DATA
VISUALIZATION

Jean-Philippe Villemin, PhD - jpvillemin@gmail.com
https://github.com/ZheFrench/Qbio

Cancer Bioinformatics and Systems Biology
Institut de Recherche en Cancérologie de Montpellier
Inserm U1194 - Université de Montpellier Faculté de Médecine - ICM Val d'Aurelle
Campus Val d'Aurelle

How the two sessions are organized ? - 1h30/1h

Generalities (Slides ~20 mins)
WhoAmI
Libraries in python
Figures in science
Guidelines
Some plots you might know

Hands-on Seaborn (Jupyter Notebook ~15 mins / topic)
Simple plots
Composite Plots
Heatmap
Multidimensionality

Handle your panel of figures /know the difference between format. (~10mins)
Inkscape / PowerPoint / Google Slides

Who Am I ?

VILLEMIN JEAN-PHILIPPE

BIOINFORMATICS SCIENTIST

Post-Doctoral Fellow

Cancer Bioinformatics and Systems Biology   2021 – Current
Institute of Cancer Research, INSERM, Montpellier, France.
Tracking preexisting anti-tumor CD8 + T cells to predict clinical response to the blocking of PD-1/PD-L1 axis.
Study of the mechanisms involved in dormancy and relapse in Non-Small Lung Cancer Cells (NSLCC) treated by TK-Inhibitor using single cells approaches.
Supervised by Jacques Colinge

PhD Student

Chromatin and Splicing / Artificial Intelligence & Gene Expression group,   2017 – 2021
Institute of Human Genetics, CNRS, Montpellier, France.
Investigation of a splicing signature in Breast Cancer during EMT.
Supervised by Reini Luco / William Ritchie

Engineer

Chromatin and Splicing group,   2016 – 2017
Institute of Human Genetics, CNRS, Montpellier, France.
Chip-Seq analysis in a time-course model of EMT.

Molecular Genetics of Rare Diseases group,   2015 – 2016
Clinical Research Institute, INSERM, Montpelier, France. *Supervised by Michel Koenig*
Variant calling pipeline set up for targeted Exome & Copy Number Detection.

Bioinformatics Platform,   2013 – 2015
Synergie Lyon Cancer, LEON BERARD CENTER, Lyon, France. *Supervised by Alain Viari*
Database & web application development for a prospective personalized medicine project in oncology.

Splicing and Tumor Progression Team,   2010 – 2013
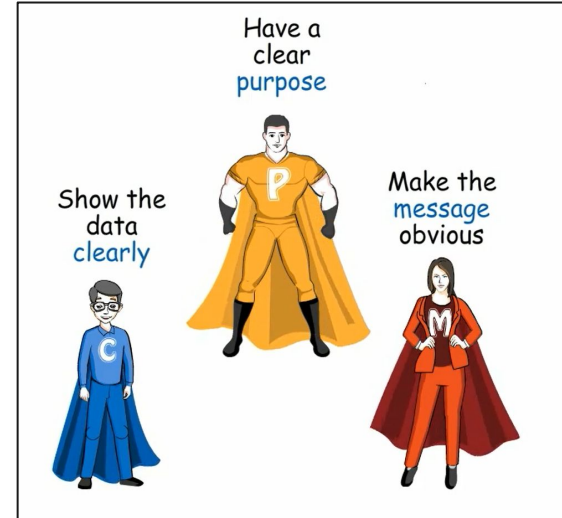**Cancer Research Center of Lyon, INSERM, Lyon, France.** *Supervised by Didier Auboeuf*
Conception of bioinformatic tools (database, webapps, pipelines) for the understanding of splicing mechanisms based on exon arrays.
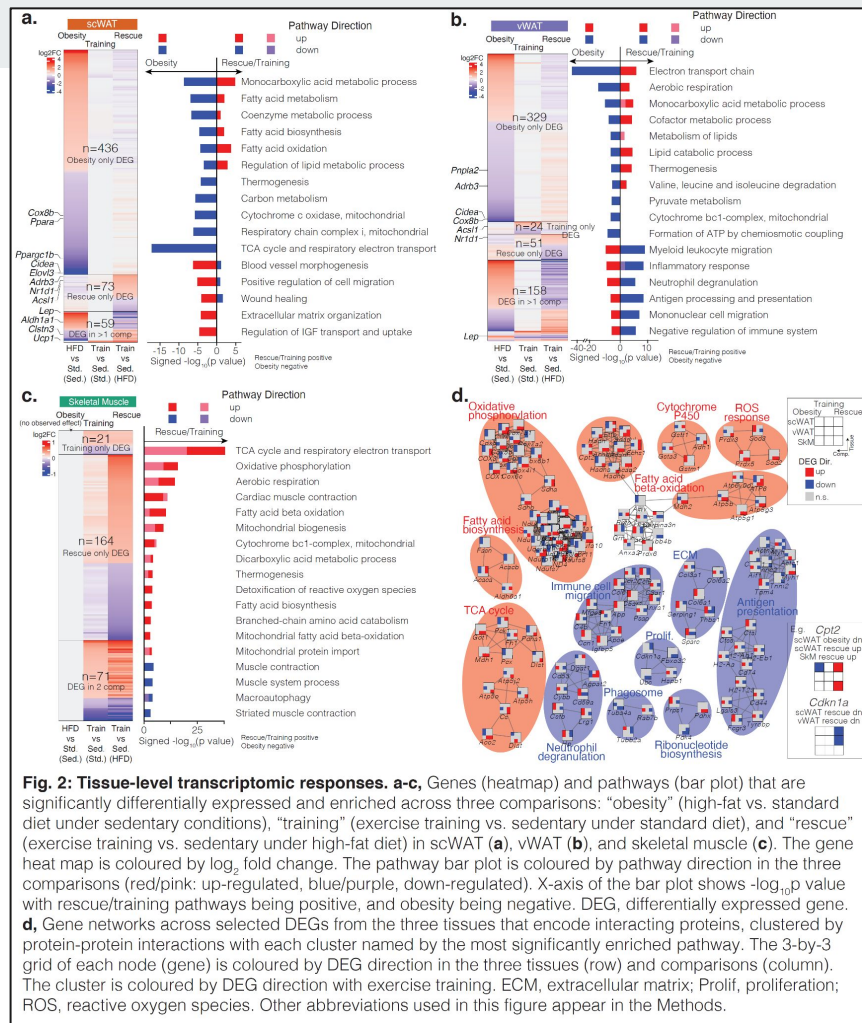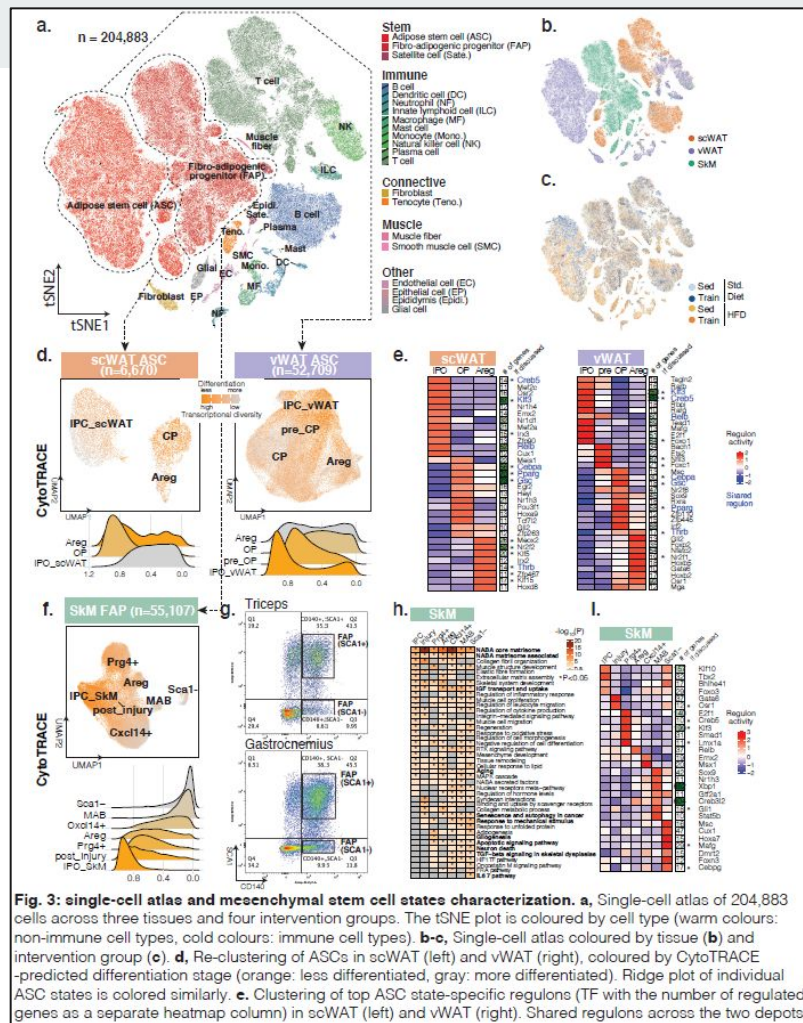
Topic of the day : Data Viz in Python

