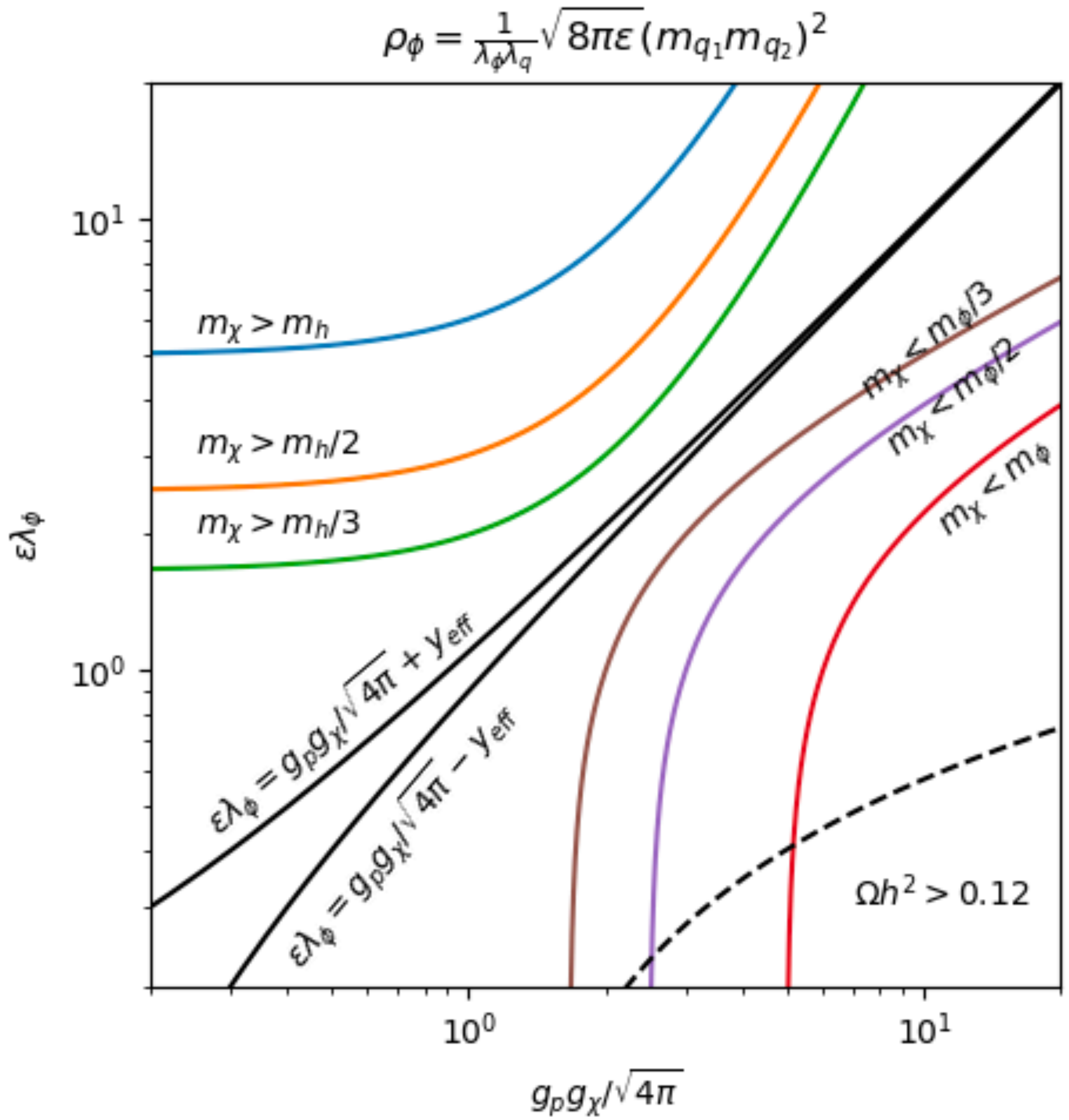
Label your plots properly

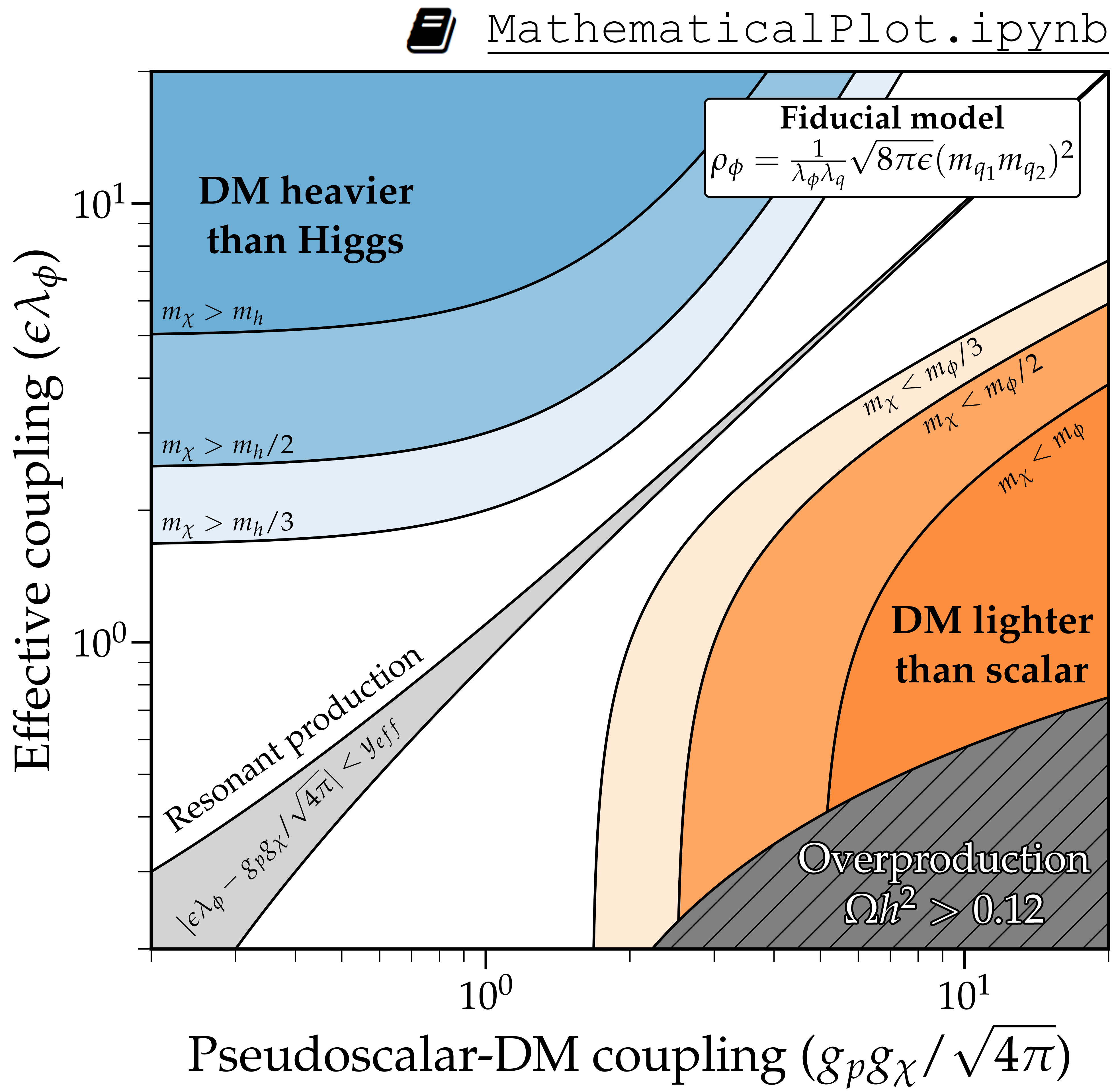
Not labelling your plot using English is the best way to make your reader's life as difficult as possible. Don't make them hunt down the definition of every variable name, just spell out what the plot is saying. This is especially important for plots used in talks.

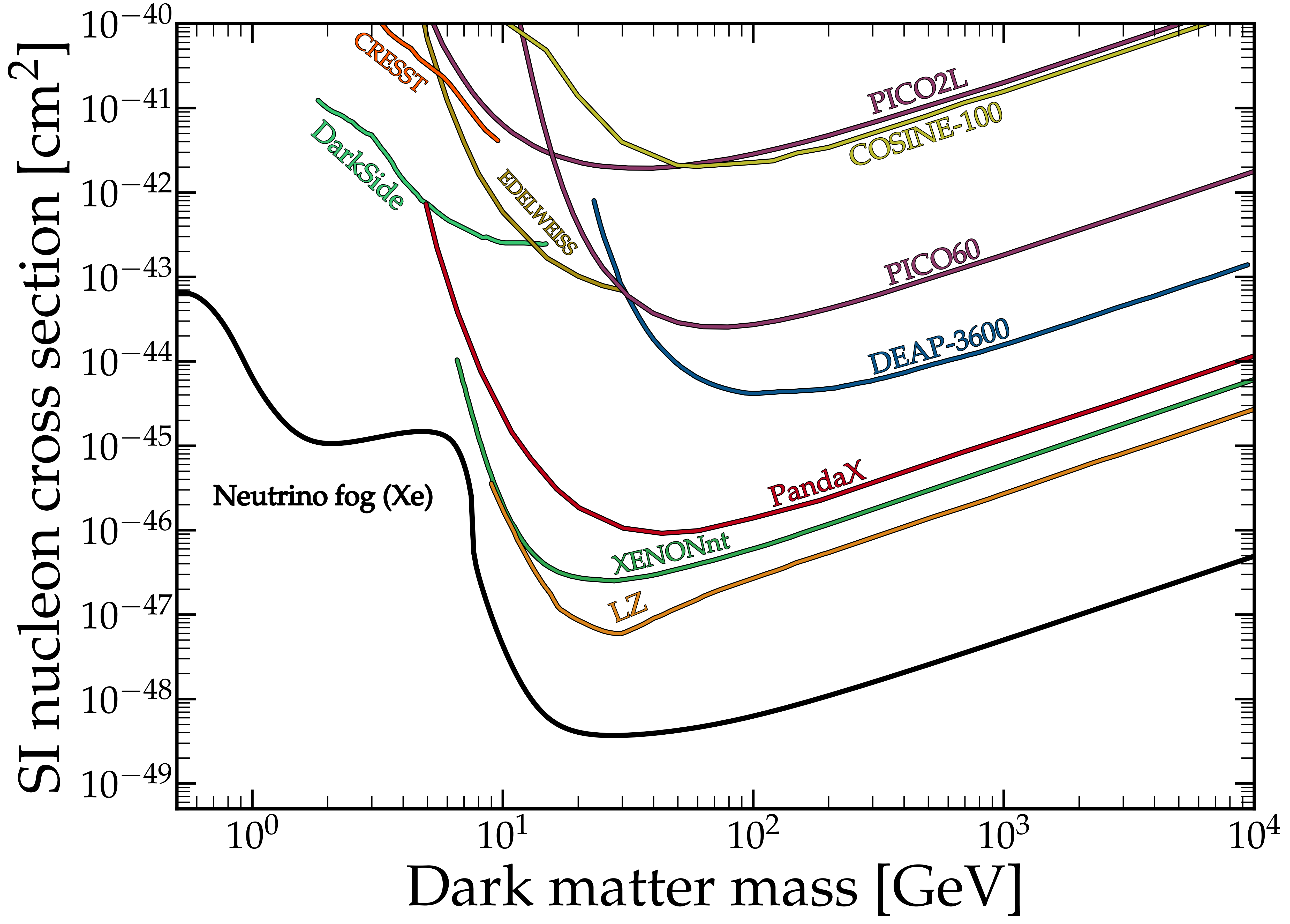


Label your plots properly

Not labelling your plot using English is the best way to make your reader's life as difficult as possible. Don't make them hunt down the definition of every variable name, just spell out what the plot is saying. This is especially important for plots used in talks.



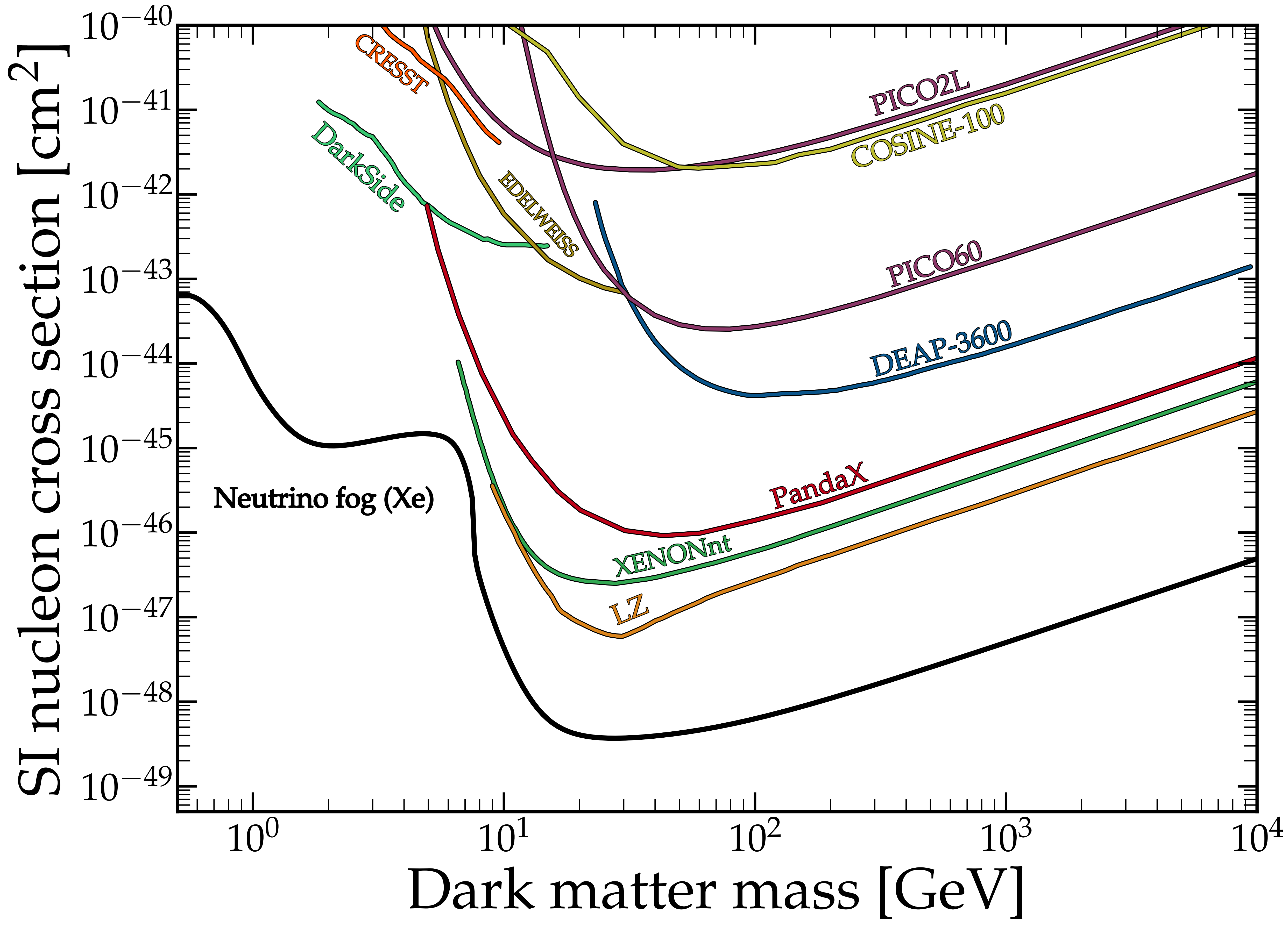




SI nucleon cross section [cm^2]
Dark matter mass [GeV]
10^{-40}
10^{-41}
10^{-42}
10^{-43}
10^{-44}
10^{-45}
10^{-46}
10^{-47}
10^{-48}
10^{-49}
10^{0}
10^{1}
10^{2}
10^{3}
10^{4}
CRESST
DarkSide
EDELWEISS
PICO2L
COSINE-100
PICO60
DEAP-3600
PandaX
XENONnt
LZ
Neutrino fog (Xe)

Inequalities and exclusion limits

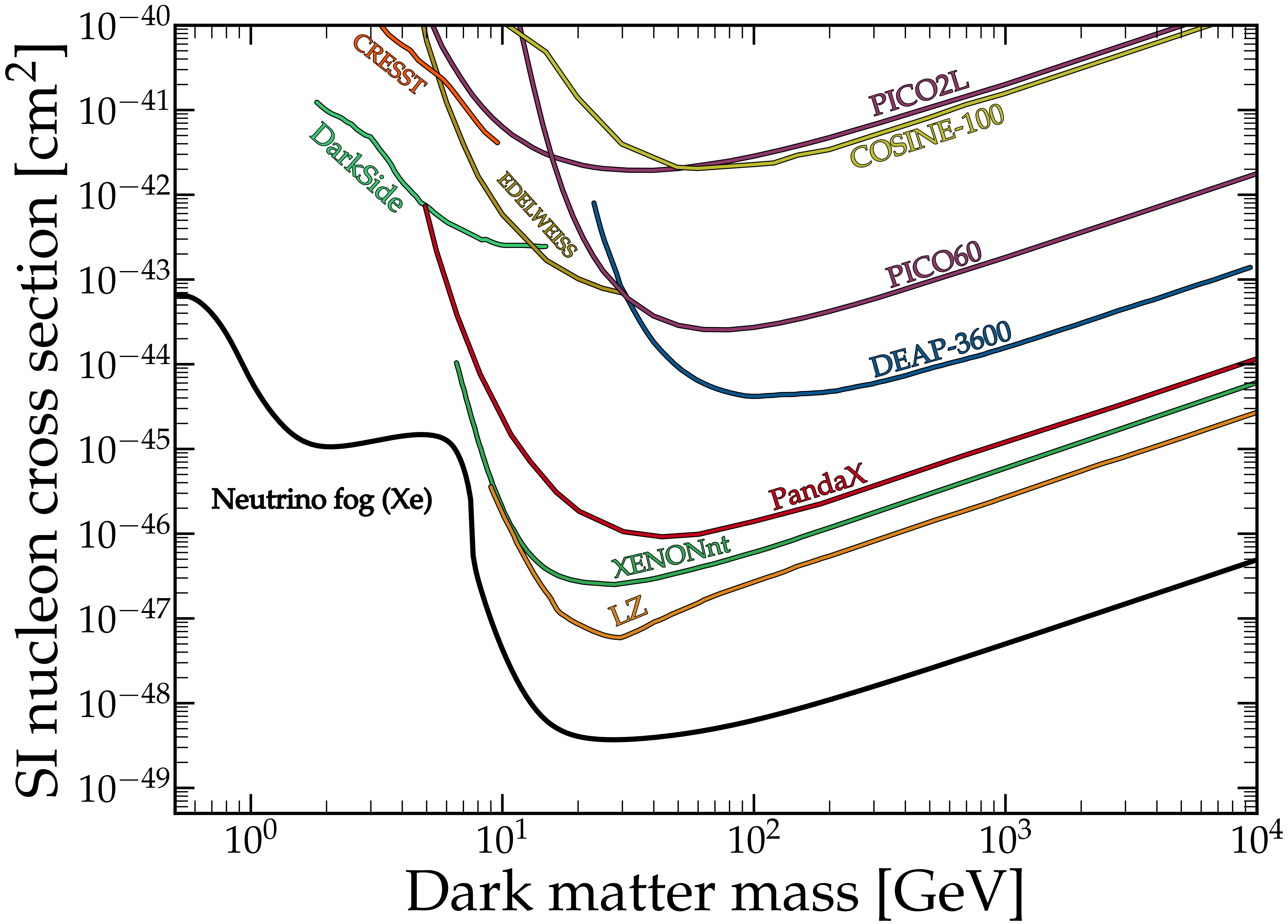
One of my pet peeves is when people use naked lines as opposed to filled regions to display inequalities, e.g. exclusion limits. How is an uninitiated reader supposed to know which part is excluded?

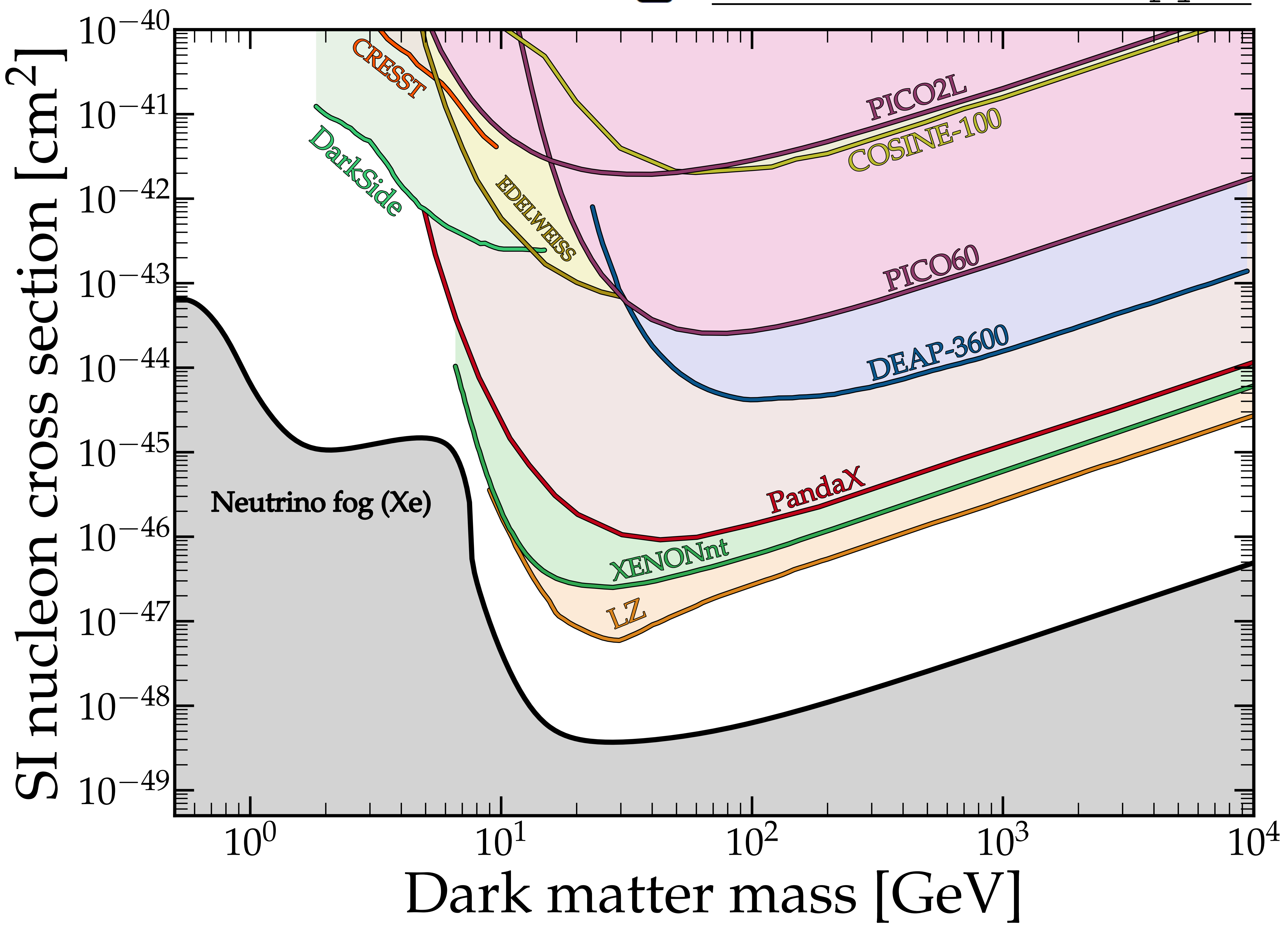


Inequalities and exclusion limits

One of my pet peeves is when people use naked lines as opposed to filled regions to display inequalities, e.g. exclusion limits. How is an uninitiated reader supposed to know which part is excluded?

ExclusionLimits.ipynb





Tables versus figures

Tables are sometimes essential, e.g. if you need to record lots of numbers and the relationships between them. However, if the takeaway message of your paper is conveyed solely via referring to numbers in a table, consider turning that message into a figure

	^{238}U (µBq/kg)	^{232}Th (µBq/kg)	^{40}K (µBq/kg)	^{60}Co (µBq/kg)	^{222}Rn (µBq/kg)
Run 1	910.3 ± 182.1	730.9 ± 146.2	1050.3 ± 210.1	720.2 ± 144.0	80.3 ± 16.1
Run 2	20.3 ± 4.1	11.2 ± 2.2	29.1 ± 5.8	13.5 ± 2.7	3.3 ± 0.7
Run 3	2.3 ± 0.3	1.2 ± 0.2	2.1 ± 0.3	1.5 ± 0.2	0.3 ± 0.0

(And it goes without saying that tables are terrible things to show during talks)

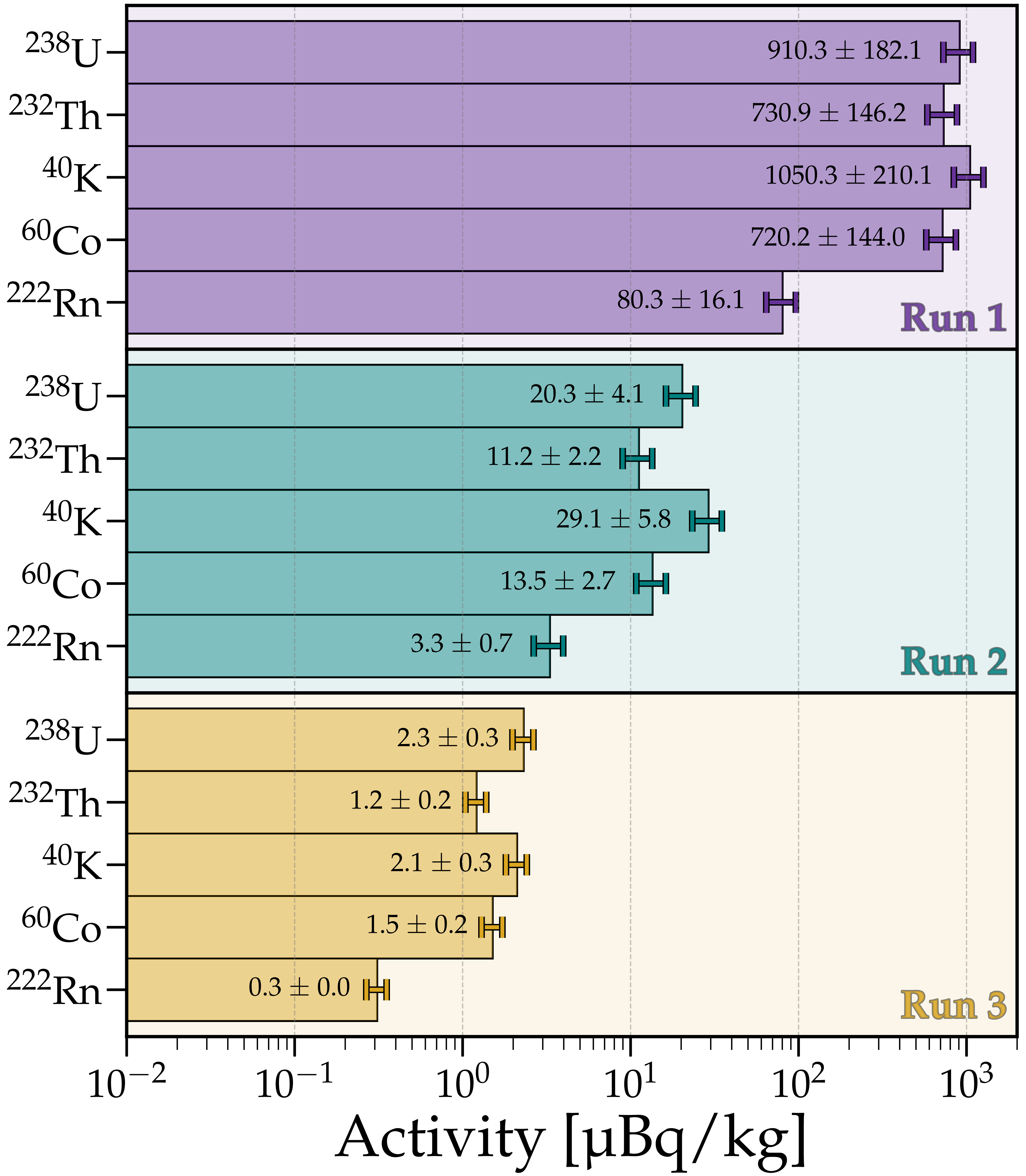
Tables versus figures

TablesVersusFigures.ipynb

Tables are sometimes essential, e.g. if you need to record lots of numbers and the relationships between them. However, if the takeaway message of your paper is conveyed solely via referring to numbers in a table, consider turning that message into a figure

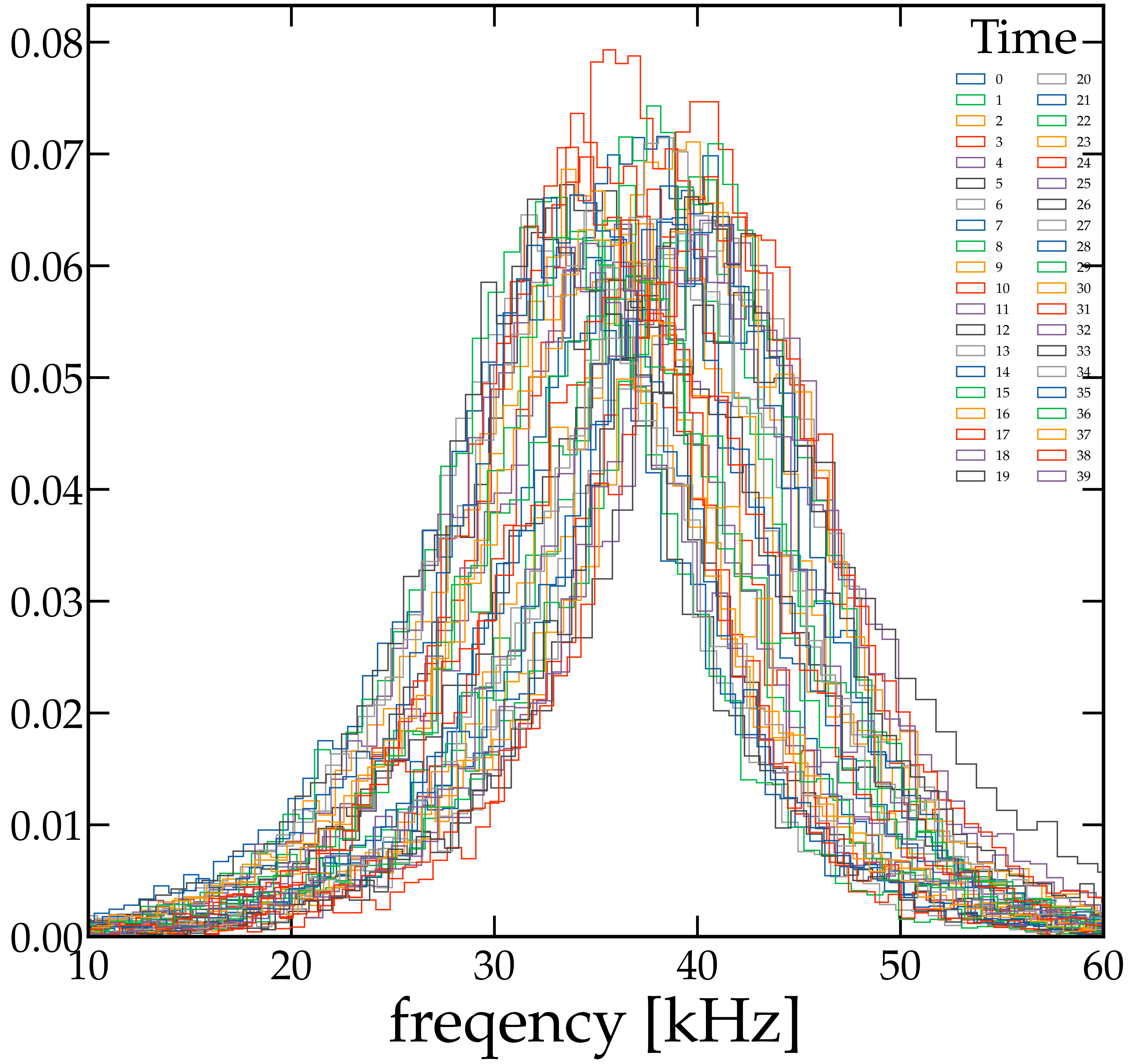
	^{238}U (μBq/kg)	^{232}Th (μBq/kg)	^{40}K (μBq/kg)	^{60}Co (μBq/kg)	^{222}Rn (μBq/kg)
Run 1	910.3 ± 182.1	730.9 ± 146.2	1050.3 ± 210.1	720.2 ± 144.0	80.3 ± 16.1
Run 2	20.3 ± 4.1	11.2 ± 2.2	29.1 ± 5.8	13.5 ± 2.7	3.3 ± 0.7
Run 3	2.3 ± 0.3	1.2 ± 0.2	2.1 ± 0.3	1.5 ± 0.2	0.3 ± 0.0

(And it goes without saying that tables are terrible things to show during talks)



Histograms

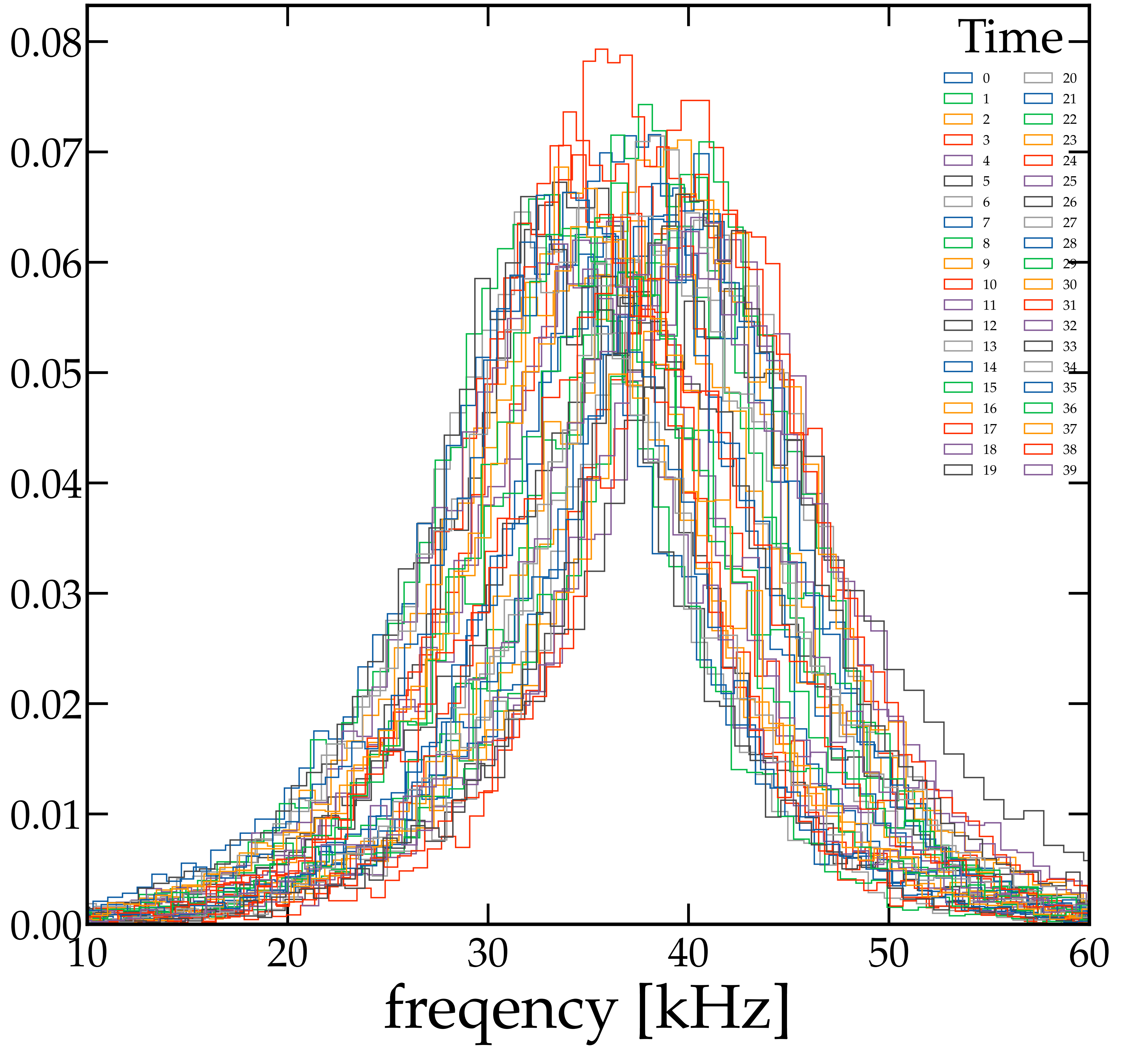
Show up often, and can end up very confusing due to cluttered visual information.



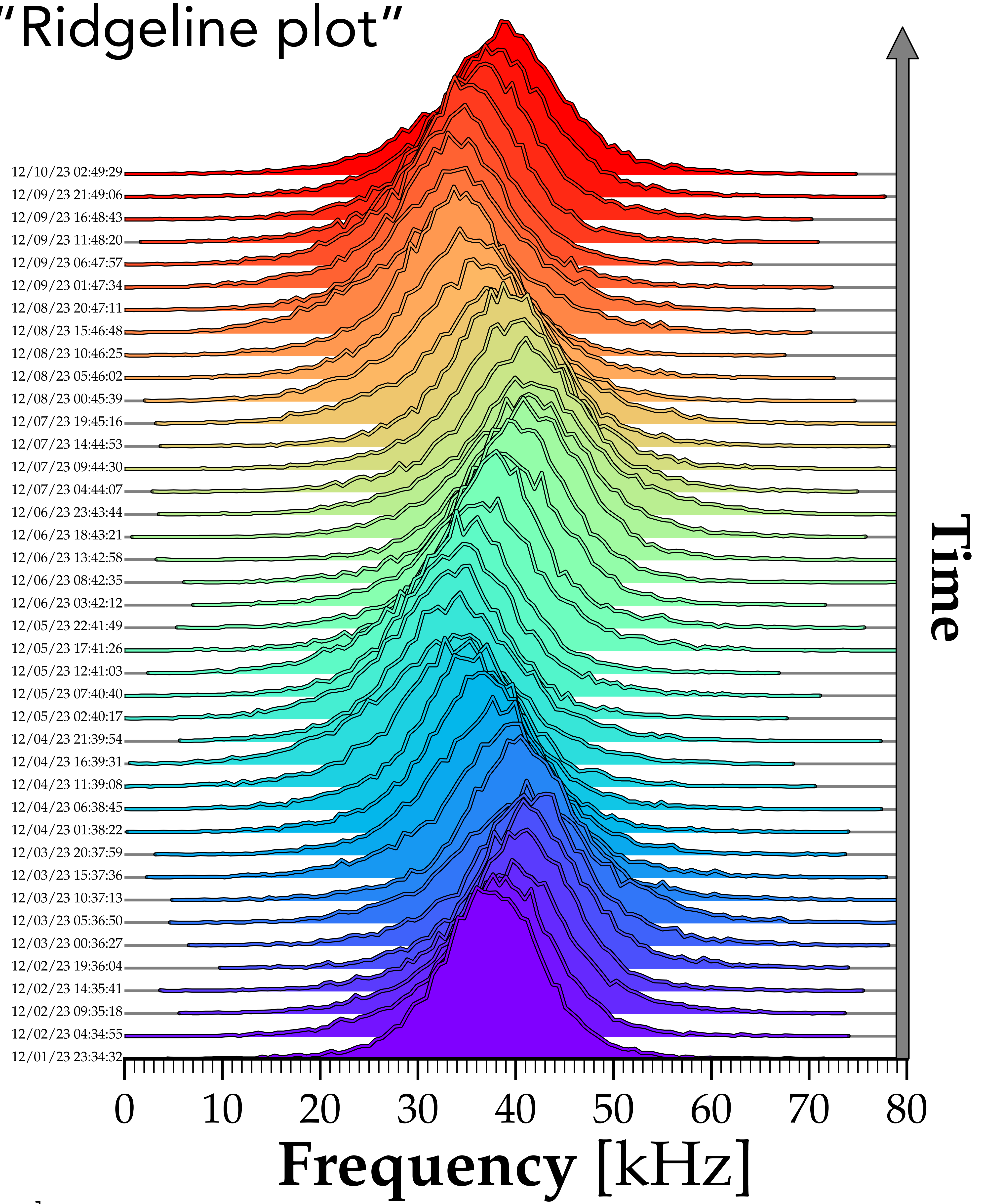
Histograms.ipynb

Histograms

Show up often, and can end up very confusing due to cluttered visual information.

