Contents

Introduction

To provide a comprehensive background for the following chapters that focus on the interaction of human mesenchymal stromal cells (hMSCs) with multiple myeloma (MM) cells, this

Aims

This project defines these aims:

- Characterize the interaction between myeloma cells and mesenchymal stromal cells
- Develop methods
- Face the challenge of time-dependent cell adhesion through
- Provide tools to analyze multidimensional cell adhesion data

Chapter 2: Semi-Automating Data Analysis with `plotastic`

Abstract

`plotastic` addresses the challenges of transitioning from exploratory data analysis to hypothesis testing in Python's data science ecosystem. Bridging the gap between `seaborn` and `pingouin`, this library offers a unified environment for plotting and statistical analysis. It simplifies the workflow with user-friendly syntax and seamless integration with familiar `seaborn` parameters (y, x, hue, row, col). Inspired by `seaborn`'s consistency, `plotastic` utilizes a `DataAnalysis` object to intelligently pass parameters to `pingouin` statistical functions. Hence, statistics and plotting are performed on the same set of parameters, so that the strength of `seaborn` in visualizing multidimensional data is extended onto statistical analysis. In essence, `plotastic` translates `seaborn` parameters into statistical terms, configures statistical protocols based on intuitive plotting syntax and returns a `matplotlib` figure with known customization options and more. This approach streamlines data analysis, allowing researchers to focus on correct statistical testing and less about specific syntax and implementations.

Introduction

The reproducibility crisis in research highlights a significant challenge in contemporary biosciences, where a substantial portion of studies faces reproducibility issues (Baker, 2016; Begley & Ioannidis, 2015; Gosselin, 2021). One critical yet often overlooked aspect contributing to this crisis is data management. The literature most often refers to *big data* as the main challenge (Gomez-Cabrero et al., 2014). However, these challenges are also present in smaller datasets, which the author refers to as *semi-big data*. This term describes datasets that – while not extensive enough to necessitate advanced computational tools typically reserved for *big data* – are sufficiently large to render manual analysis very time-intensive. Semi-big data is often generated by methods like automated microscopy or multiplex qPCR, which produce volumes of data that are manageable on a surface level, but pose substantial barriers for in-depth, manual reproducibility (Bustin, 2014; Incerti et al., 2019). This is further complicated by the complexity inherent in multidimensional datasets (Krzywinski & Savig, 2013): Modern biosciences describe processes (e.g. cell-adhesion) that are highly dependent on multiple experimental parameters, like time or kinds of treatments (Rebl et al., 2010; McKay et al., 1997). Without a clearly documented data analysis protocol and standardized data formats, such analyses become nontransparent and too overwhelming for reproduction (Bustin, 2014).

The evolving standards in data analysis advocate for the standardization of analytical pipelines, rationalization of sample sizes, and enhanced infrastructure for data storage, addressing some of these challenges (Goodman et al., 2016; Wilkinson et al., 2016). However, these advancements can place undue pressure on researchers, particularly those with limited training in statistics, underscoring the need for intuitive, user-friendly analytical tools (Federer et al., 2016; Lakhlifi et al., 2023; Armstrong, 2014; Gómez-López et al., 2019; Leek & Peng, 2015)

In this context, `plotastic` emerges as a tool designed to democratize access to sophisticated statistical analysis, offering a user-centric interface that caters to researchers across varying levels of statistical proficiency.

`plotastic` simplifies inferential statistics based on the idea that statistical analyses are often performed based on how the data is visualized, and automates the process based on the same intuitive plotting parameters used to structure the figure. By integrating robust statistical methodologies within an accessible framework, `plotastic` could to contribute to enhancing the reproducibility of research in the biosciences (Gomez-Cabrero et al., 2014).

The user-centric approach of `plotastic` distinguishes itself from the fully automated pipelines used for big data, which are designed to handle extensive computational tasks. Instead, `plotastic` focuses on intermediate outputs and ease-of-use, a concept the author refers to as *semi-automation* (Tab. 1).