- **Matplotlib** is a data visualization library and 2-D plotting library of Python It was initially released in 2003 and it is built on NumPy arrays. It is thought to be the most popular and widely-used plotting library in the Python community.

- **Seaborn** is a library built on top of matplotlib and integrates closely with pandas data structures .

- **Bokeh** is mainly famous for its interactive charts visualization. Bokeh renders its plots using HTML and JavaScript.It's an interactive visualization library for modern web browsers.

- **Plotly** is another famous interactive library for charts and maps for Python, R, julia. It's is a more platform agnostic framework.
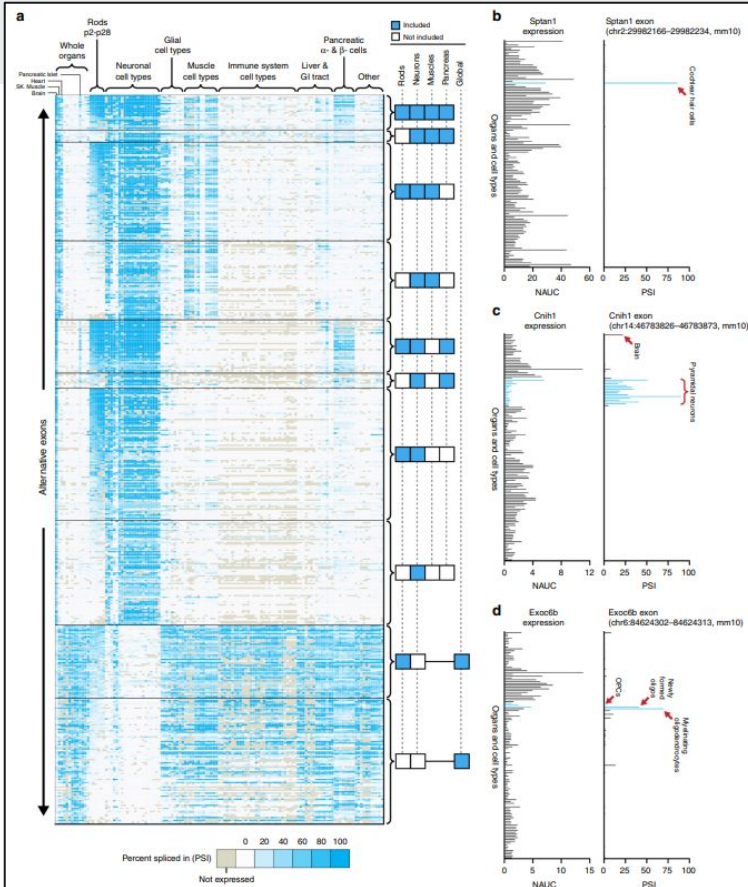

Have a clear **purpose**

Show the data **clearly**

Make the **message** obvious

https://graphicsprinciples.github.io/

Pros and Cons

Fig. 3: **single-cell atlas and mecenchymal stem cell states characterization. a,** Single-cell atlas of 204,883 cells across three tissues and four intervention groups. The tSNE plot is coloured by cell type (warm colours: non-immune cell types, cold colours: immune cell types). **b-c,** Single-cell atlas coloured by tissue (**b**) and intervention group (**c**). **d,** Re-clustering of ASCs in scWAT (left) and vWAT (right), coloured by CytoTRACE -predicted differentiation stage (orange: less differentiated, gray: more differentiated). Ridge plot of individual ASC states is colored similarly. **e.** Clustering of top ASC state-specific regulons (TF with the number of regulated genes as a separate heatmap column) in scWAT (left) and vWAT (right). Shared regulons across the two depots