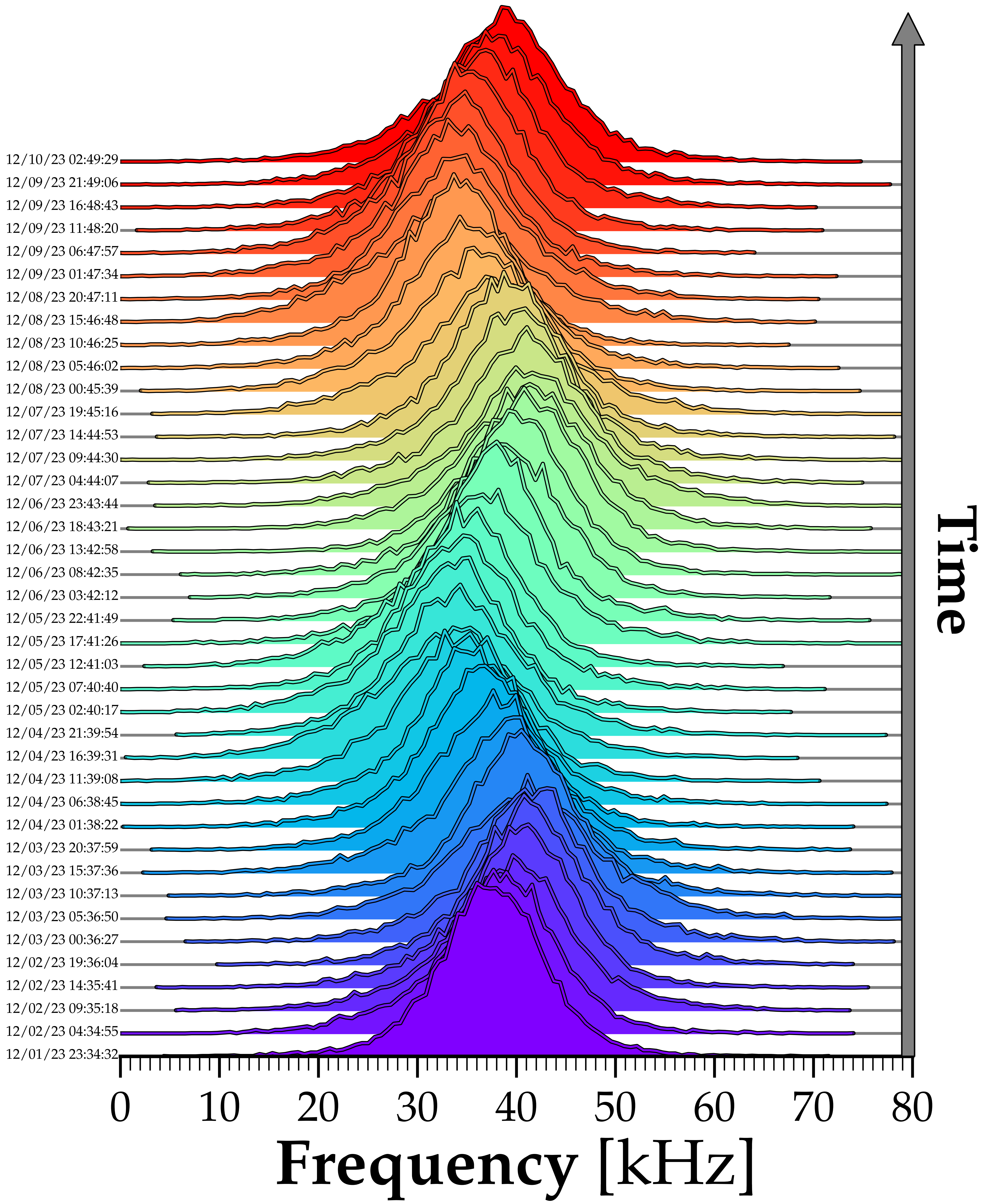


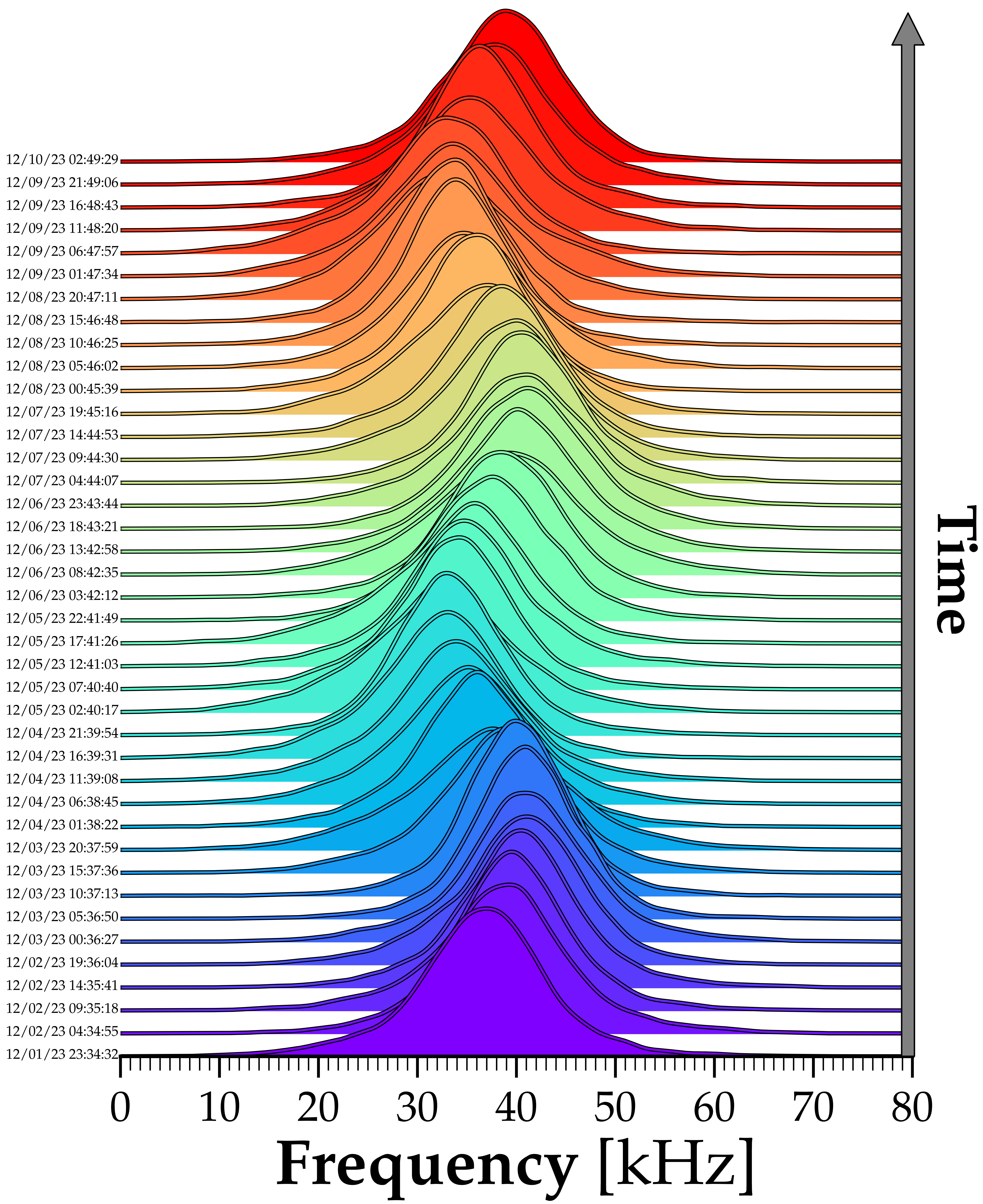
"Ridgeline plot"



Histograms.ipynb

You can improve clarity further by doing a kernel density estimation (kde) of the histogram

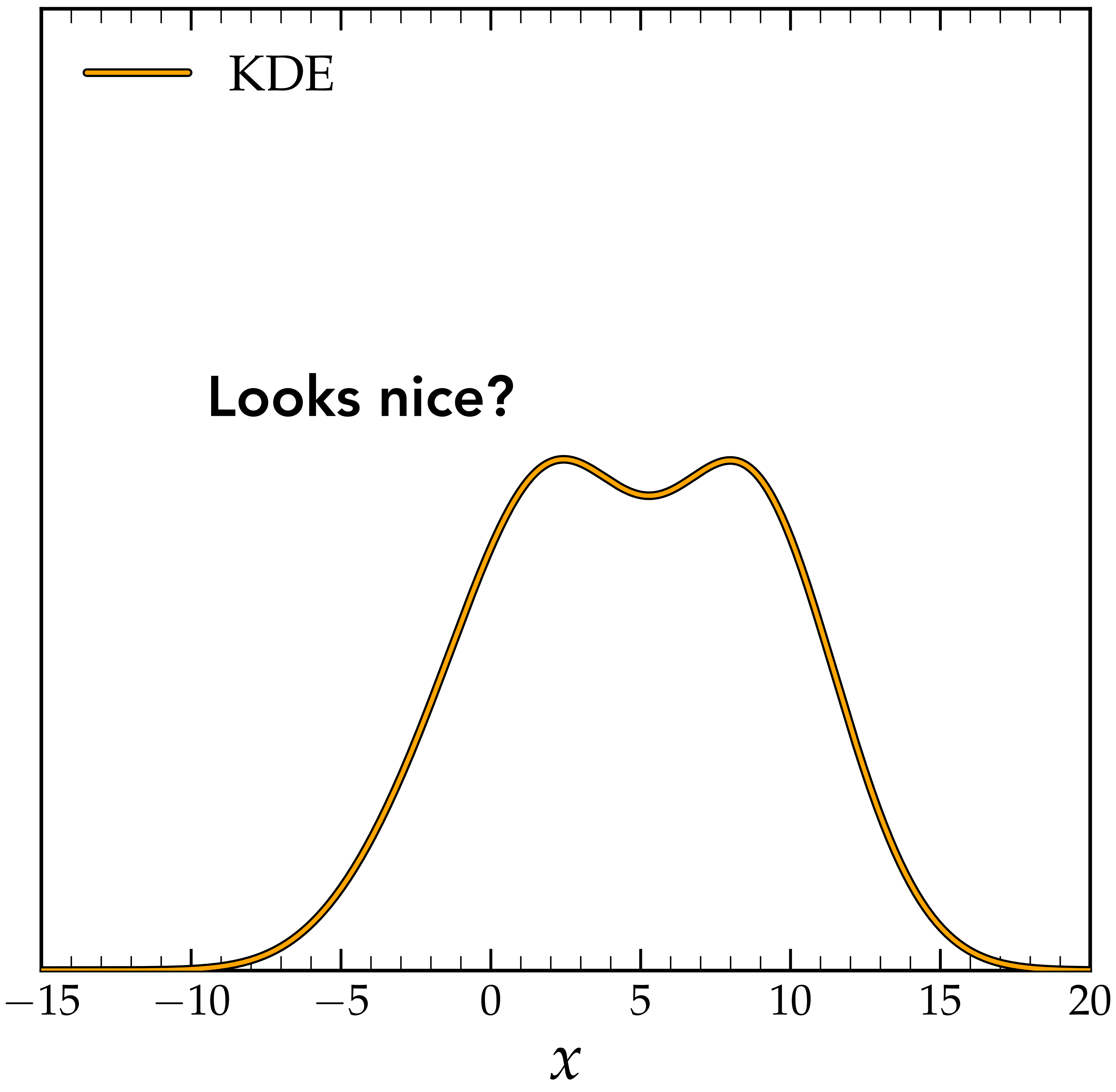




Histograms.ipynb

Lying by omission

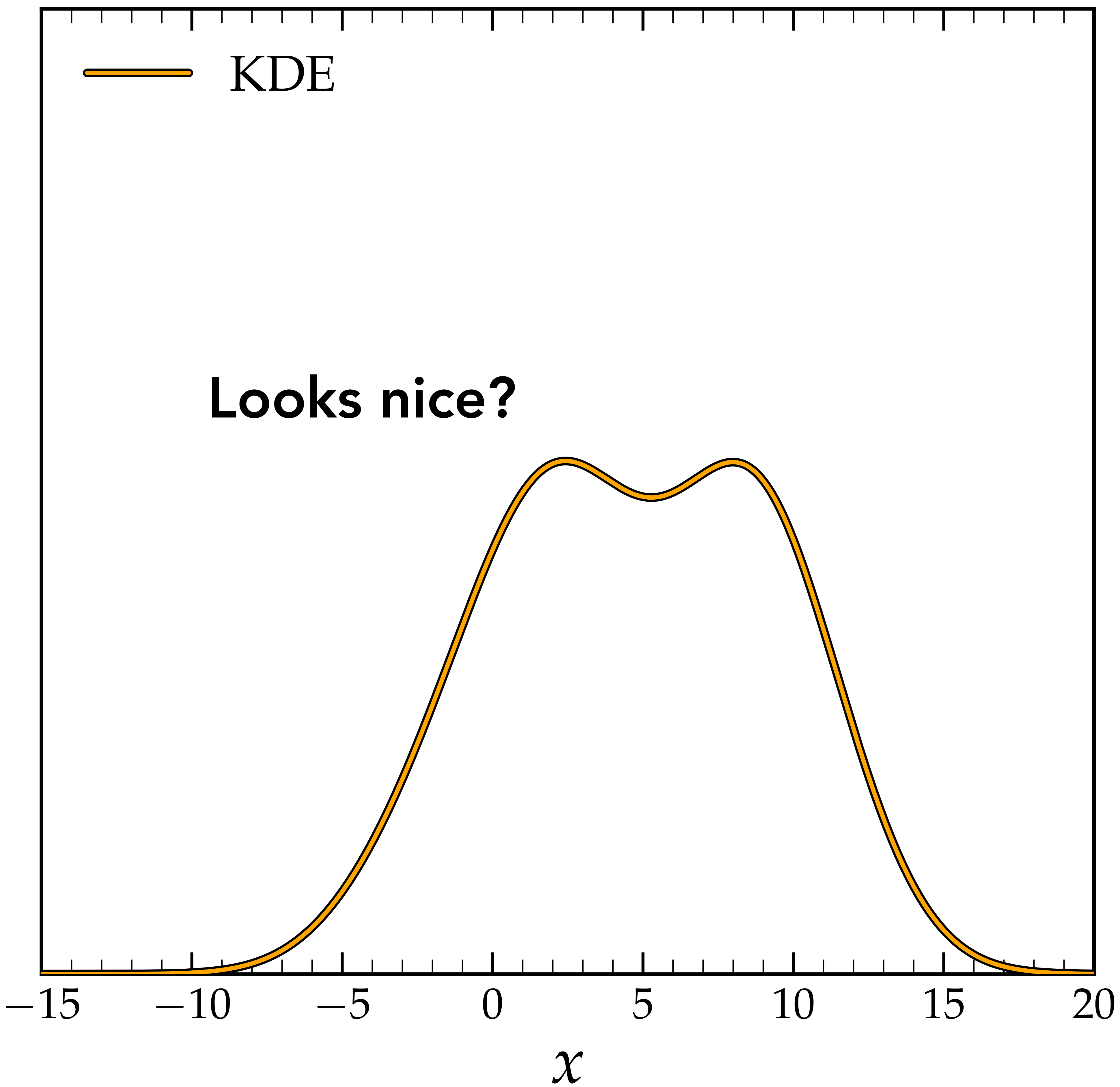
Tempting to use kde to smooth out messy or sparse data. But good practice is to always show the underlying data when doing smoothing like this.

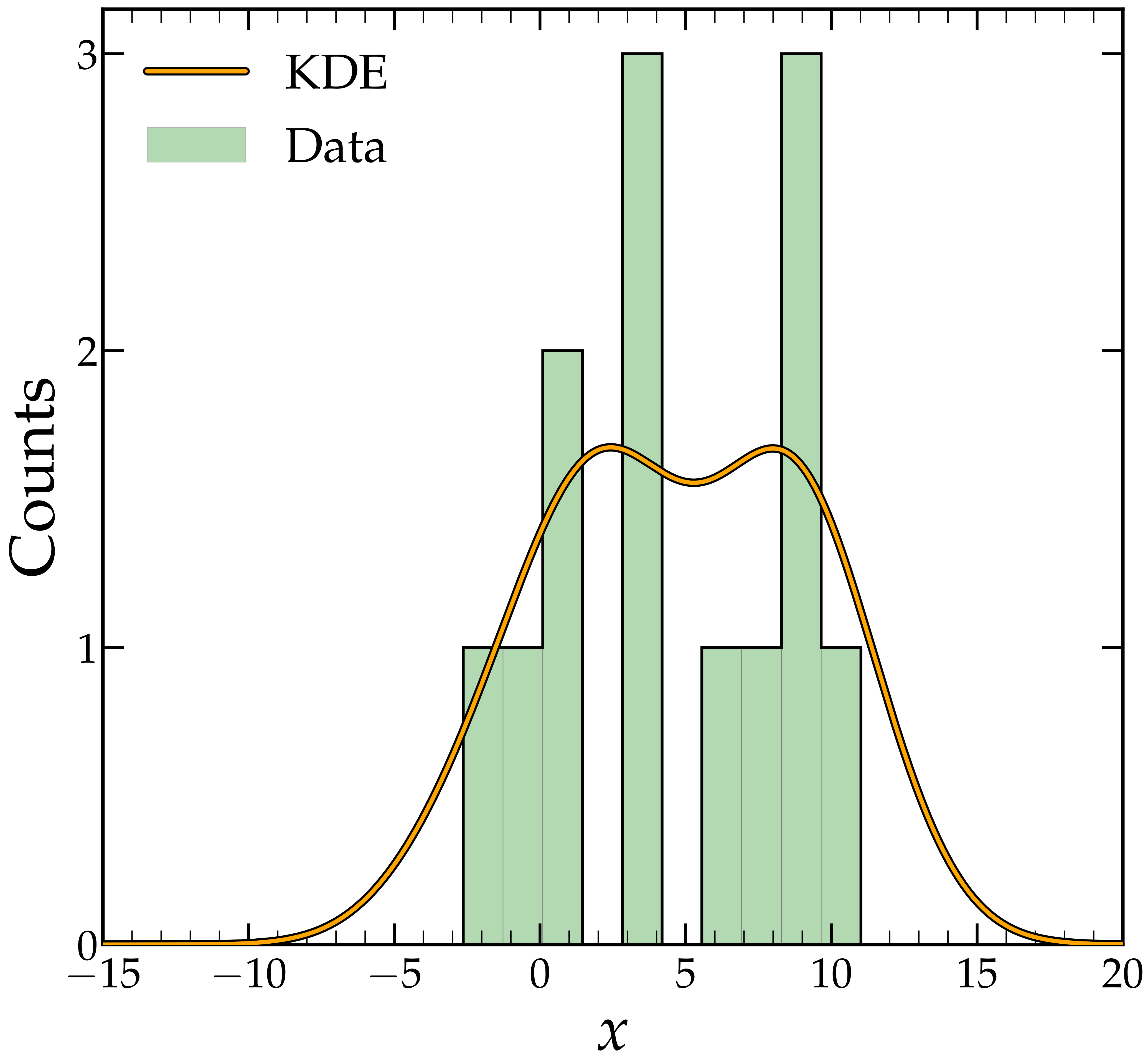


Lying by omission

Tempting to use kde to smooth out messy or sparse data. But good practice is to always show the underlying data when doing smoothing like this.

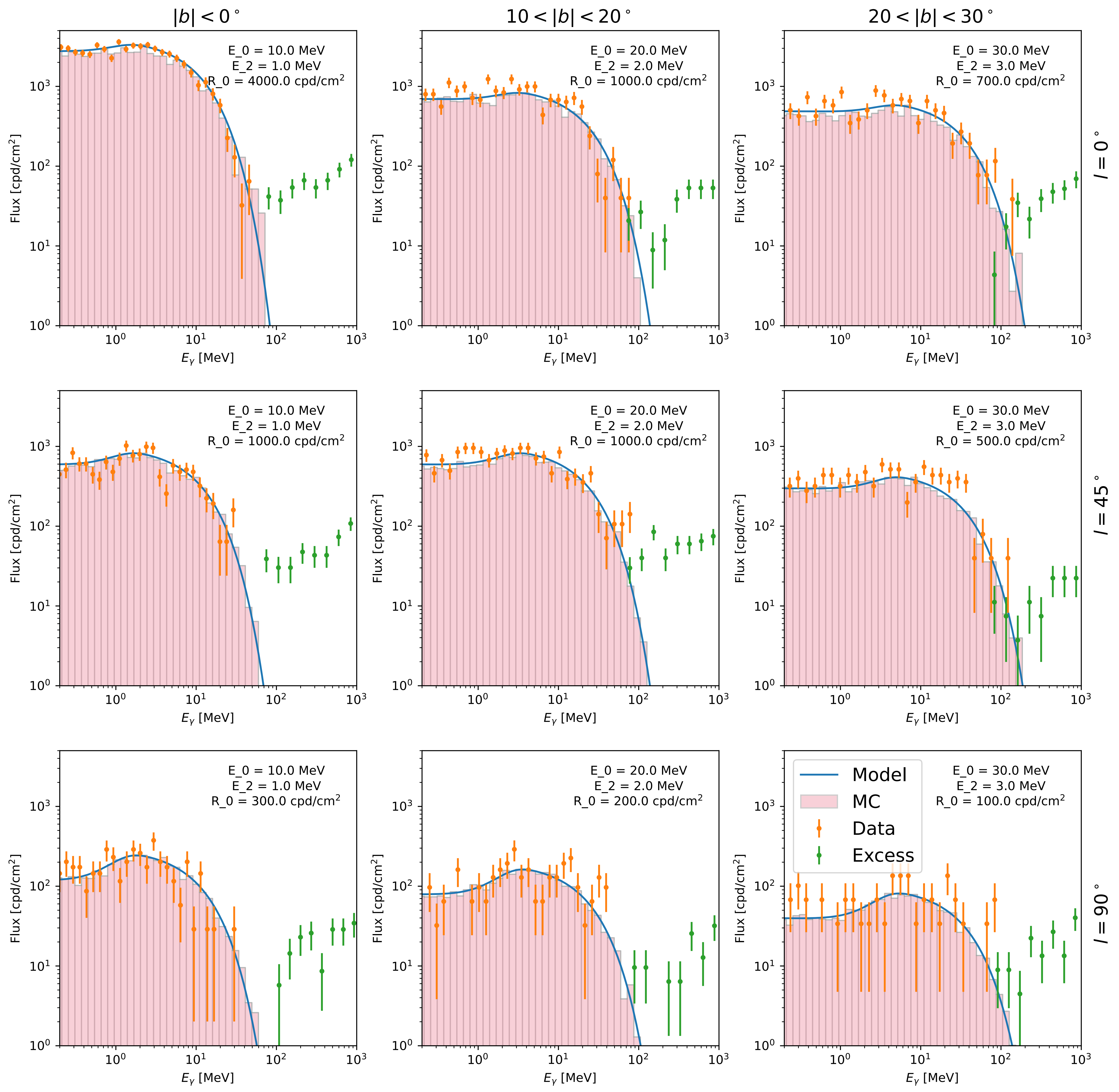
Histograms.ipynb





Complicated plots

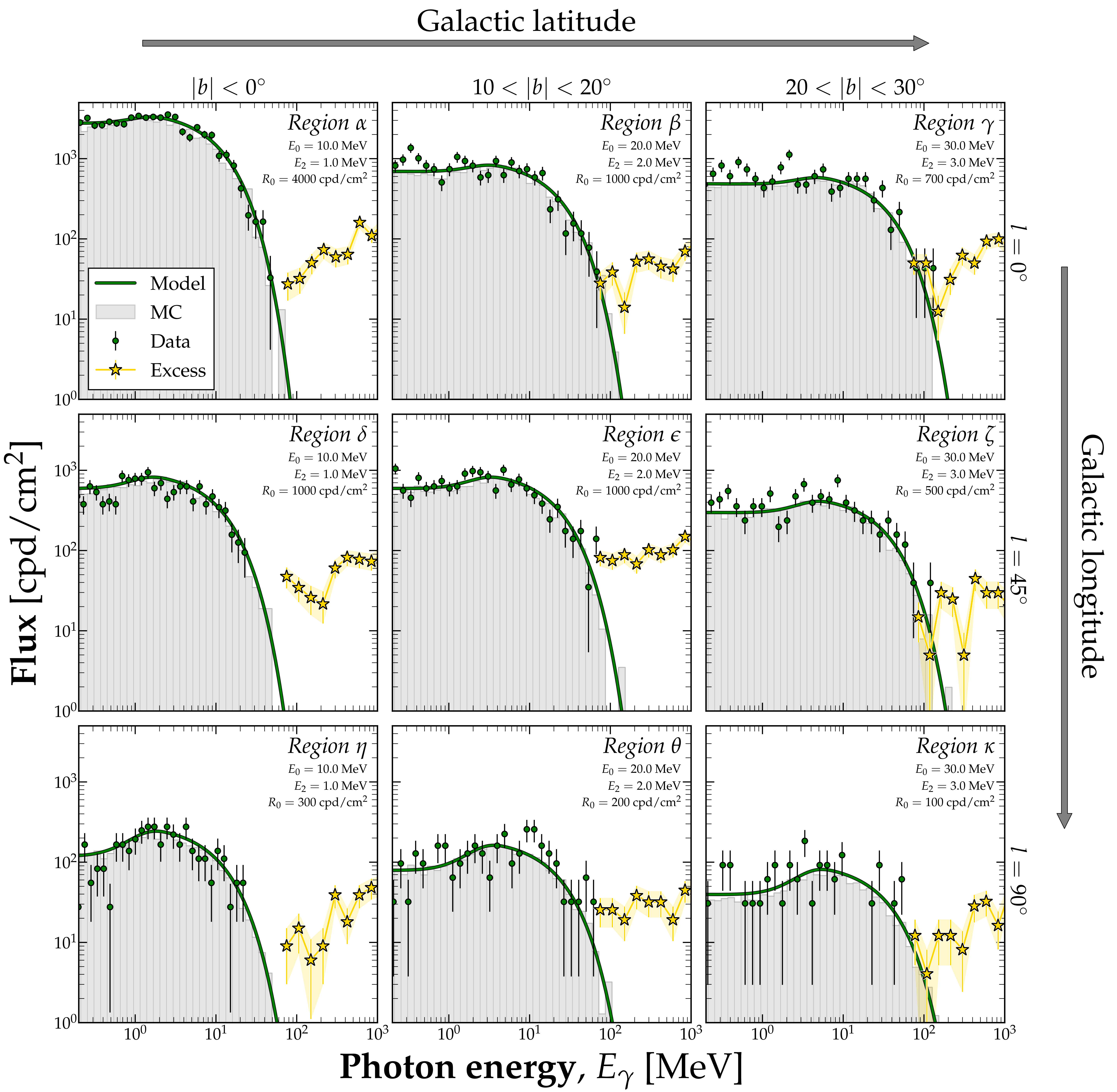
- Step 1: Avoid making complicated plots. Why does your plot need to be so complicated? Are you trying to impress someone? Because you won't - you will just confuse or annoy them. Simplify. Draw out the message you are trying to say and just show that.
- That said, sometimes plots in papers do get complicated. In those cases, you still want to maximise the time people spend thinking about your message rather than just figuring out your plot's internal logic.
- Use pre-attentive attributes to express multiple layers of information that the reader can appreciate in stages.
- Use labels liberally. Do not force the reader to play tennis between the figure and the caption



📓 ComplicatedPlots.ipynb

Complicated plots

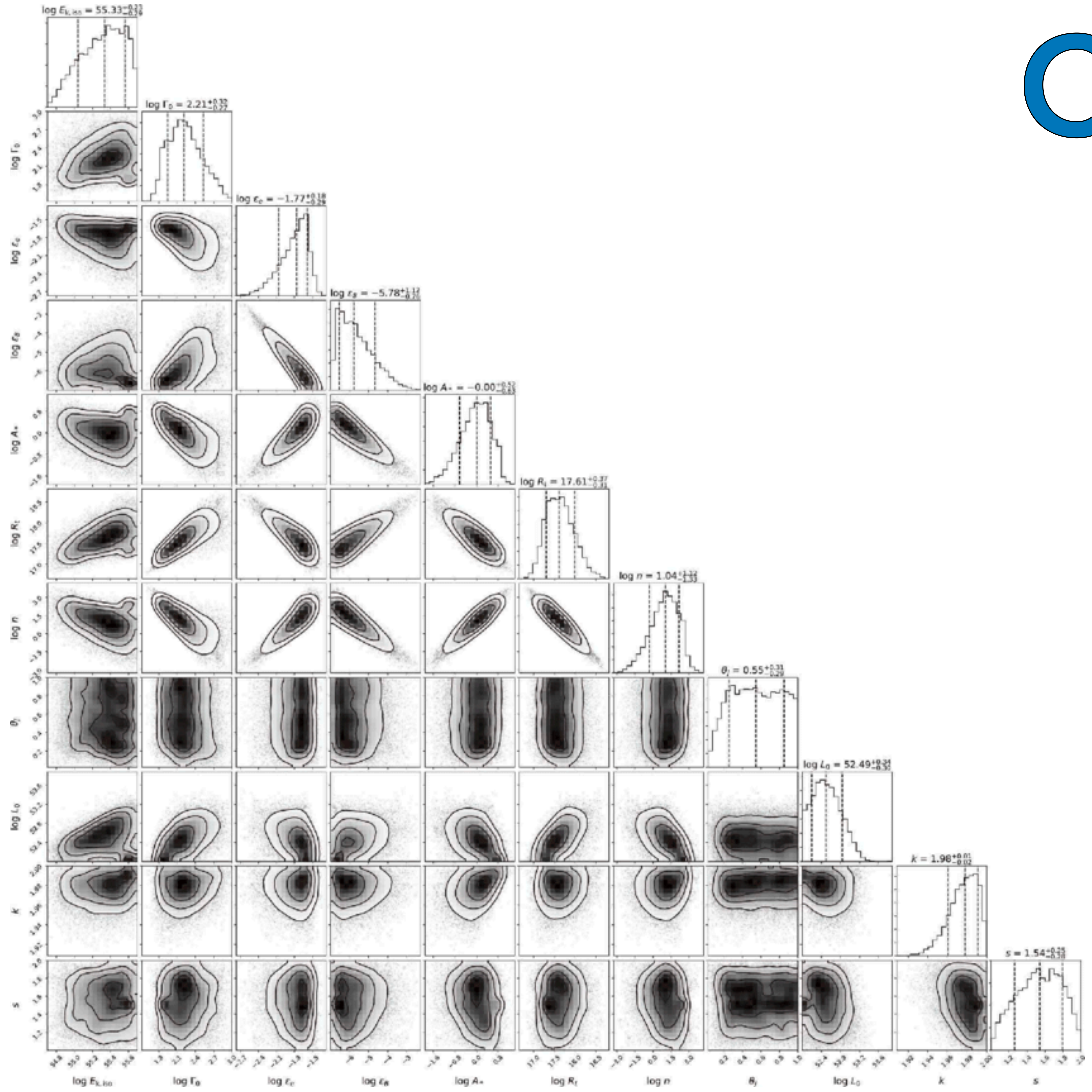
- Step 1: Avoid making complicated plots. Why does your plot need to be so complicated? Are you trying to impress someone? Because you won't - you will just confuse or annoy them. Simplify. Draw out the message you are trying to say and just show that.
- That said, sometimes plots in papers do get complicated. In those cases, you still want to maximise the time people spend thinking about your message rather than just figuring out your plot's internal logic.
- Use pre-attentive attributes to express multiple layers of information that the reader can appreciate in stages.
- Use labels liberally. Do not force the reader to play tennis between the figure and the caption



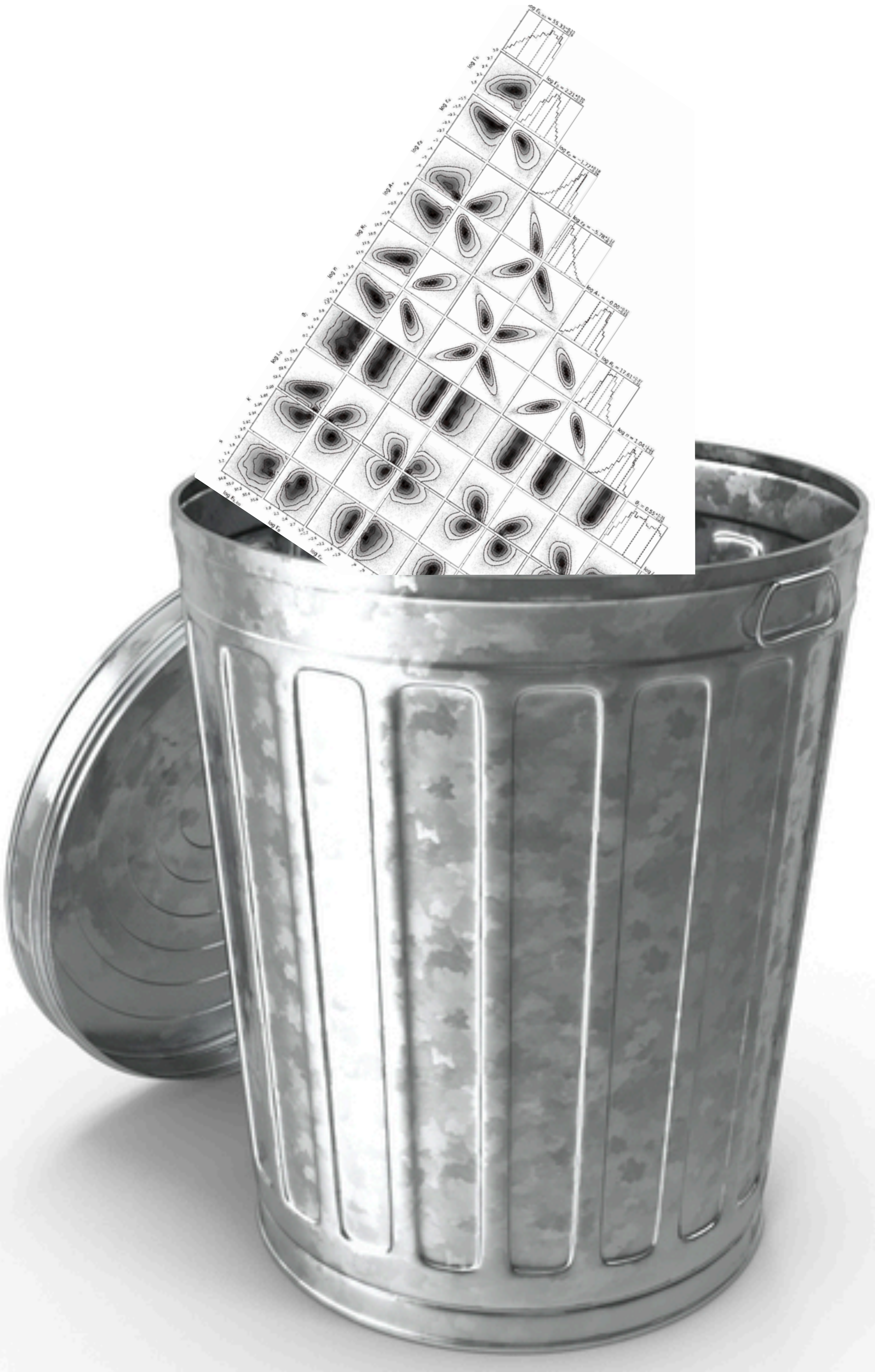
ComplicatedPlots.ipynb

Corner plots

…

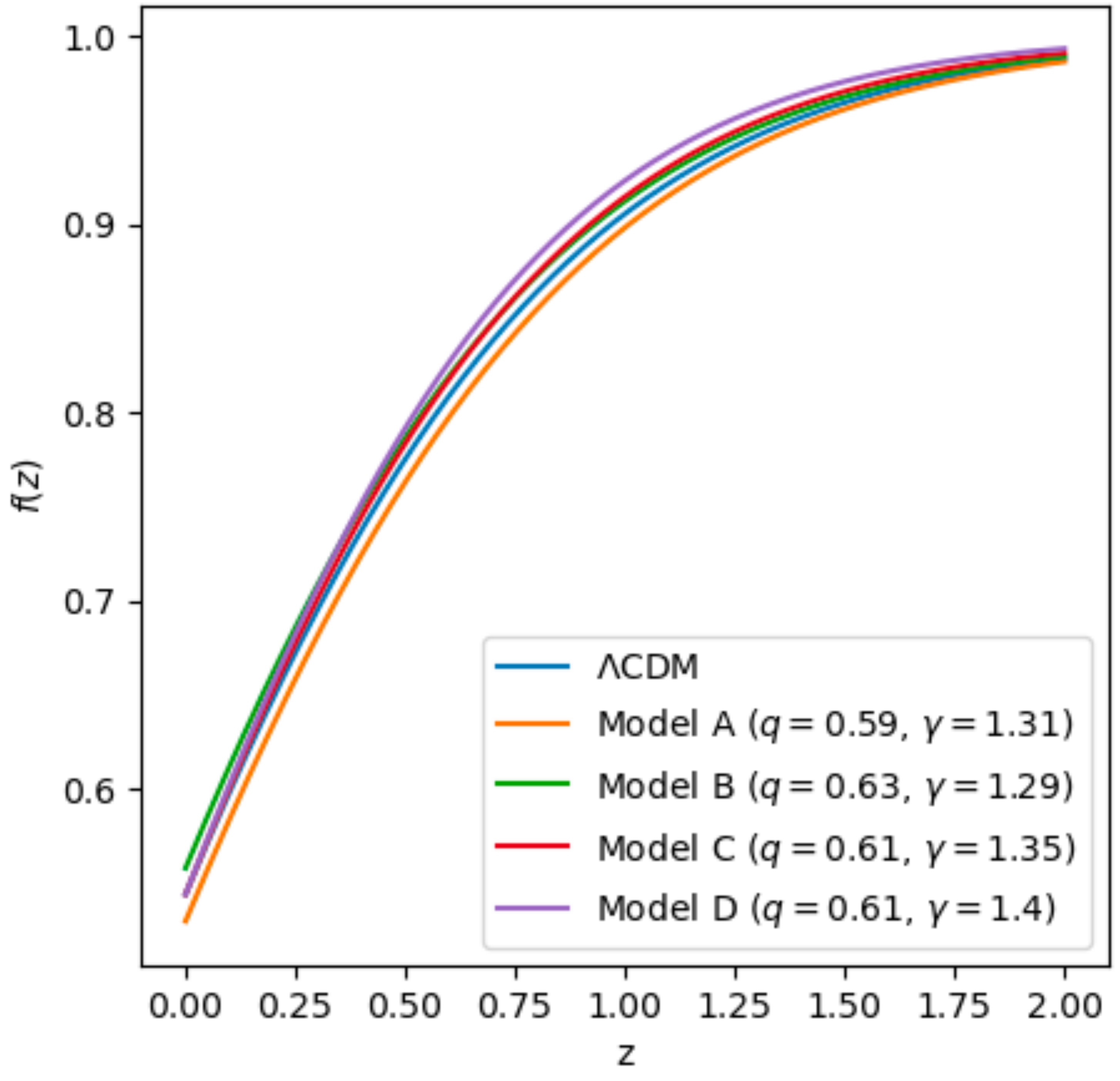


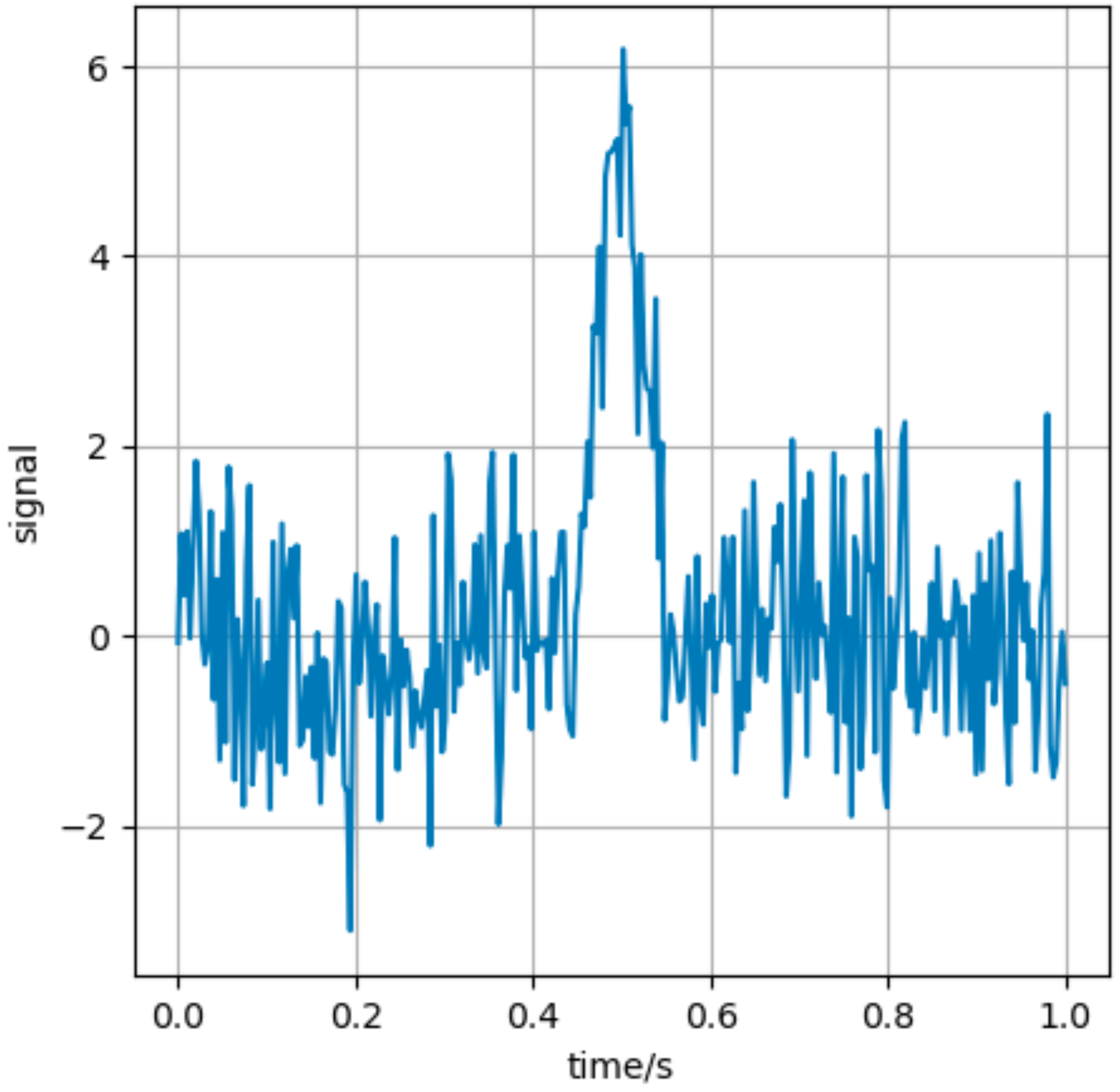
Corner plots

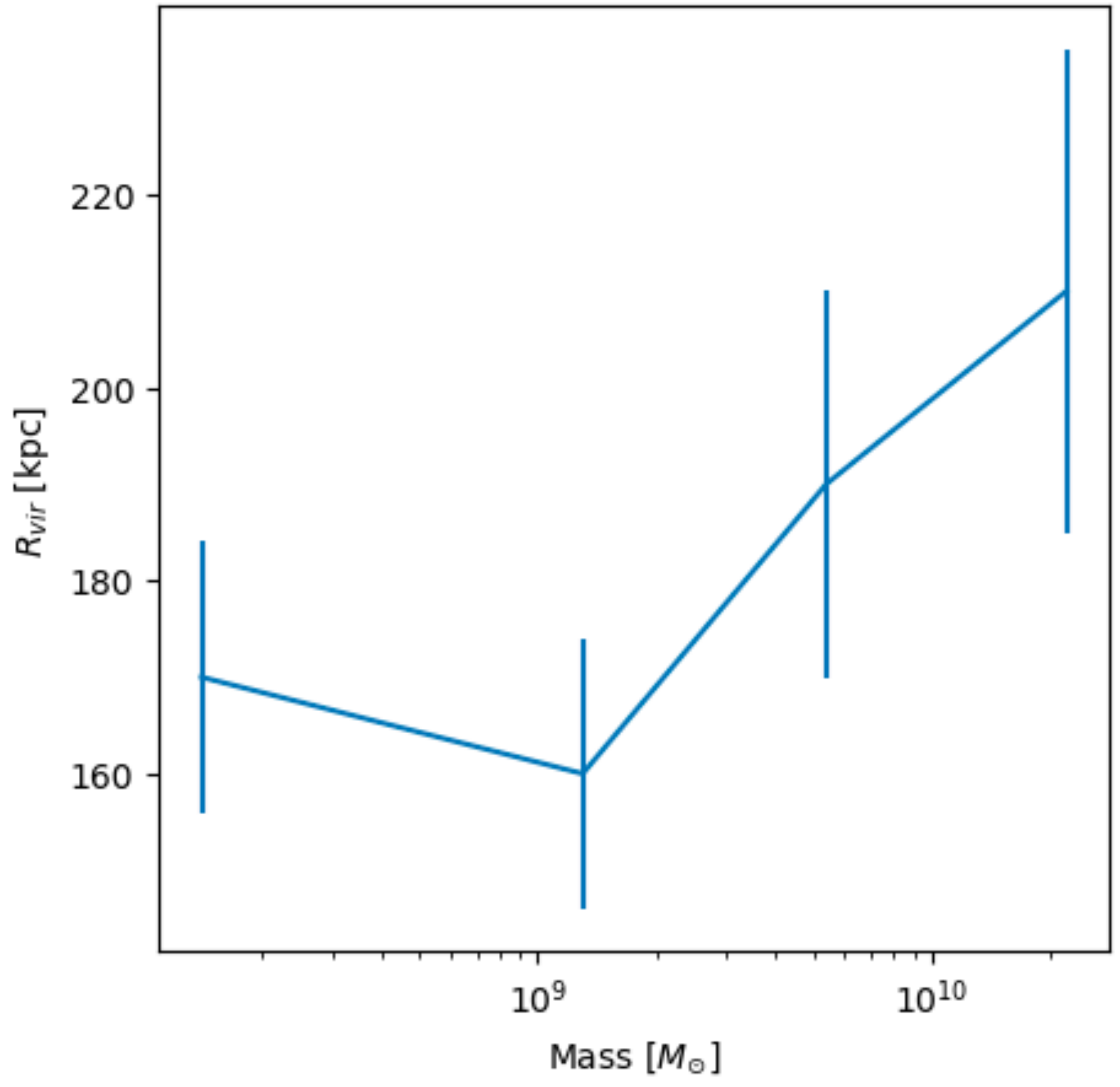


Sometimes your data is just ugly or uninteresting

It's hard to get excited by an ugly plot, even if the point it is making is exciting.