Fig. 2: **Tissue-level transcriptomic responses. a-c,** Genes (heatmap) and pathways (bar plot) that are significantly differentially expressed and enriched across three comparisons: "obesity" (high-fat vs. standard diet under sedentary conditions), "training" (exercise training vs. sedentary under standard diet), and "rescue" (exercise training vs. sedentary under high-fat diet) in scWAT (**a**), vWAT (**b**), and skeletal muscle (**c**). The gene heat map is coloured by $\log_2$ fold change. The pathway bar plot is coloured by pathway direction in the three comparisons (red/pink: up-regulated, blue/purple, down-regulated). X-axis of the bar plot shows -$\log_{10}$p value with rescue/training pathways being positive, and obesity being negative. DEG, differentially expressed gene. **d,** Gene networks across selected DEGs from the three tissues that encode interacting proteins, clustered by protein-protein interactions with each cluster named by the most significantly enriched pathway. The 3-by-3 grid of each node (gene) is coloured by DEG direction in the three tissues (row) and comparisons (column). The cluster is coloured by DEG direction with exercise training. ECM, extracellular matrix; Prolif, proliferation; ROS, reactive oxygen species. Other abbreviations used in this figure appear in the Methods.

Fig. 1 Alternative exons enriched in the nervous system (MESA compilation). a Mouse RNA-Seq data sets were manually curated from the SRA, covering a broad range of cell types and organs. Cell type data sets were generated from various independent labs using FACS or affinity isolation. To test our algorithm, we identified alternative exons that were differentially spliced between neuronal cell types and other cell types in the body and found that exons could be generally clustered by their inclusion or exclusion in rods, neurons, muscles, pancreas, or global non-neuronal (right columns). Each row is an individual exon, and exon utilization is measured by a percent spliced in (PSI) ratio as indicated by gradient legend (bottom). The overlap between neuronal exons and muscle cell types agrees with previous observations from our work[51] and others[58–60], suggesting that these exons are at least partially activated by Ptbp1 downregulation. There is only partial overlap between rod exons and neuron-enriched exons, which is not unexpected since rods do not express many neuron-enriched splicing factors (Supplementary Fig. 3). **b–d** Our splicing analysis method reliably identifies alternative exons that are unique to specific cell types. For example, an exon in Sptan1 is specifically enriched in cochlear hair cells, despite ubiquitous expression across all organs and cell types (**b**). Likewise, an exon in Cnih1 is specifically enriched in pyramidal neurons (**c**) and an exon in Exoc6b is selectively enriched in myelinating oligodendrocytes (**d**).