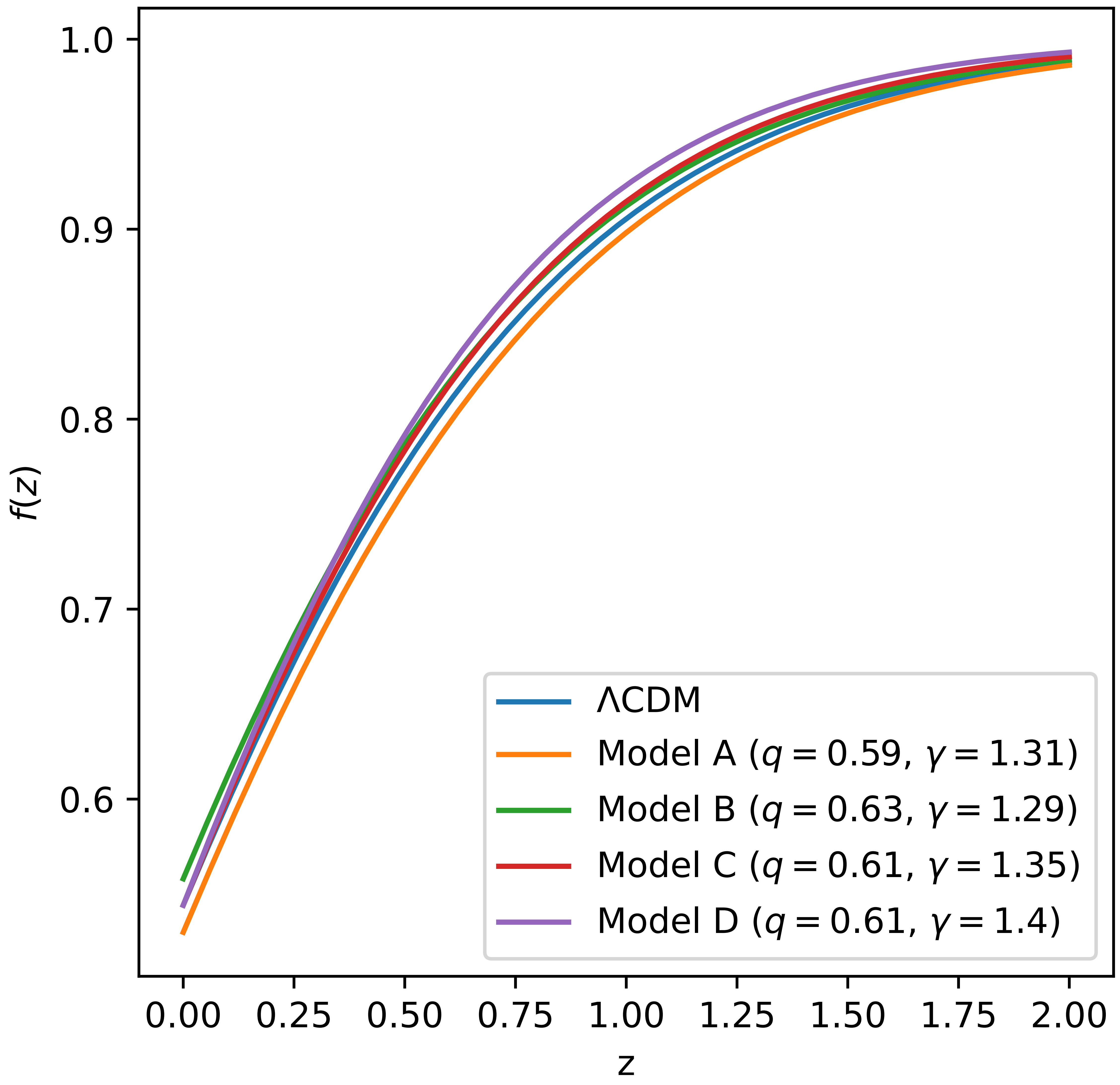






Remedy these cases by making the plot as pretty as possible and using the fact that the data is uninteresting to make the message even clearer. Some of the best plots are actually the simplest

Example 1

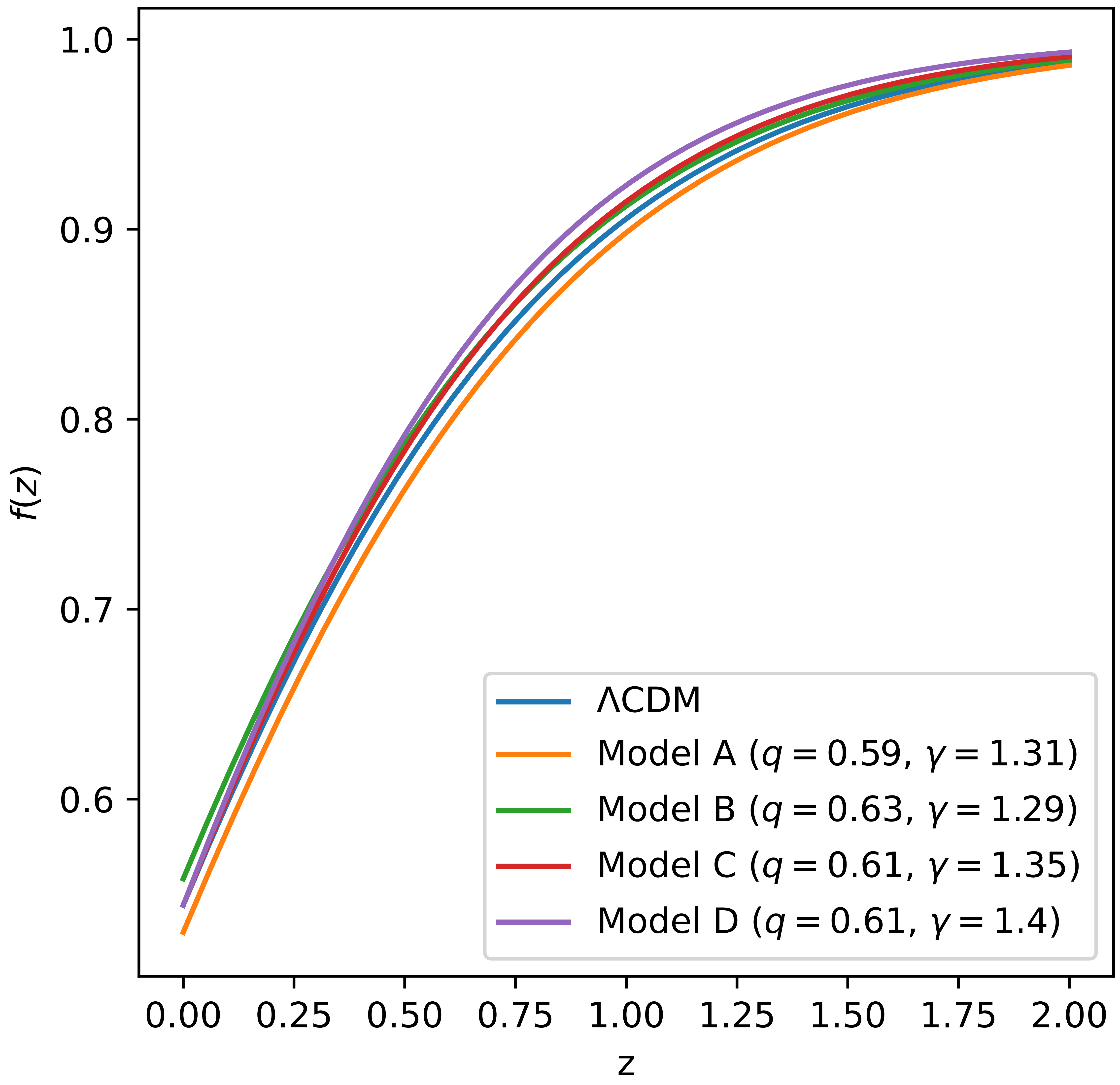


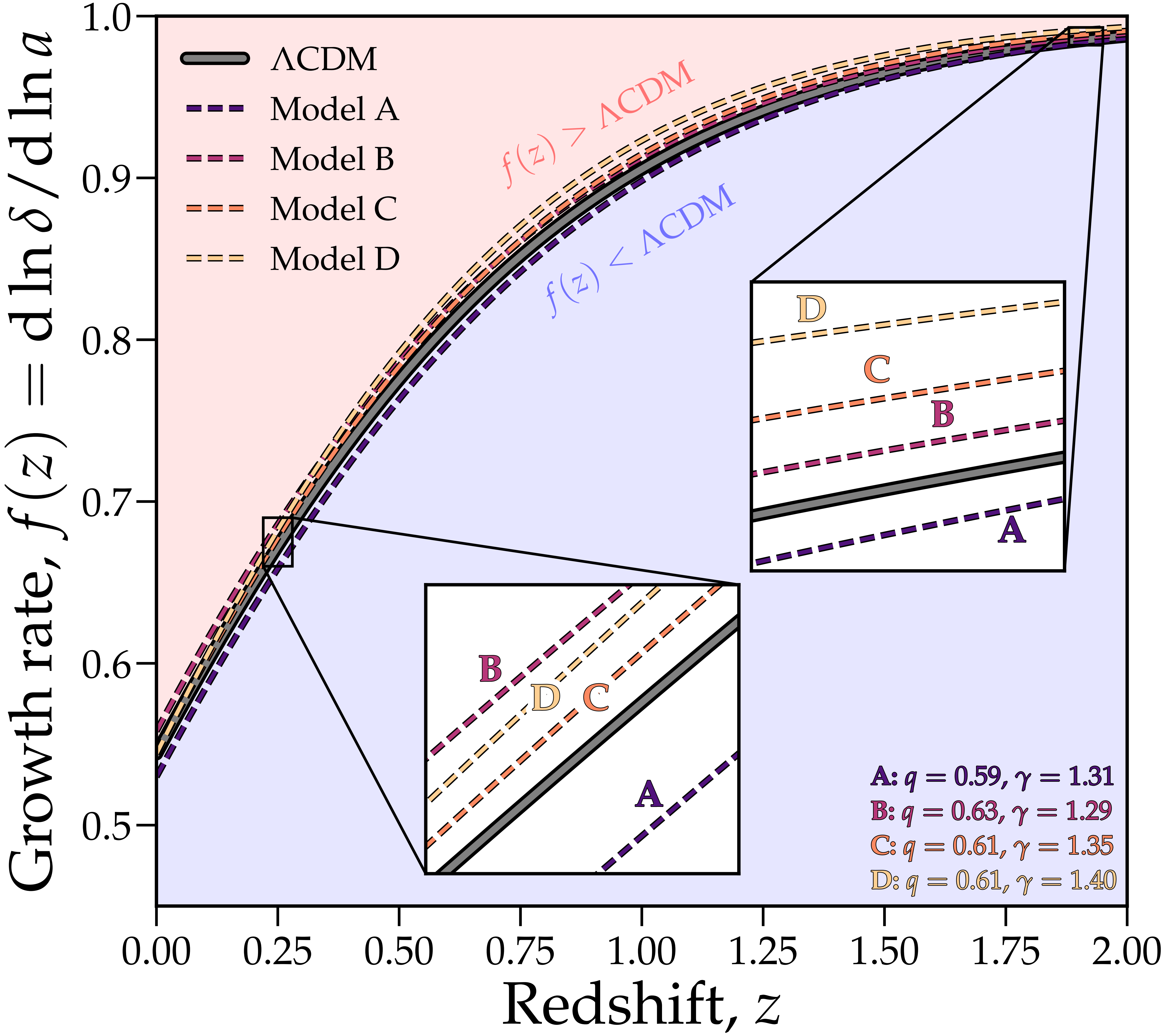
1.0
0.9
0.8
0.7
0.6
f(z)
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
z
ΛCDM
Model A ($q = 0.59$, $\gamma = 1.31$)
Model B ($q = 0.63$, $\gamma = 1.29$)
Model C ($q = 0.61$, $\gamma = 1.35$)
Model D ($q = 0.61$, $\gamma = 1.4$)

Example 1

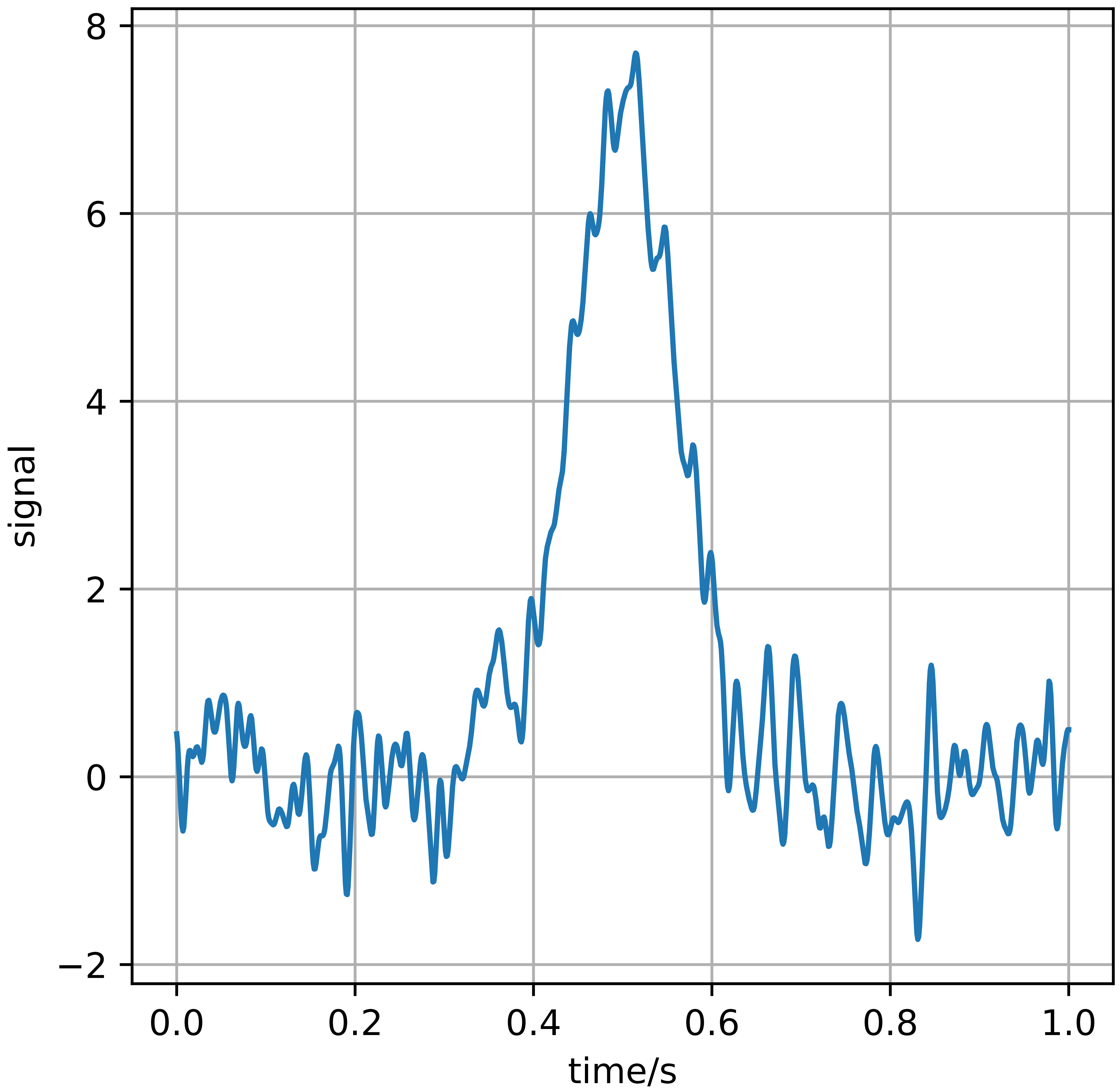
Tip: Adding inset axes to plots in matplotlib is much easier than it seems
See: https://matplotlib.org/stable/gallery/axes_grid1/inset_locator_demo.html

BoringPlots.ipynb

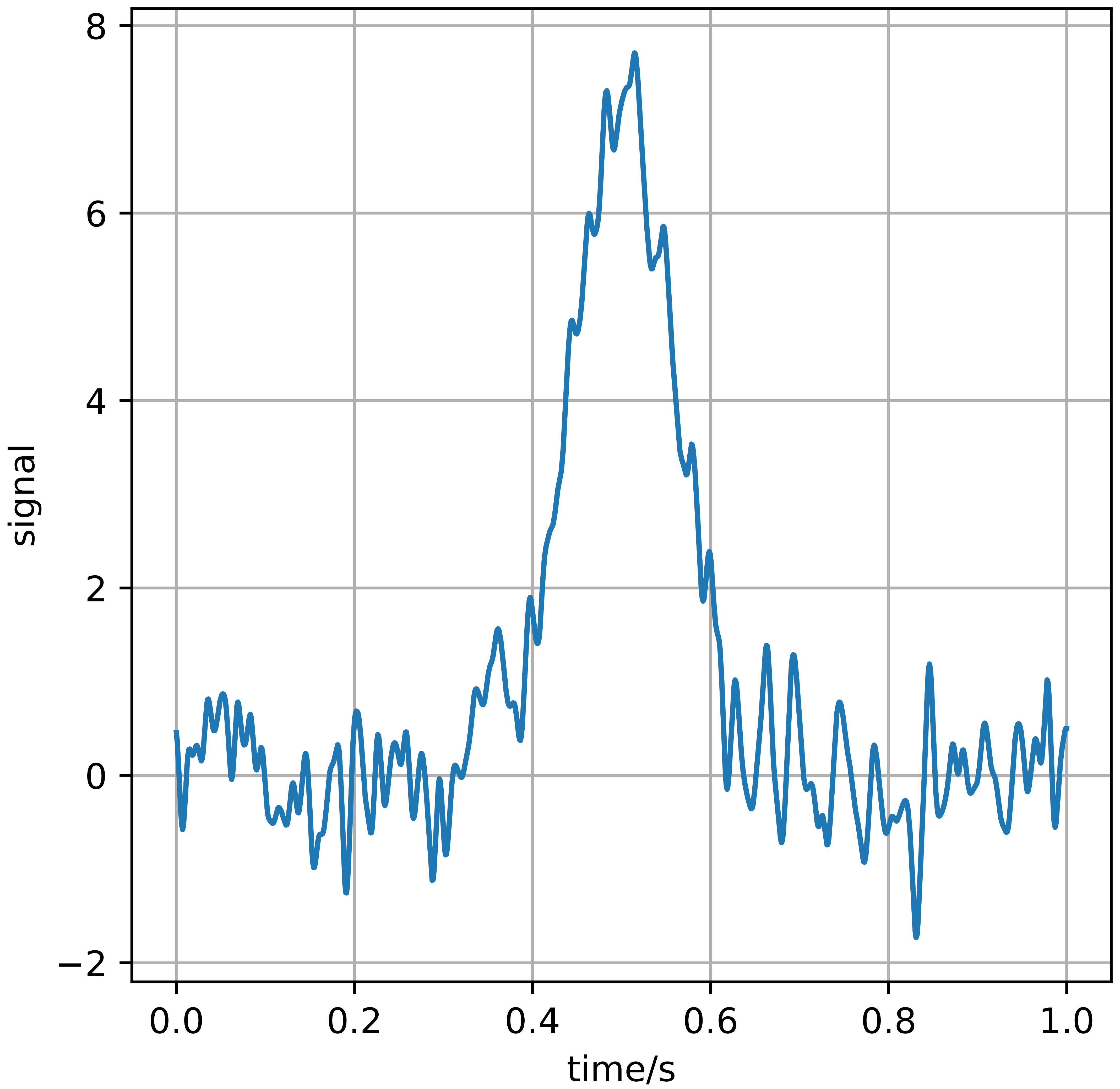




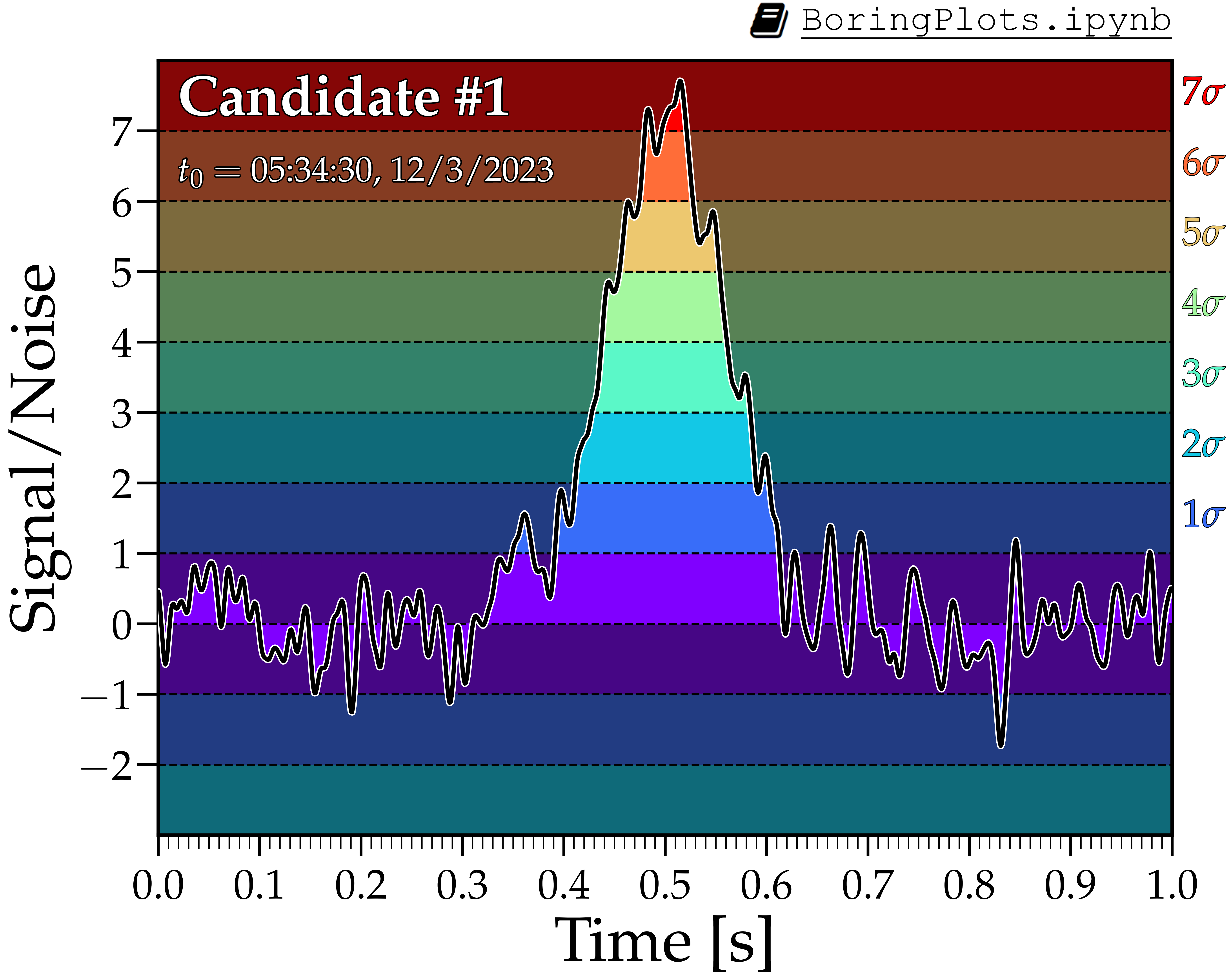
Example 2



Example 2

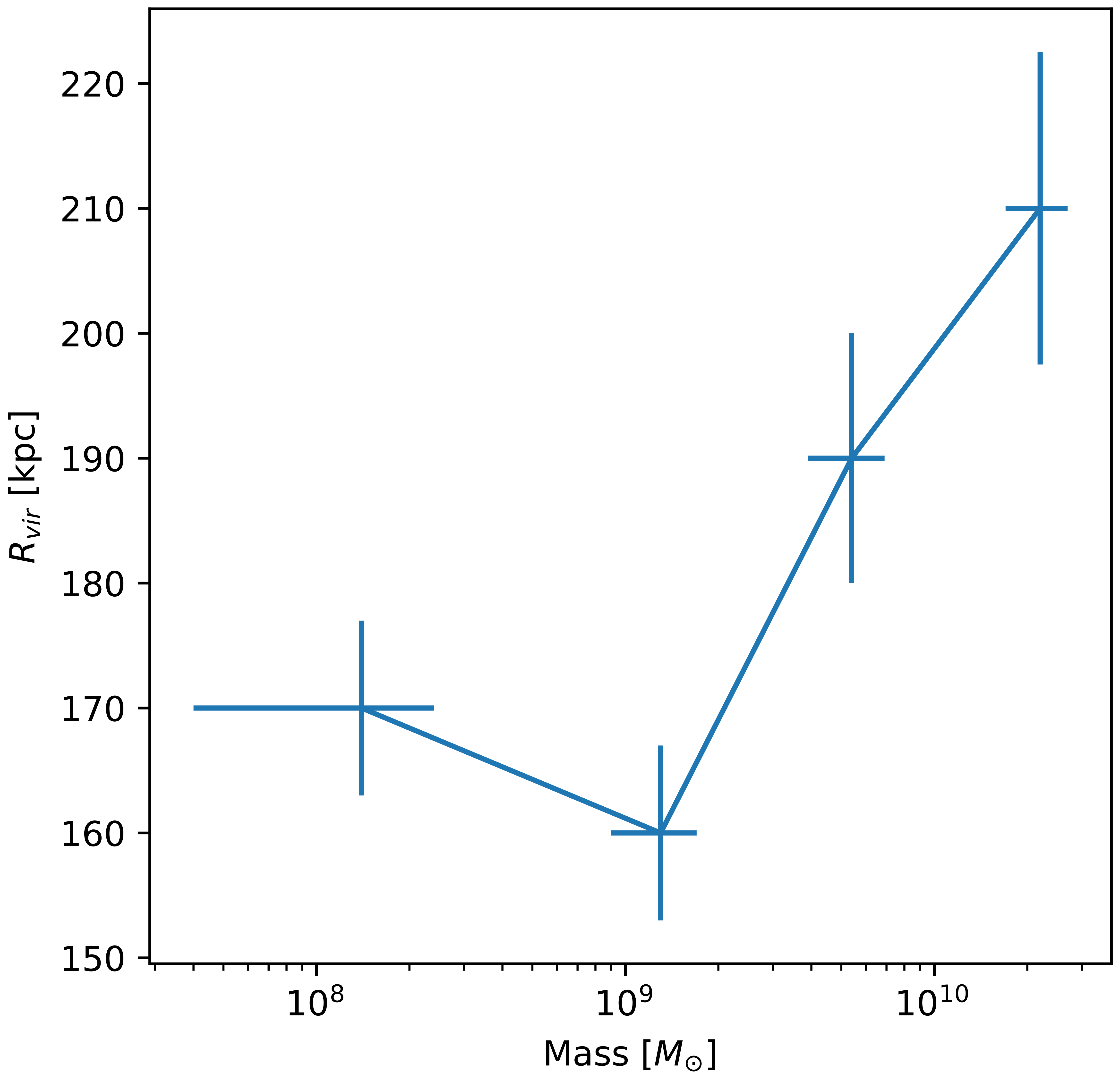


signal
time/s



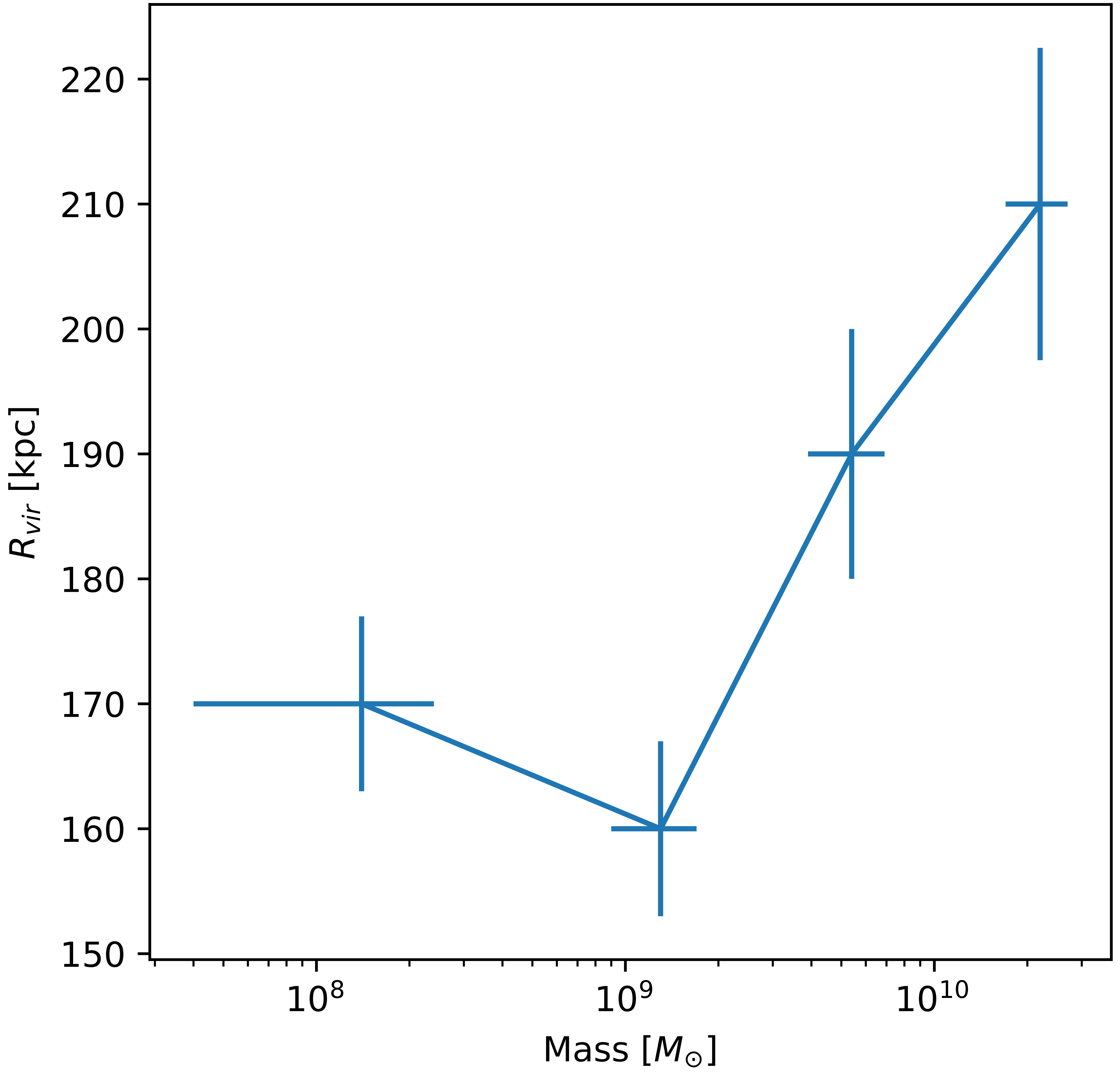
BoringPlots.ipynb
Candidate #1
t_0 = 05:34:30, 12/3/2023
Signal/Noise
Time [s]
7σ
6σ
5σ
4σ
3σ
2σ
1σ

Example 3

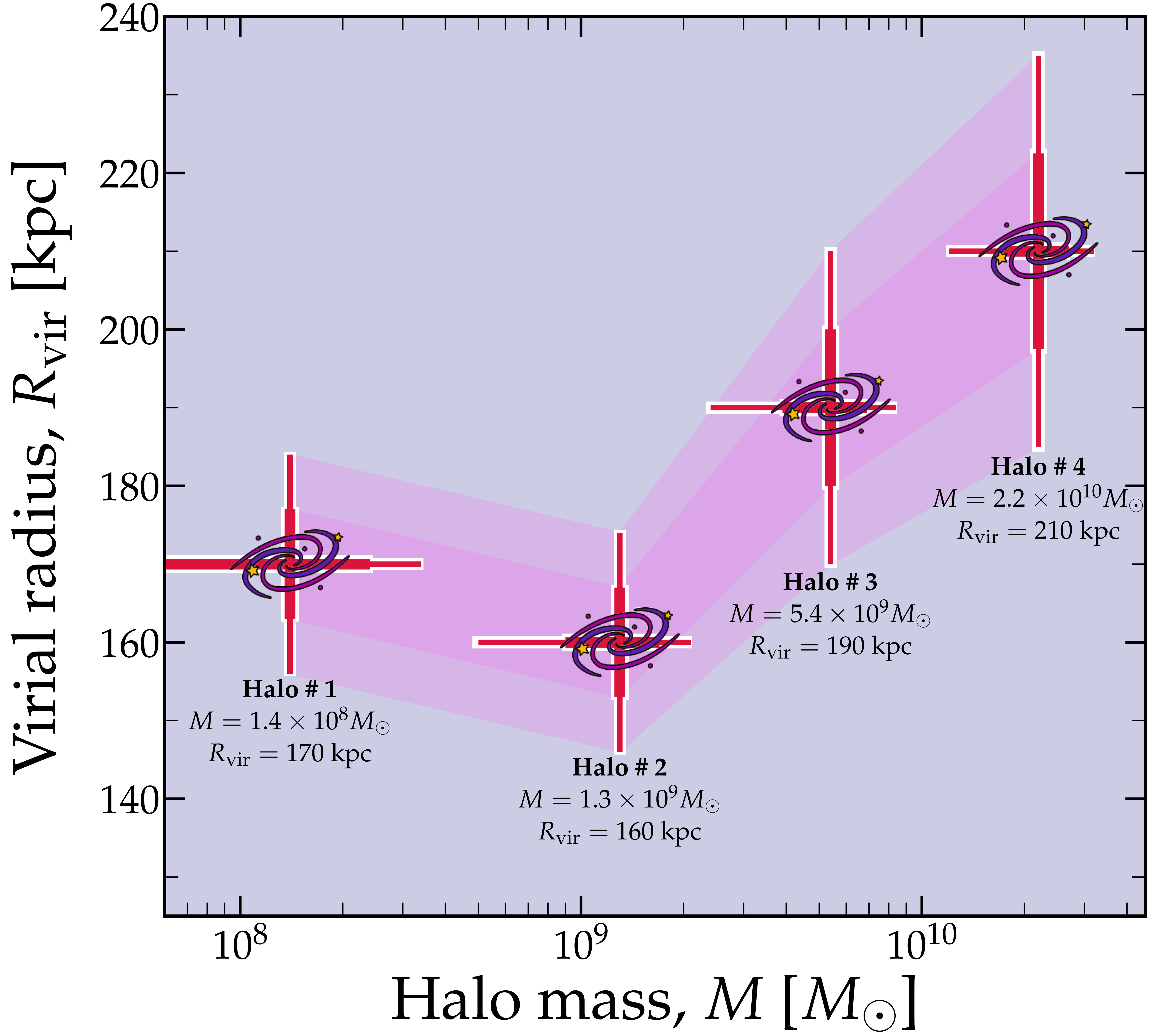


220
210
200
190
180
170
160
150
R_{vir} [kpc]
10^8
10^9
10^{10}
Mass [$M_\odot$]

Example 3



BoringPlots.ipynb



Practical/stylistic tips

- Defaults
- Style sheets
- Colours
- Fonts

Disclaimers

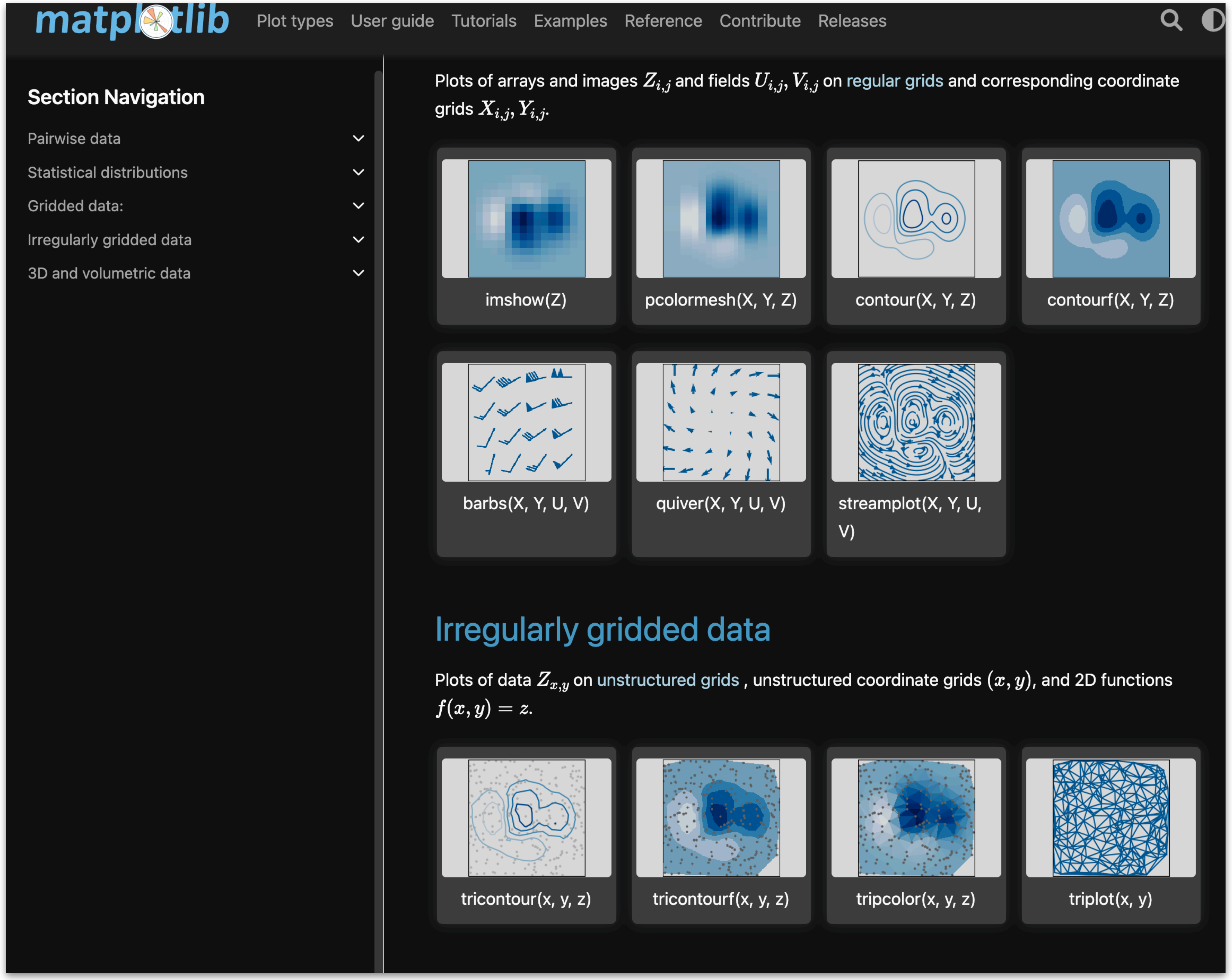
- I will use the example of python/matplotlib for concreteness, though basically all of this advice can be translated to other software like R, Mathematica, MATLAB, ROOT, etc.