https://doi.org/10.1016/j.cell.2022.01.003
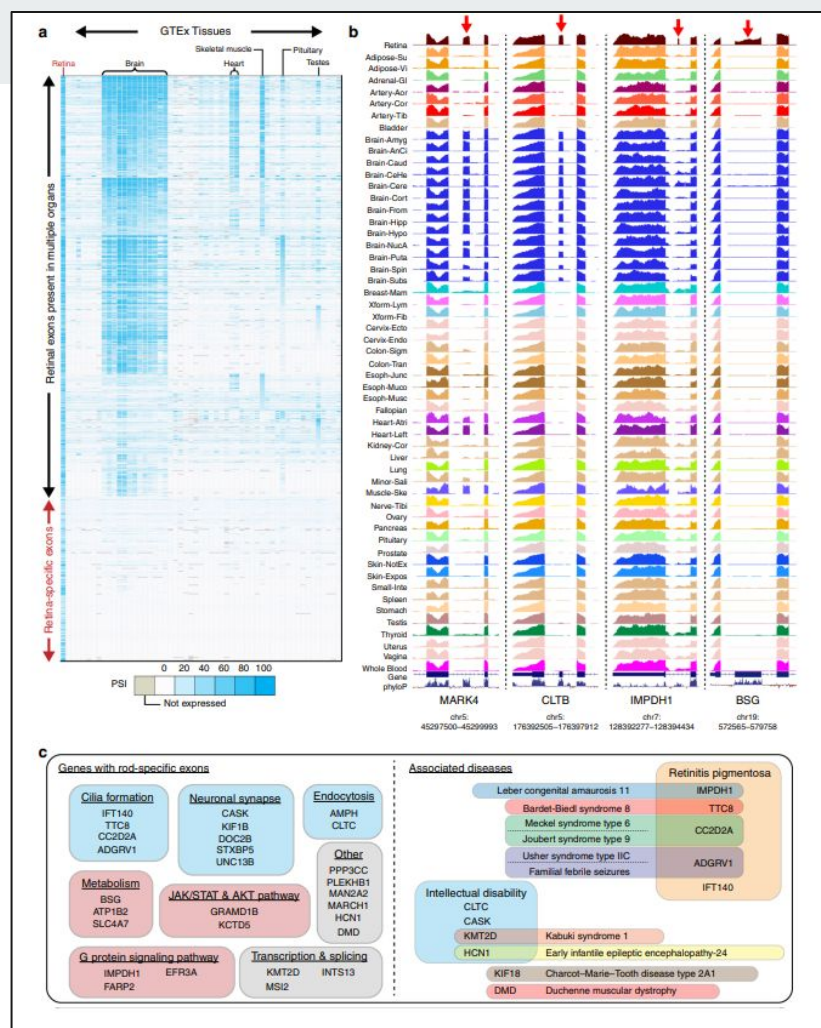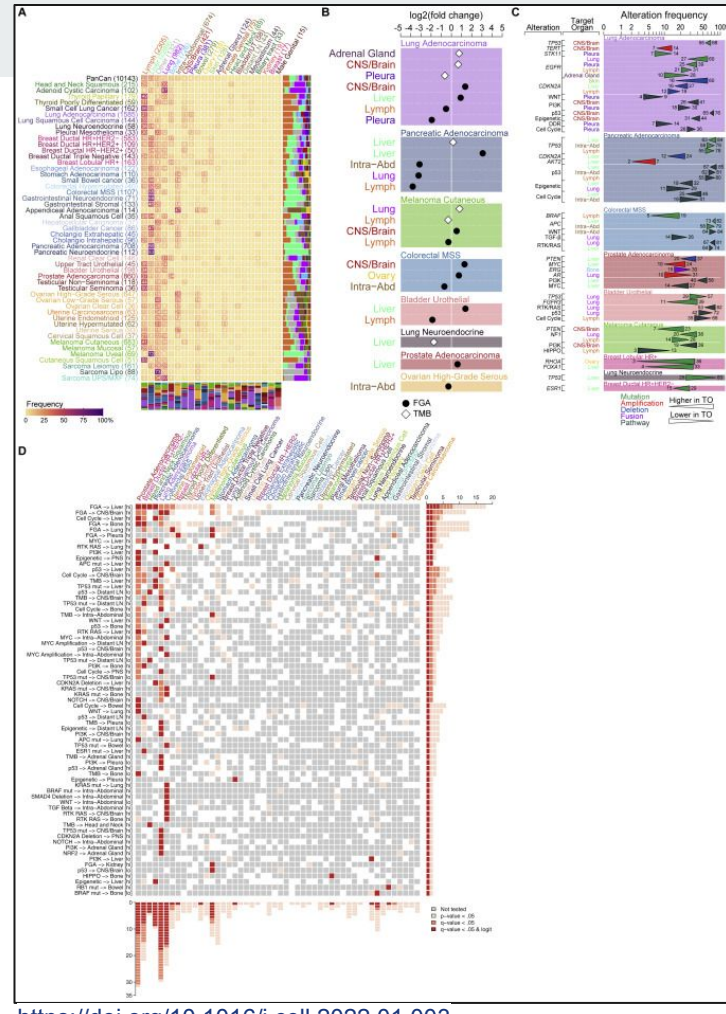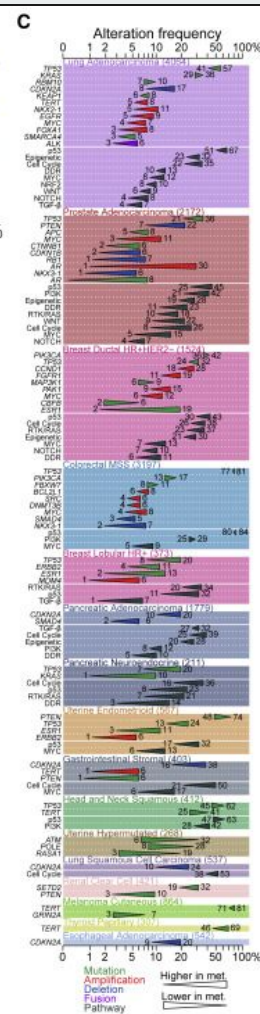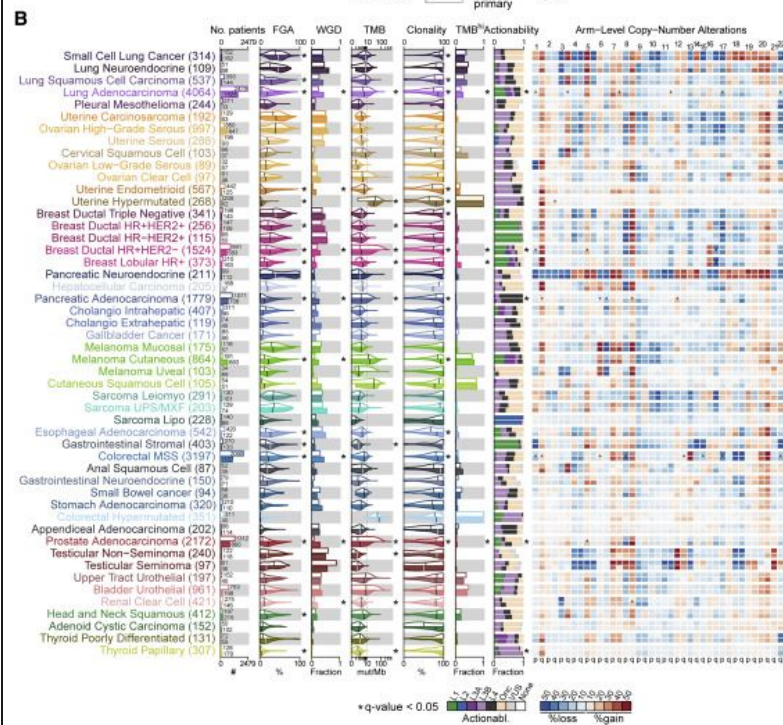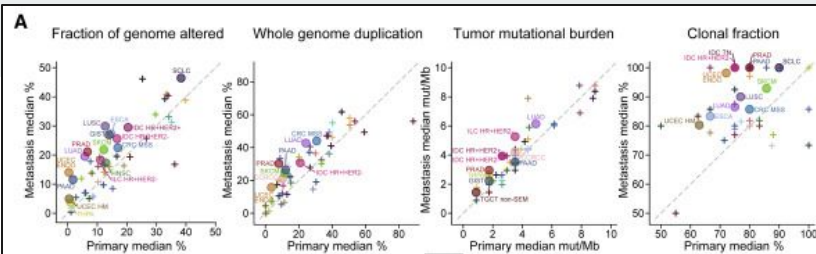
A forest of plots and color palettes

Which ones do you know ?
Cite some of them ... or we will be stuck on this slide forever.

A forest of plots and color palettes

Heatmap	Circular Plot	Network Graph
Venn Diagram	Raincloud	Sankey Diagram
Barplot	Manhattan Plot	Parallel Coordinates Plot
Histogram	Kaplan Meier Curve	Lollipop Plot
Density Plot	Violin Plot	PCA
Ridge plot	Bubble Charts	UMAP
Radar Plot	Line Charts	T-Sne
Scatter Plot	UpSet Plot	Correlogram
BoxPlot	Roc Curve	Volcano Plot

https://www.python-graph-gallery.com/

https://seaborn.pydata.org/tutorial/color_palettes.html
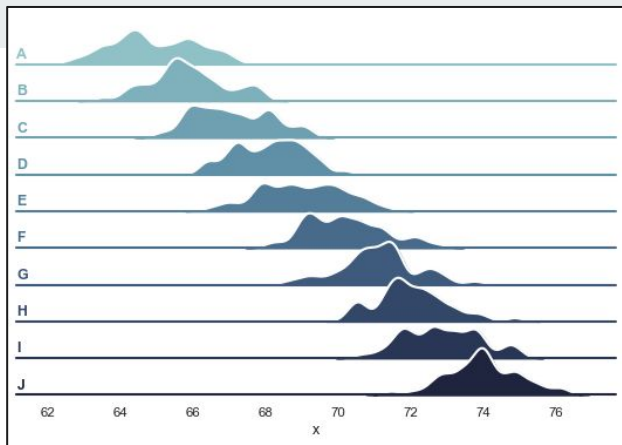
https://seaborn.pydata.org/examples/kde_ridgeplot.html