- For the sake of your own sanity, I recommend using notebooks (e.g. jupyter) to make plots, at least when you are still in the creation phase

  (NB: I do not condone the use notebooks for all your work)

General tips for matplotlib

- Matplotlib is kind of annoying. This isn't a tip, but if you're struggling don't worry, it's probably not your fault.
- Documentation is quite arcane, but they have tons of examples. You'll get much better mileage by extending pre-existing code that approximates what you want.
- Google images is a good way to find those examples if you don't know exactly the right words to describe what you want.
- ChatGPT is pretty good for the initial setup of plots and to get over any syntax barriers



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

Matplotlib style sheets

- Style sheets (`.mplstyle` files) are a way to change any of the default settings for making plots and have them hold over to all subsequent plots.

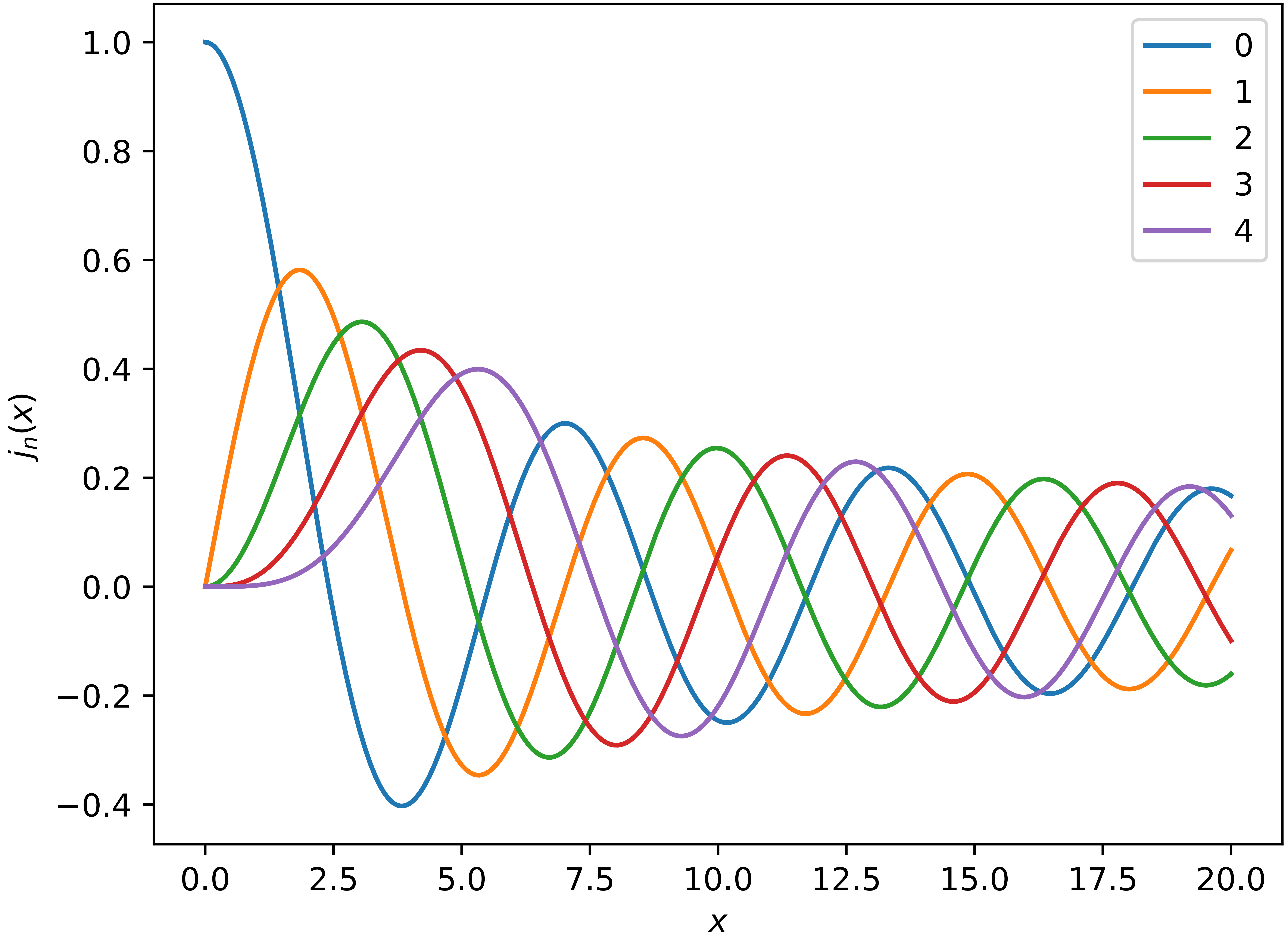
- See here for more details on how to construct them. Just about every default setting can be changed, including tick, axis and figure properties, default line colours, fonts etc.

```
# Set default figure size
figure.figsize : 13, 12

# Set x axis
xtick.major.size : 15
xtick.major.width : 2
xtick.minor.size : 10
xtick.minor.width : 1
xtick.direction : in
xtick.top : True

# Set y axis
ytick.major.size : 15
```

Matplotlib default style

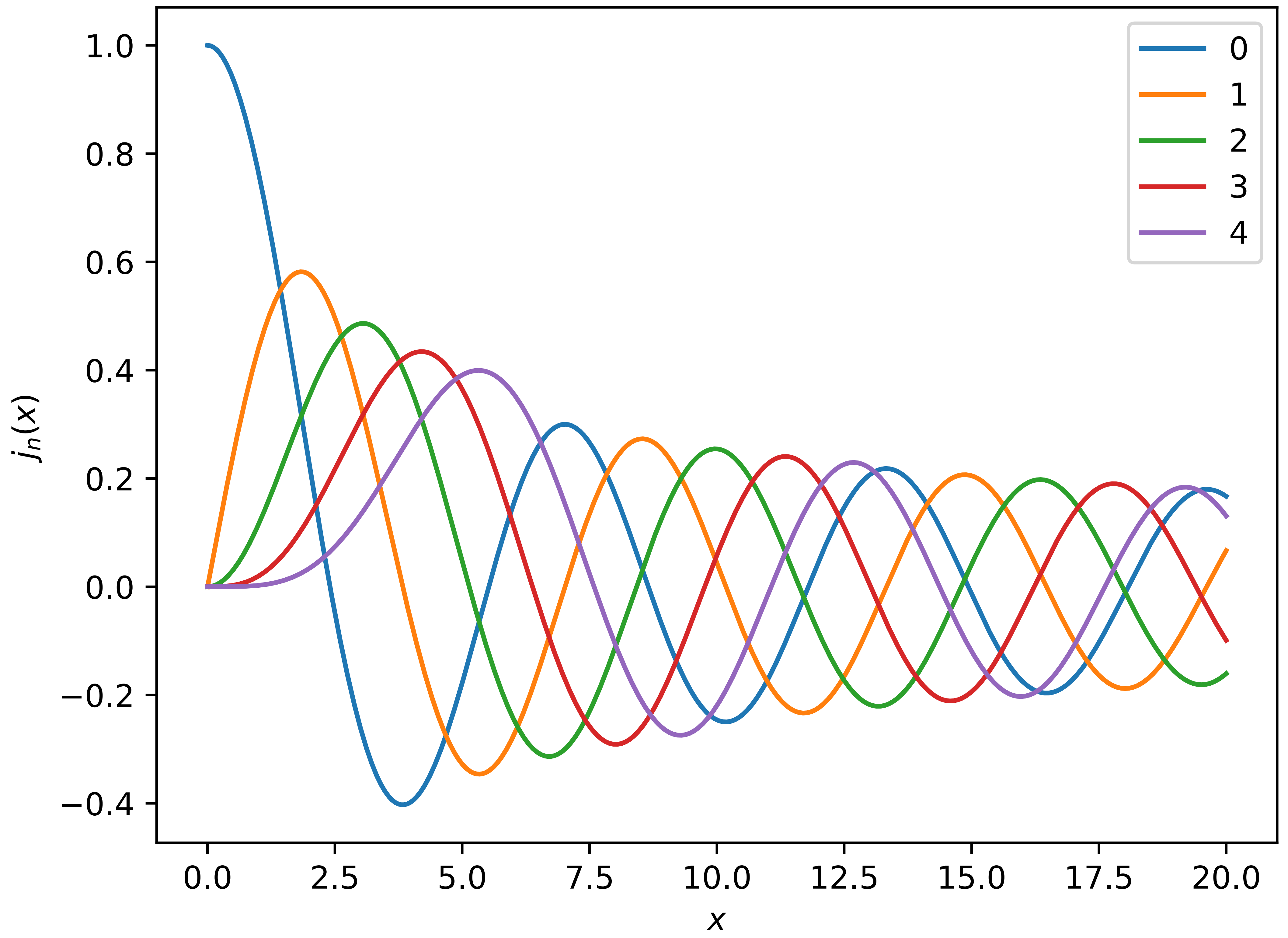


There is nothing really *wrong* with this plot, but there is also nothing special about it either. Its greatest crime is just that it is relying on the default Matplotlib settings, which is immediately obvious to anyone who uses python

Example of a style sheet

Using my style sheet ('sty.mplstyle' provided in the Confluence materials), you can improve the look of your plot with a single line of code.

Default

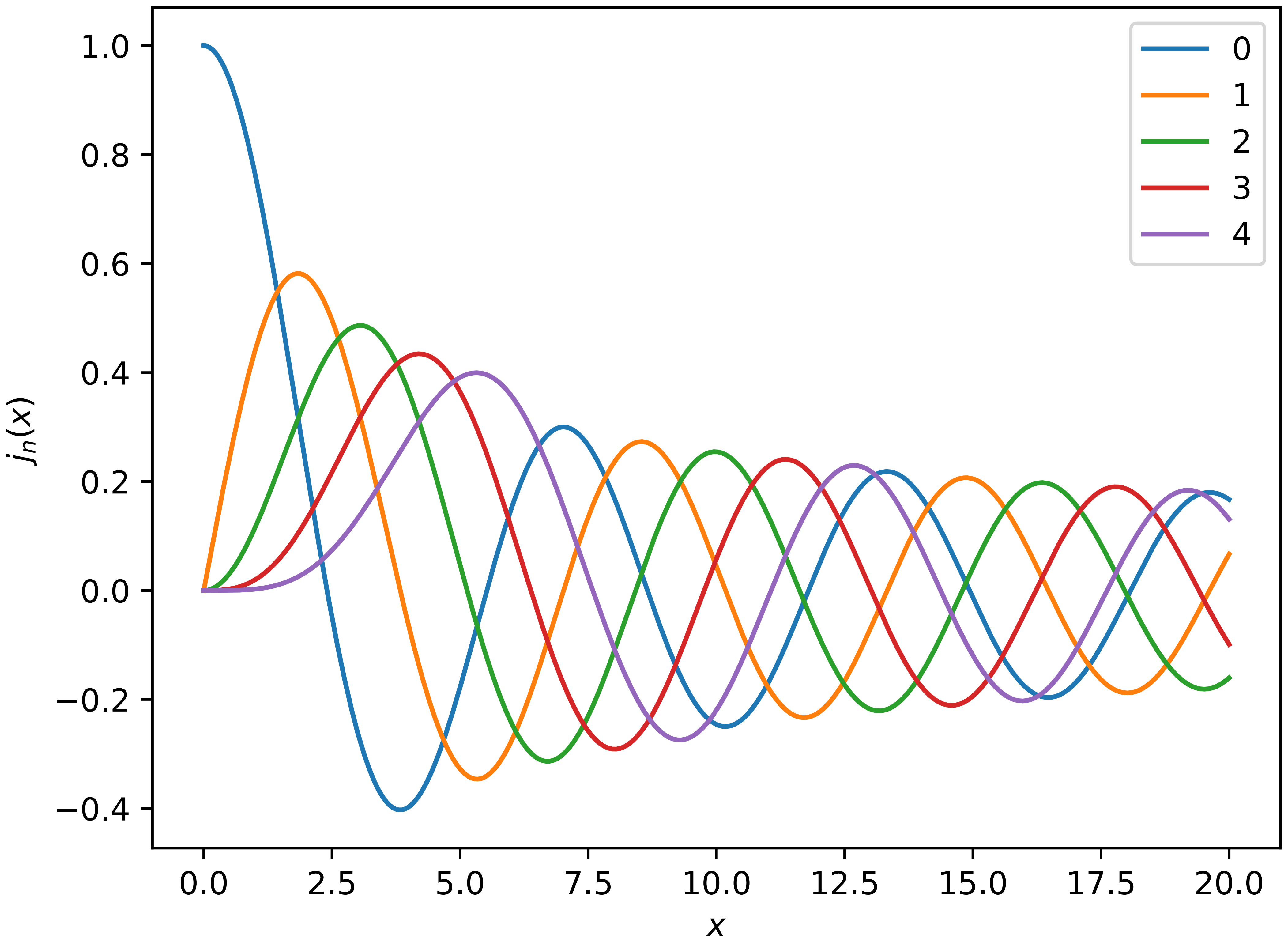


`Defaults.ipynb`

Example of a style sheet

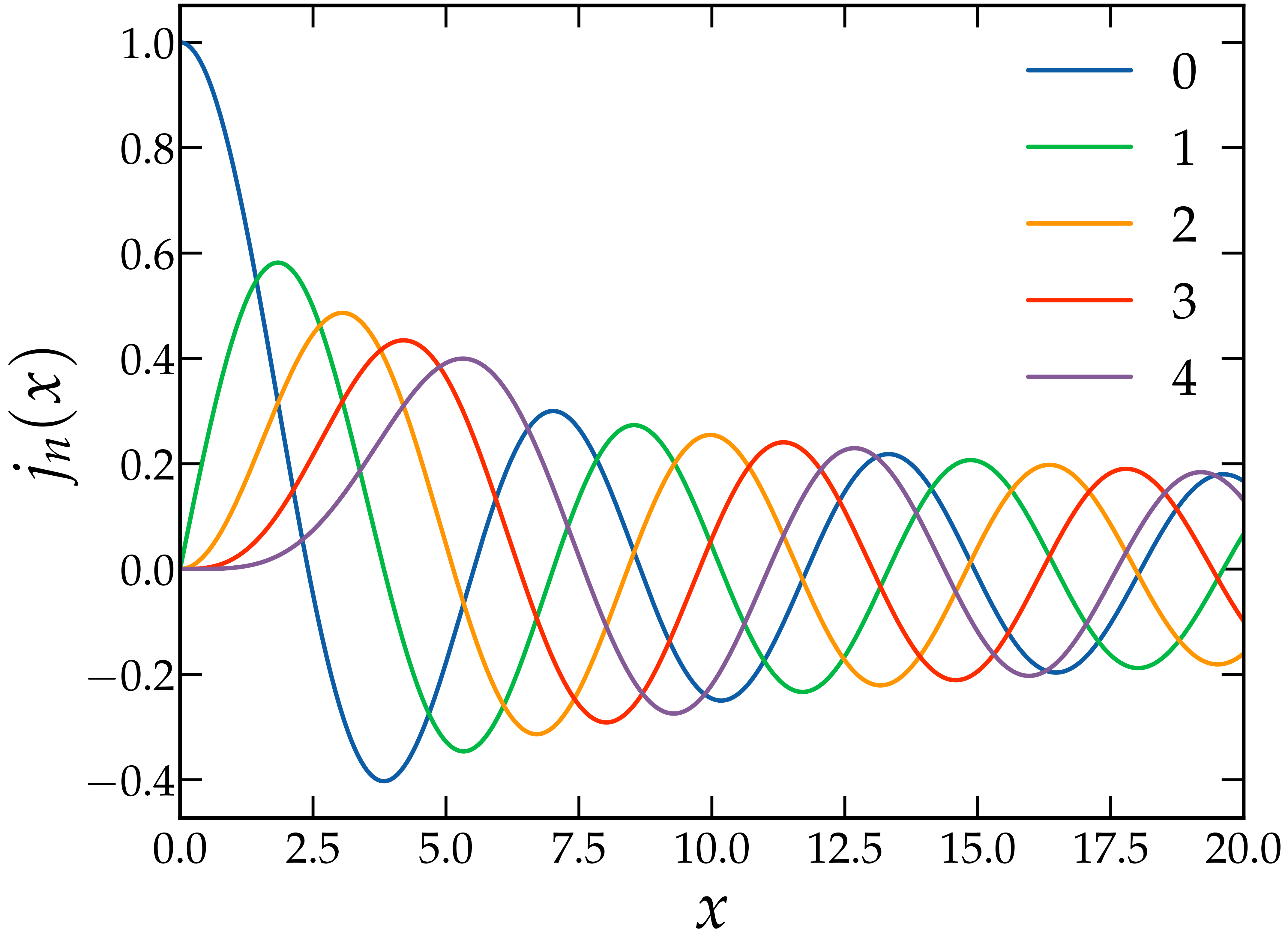
Using my style sheet ('sty.mplstyle' provided in the Confluence materials), you can improve the look of your plot with a single line of code.

Default



Style sheet

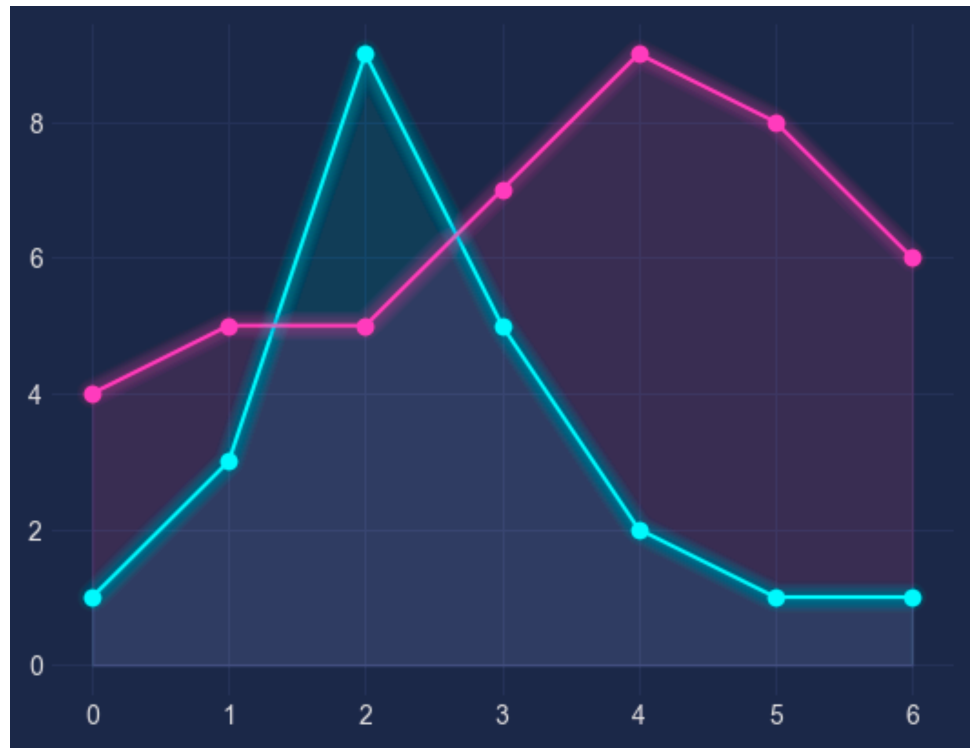
→ `plt.style.use('sty.mplstyle')`

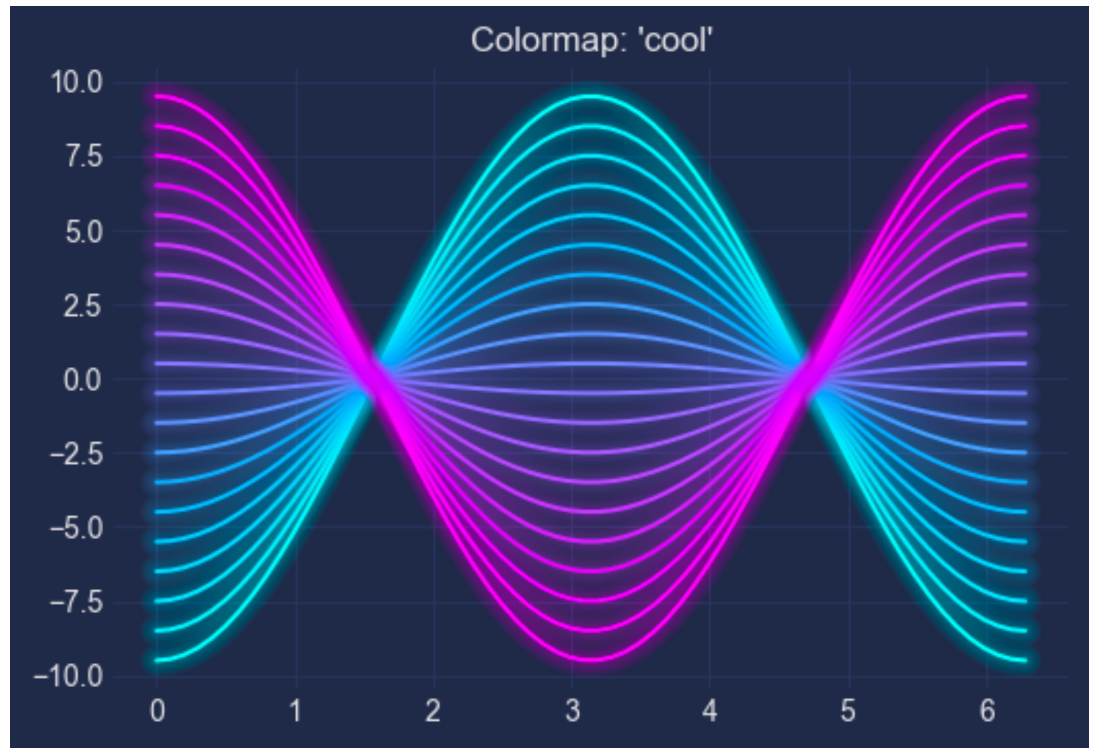


`Defaults.ipynb`

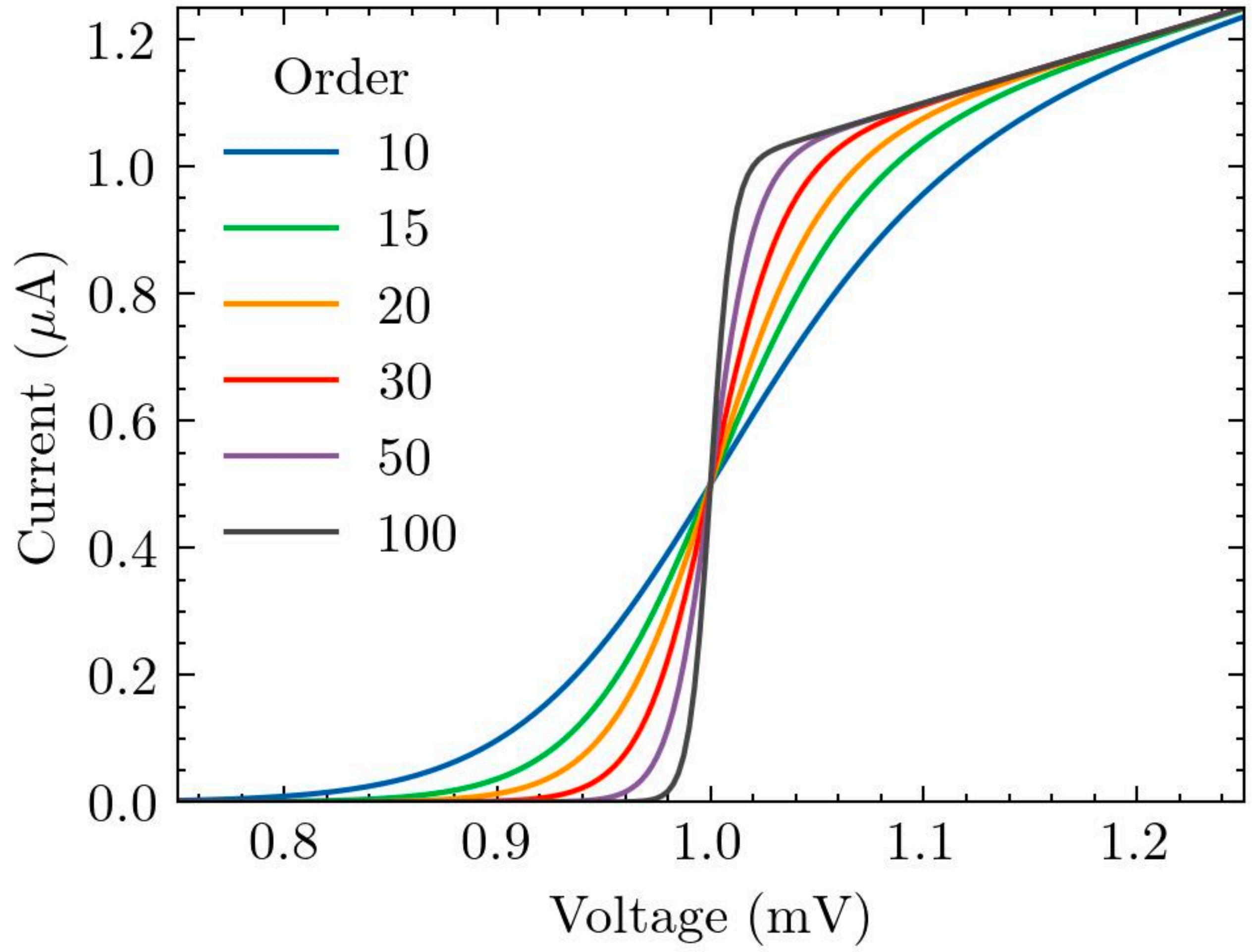
Alternative styles

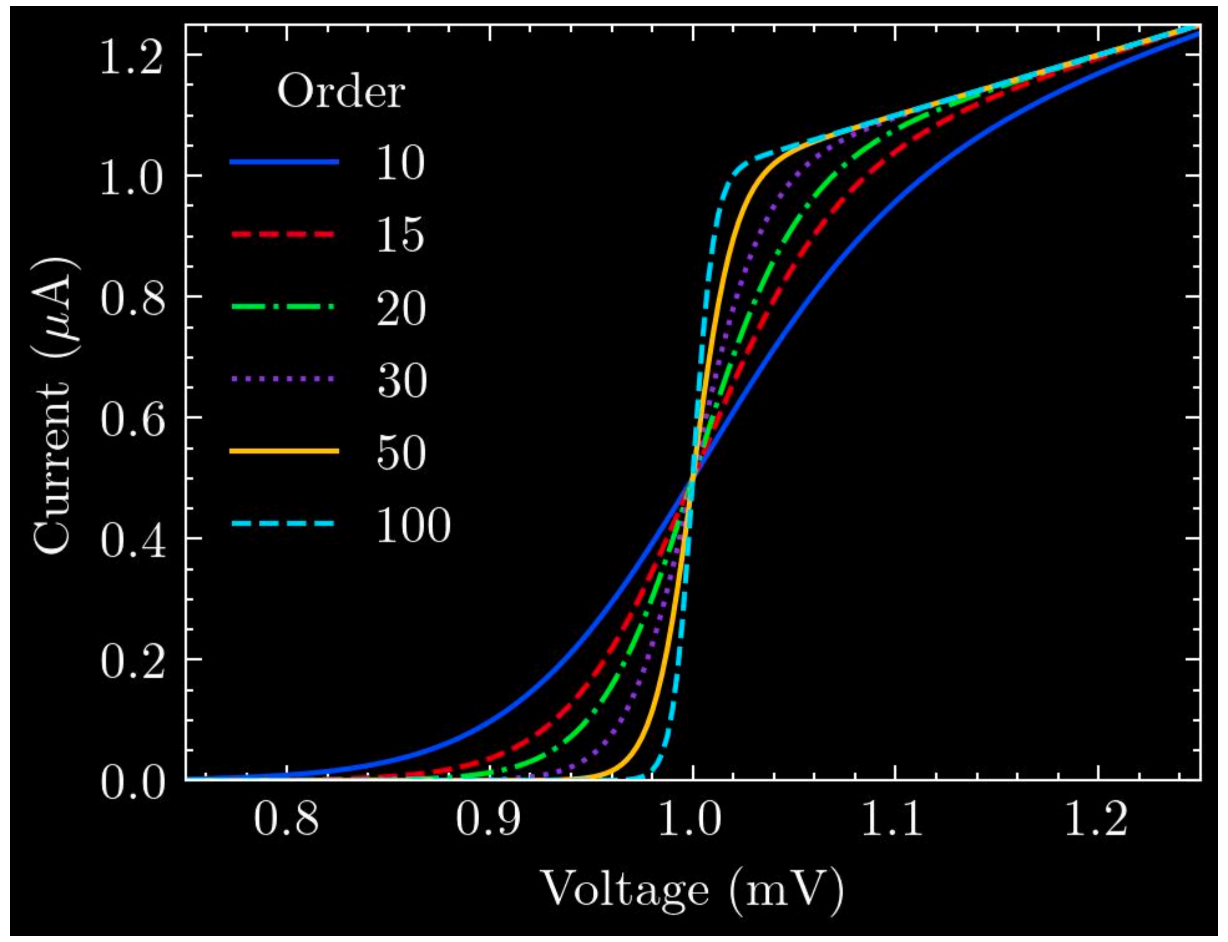
Cyperpunk
https://github.com/dhaitz/mplcyberpunk



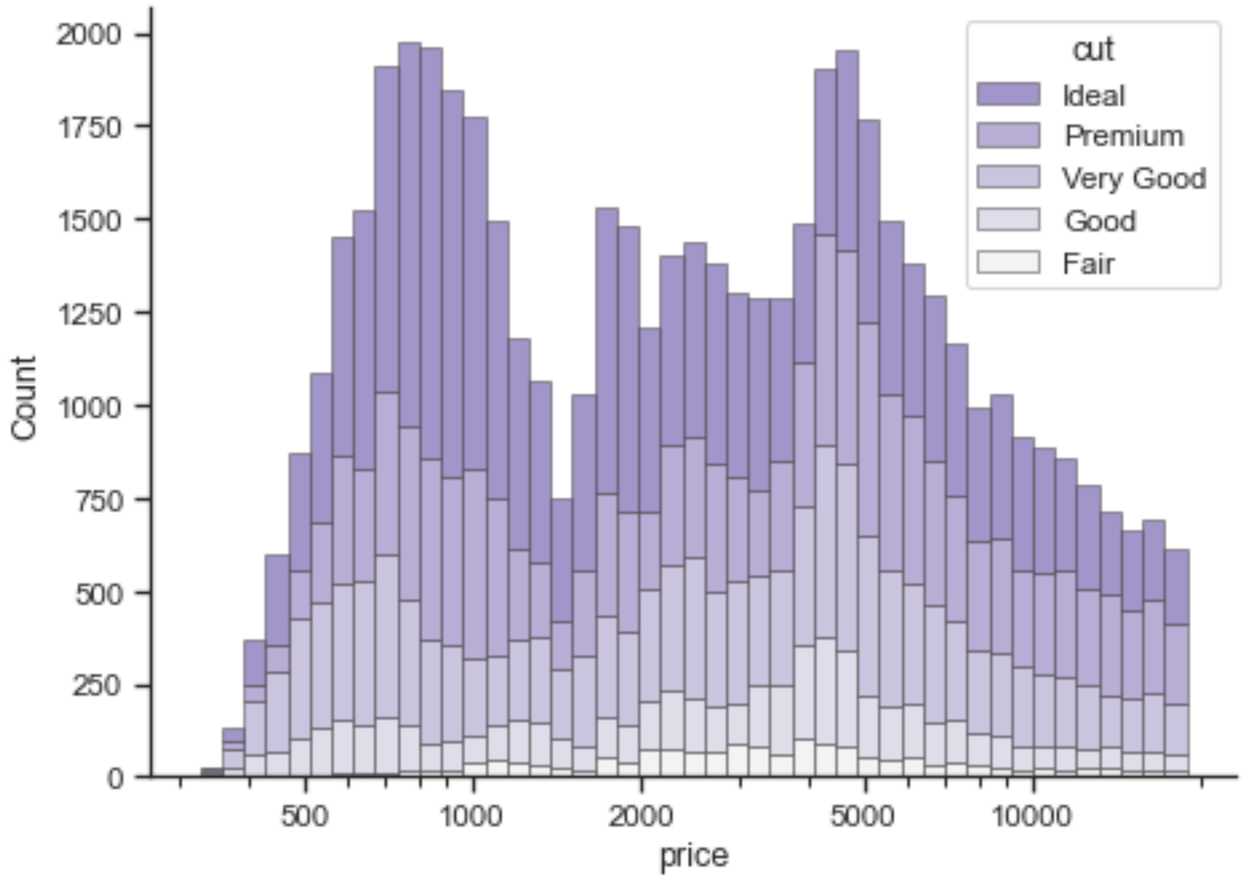


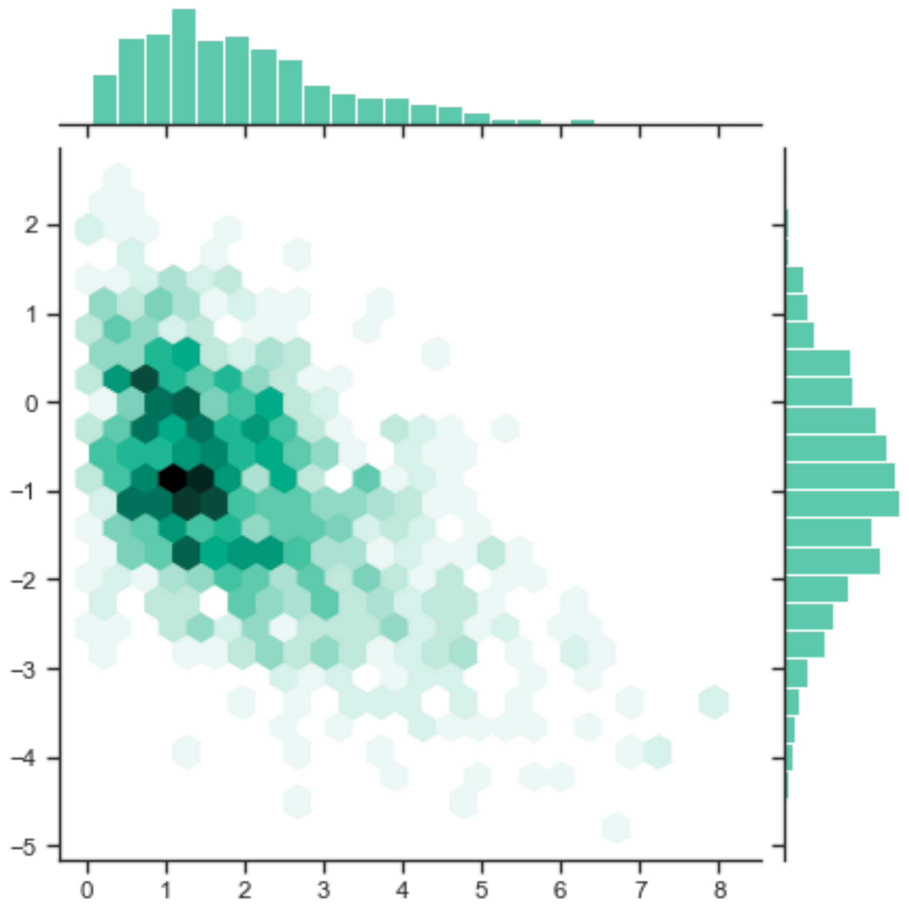
Science plots
https://github.com/garrettj403/SciencePlots



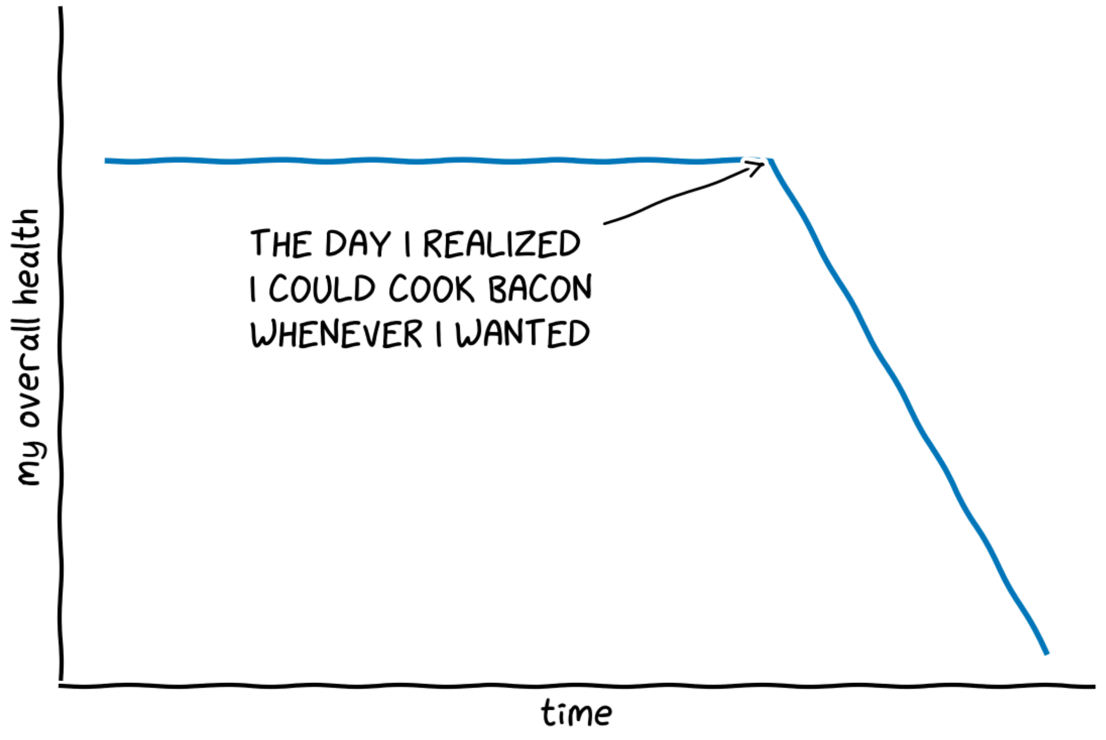


Seaborn
https://seaborn.pydata.org/

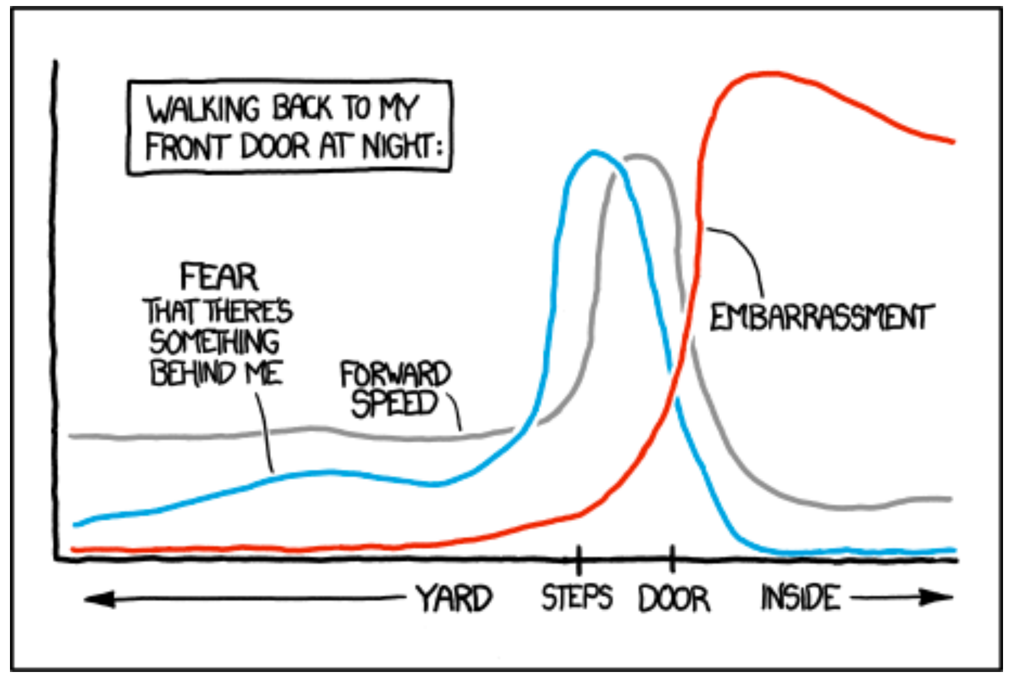




xkcd
https://matplotlib.org/stable/gallery/showcase/xkcd.html

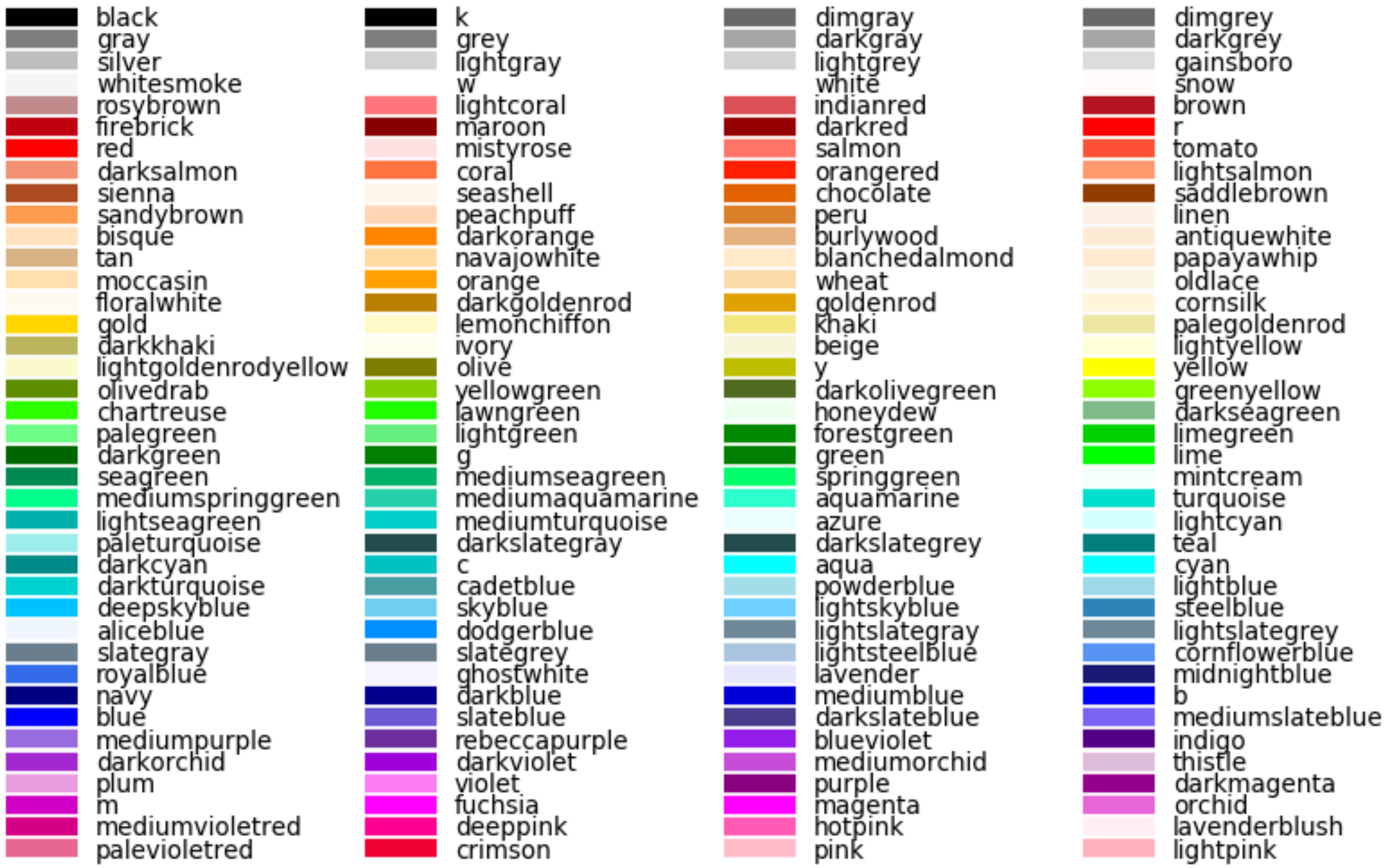


"Stove Ownership" from xkcd by Randall Munroe



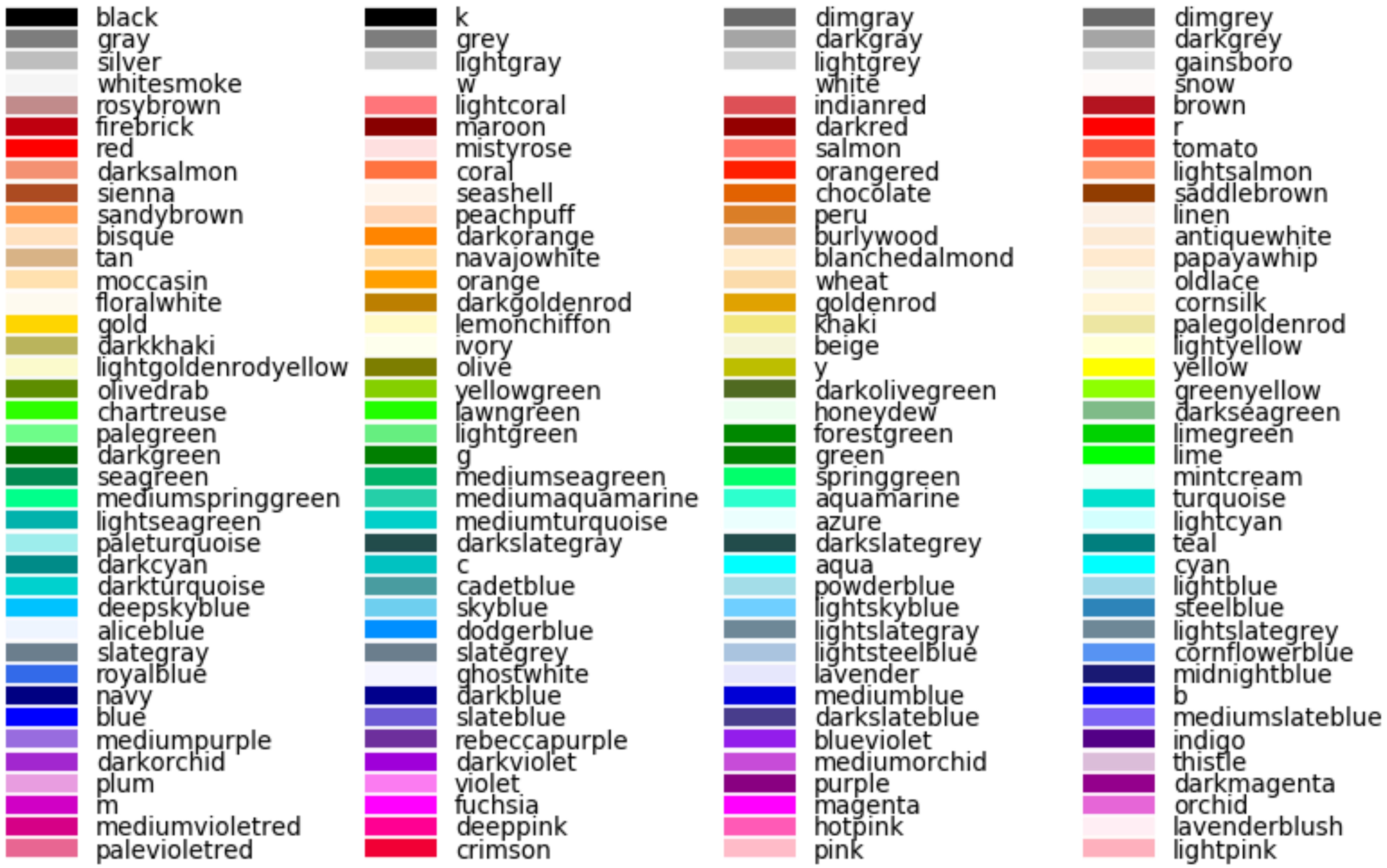
Picking colours

Matplotlib has a long list of confusingly named colours:

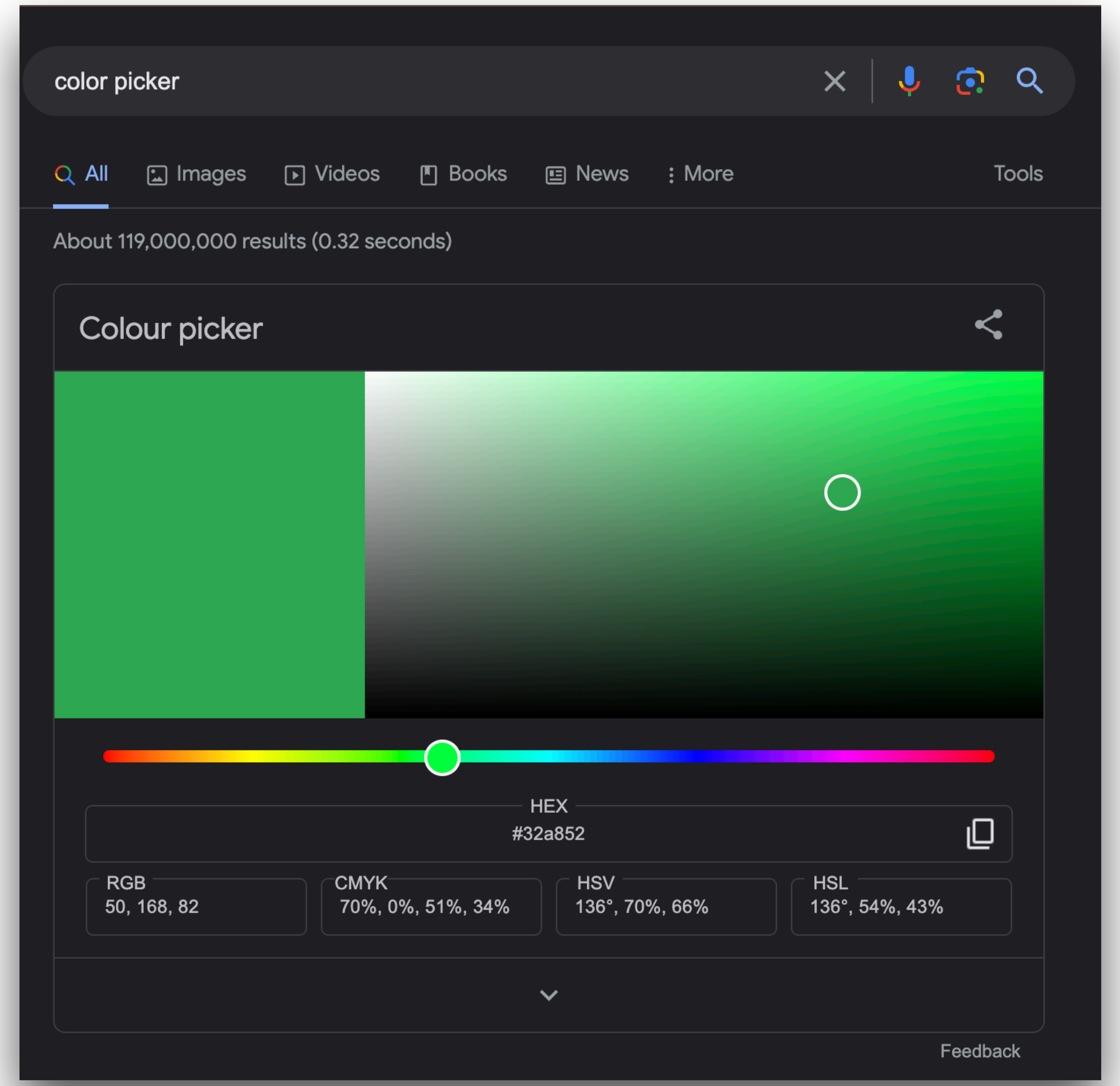


Picking colours

Matplotlib has a long list of confusingly named colours:



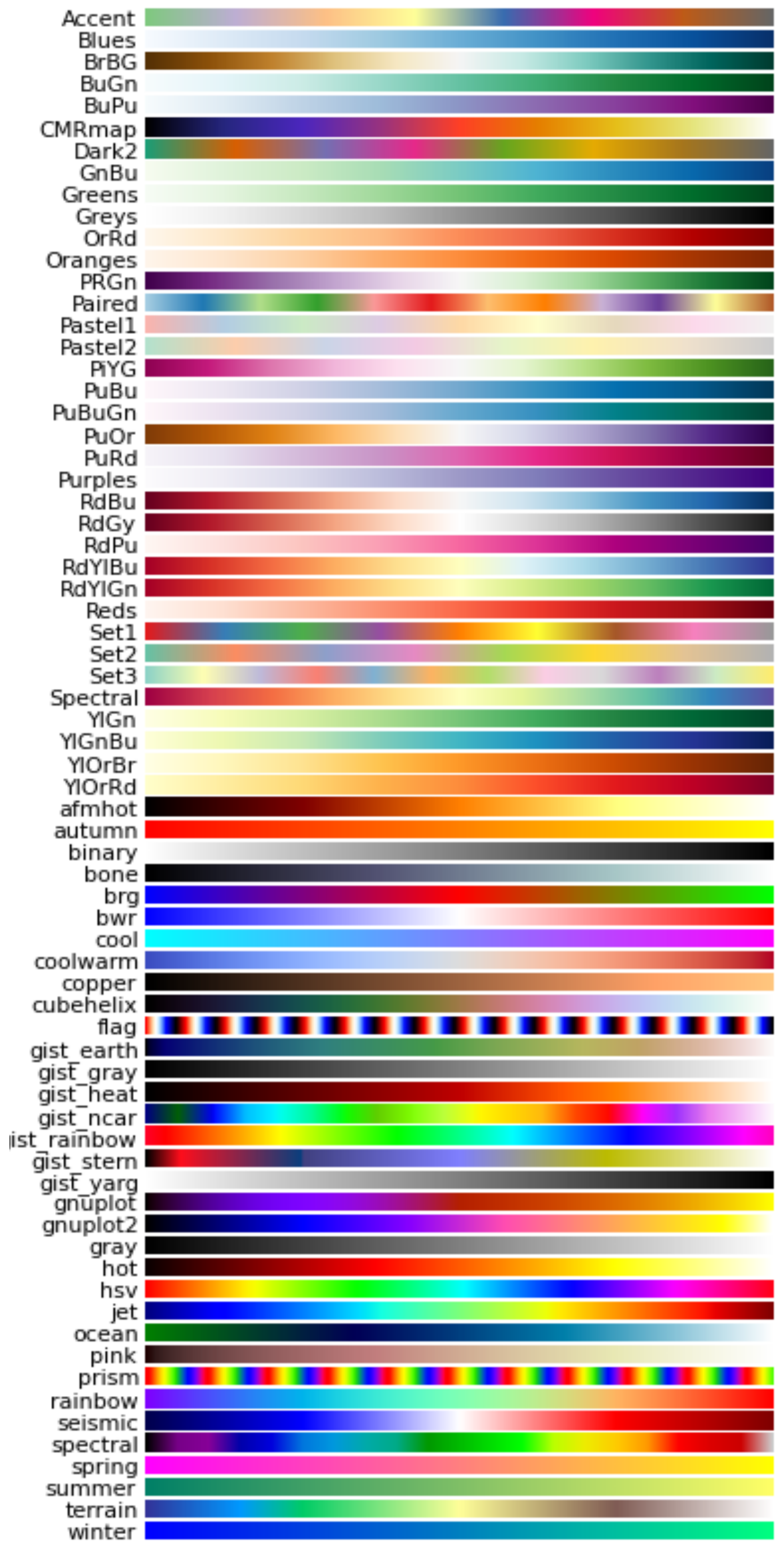
Better way: google "*color picker*" and you can select the HEX code of any colour you want



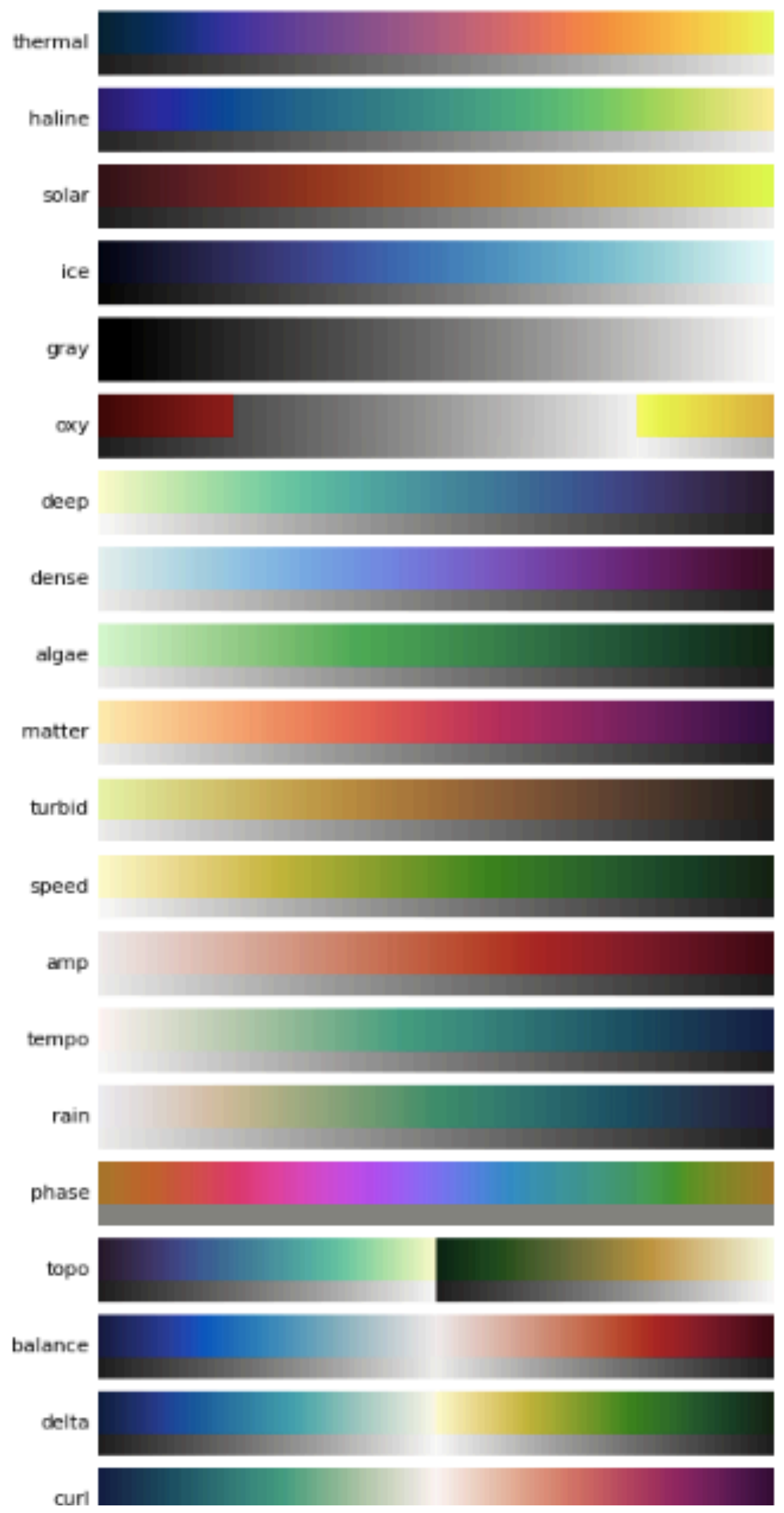
Colormaps

Many options available, don't have to limit yourself to the ones provided by Matplotlib by default (although the default ones are default for a reason)

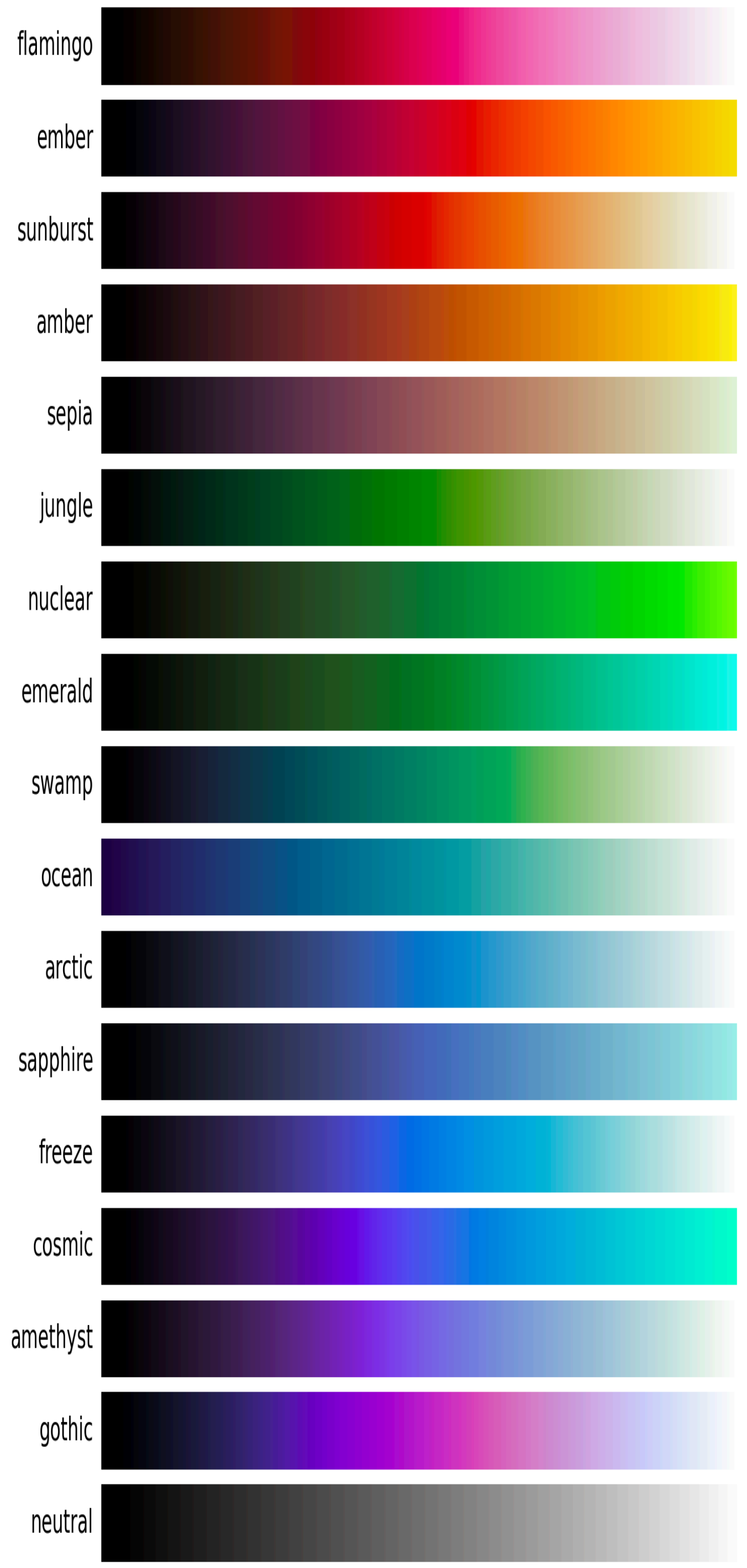
Matplotlib



cmocean



cmasher



Custom (e.g. Img2cmap)

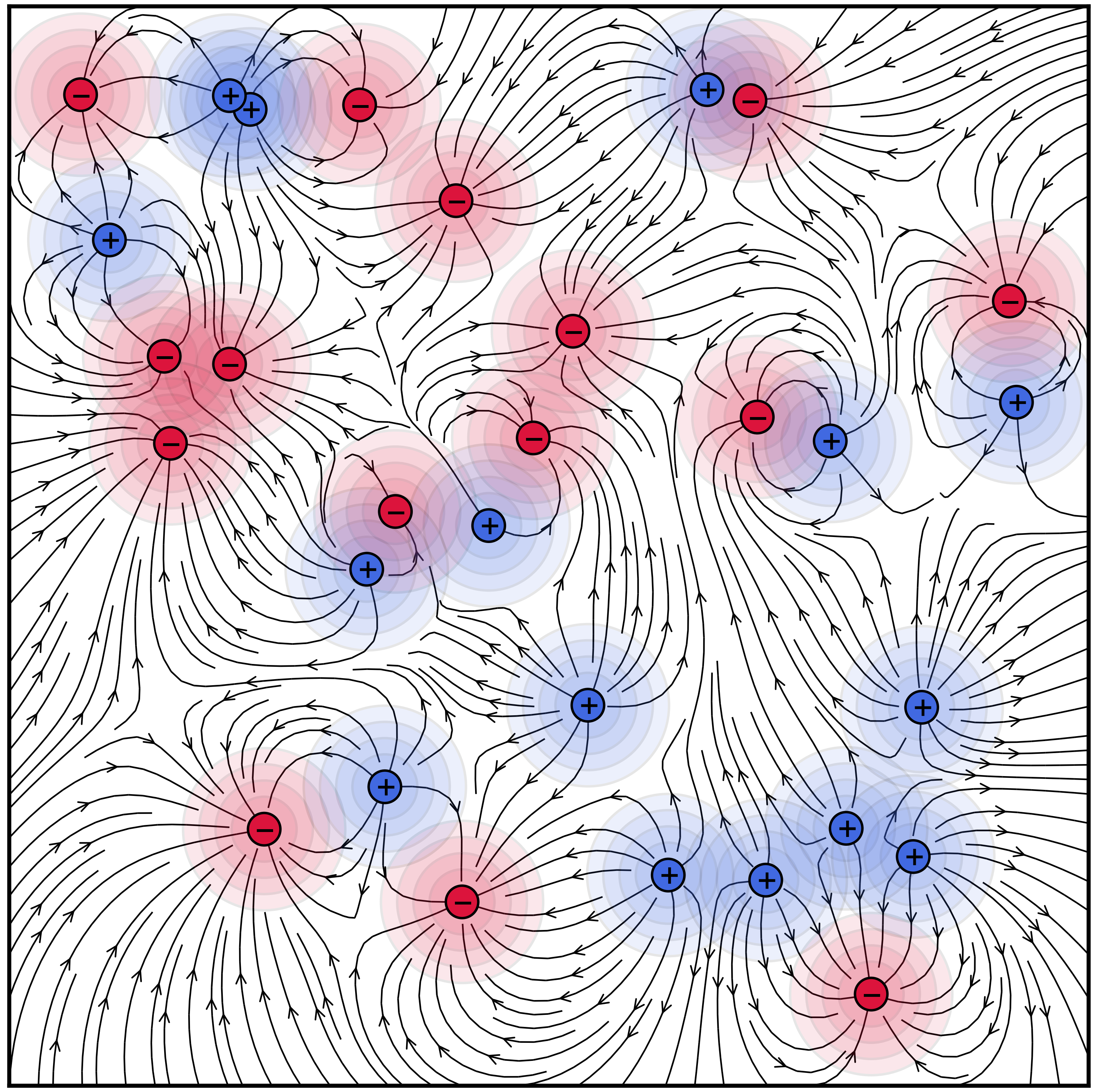
https://github.com/arvkevi/img2cmap

See how to make a colormap out of any list of colors: https://matplotlib.org/stable/gallery/color/custom_cmap.html

Colormaps

Tip: watch the file size!
If you use, `scatter`, `imshow`, `pcolormesh`, etc. to display large datasets the file size can get pretty huge. To fix that set “`rasterized=True`” inside the function, e.g. `plt.pcolormesh(… rasterized=True)`

You cannot solely opt for aesthetics, you must choose the correct type for your data

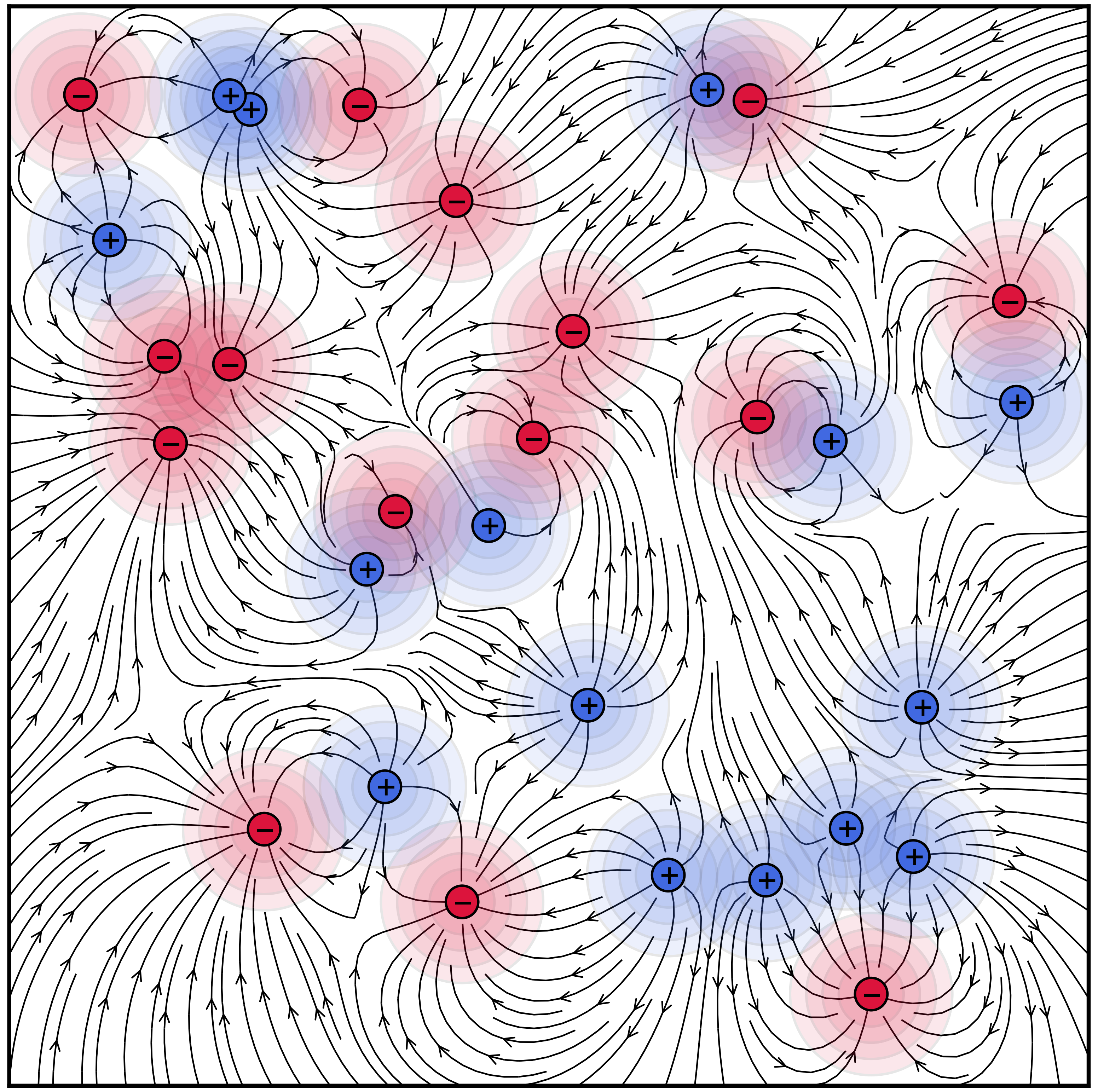


Charges.ipynb

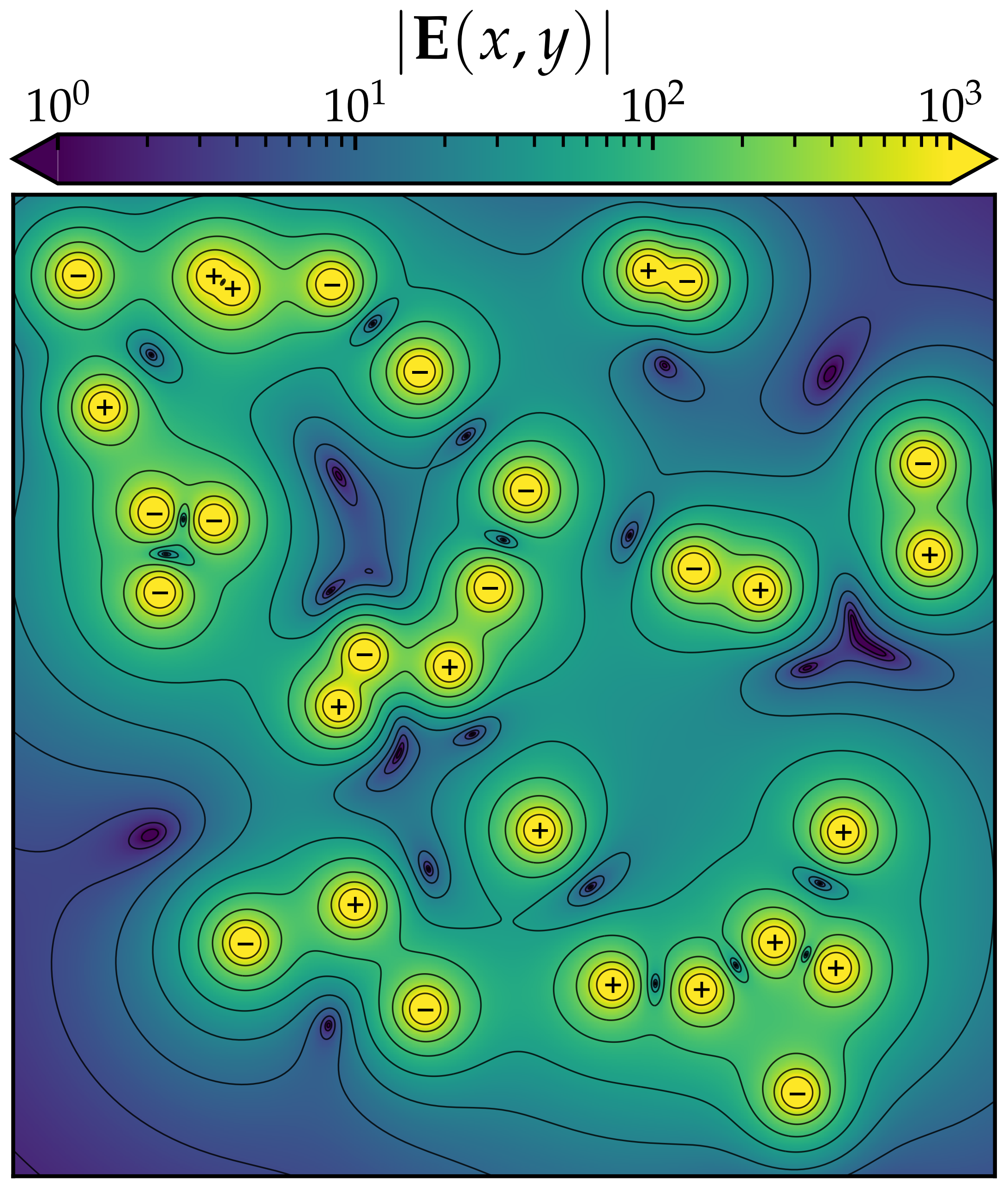
Colormaps

Tip: watch the file size!
If you use, `scatter`, `imshow`, `pcolormesh`, etc. to display large datasets the file size can get pretty huge. To fix that set "`rasterized=True`" inside the function, e.g. `plt.pcolormesh(… rasterized=True)`

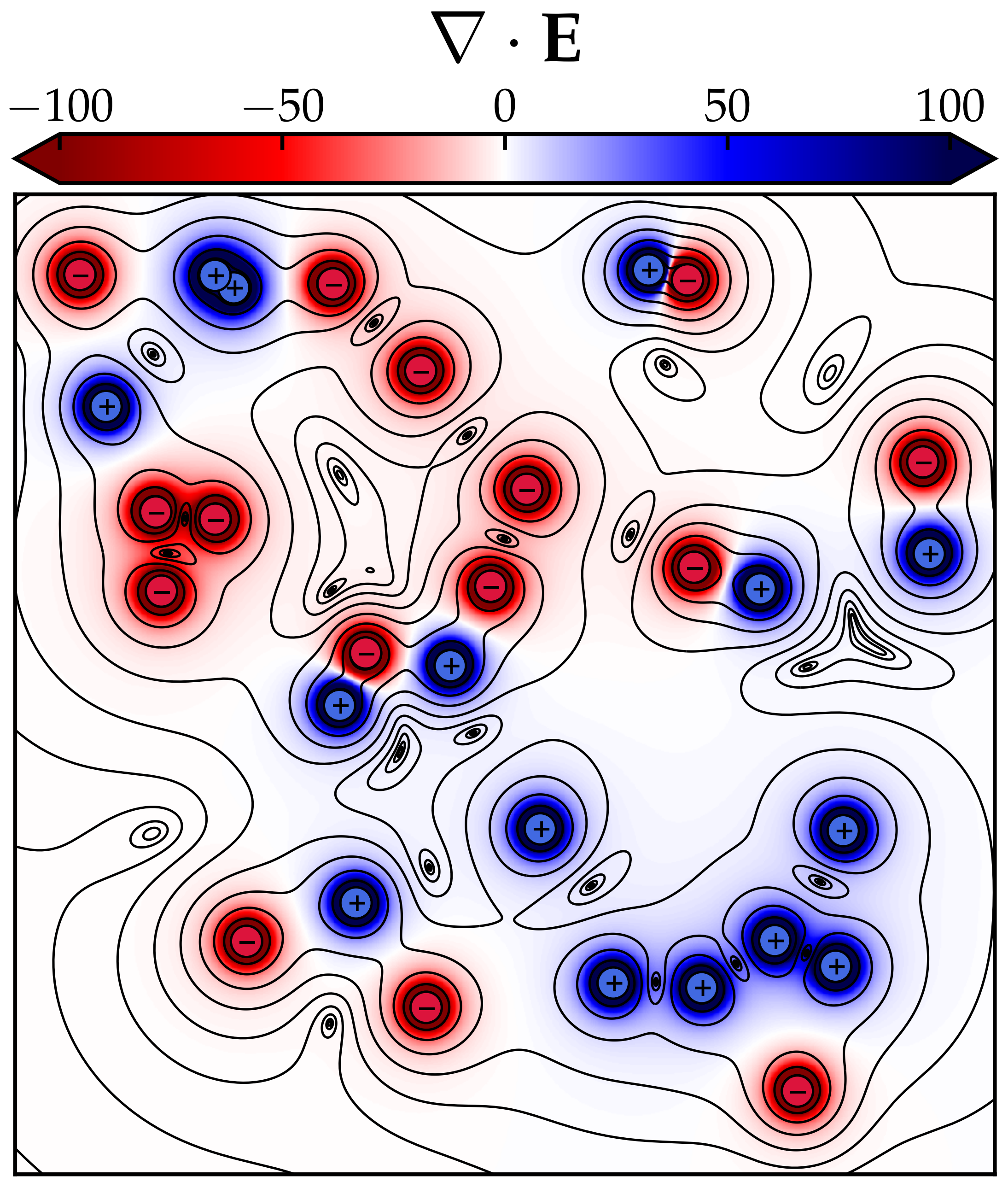
You cannot solely opt for aesthetics, you must choose the correct type for your data



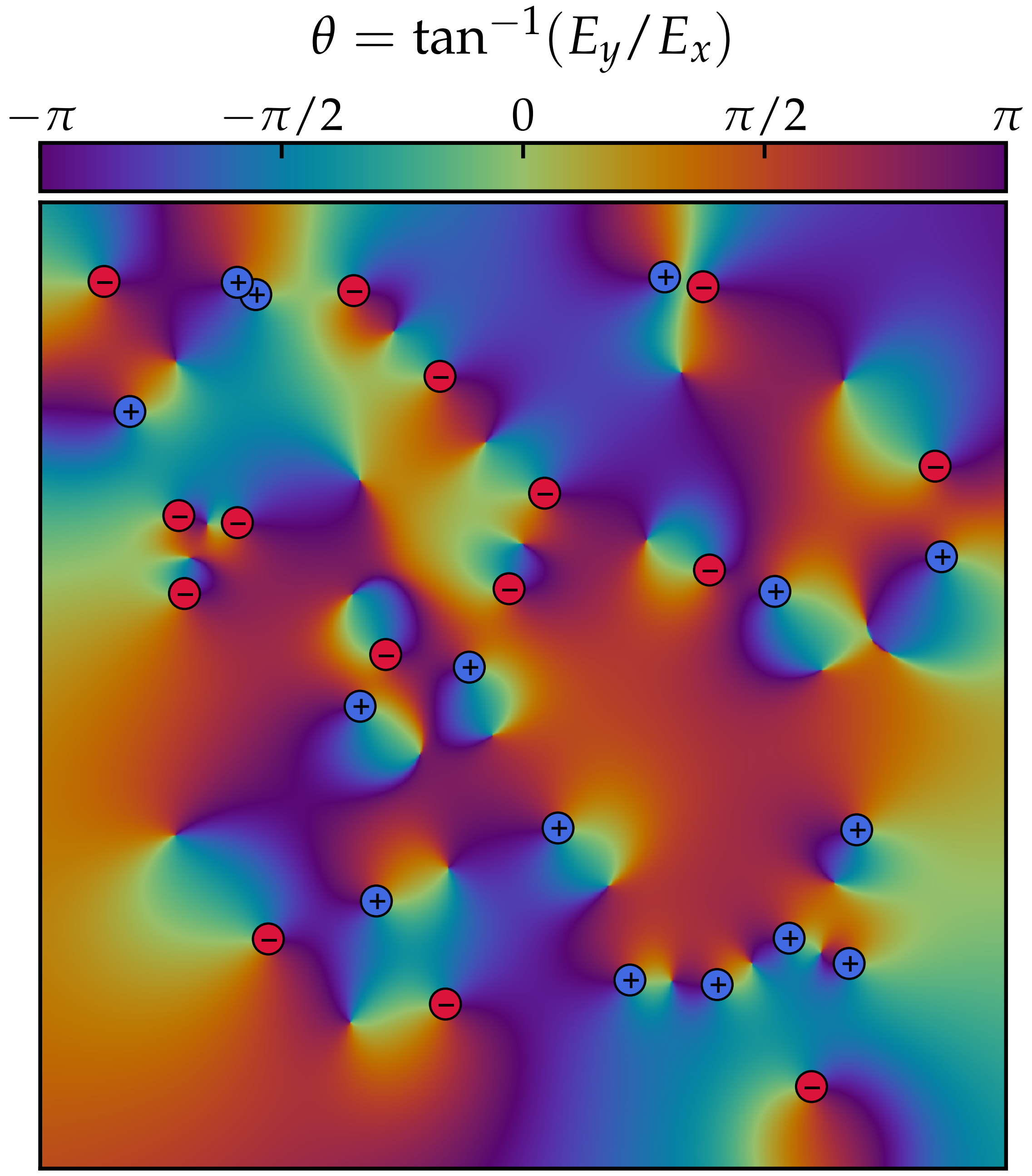
Charges.ipynb



Sequential (cm.viridis)



Diverging (cm.seismic)

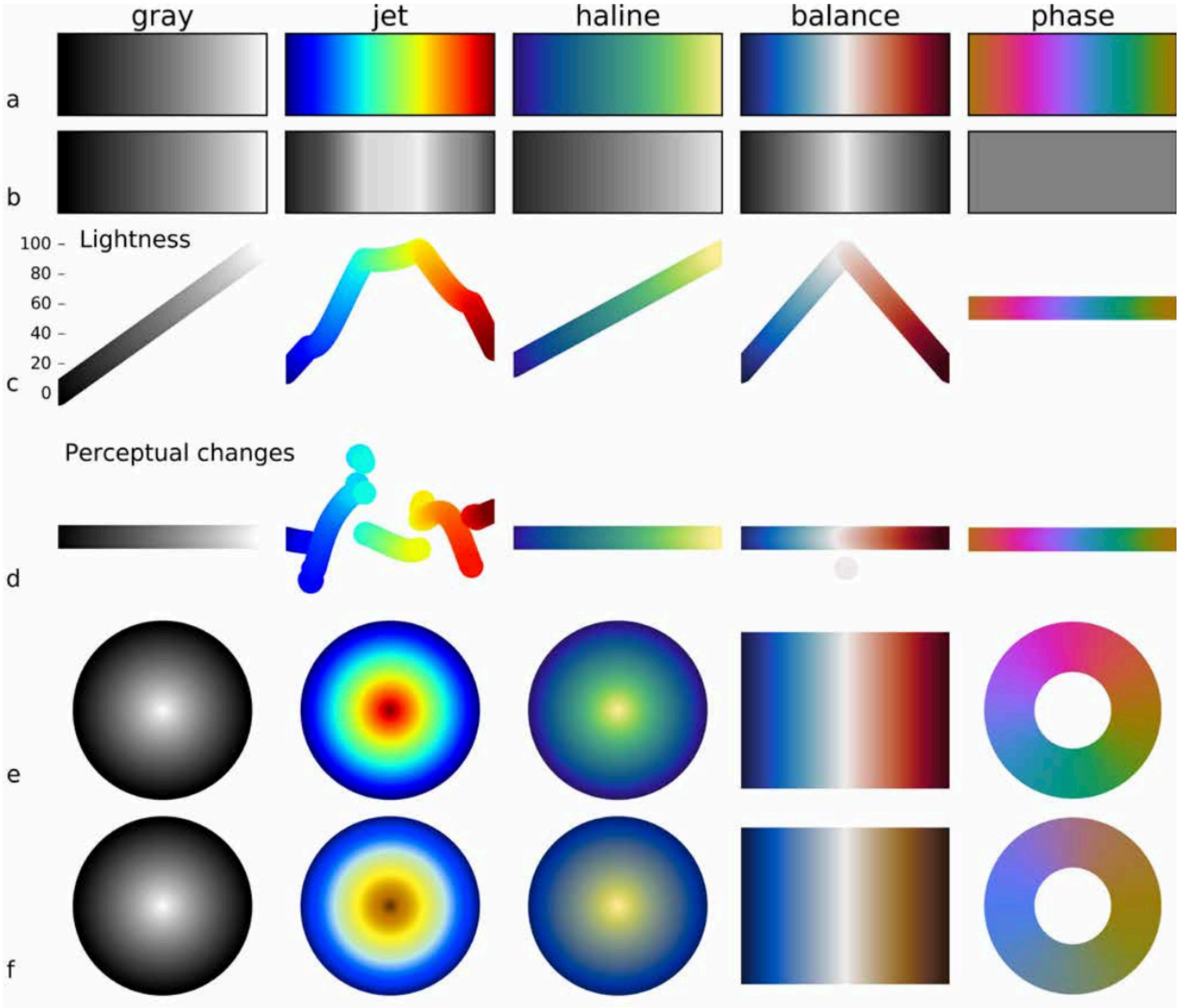


Cyclic (cmasher.infinity)

Visually uniform colormaps

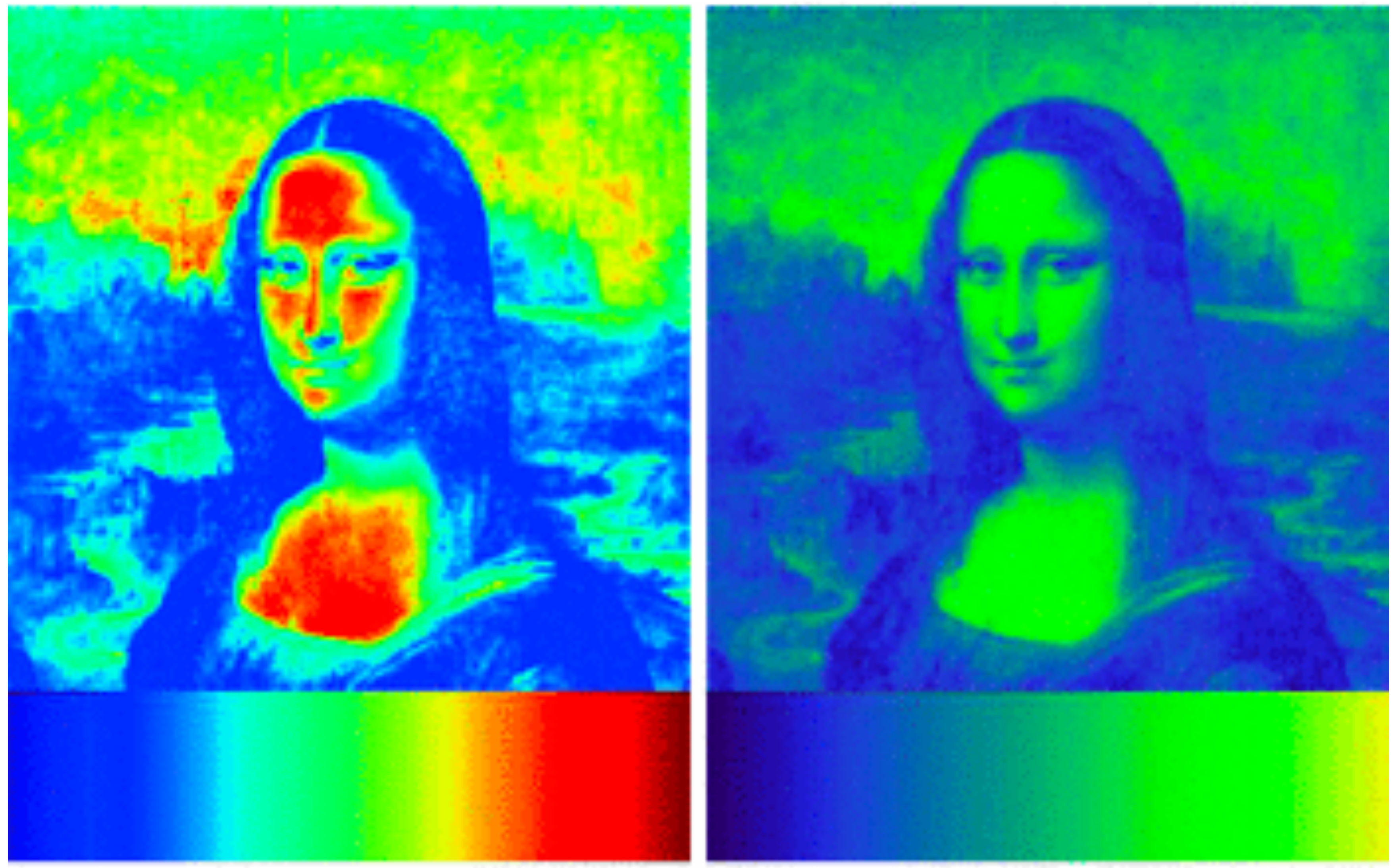
https://colorcet.holoviz.org/

For sequential colormaps you want to have a uniform perceptual change as a function of distance through the colormap. Not all colormaps have this (the default viridis does)



https://tos.org/oceanography/article/true-colors-of-oceanography-guidelines-for-effective-and-accurate-colormap

Some colormaps induce artificial levels in the data due to perceptual discontinuities
→ Bad! Don't use them!