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

https://seaborn.pydata.org/examples/spreadsheet_heatmap.html
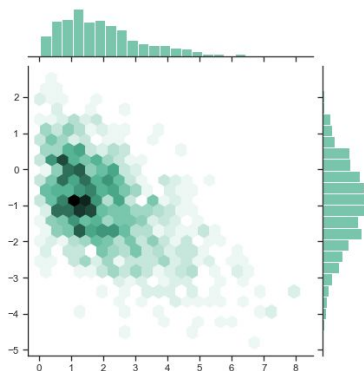
https://seaborn.pydata.org/examples/scatterplot_sizes.html
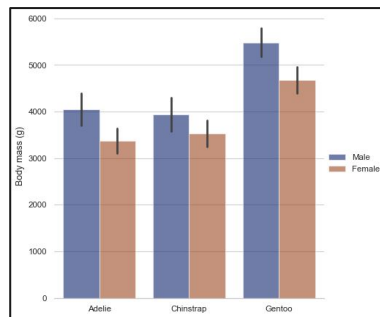
https://seaborn.pydata.org/examples/grouped_barplot.html
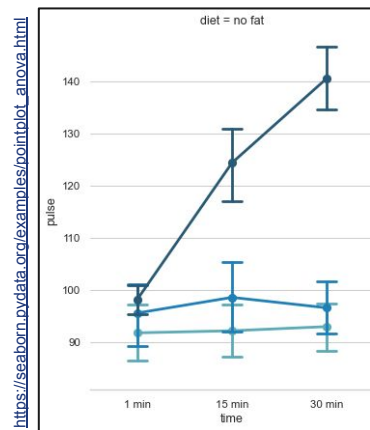
https://seaborn.pydata.org/examples/pointplot_anova.html

Genome Biology Guidelines

When preparing figures, please follow the formatting instructions below.

- Figures s**hould be numbered in the order they are first mentioned in the text,** and uploaded in this order. Multi-panel figures (those with parts a, b, c, d etc.) should be submitted as a single composite file that contains all parts of the figure.
- Figures should be uploaded in the correct orientation.
- **Figure titles (max 15 words) and legends (max 300 words)** should be provided in the main manuscript, not in the graphic file.
- Figure keys should be incorporated into the graphic, not into the legend of the figure.
- Each fi**gure should be closely cropped to minimize the amount of white space surrounding the illustration**. Cropping figures improves accuracy when placing the figure in combination with other elements when the accepted manuscript is prepared for publication on our site. For more information on individual figure file formats, see our detailed instructions.
- **Individual figure files should not exceed 10 MB**. If a suitable format is chosen, this file size is adequate for extremely high quality figures.

Figure file types

We accept the following file formats for figures:

- EPS (suitable for diagrams and/or images)
- PDF (suitable for diagrams and/or images)
- Microsoft Word (suitable for diagrams and/or images, figures must be a single page)
- PowerPoint (suitable for diagrams and/or images, figures must be a single page)
- TIFF (suitable for images)
- JPEG (suitable for photographic images, less suitable for graphical images)
- PNG (suitable for images)
- BMP (suitable for images)
- CDX (ChemDraw - suitable for molecular structures)

https://genomebiology.biomedcentral.com/submission-guidelines/preparing-your-manuscript

Figure size and resolution

Figures in the final PDF version:

- width of 85 mm for half page width figure
- width of **170 mm for full page** width figure
- **maximum height of 225 mm** for figure and legend
- image resolution of approximately 300 dpi (dots per inch) at the final size

To make life easy to everyone, **in Nature journals, a page is 183mm wide and 247mm deep**.

Figures should be designed such that all information, including text, is legible at these dimensions. All lines should be wider than 0.25 pt when constrained to standard figure widths. All fonts must be embedded.

Nature Communications Guidelines

Production-quality figures are not required at initial submission, but to avoid potential substantial revisions at later stages you may wish to note some of the guidelines below even at the initial submission stage.

It is recommended that you convert all your figures to JPEG before generating PDFs or uploading individual files. This will reduce the file sizes and the amount of time it takes the files to upload to our submission site and will also give you a closer approximation to the way your figures will appear on our site. If you choose to submit your files in PowerPoint format, please do not make a JPEG of these within PowerPoint. The conversion is more successful when a raw PowerPoint file is submitted.

General Figure Guidelines

- Use distinct colors with comparable visibility and **consider colorblind individuals** by avoiding the use of red and green for contrast. Recoloring primary data, such as fluorescence images, to color-safe combinations such as green and magenta, turquoise and red, yellow and blue or other accessible color palettes is strongly encouraged. Use of the rainbow color scale should be avoided.
- Use solid color for filling objects and avoid hatch patterns.
- Avoid background shading.
- Figures divided into parts should be labeled with a lower-case, boldface 'a', 'b', etc in the top left-hand corner. Labeling of axes, keys and so on should be in 'sentence case' (first word capitalized only) with no full stop. Units must have a space between the number and the unit, and follow the nomenclature common to your field.
- Commas should be used to separate thousands.
- Unusual units or abbreviations should be spelled out in full, or defined in the legend.

Final Figure Submission Guidelines

Should your manuscript be accepted, you will receive more extensive instructions for final submission of display items. However, a summary of our guidelines for final figure preparation are included here.

- **Images should be saved in RGB color mode at 300 dpi** or higher resolution.
- **Use the same typeface** (**Arial, Helvetica or Times New Roman)** for all figures. Use symbol font for Greek letters.
- We prefer vector files with editable layers. Acceptable formats are: .ai, .eps, .pdf, and .ps for fully editable vector-based art; layered .psd and .tif for editable layered art; .psd, .tif, .png and .jpg for bitmap images; .ppt if fully editable and without styling effects; ChemDraw (.cdx) for chemical structures. We are unable to support the following formats: .svg, .cvs, .xml, .cdr, .doc, .docx, .emf, .ibw, .opj, .vsd
- Figures are best prepared at the size you would expect them to appear in print. At this size, the **optimum font size is 8p**t and no lines should be thinner than 0.25 pt (0.09 mm).

https://mts-ncomms.nature.com/cgi-bin/main.plex?form_type=display_auth_instructions

Science Guidelines

Creating your figures It is best to create your figures as vector-based files such as those produced by Adobe Illustrator. Vector-based files will give us maximum flexibility for sizing your figures properly without losing resolution, as they can be altered in size while maintaining high print-quality resolution. **We cannot accept PowerPoint files or files that are not readable by Adobe Photoshop, Macromedia Freehand, or Adobe Illustrator.** To keep file sizes reasonable, please save art at a resolution of 150 to 300 dots per inch (dpi) for initial submission. A higher resolution applies for figures submitted at the revision stage – see instructions for preparing a revised manuscript. Digital color art should be submitted as CMYK (Cyan, Magenta, Yellow, Black) rather than RGB (Red, Green, Blue).

Paper The width of figures, when printed, will usually be 5.5 cm (2.25 inches or 1 column) or 12.0 cm (4.75 inches or 2 columns). Bar graphs, simple line graphs, and gels may be reduced to a smaller width. Symbols and lettering should be large enough to be legible after reduction [a reduced size of about 7 points (2 mm) high, and not smaller than 5 points]. Avoid wide variation in type size within a single figure. In laying out information in a figure, the objective is to maximize the space given to presentation of the data. Avoid wasted white space and clutter.

The figure's title should be at the beginning of the figure legend, not in the figure itself.

Include the figure's identifying number (e.g., "Fig. 1") on the same manuscript page that includes the figure.

Keys to symbols, if needed, should be kept as simple as possible and be positioned so they do not needlessly enlarge the figure. Details can be put into the captions.

Use solid symbols for plotting data if possible (unless data overlap or there are multiple symbols). Size symbols so that they will be distinguishable when the figure is reduced (6 pt minimum). Line widths should be legible upon reduction (minimum of 0.5 pt at the final reduced size).

Panels should be set close to each other, and common axis labels should not be repeated.

Scales or axes should not extend beyond the range of the data plotted.

Use scale bars in place of, or in addition to, magnifications. Do not use minor tick marks in scales or grid lines. Avoid using y-axis labels on the right that repeat those on the left.

Color-mix and contrast considerations

Avoid using red and green together. Color blind individuals will not be able read the figure.

Please do not use colors that are close in hue to identify different parts of a figure.

https://www.science.org/content/page/instructions-preparing-initial-manuscript#preparation-of-figures

Avoid using grayscale.

Use white type and scale bars over darker areas of images.

Units should be metric and follow SI convention.

Typefaces and labels

Please observe the following guidelines for labels on graphs and figures:

Use a sans-serif font whenever possible (we prefer Helvetica).

Simple solid or open symbols reduce well.

Label graphs on the ordinate and abscissa with the parameter or variable being measured, the units of measure in parentheses, and the scale. Scales with large or small numbers should be presented as powers of 10.

Avoid the use of light lines and screen shading. Instead, use black-and-white, hatched, and cross-hatched designs for emphasis.

Capitalize the first letter in a label only, not every word (and proper nouns, of course).

Units should be included in parentheses. Use SI notation. If there is room, write out variables – e.g., Pressure (MPa), Temperature (K).

Variables are always set in italics or as plain Greek letters (e.g., P, T, m). The rest of the text in the figure should be plain or bold text.

Type on top of color in a color figure should be in bold face. Avoid using color type.

When figures are assembled from multiple gels or micrographs, a line or space should indicate the border between two original images.

Use leading zeros on all decimals – e.g., 0.3, 0.55 – and only report significant digits.

U**se capital letters for part labels in multipart figures – A, B, C, etc. These should be 9 pt and bold in the final figur**e. When possible, place part labels at the upper left-hand corner of each figure part; if a part is an image, set labels inside the perimeter so as not to waste space.

If one day you get bored with Python for Viz...

- https://www.ibm.com/cloud/blog/python-vs-r

- https://www.datacamp.com/community/blog/when-to-use-python-or-r

- https://towardsdatascience.com/python-vs-r-for-data-science-cf2699dfff4b

- https://medium.com/@datadrivenscience/python-vs-r-for-data-science-and-the-winner-is-3ebb1a968197

- https://www.r-graph-gallery.com/

- https://upset.app/#upset-vs-venn-diagrams

Ressources

- [Principles of Effective Data Visualization](#) (Scientific Article)

- [Kimberly Fessel Youtube Channel](#) (Youtube Channel)

- [Cedric Scherer](#) (Blog)

- [Hands-On Data Visualization](#) (Free Book)

- [Fundamentals of Data Visualization](#) (Free Book)

- [Python Data Viz Tutorial](#) (Youtube Video)

Go Fast / Tricks : Online Tools

- https://software.broadinstitute.org/morpheus/

- https://bioinformatics.psb.ugent.be/webtools/Venn/

- https://sankeymatic.com/

Python For Data Science *Cheat Sheet*

Seaborn

Learn Data Science Interactively at www.DataCamp.com

Statistical Data Visualization With Seaborn

The Python visualization library **Seaborn** is based on matplotlib and provides a high-level interface for drawing attractive statistical graphics.