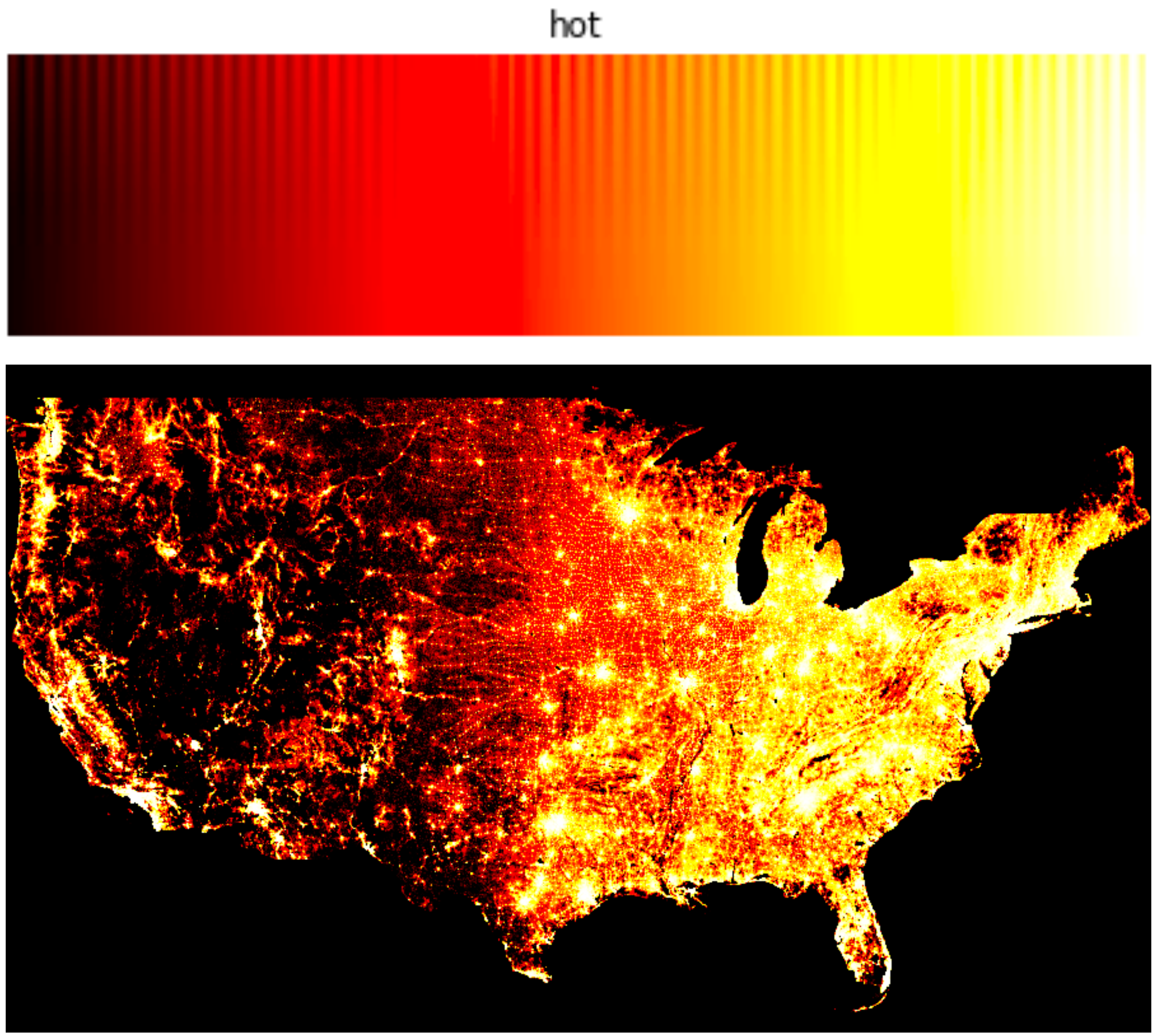
(a) Jet colormap.

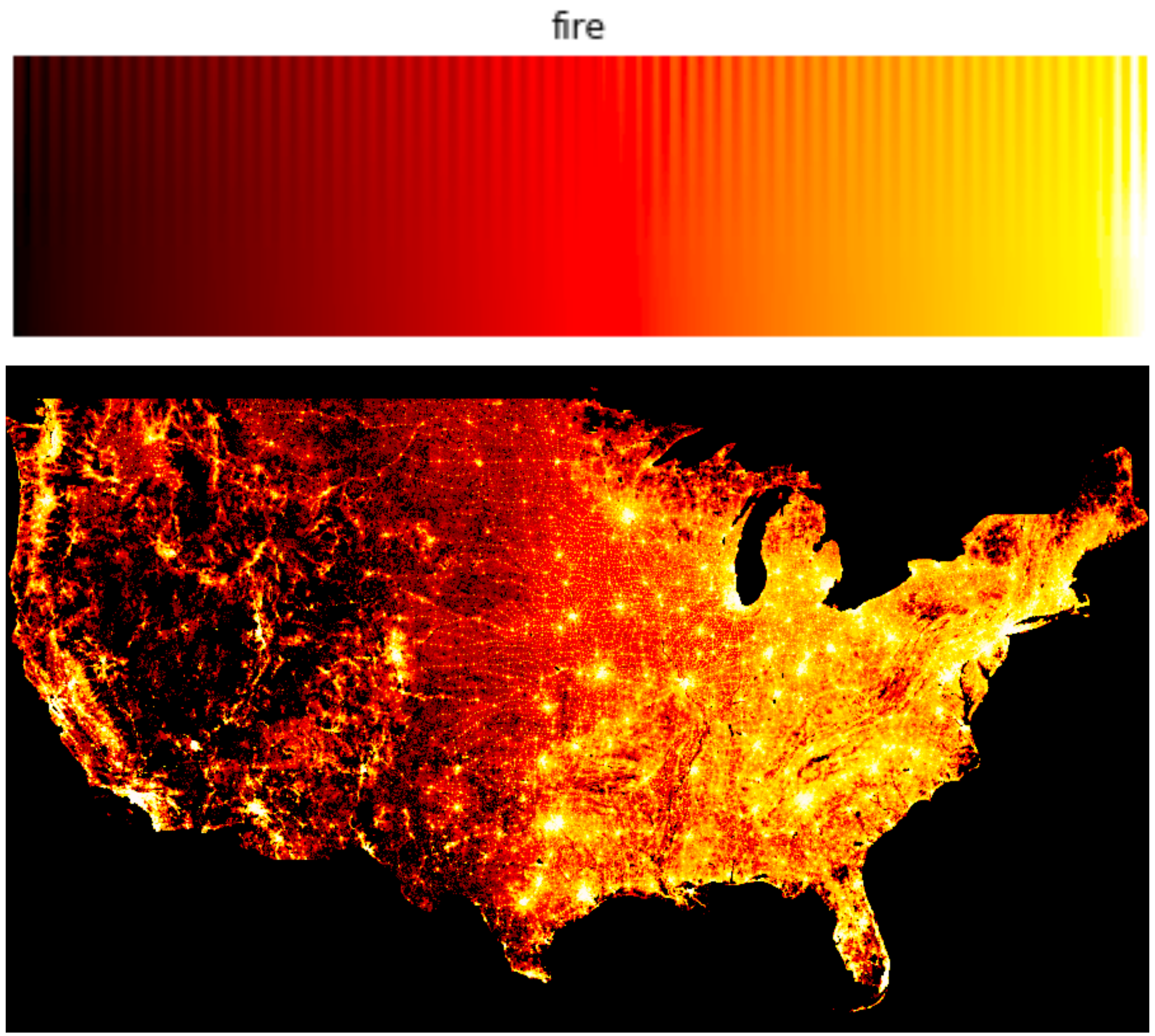
(b) Viridis colormap.

Visually uniform colormaps

https://colorcet.holoviz.org/

For sequential colormaps you want to have a unifiorm perceptual change as a function of distance through the colormap. Not all colormaps have this (the default *viridis* does)



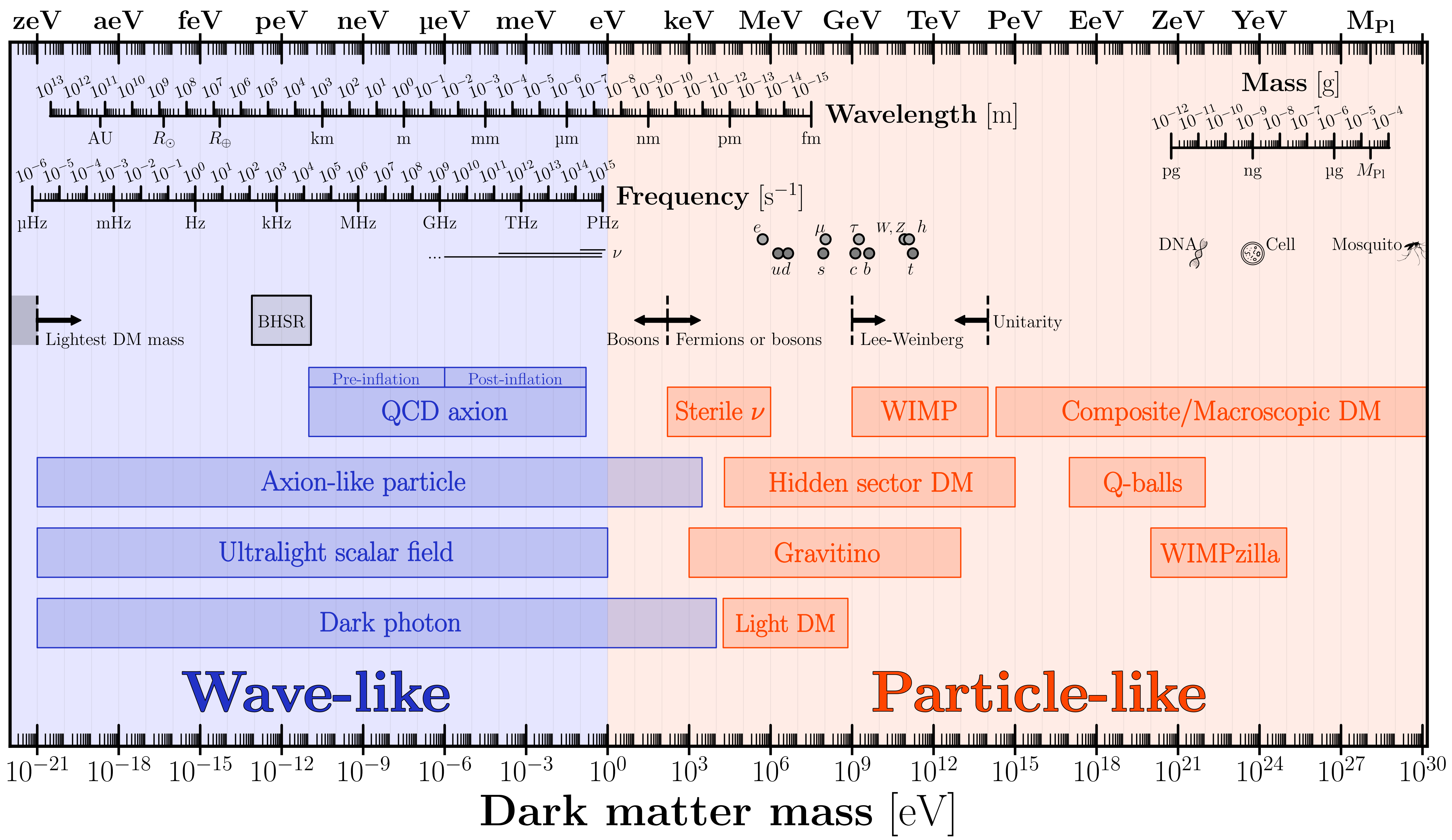


Text and fonts

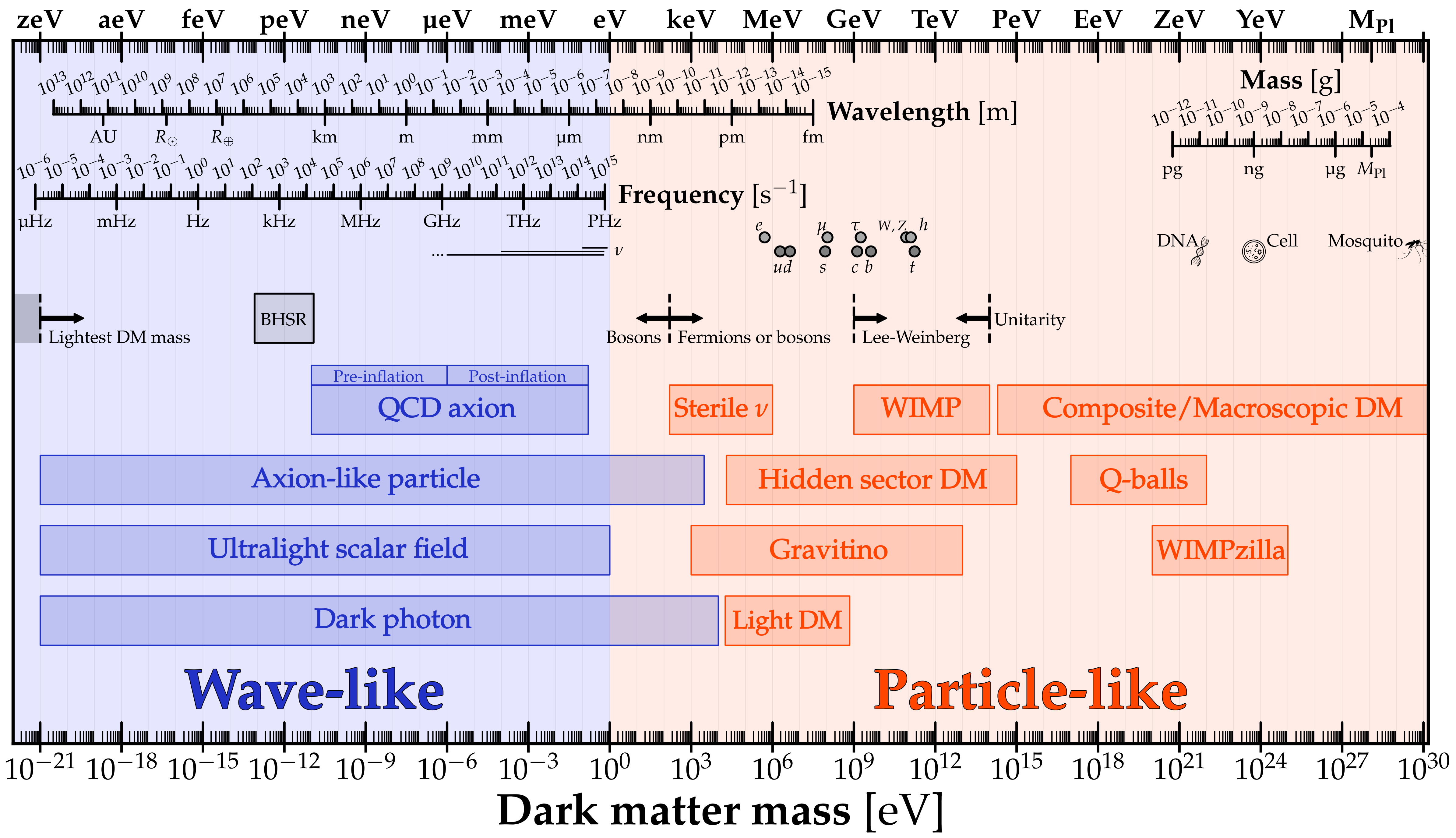
Tip: Placing text labels
→ `plt.text(x,y,'label')` will add a text label at the point (x,y) defined by whatever your axis coordinates are.
→`plt.gcf().text(x,y,'label')` will add a label to the figure itself where (x,y) are defined with respect to the bottom left corner, e.g. (0.5,0.5) is the middle of the figure.

For papers your plot will look best if all text (including labels, tick marks, etc.) is rendered as TeX. I am a big advocate for the "Palatino" font.

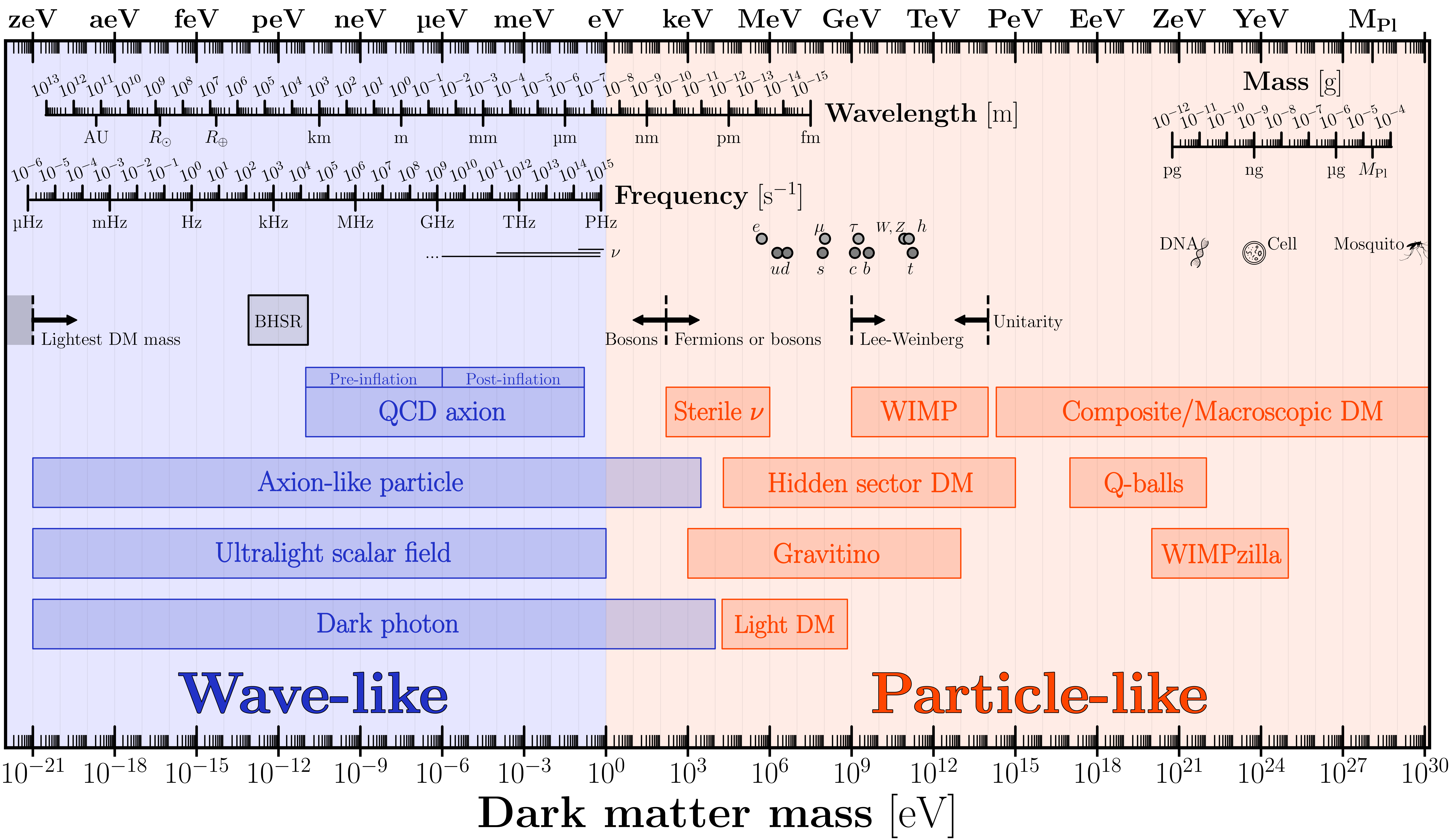
CM



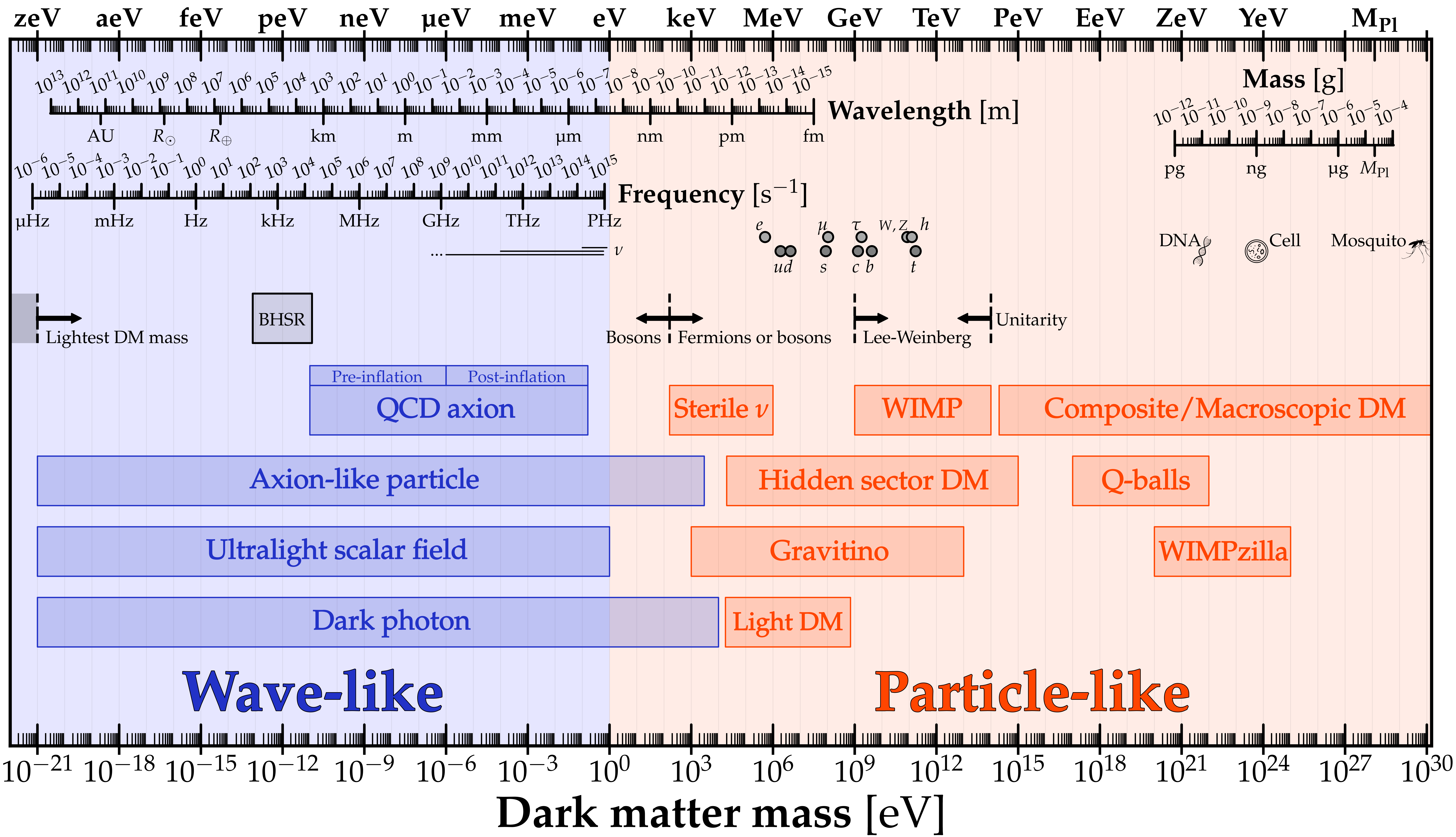
Palatino



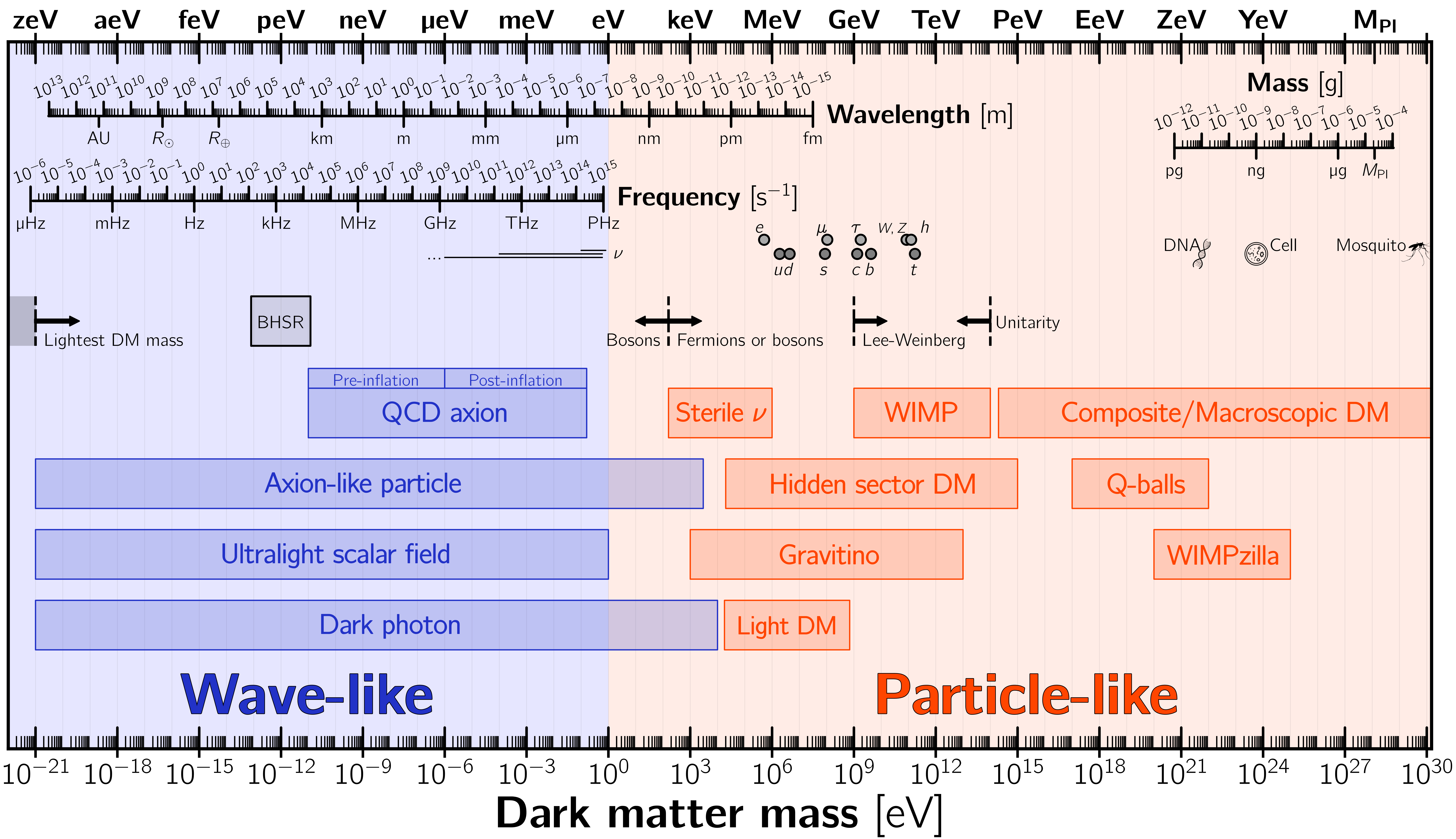
DarkMatterLandscape.ipynb



zeV aeV feV peV neV μeV meV eV keV MeV GeV TeV PeV EeV ZeV YeV M_Pl
Wavelength [m]
AU
km
m
mm
μm
nm
pm
fm
Frequency [s^-1]
μHz
mHz
Hz
kHz
MHz
GHz
THz
PHz
Mass [g]
pg
ng
μg
DNA
Cell
Mosquito
Lightest DM mass
BHSR
Bosons
Fermions or bosons
Lee-Weinberg
Unitarity
Pre-inflation
Post-inflation
QCD axion
Sterile ν
WIMP
Composite/Macroscopic DM
Axion-like particle
Hidden sector DM
Q-balls
Ultralight scalar field
Gravitino
WIMPzilla
Dark photon
Light DM
Wave-like
Particle-like
Dark matter mass [eV]



zeV aeV feV peV neV μeV meV eV keV MeV GeV TeV PeV EeV ZeV YeV M_{Pl}
Wavelength [m]
AU $R_\odot$ $R_\oplus$ km m mm μm nm pm fm
Frequency [s^{-1}]
μHz mHz Hz kHz MHz GHz THz PHz
Mass [g]
pg ng μg M_{Pl}
e μ τ W, Z h ud s c b t ν
DNA Cell Mosquito
Lightest DM mass
BHSR
Bosons
Fermions or bosons
Lee-Weinberg
Unitarity
Pre-inflation
Post-inflation
QCD axion
Sterile ν
WIMP
Composite/Macroscopic DM
Axion-like particle
Hidden sector DM
Q-balls
Ultralight scalar field
Gravitino
WIMPzilla
Dark photon
Light DM
Wave-like
Particle-like
Dark matter mass [eV]



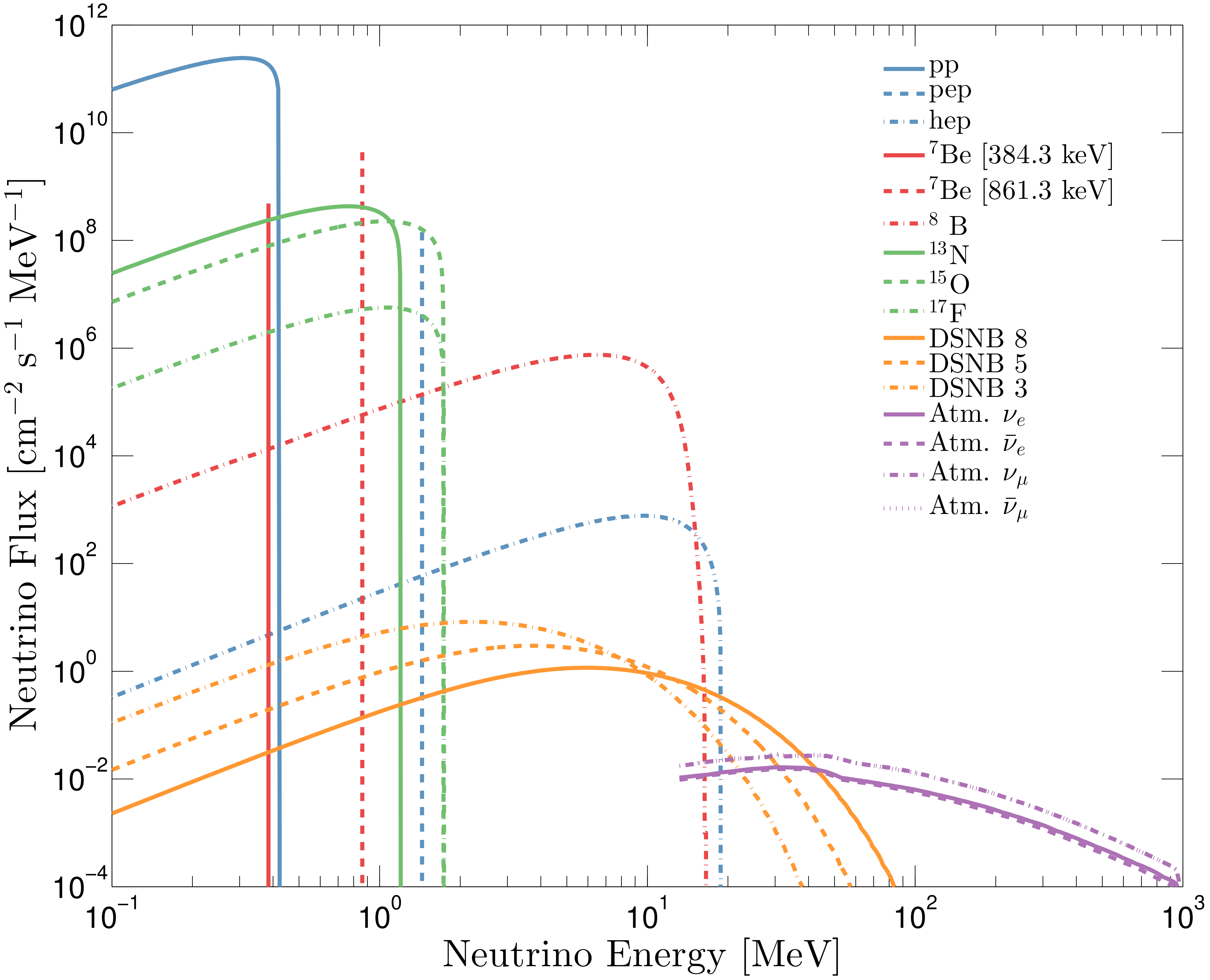
zeV aeV feV peV neV μeV meV eV keV MeV GeV TeV PeV EeV ZeV YeV M_Pl
Wavelength [m]
AU
R_⊙
R_⊕
km m mm μm nm pm fm
Mass [g]
pg ng μg M_Pl
Frequency [s^−1]
μHz mHz Hz kHz MHz GHz THz PHz
ν
e μ τ W, Z h
u d s c b t
DNA
Cell
Mosquito
Lightest DM mass
BHSR
Bosons
Fermions or bosons
Lee-Weinberg
Unitarity
Pre-inflation
Post-inflation
QCD axion
Sterile ν
WIMP
Composite/Macroscopic DM
Axion-like particle
Hidden sector DM
Q-balls
Ultralight scalar field
Gravitino
WIMPzilla
Dark photon
Light DM
Wave-like
Particle-like
Dark matter mass [eV]

Examples in the wild

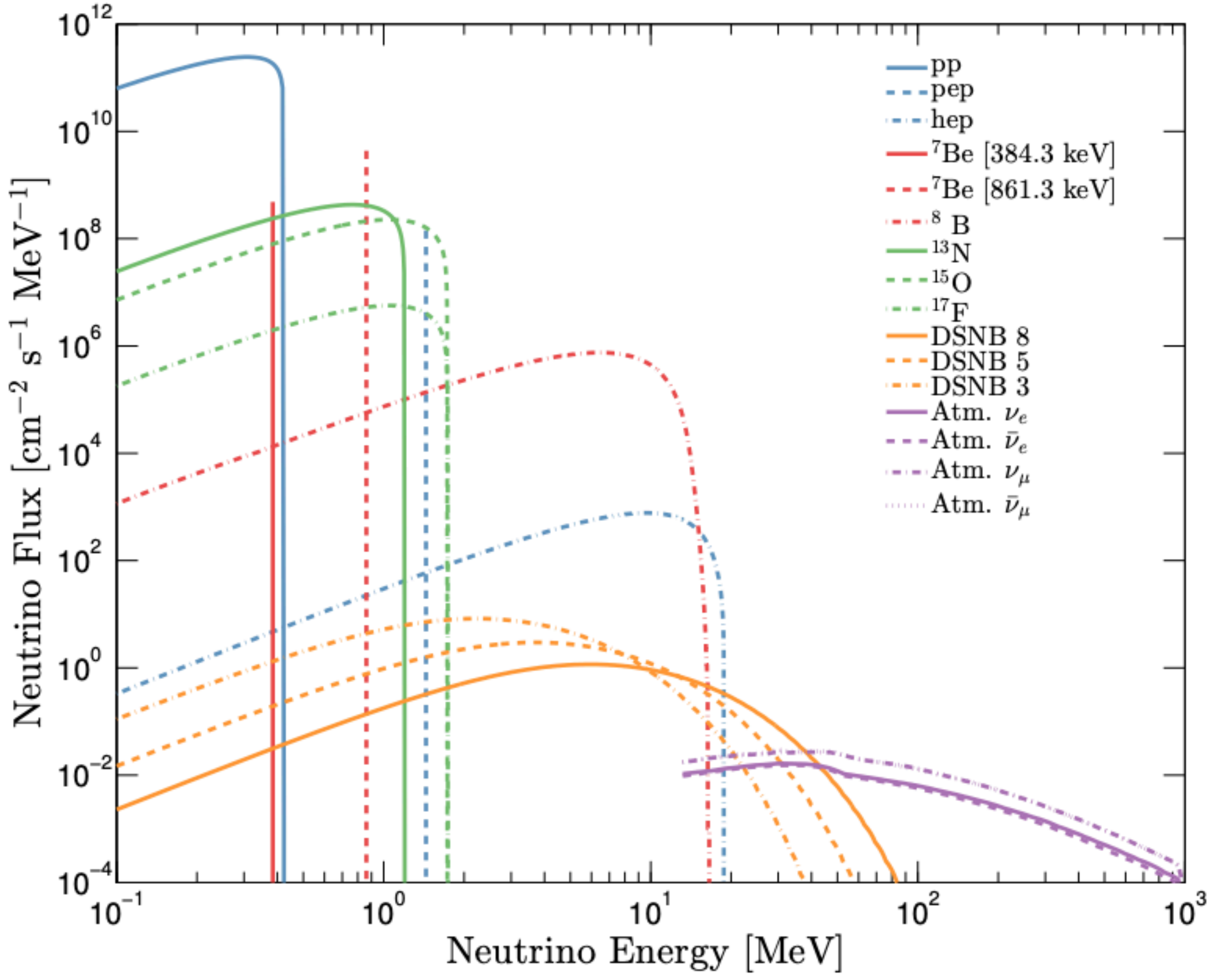
Examples of bad practice: my own plot

In what ways is this plot bad?