Make use of the following aliases to import the libraries:

```python
>>> import matplotlib.pyplot as plt
>>> import seaborn as sns
```

The basic steps to creating plots with Seaborn are:
1. Prepare some data
2. Control figure aesthetics
3. Plot with Seaborn
4. Further customize your plot

```python
>>> import matplotlib.pyplot as plt
>>> import seaborn as sns
>>> tips = sns.load_dataset("tips")          Step 1
>>> sns.set_style("whitegrid")               Step 2
>>> g = sns.lmplot(x="tip",                  Step 3
                   y="total_bill",
                   data=tips,
                   aspect=2)
>>> g = (g.set_axis_labels("Tip","Total bill(USD)").
    set(xlim=(0,10),ylim=(0,100)))           Step 4
>>> plt.title("title")
>>> plt.show(g)                              Step 5
```

1 Data
Also see Lists, NumPy & Pandas

```python
>>> import pandas as pd
>>> import numpy as np
>>> uniform_data = np.random.rand(10, 12)
>>> data = pd.DataFrame({'x':np.arange(1,101),
                         'y':np.random.normal(0,4,100)})
```

Seaborn also offers built-in data sets:

```python
>>> titanic = sns.load_dataset("titanic")
>>> iris = sns.load_dataset("iris")
```

2 Figure Aesthetics
Also see Matplotlib

```python
>>> f, ax = plt.subplots(figsize=(5,6))     Create a figure and one subplot
```

Seaborn styles

```python
>>> sns.set()                          (Re)set the seaborn default
>>> sns.set_style("whitegrid")         Set the matplotlib parameters
>>> sns.set_style("ticks",             Set the matplotlib parameters
                  {"xtick.major.size":8,
                   "ytick.major.size":8})
>>> sns.axes_style("whitegrid")        Return a dict of params or use with
                                       with to temporarily set the style
```

Context Functions

```python
>>> sns.set_context("talk")            Set context to "talk"
>>> sns.set_context("notebook",        Set context to "notebook",
                    font_scale=1.5,    scale font elements by 1.5,
                    rc={"lines.linewidth":2.5})  override param mapping
```

Color Palette

```python
>>> sns.set_palette("husl",3)          Define the color palette
>>> sns.color_palette("husl")          Use with with to temporarily set palette
>>> flatui = ["#9b59b6","#3498db","#95a5a6","#e74c3c","#34495e","#2ecc71"]
>>> sns.set_palette(flatui)            Set your own color palette
```

3 Plotting With Seaborn

Axis Grids

```python
>>> g = sns.FacetGrid(titanic,              Subplot grid for plotting conditional
                      col="survived",       relationships
                      row="sex")
>>> g = g.map(plt.hist,"age")
>>> sns.factorplot(x="pclass",              Draw a categorical plot onto a
                   y="survived",            Facetgrid
                   hue="sex",
                   data=titanic)
>>> sns.lmplot(x="sepal_width",             Plot data and regression model fits
               y="sepal_length",            across a FacetGrid
               hue="species",
               data=iris)
```

```python
>>> h = sns.PairGrid(iris)                  Subplot grid for plotting pairwise
>>> h = h.map(plt.scatter)                  relationships
>>> sns.pairplot(iris)                      Plot pairwise bivariate distributions
>>> i = sns.JointGrid(x="x",                Grid for bivariate plot with marginal
                      y="y",                univariate plots
                      data=data)
>>> i = i.plot(sns.regplot,
               sns.distplot)
>>> sns.jointplot("sepal_length",           Plot bivariate distribution
                  "sepal_width",
                  data=iris,
                  kind='kde')
```

Categorical Plots

Scatterplot

```python
>>> sns.stripplot(x="species",              Scatterplot with one
                  y="petal_length",         categorical variable
                  data=iris)
>>> sns.swarmplot(x="species",              Categorical scatterplot with
                  y="petal_length",         non-overlapping points
                  data=iris)
```

Bar Chart

```python
>>> sns.barplot(x="sex",                    Show point estimates and
                y="survived",               confidence intervals with
                hue="class",                scatterplot glyphs
                data=titanic)
```

Count Plot

```python
>>> sns.countplot(x="deck",                 Show count of observations
                  data=titanic,
                  palette="Greens_d")
```

Point Plot

```python
>>> sns.pointplot(x="class",                Show point estimates and
                  y="survived",             confidence intervals as
                  hue="sex",                rectangular bars
                  data=titanic,
                  palette={"male":"g",
                           "female":"m"},
                  markers=["^","o"],
                  linestyles=["-","--"])
```

Boxplot

```python
>>> sns.boxplot(x="alive",                  Boxplot
                y="age",
                hue="adult_male",
                data=titanic)
>>> sns.boxplot(data=iris,orient="h")       Boxplot with wide-form data
```

Violinplot

```python
>>> sns.violinplot(x="age",                 Violin plot
                   y="sex",
                   hue="survived",
                   data=titanic)
```

Regression Plots

```python
>>> sns.regplot(x="sepal_width",            Plot data and a linear regression
                y="sepal_length",           model fit
                data=iris,
                ax=ax)
```

Distribution Plots

```python
>>> plot = sns.distplot(data.y,             Plot univariate distribution
                        kde=False,
                        color="b")
```

Matrix Plots

```python
>>> sns.heatmap(uniform_data,vmin=0,vmax=1)   Heatmap
```

4 Further Customizations
Also see Matplotlib

Axisgrid Objects

```python
>>> g.despine(left=True)                    Remove left spine
>>> g.set_ylabels("Survived")               Set the labels of the y-axis
>>> g.set_xticklabels(rotation=45)          Set the tick labels for x
>>> g.set_axis_labels("Survived",           Set the axis labels
                      "Sex")
>>> h.set(xlim=(0,5),                       Set the limit and ticks of the
          ylim=(0,5),                       x-and y-axis
          xticks=[0,2.5,5],
          yticks=[0,2.5,5])
```

Plot

```python
>>> plt.title("A Title")           Add plot title
>>> plt.ylabel("Survived")         Adjust the label of the y-axis
>>> plt.xlabel("Sex")              Adjust the label of the x-axis
>>> plt.ylim(0,100)                Adjust the limits of the y-axis
>>> plt.xlim(0,10)                 Adjust the limits of the x-axis
>>> plt.setp(ax,yticks=[0,5])      Adjust a plot property
>>> plt.tight_layout()             Adjust subplot params
```

5 Show or Save Plot
Also see Matplotlib

```python
>>> plt.show()                     Show the plot
>>> plt.savefig("foo.png")         Save the plot as a figure
>>> plt.savefig("foo.png",         Save transparent figure
                transparent=True)
```

Close & Clear
Also see Matplotlib

```python
>>> plt.cla()                      Clear an axis
>>> plt.clf()                      Clear an entire figure
>>> plt.close()                    Close a window
```

DataCamp
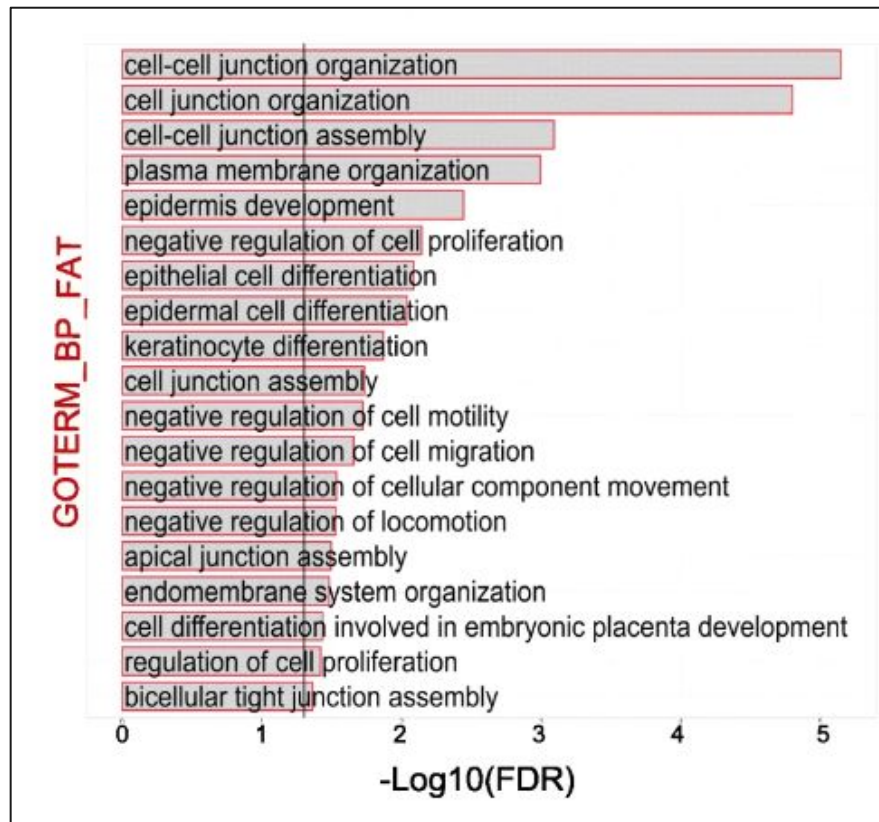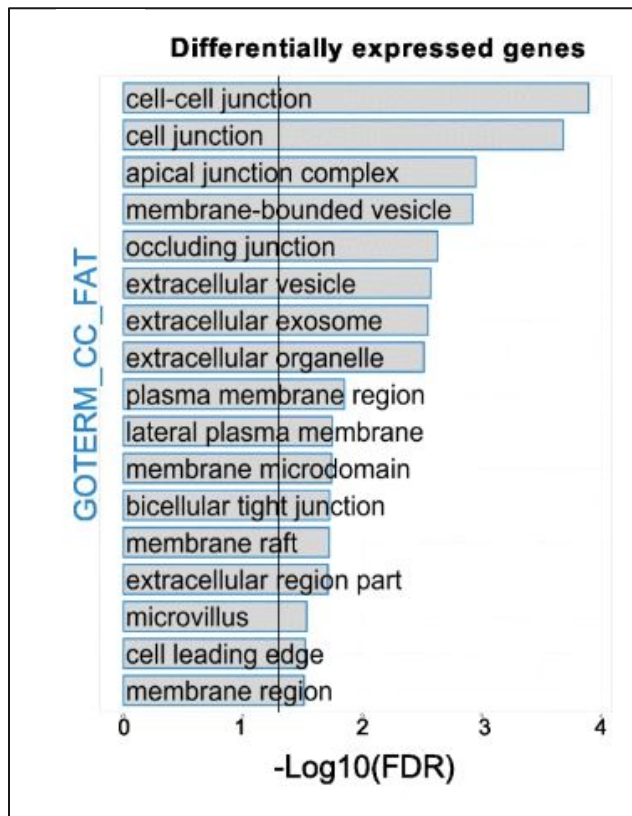Learn Python for Data Science Interactively

Few take home messages

- Basics are always good. (Heatmap, Barplot, Boxplot)

- Added statistics are always more than welcome.

- Try to make plots with legend that everyone can read easily in a lab meeting before thinking about a publication ready graph.

- Think about the size of your data before drawing.

- Describe univariate/correlation first to get the trend.

- Multidimensionality : PCA, TSNE & UMAP (Playing with the parameters & scaling can make a real change)

- Each journal has its own specificities.

- You should be able to reproduce your figures easily. Be organized.

- Scientific Twitter in your field can be good to follow the bibliography and discovers news stuffs to stay up-to-date easily (labs, softwares, new scientific publications)

- Don't be stuck with Python. Be open Minded.

- Be kind. Try to use colorblind palettes.

Exercise : can you reproduce this plot ?

Gene Ontology Over-Representation

David :Functional Annotation Chart Output

https://github.com/ZheFrench/Qbio

How the two sessions are organized ? - 1h30/1h

Generalities (Slides ~20 mins)
> WhoAmI
> Libraries in python
> Figures in science
> Guidelines
> Some plots you might know

Hands-on Seaborn (Jupyter Notebook ~15 mins / topic)
> Simple plots
> Composite Plots
> Heatmap
> Multidimensionality

Generally for the next session

Handle your panel of figures /know the difference between format. (~10mins)
> Inkscape / PowerPoint / Google Slides

Sources

- https://towardsdatascience.com/visualising-high-dimensional-datasets-using-pca-and-t-sne-in-python-8ef87e7915b

- https://medium.com/@violante.andre/an-introduction-to-t-sne-with-python-example-47e6ae7dc58f

- https://scipy-lectures.org/packages/scikit-learn/auto_examples/plot_digits_simple_classif.html

- https://shiva1gandluri.medium.com/principal-component-analysis-pca-in-machine-learning-c3f239249b73

- https://www.kaggle.com/uciml/breast-cancer-wisconsin-data/code

- Tsne Parameters Player https://distill.pub/2016/misread-tsne/

- https://levelup.gitconnected.com/statistics-on-seaborn-plots-with-statannotations-2bfce0394c00

- https://towardsdatascience.com/beautiful-boxplots-with-statistical-significance-annotation-e1b314927fc5

- https://github.com/webermarcolivier/statannot

- https://dev.to/thalesbruno/subplotting-with-matplotlib-and-seaborn-5ei8