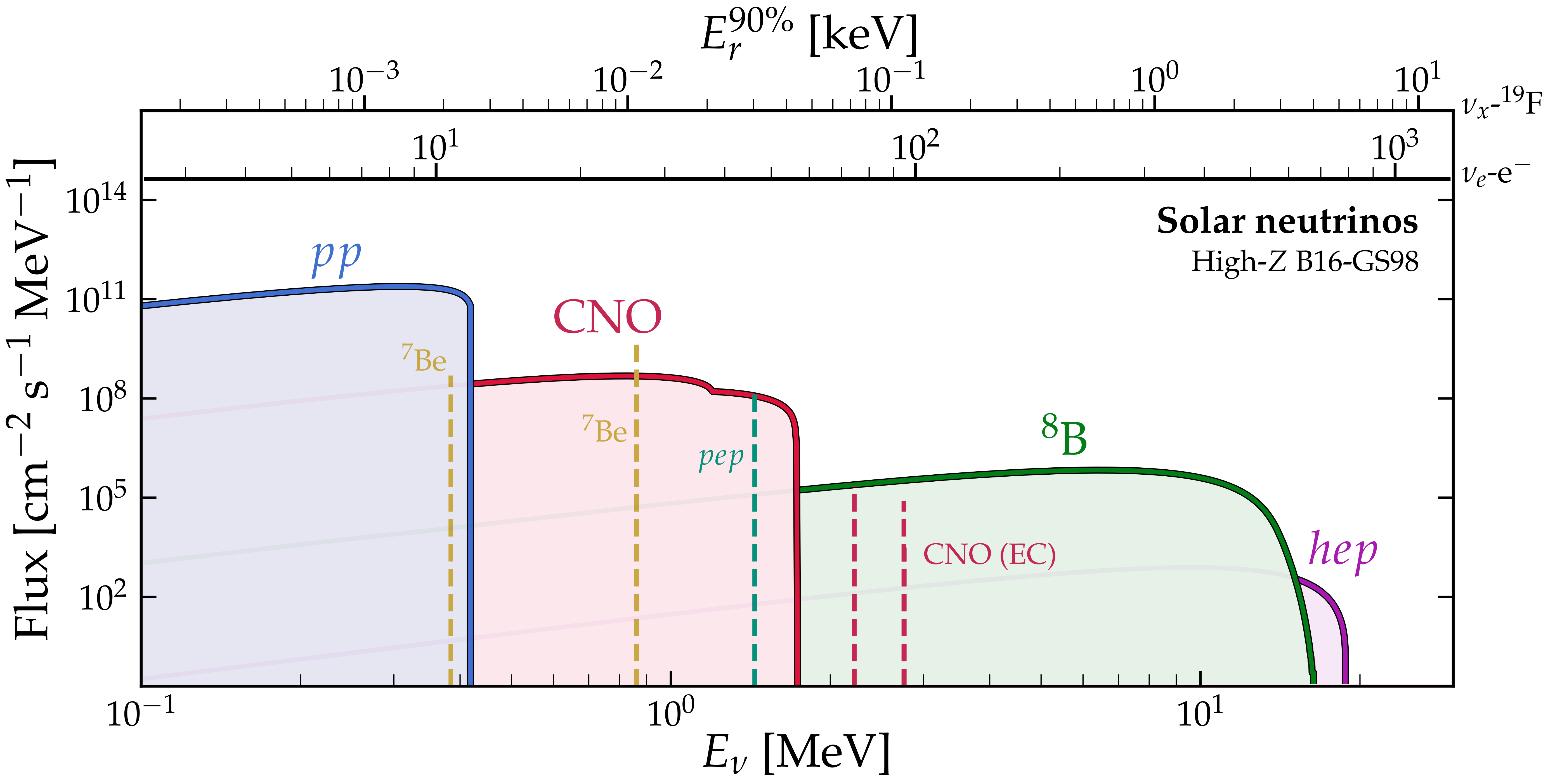
Plot from my 2nd paper (2015)



2015 version



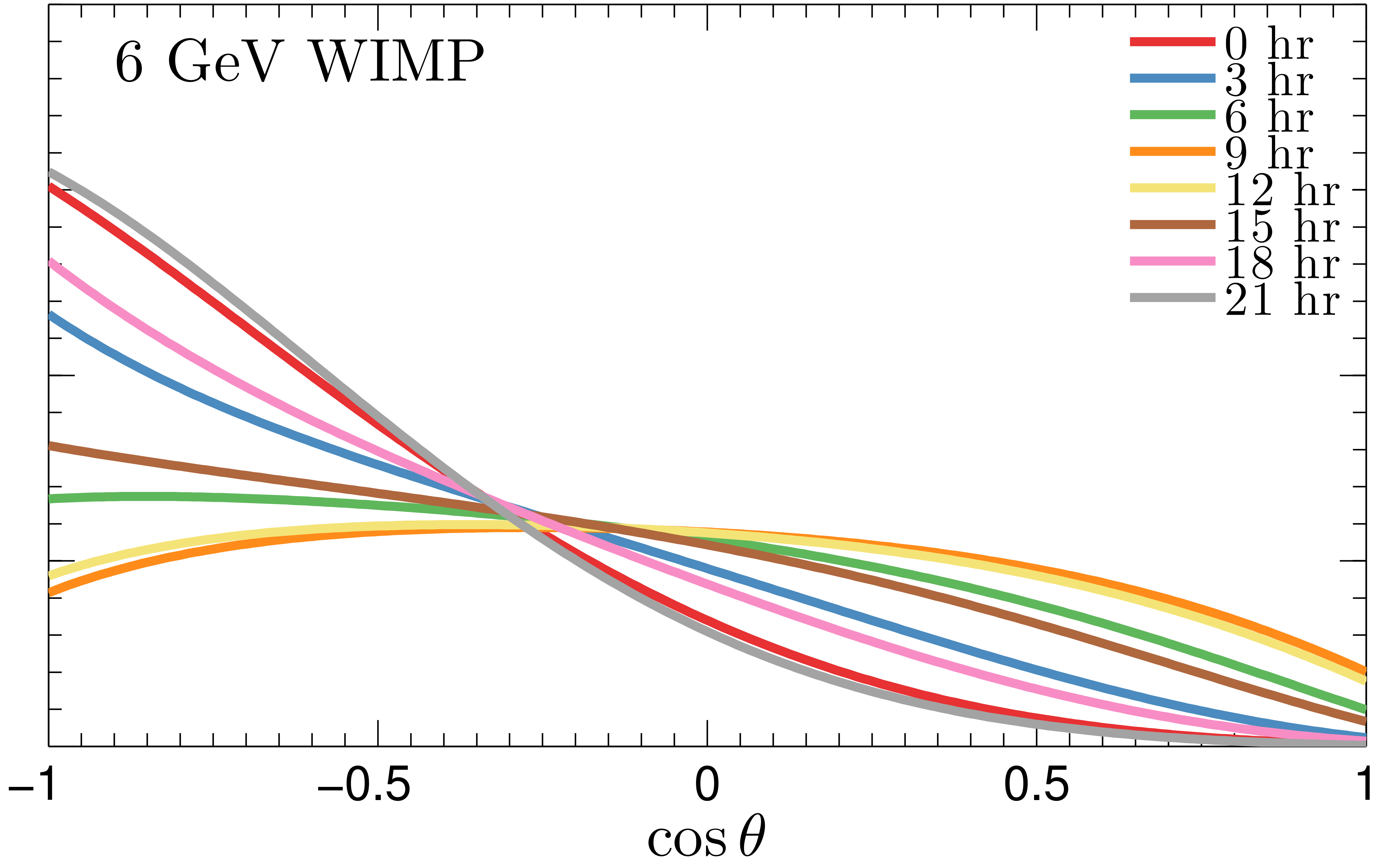
2022 version



Lesson: if you cringe at plots you made more than a year ago, that is a good sign

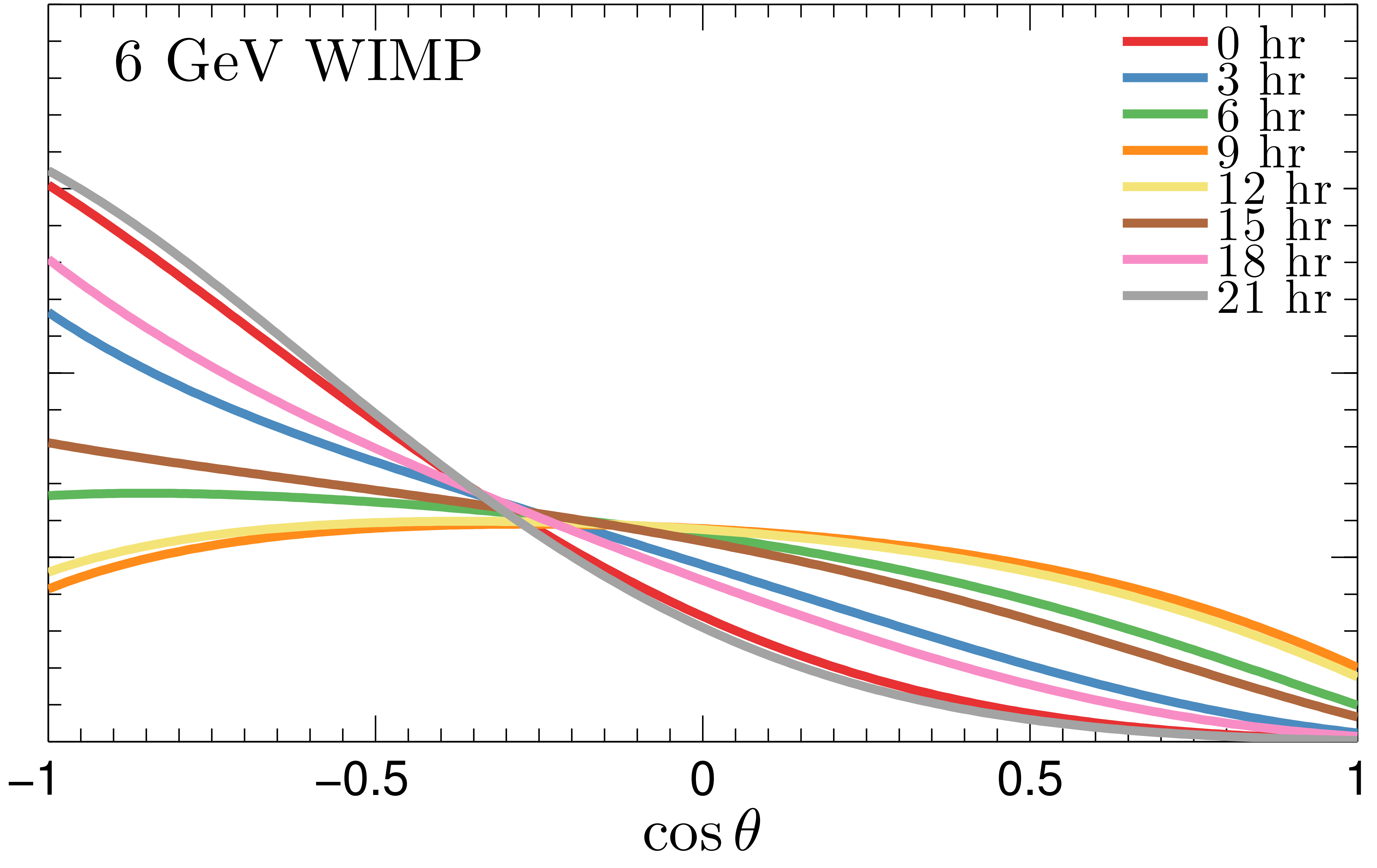
Examples of bad practice: my own plots

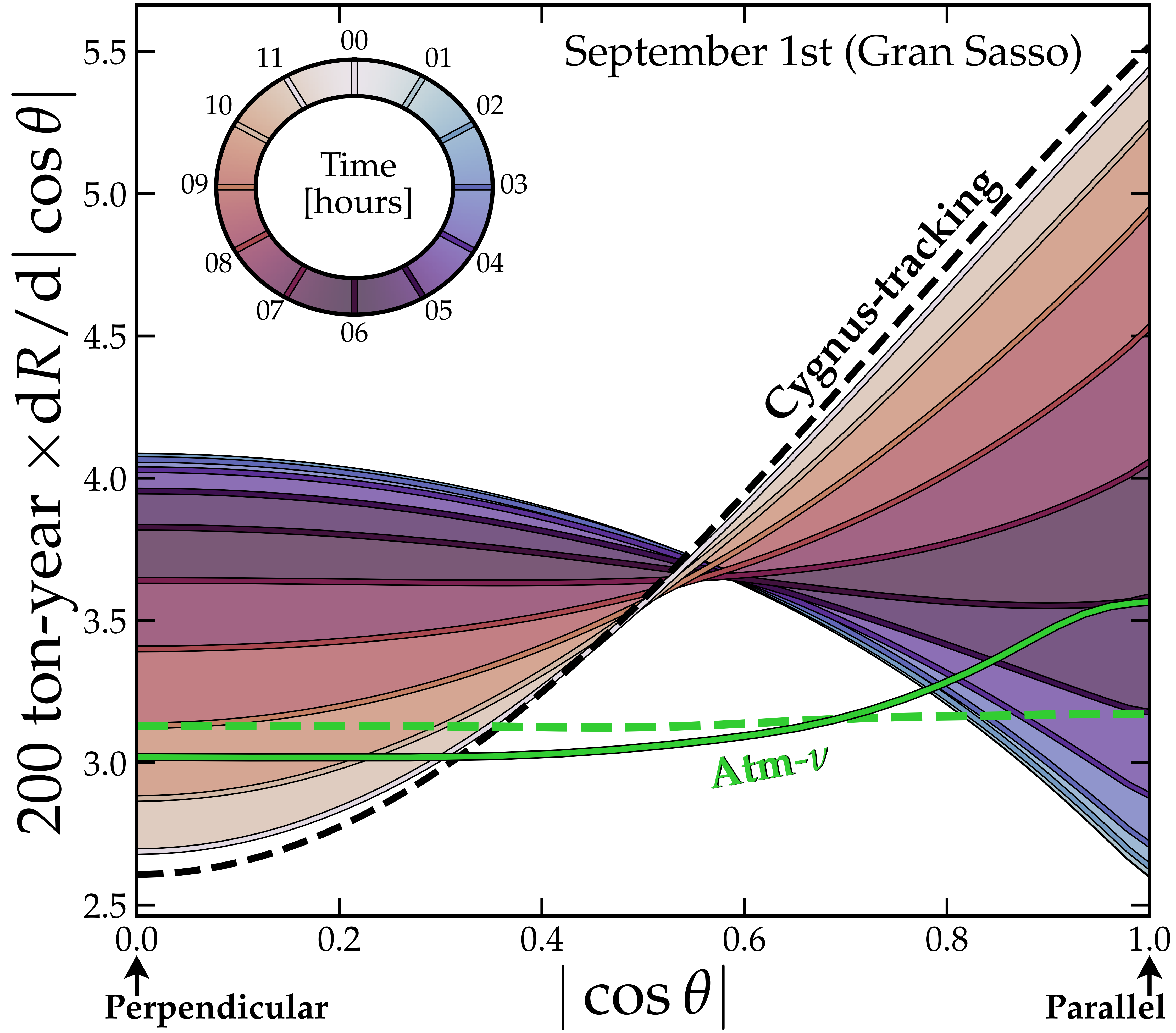
In what ways is this plot bad?



Examples of bad practice: my own plots

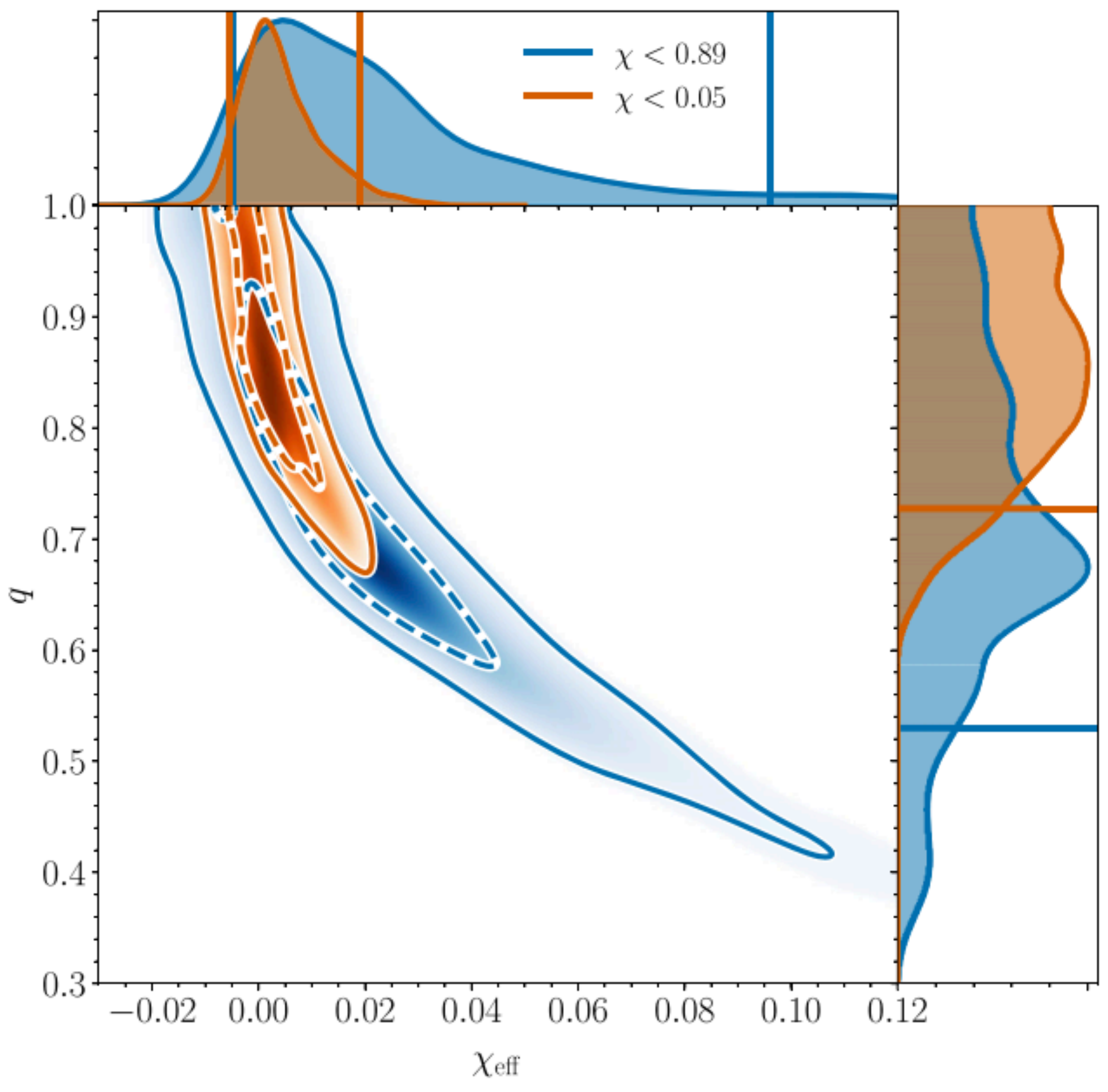
In what ways is this plot bad?

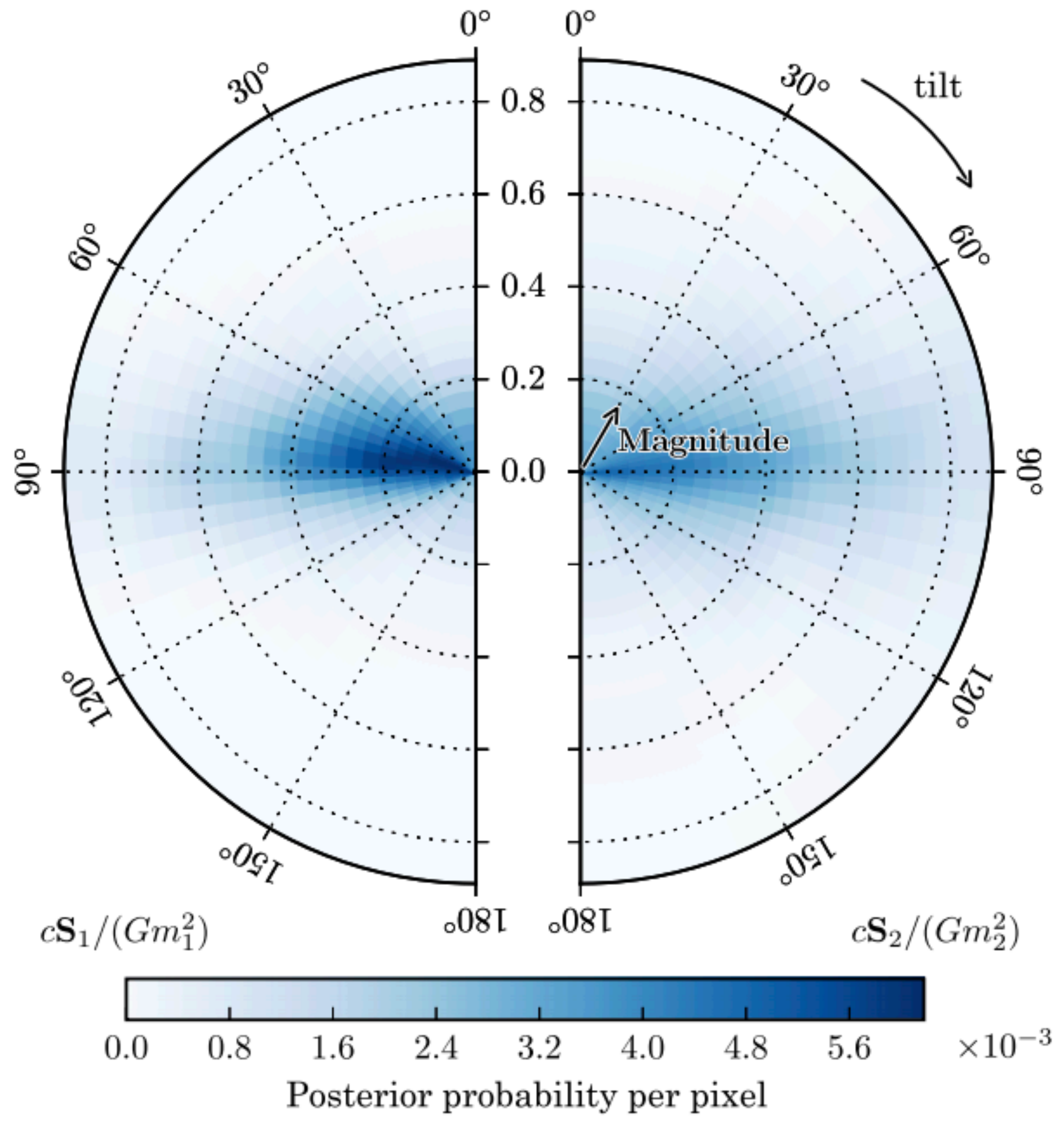


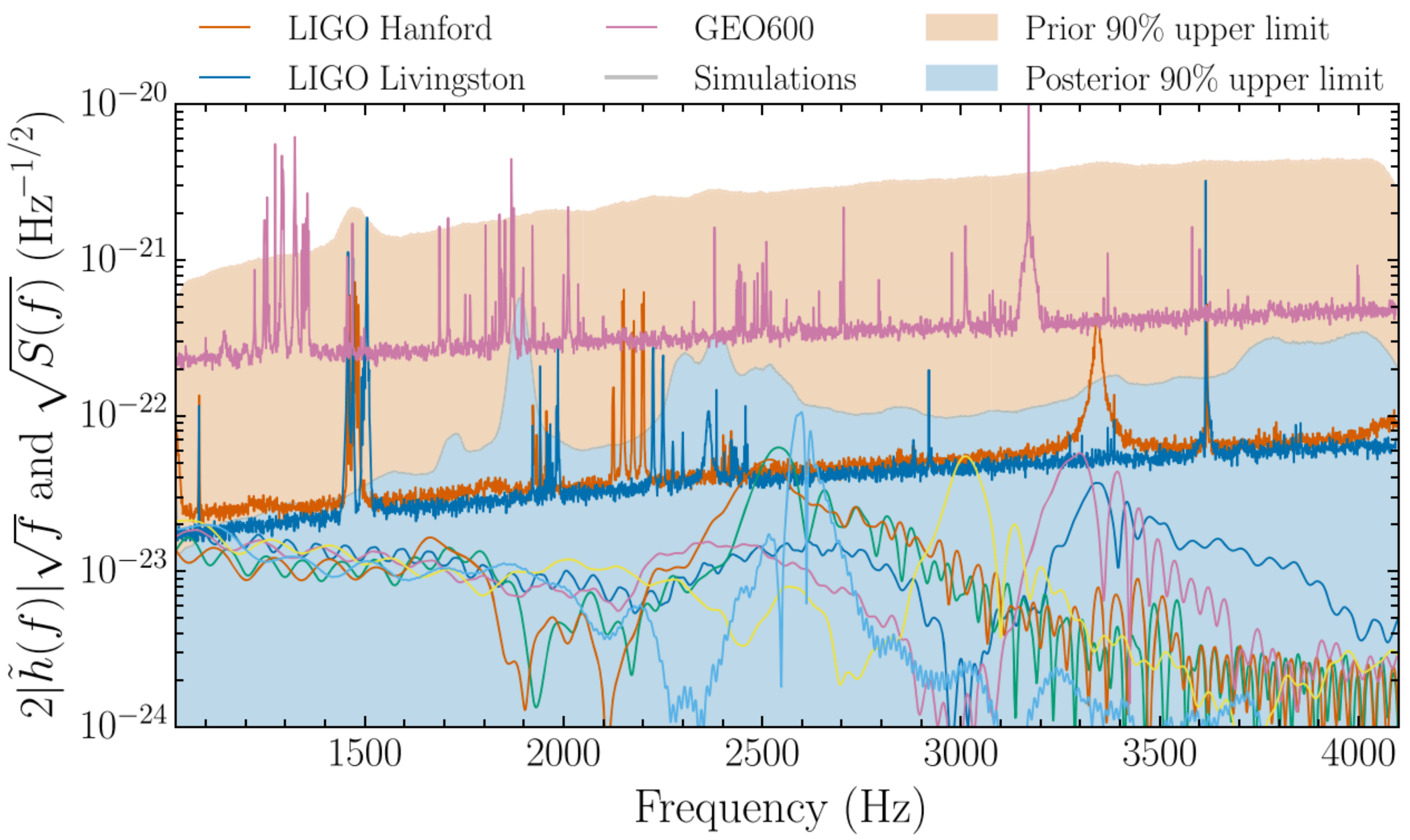


Example of good practice: LIGO

- Always have attractive figures, look in any of their papers, e.g. arxiv.org/abs/1805.11579
- High attention to detail
- Clear and uncluttered, even for complicated plots
- Well-labelled and can be understood at a glance
- Often opt for complementary colours of orange/blue used consistently and meaningfully, i.e. blue for Livingston and orange for Hanford



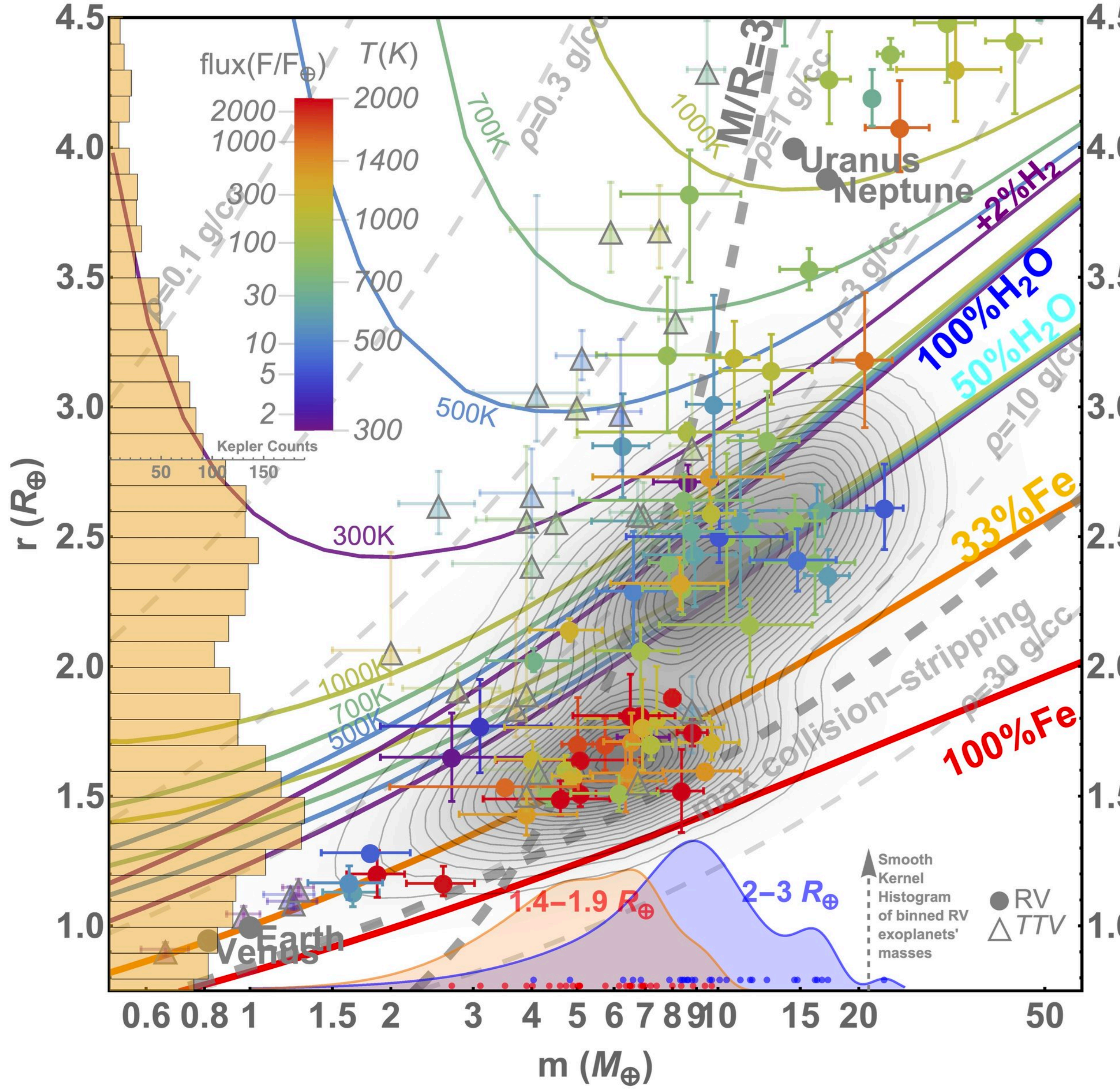


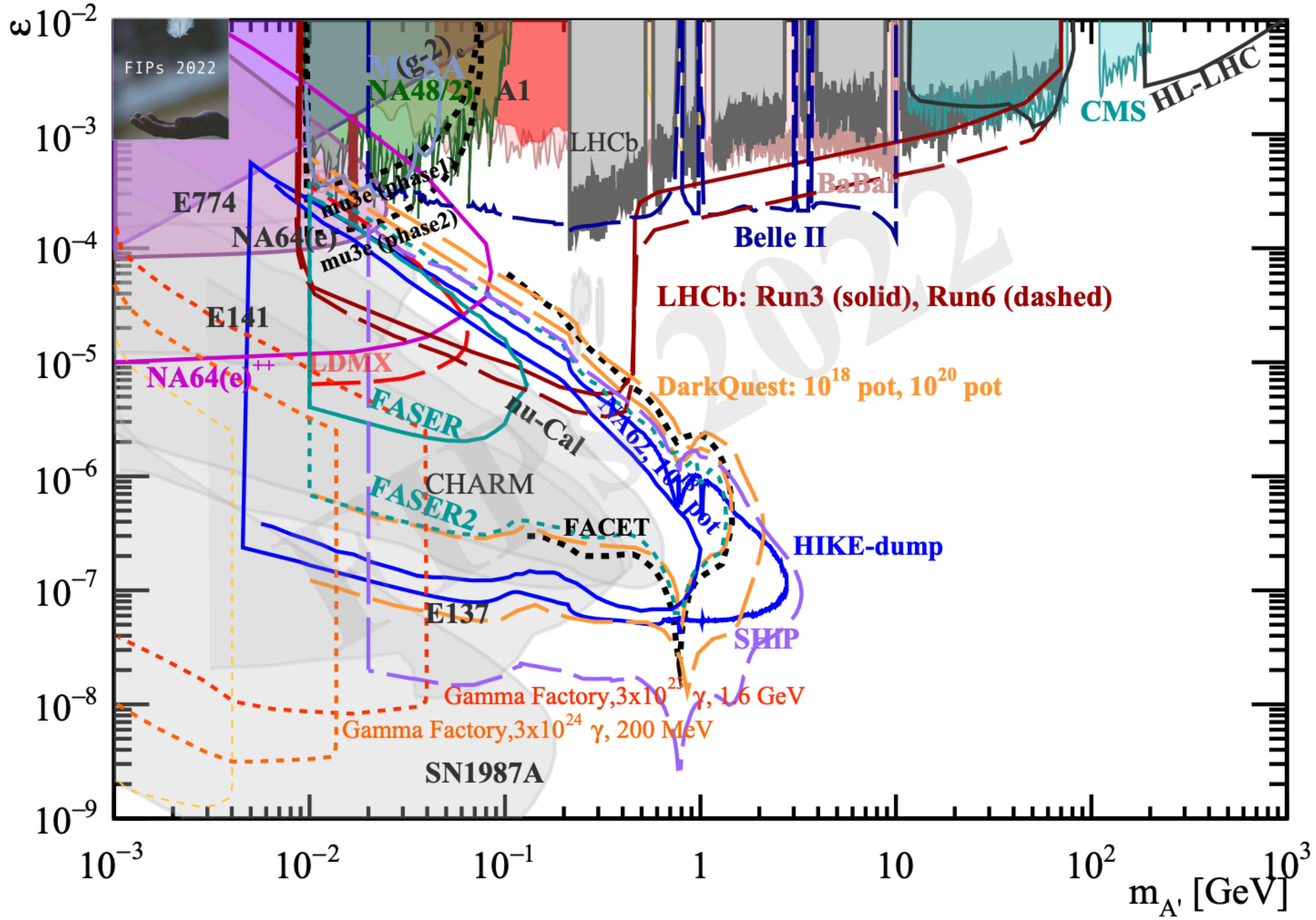


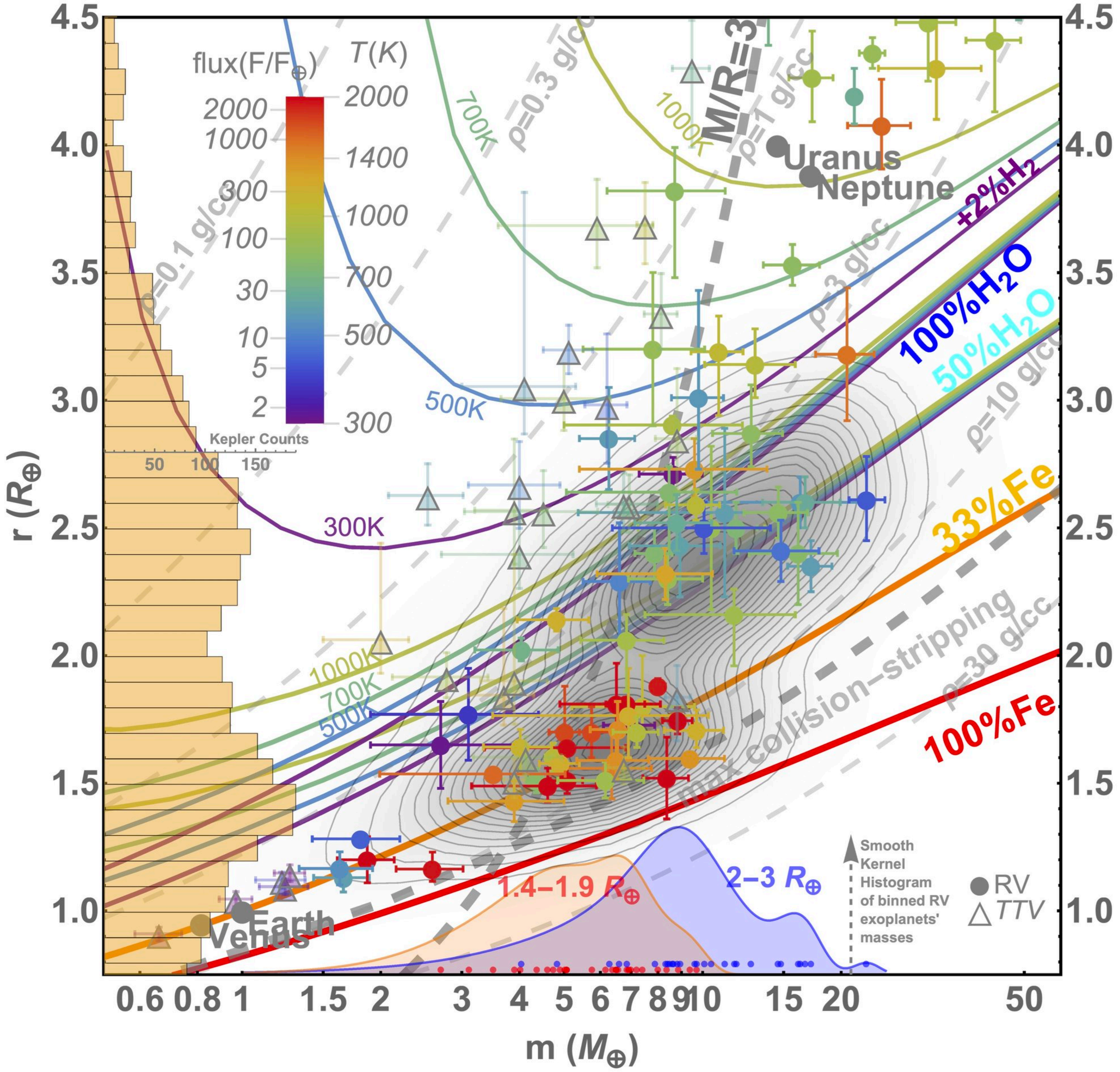
How to break the rules (aka know your audience)

Ultimately style is subjective, all that matters is your audience. You can always break the rules if you know why they are there and you break them intentionally.

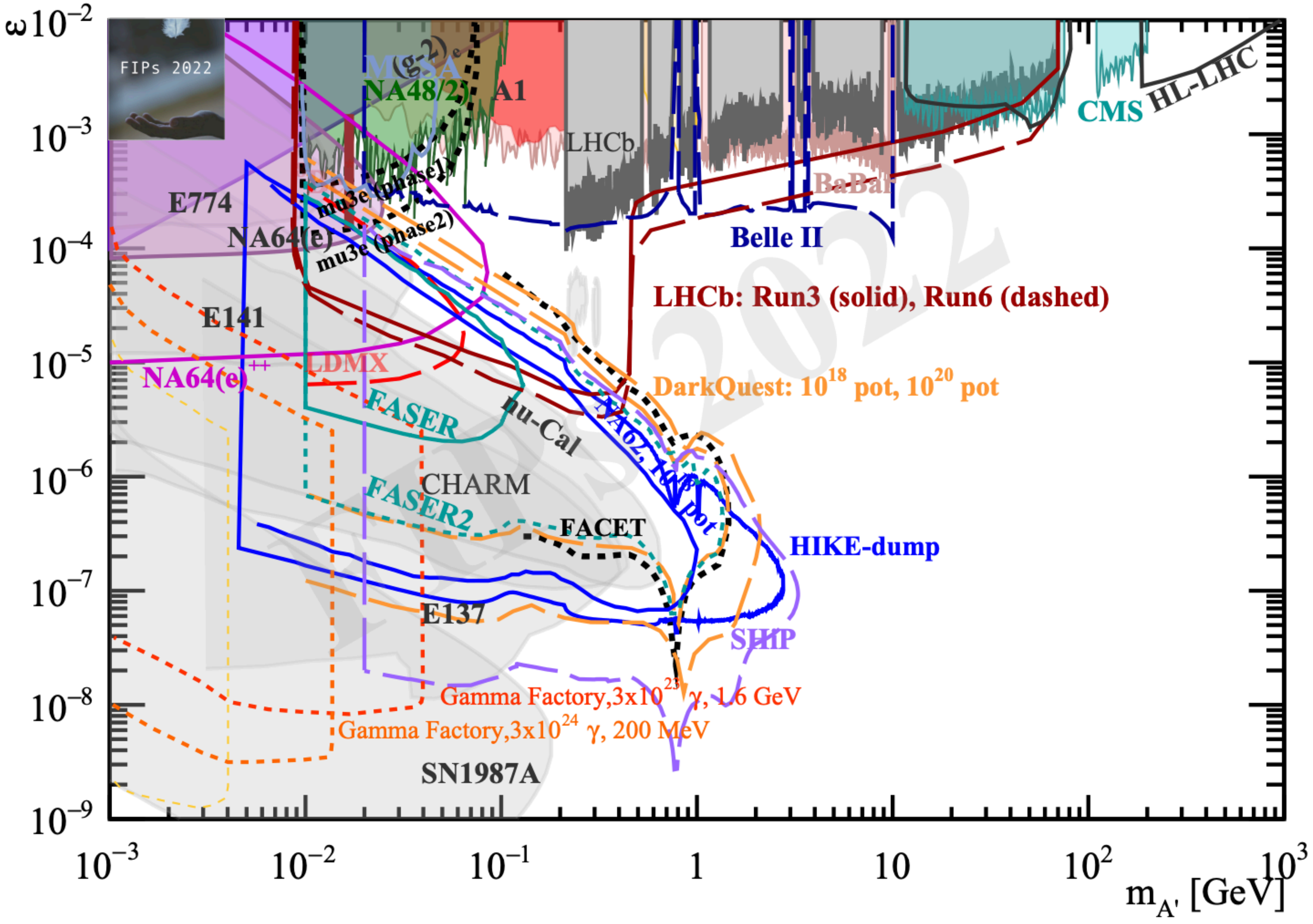
Which of these plots is more effective?







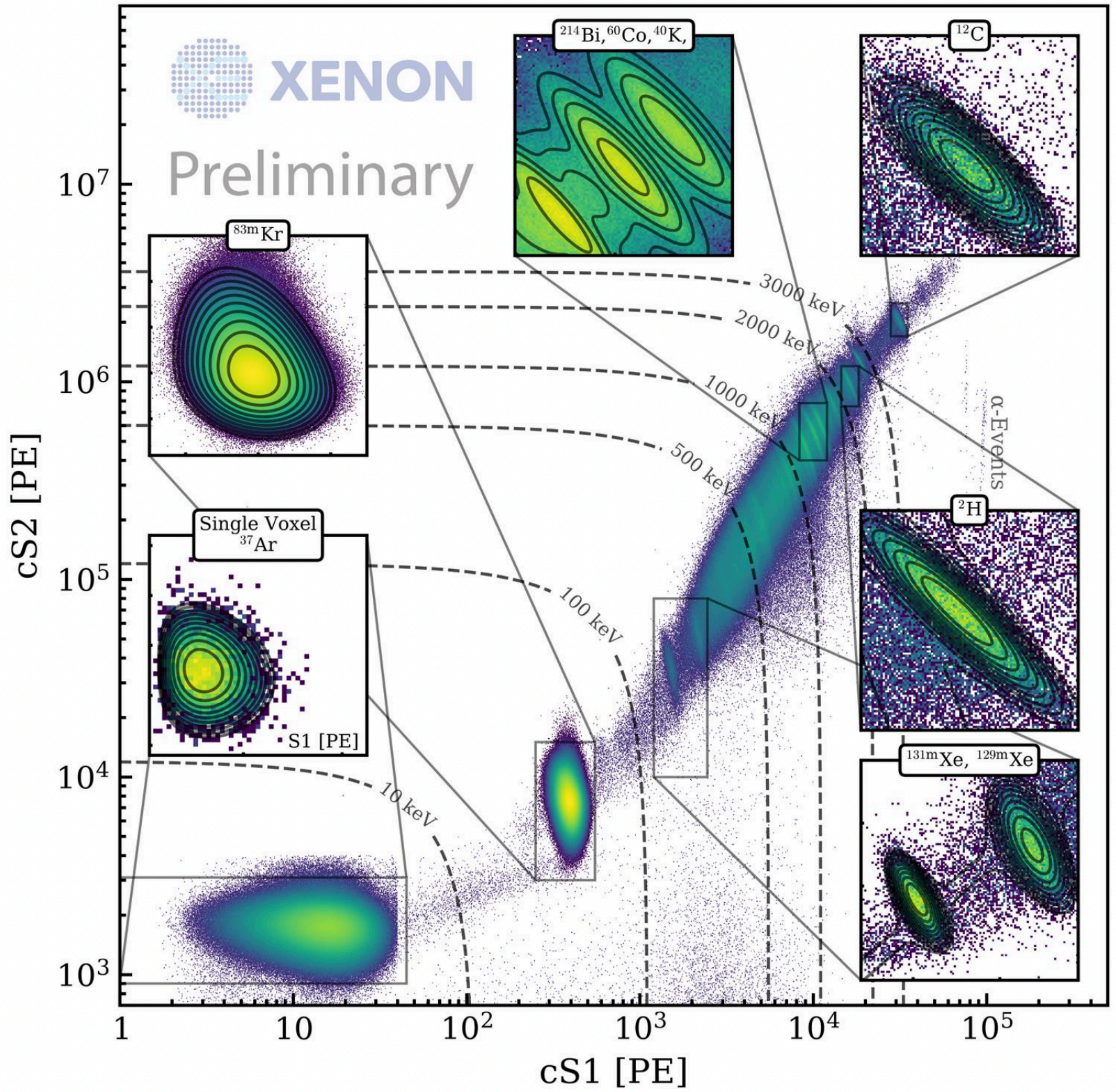
This is a highly detailed plot that conveys a large amount of information (it's basically seven plots stacked on top of each other). It works because it is based on a type of plot that is extremely familiar to a specific audience (exoplanets)



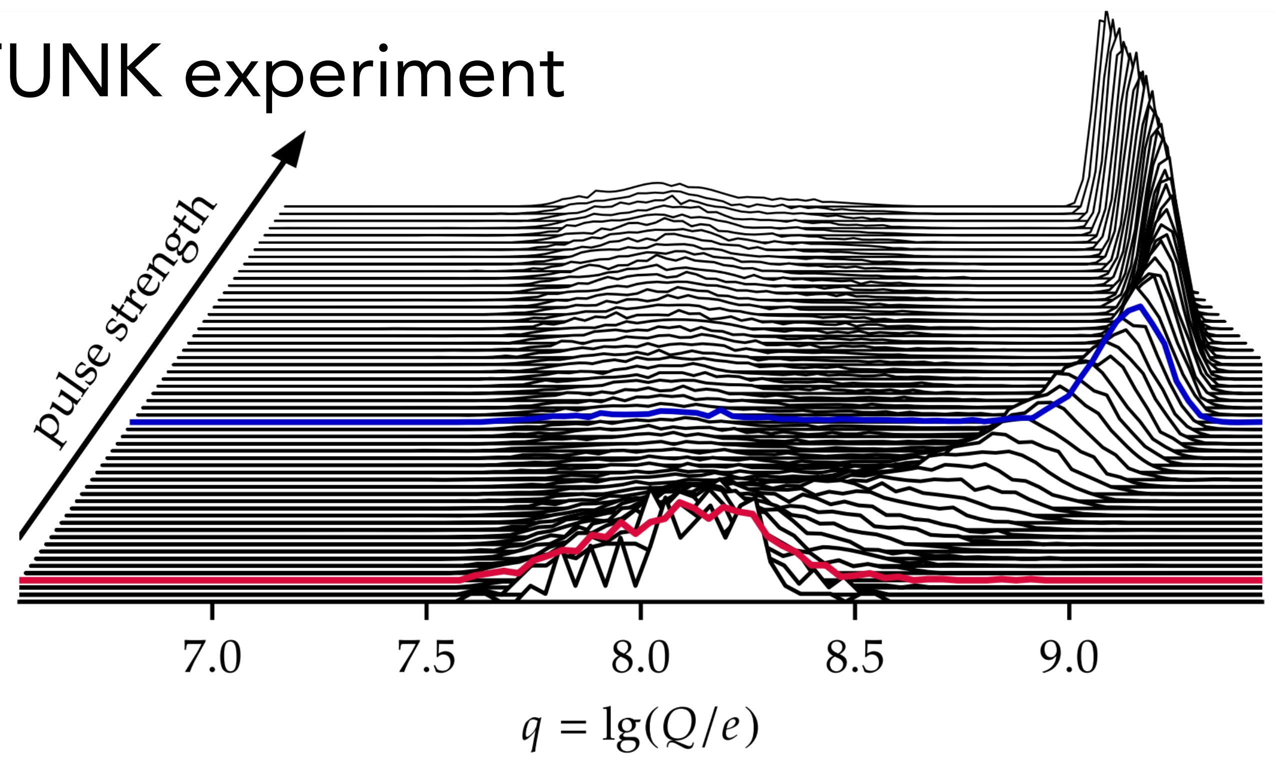
This is a comparatively simple plot (just the parameter space of a dark photon) that is near-unintelligible due to poor design and the inclusion of unnecessary elements.

More plots that I like…

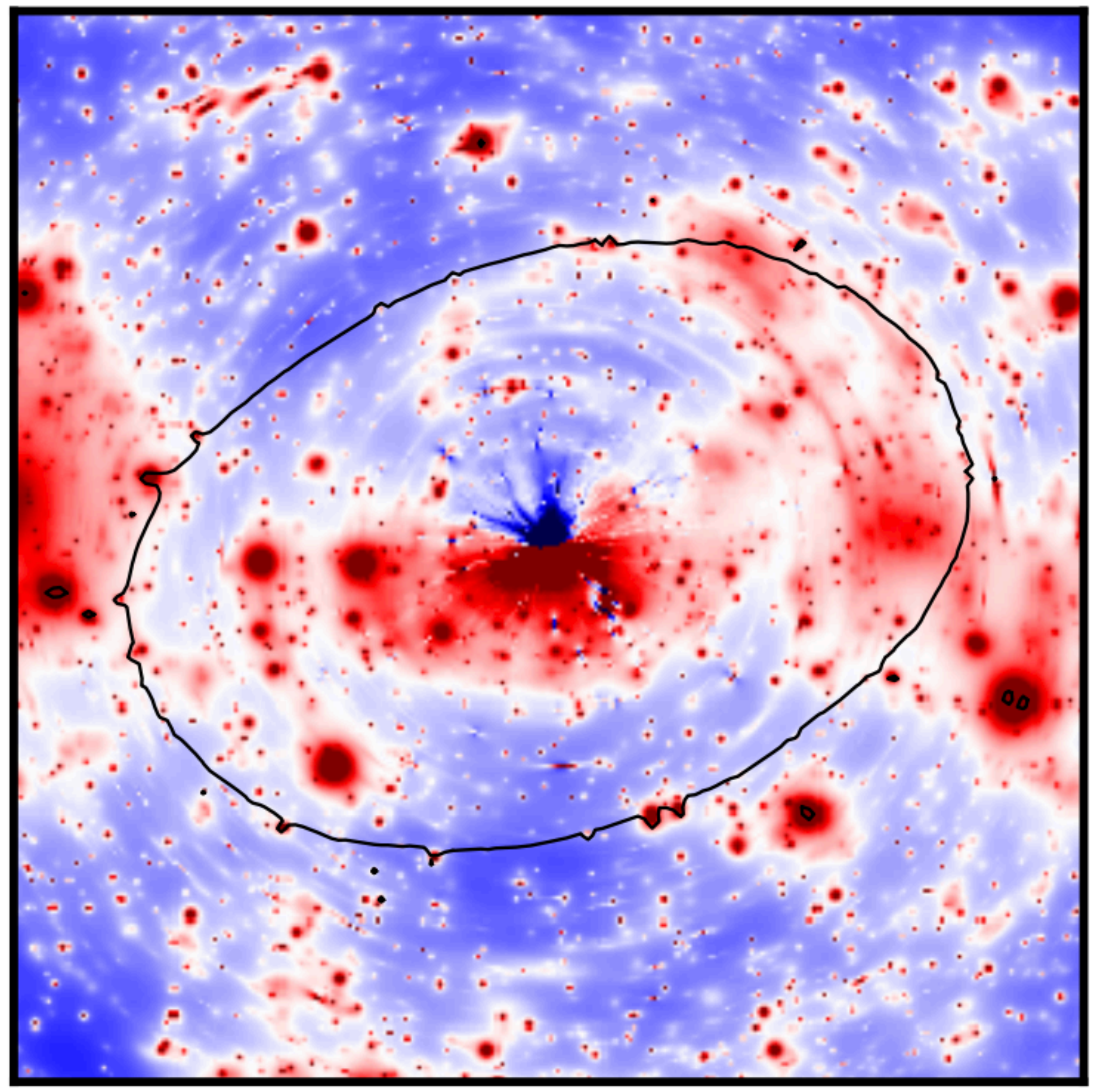
Proof that if someone likes your plot
they are more likely to use it in their talk

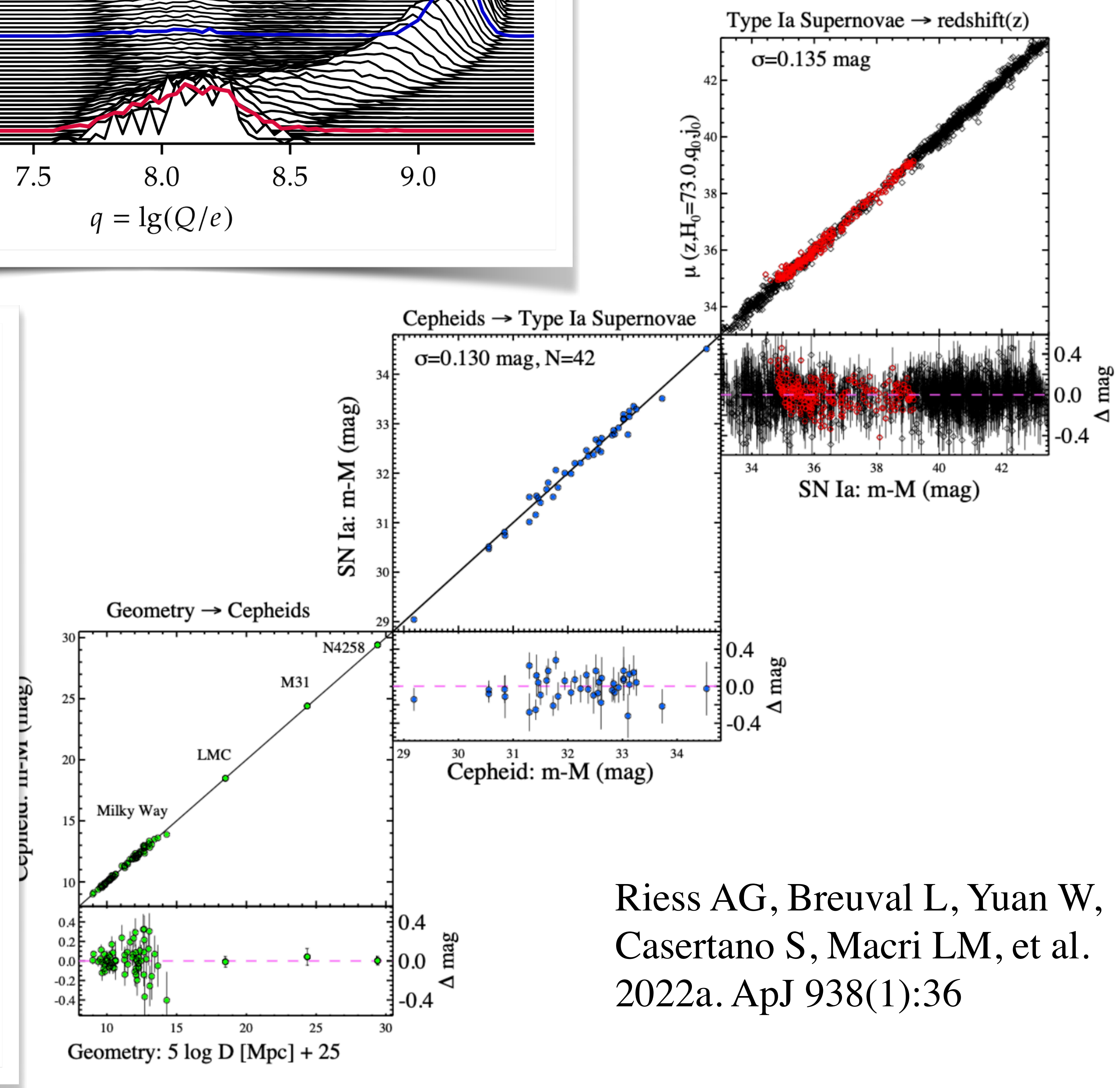


FUNK experiment



Laroche [2206.11269]

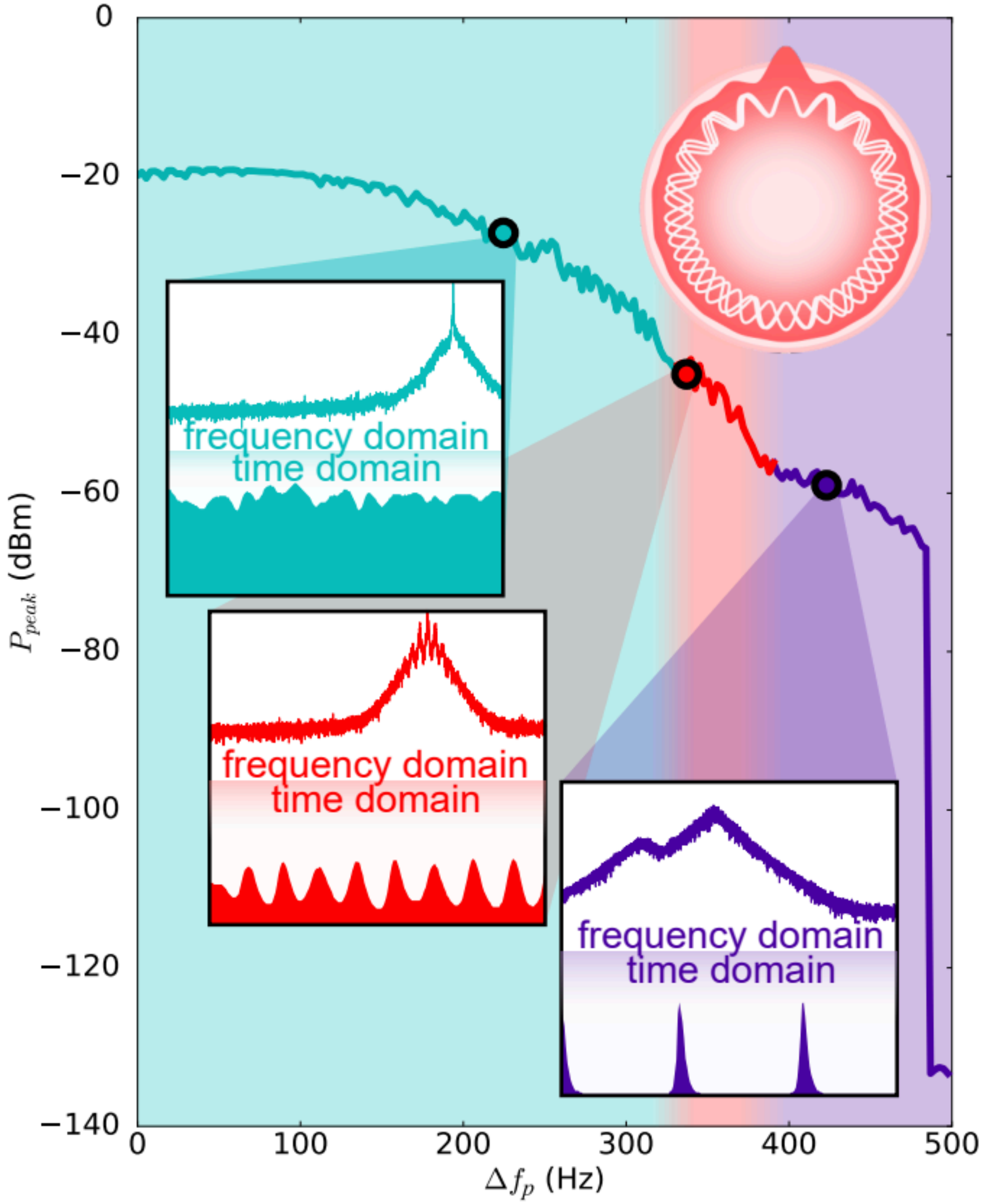




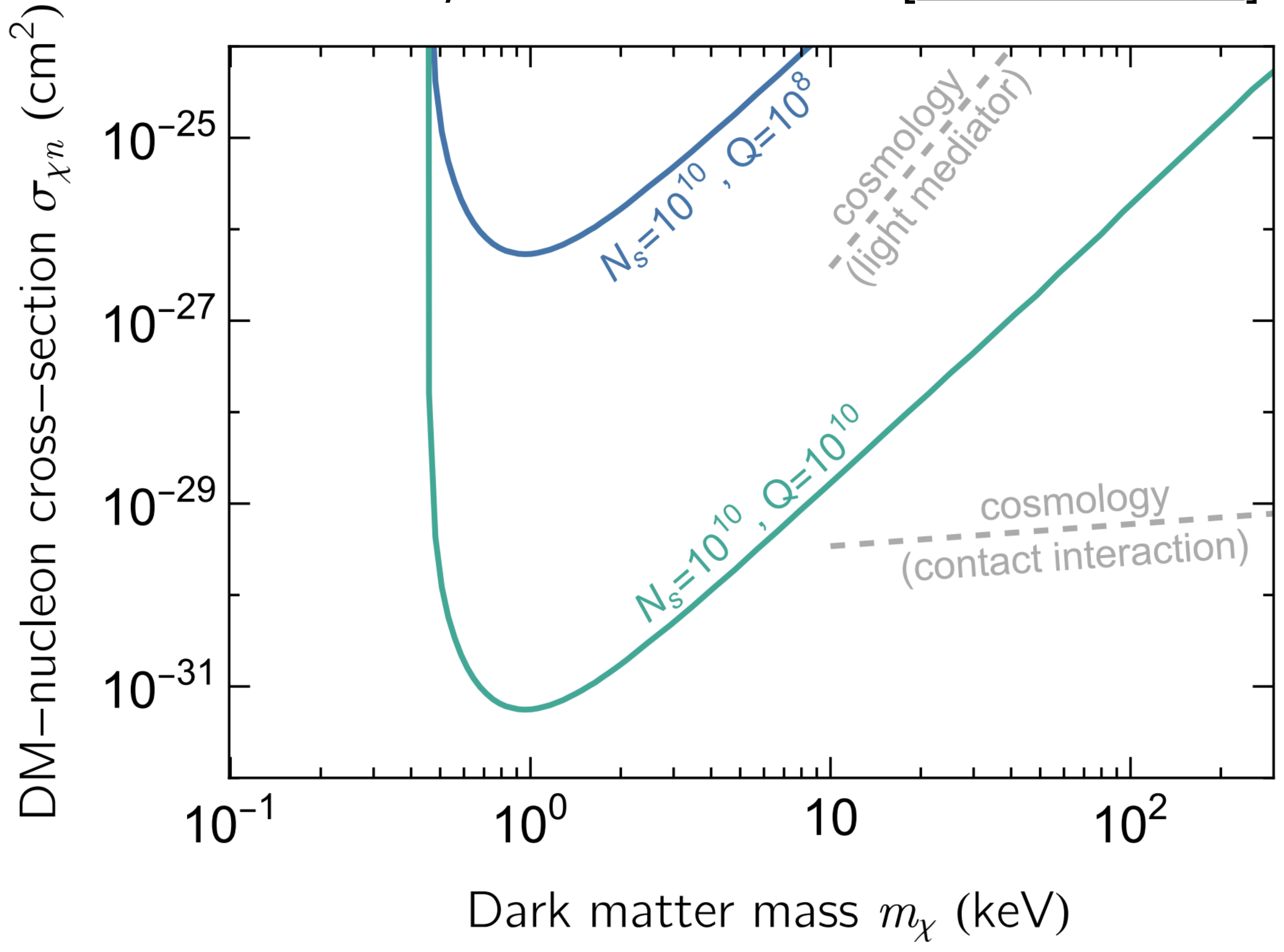
Riess AG, Breuval L, Yuan W, Casertano S, Macri LM, et al. 2022a. ApJ 938(1):36

Examples of good practice: you!

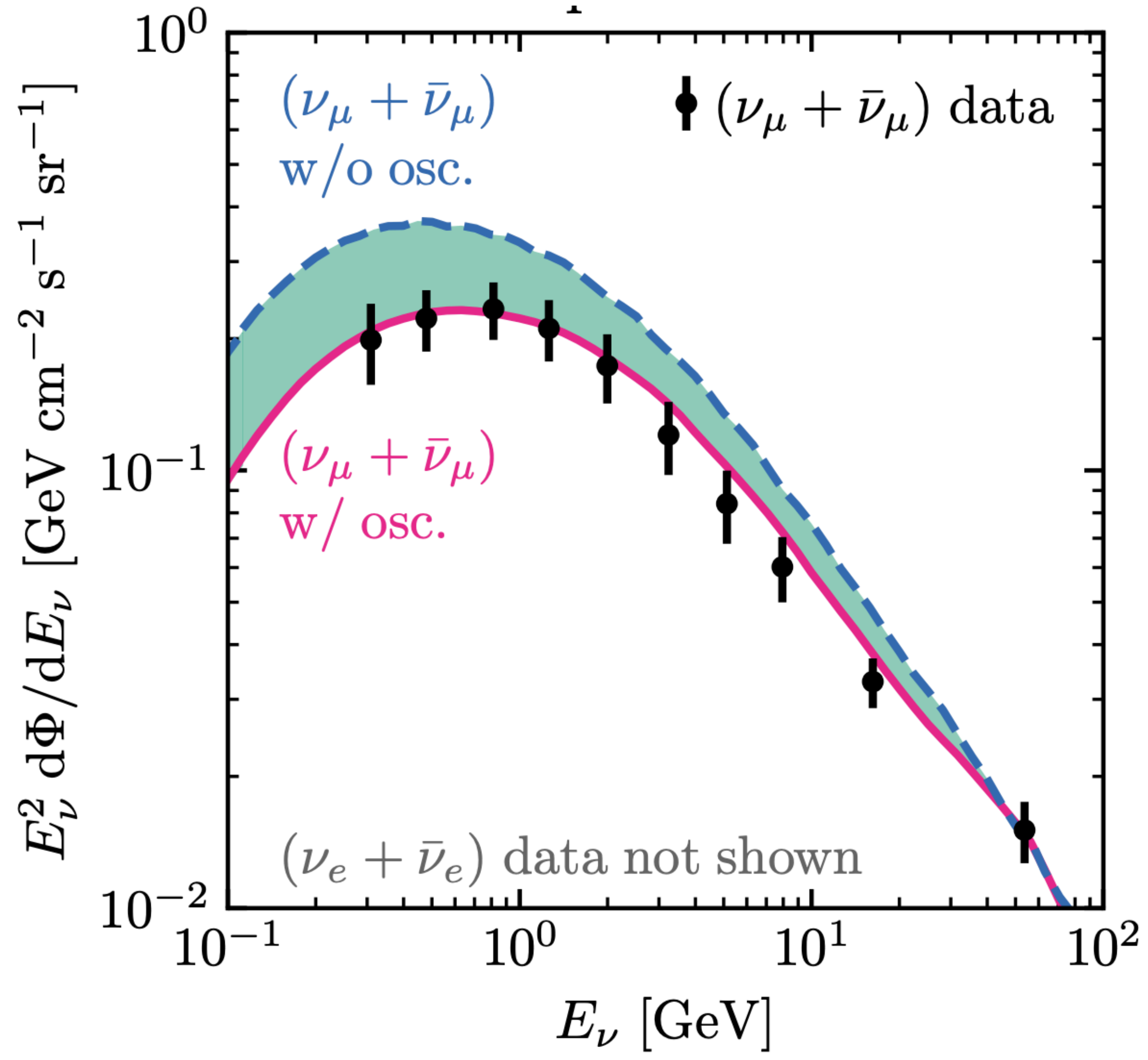
C. A. Thomson et al [2105.11180]

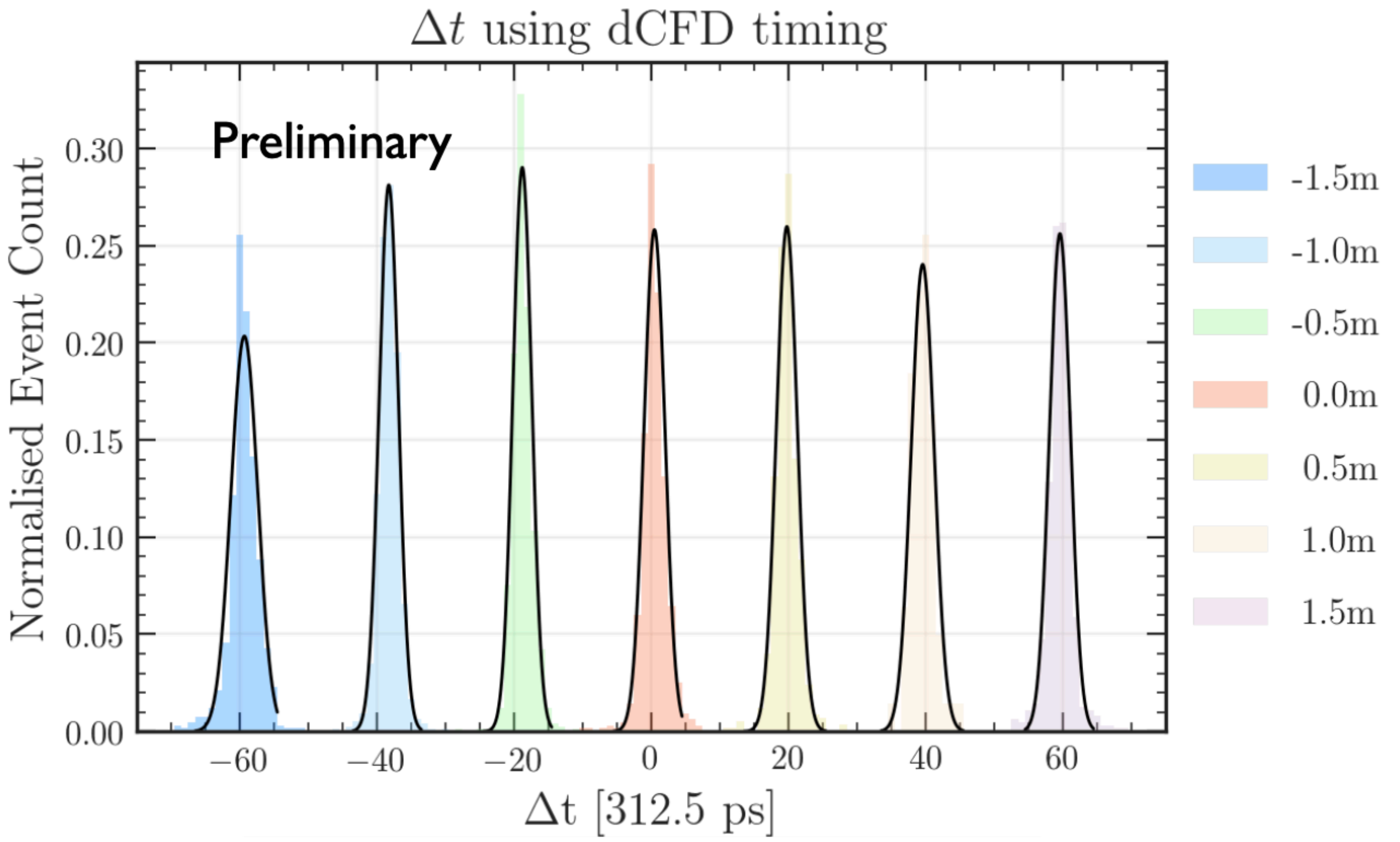


P. Cox, M. Dolan et al. [2306.09726]



S. Meighan-Berger et al. [2311.01667]





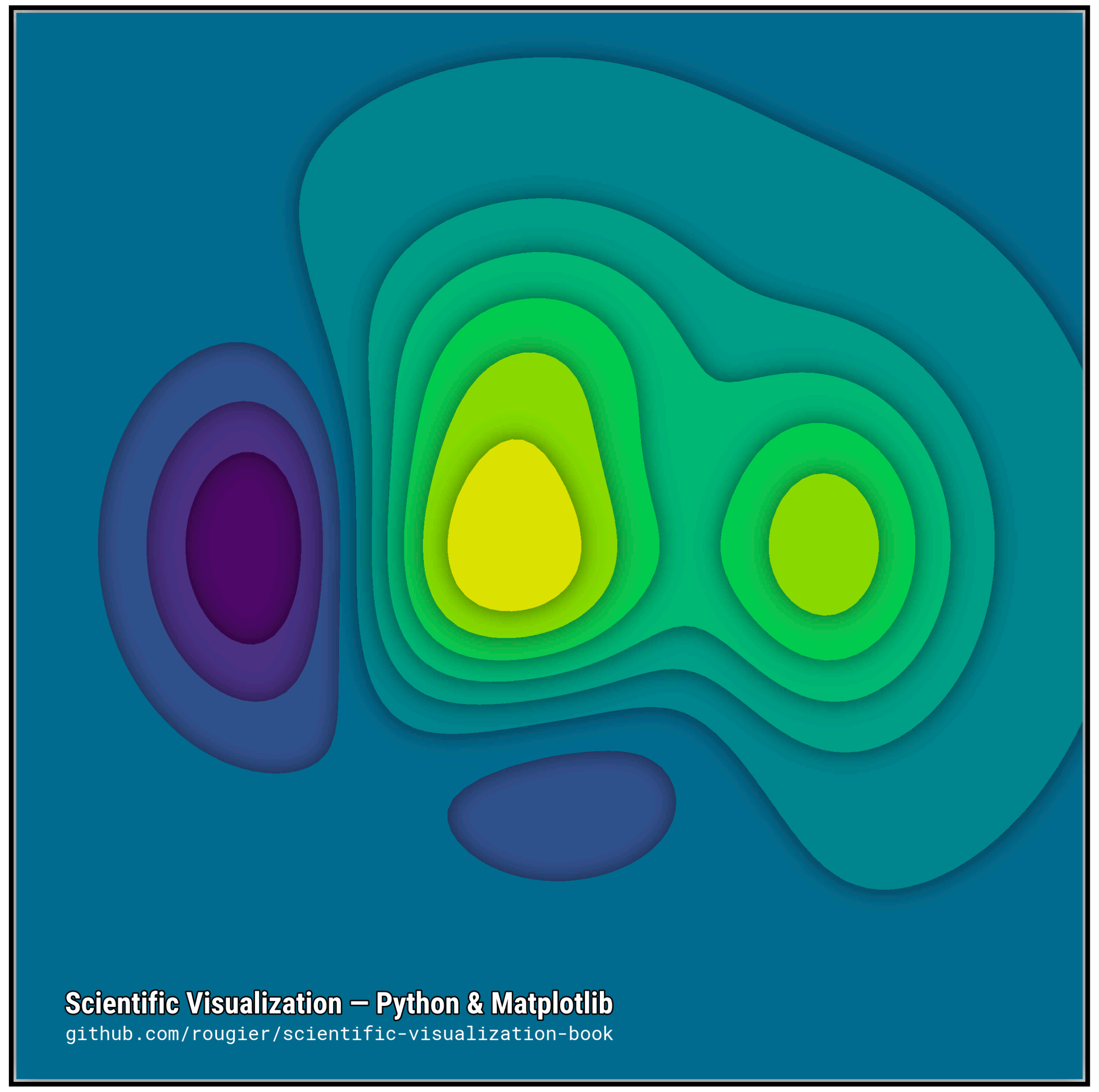
Summary/takeaway messages

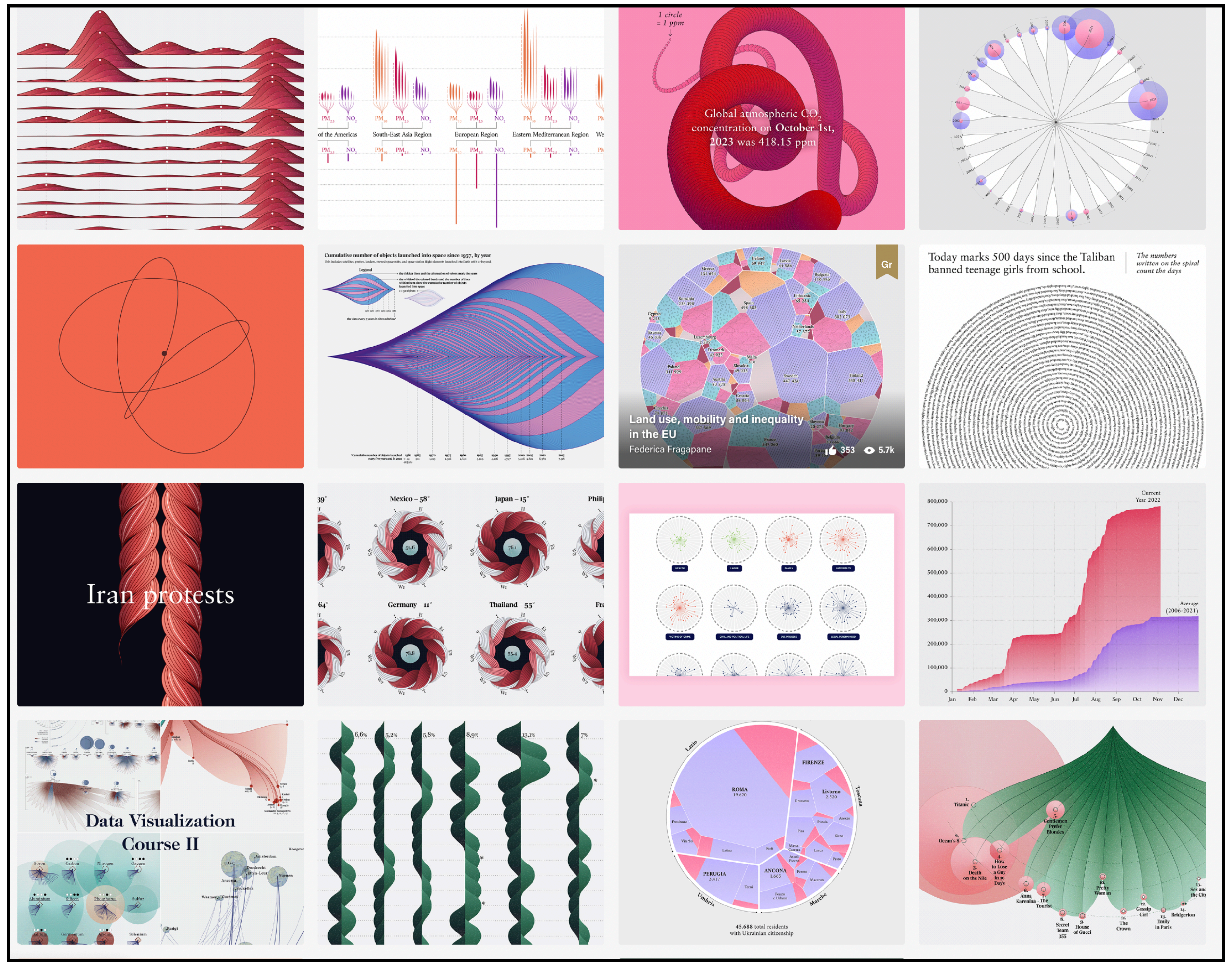
Summary/takeaway messages

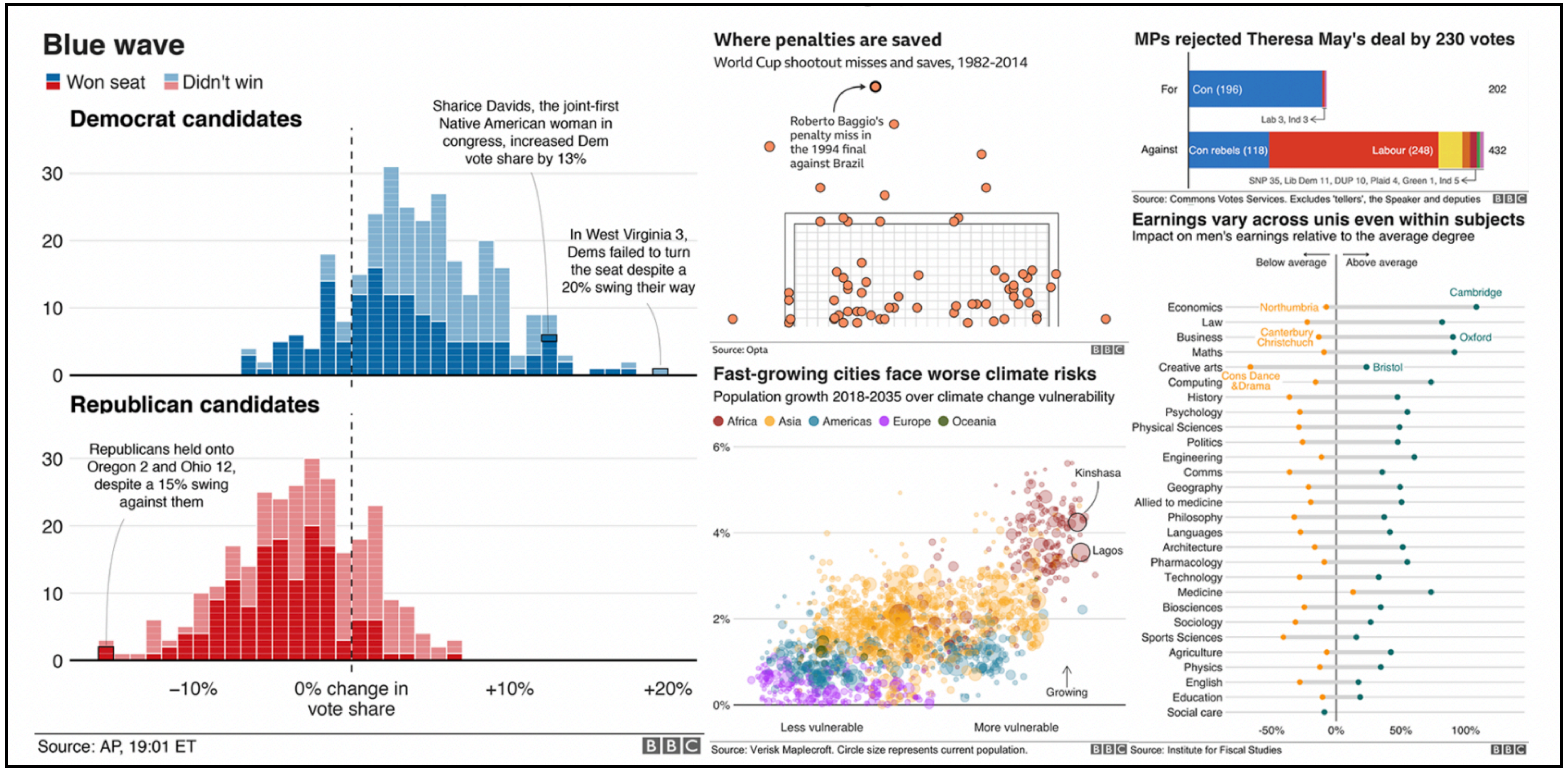
1. Just put effort in

Inspiration from the professionals

Look at what data visualisation specialists do. Their audiences and fields will be completely different, but the underlying principles carry over: you want to capture attention and convey a complex, quantitative message clearly and efficiently.







Matplotlib cheat sheets

https://matplotlib.org/cheatsheets/