Table 1: Key Principles of Semi-Automation and their Implementation in Plotastic

No.	Principle	Implementation in `plotastic`
1	**Standardized input:** The data to-be-analyzed follows a strict standardized format. The user should be able to convert their data into that format.	Long-format `pandas` `DataFrames` are used as input
2	**Automation over flexibility:** If there is an obvious way to do things, automate it and minimize user input. User options should be added with good reason. Avoid situations where the user is asked to pass the same parameter twice. This reduces the risk of human error, confusion and time spent on configuration.	E.g. passing the parameter "subject" once makes the rest of the pipeline switch automatically to the paired versions of statistical tests.
3	***Out of the box* functionality:** The software's default configuration should provide acceptable (but potentially sub-optimal) results. Beginners shouldn't need to learn custom configurations. Options are still available to allow feature-rich adaptions according to the needs of both data and user.	Default tests are standard unpaired t-tests and ANOVA
4	**Focus on intermediate outputs:** The user composes the analysis pipeline using smaller commands that are each designed to provide human-readable output of an intermediate result. Each step is a stage to control quality, allowing quick error detection and troubleshooting.	Processing steps are separated into main steps: assumption tests, factor analysis, post-hoc analysis and plotting
5	**Highly useful error messages:** Never leave the user hanging. Tell him what went wrong *and* what the software was expecting.	E.g.: `ValueError: User passed 'subect' as subject, please choose one of ['subject', 'event', 'region']`

The need for `plotastic` in this specific project arose from two main challenges (for further details, see summarizing discussion). The first is the author's need for a tool that could handle the complex, multidimensional data generated by e.g. qPCR experiments. These experiments involved the analysis of multiple outcomes across multiple genes, timepoints, method variations, cell-types, biological replicates, technical replicates etc., resulting in datasets that are challenging to analyse manually. Such complexity was necessary, since establishing new methods required extensive controls and creative variation of the experimental setup. Data analysis had to be automated somehow, since the lab-work itself was already time-intensive. The second challenge was to ignore the potential of plot-configured statistical analyses. The author believes that the way data is visualized is often the way it should be analyzed. This is a principle that is not only intuitive but also statistically sound. The author's vision was to create a generally useful tool that would automate the statistical analysis, contributing a powerful tool to the scientific community.

Statement of Need

Python's data science ecosystem provides powerful tools for both visualization and statistical testing. However, the transition from exploratory data analysis to hypothesis testing can be cumbersome, requiring users to switch between libraries and adapt to different syntaxes. `seaborn` has become a popular choice for plotting in Python, offering an intuitive interface. Its statistical functionality focuses on descriptive plots and bootstrapped confidence intervals (Waskom, 2021). The library `pingouin` offers an extensive set of statistical tests, but it lacks integration with common plotting capabilities (Vallat, 2018). `statannotations` integrates statistical testing with plot annotations, but uses a complex interface and is limited to pairwise comparisons (Charlier et al., 2022).

`plotastic` addresses this gap by offering a unified environment for plotting and statistical analysis. With an emphasis on user-friendly syntax and integration of familiar `seaborn` parameters, it simplifies the process for users already comfortable with `seaborn`. The library ensures a smooth workflow, from data import to hypothesis testing and visualization.

Example

The following code demonstrates how plotastic analyzes the example dataset “fmri”, similar to Waskom (2021) (Fig. 1).

```python
### IMPORT PLOTASTIC
import plotastic as plst

# IMPORT EXAMPLE DATA
DF, _dims = plst.load_dataset("fmri", verbose = False)

# EXPLICITLY DEFINE DIMENSIONS TO FACET BY
dims = dict(
      y = "signal",    # y-axis, dependent variable
      x = "timepoint", # x-axis, independent variable (within-subject factor)
      hue = "event",   # color,  independent variable (within-subject factor)
      col = "region"   # axes,   grouping variable
)
# INITIALIZE DATAANALYSIS OBJECT
DA = plst.DataAnalysis(
      data=DF,           # Dataframe, long format
      dims=dims,         # Dictionary with y, x, hue, col, row
      subject="subject", # Datapoints are paired by subject (optional)
      verbose=False,     # Print out info about the Data (optional)
)
# STATISTICAL TESTS
DA.check_normality()   # Check Normality
DA.check_sphericity()  # Check Sphericity
DA.omnibus_rm_anova()  # Perform RM-ANOVA
DA.test_pairwise()     # Perform Posthoc Analysis

# PLOTTING
(DA
.plot_box_strip()     # Pre-built plotting function initializes plot
.annotate_pairwise(   # Annotate results from DA.test_pairwise()
      include="__HUE" # Use only significant pairs across each hue
      )
)
```

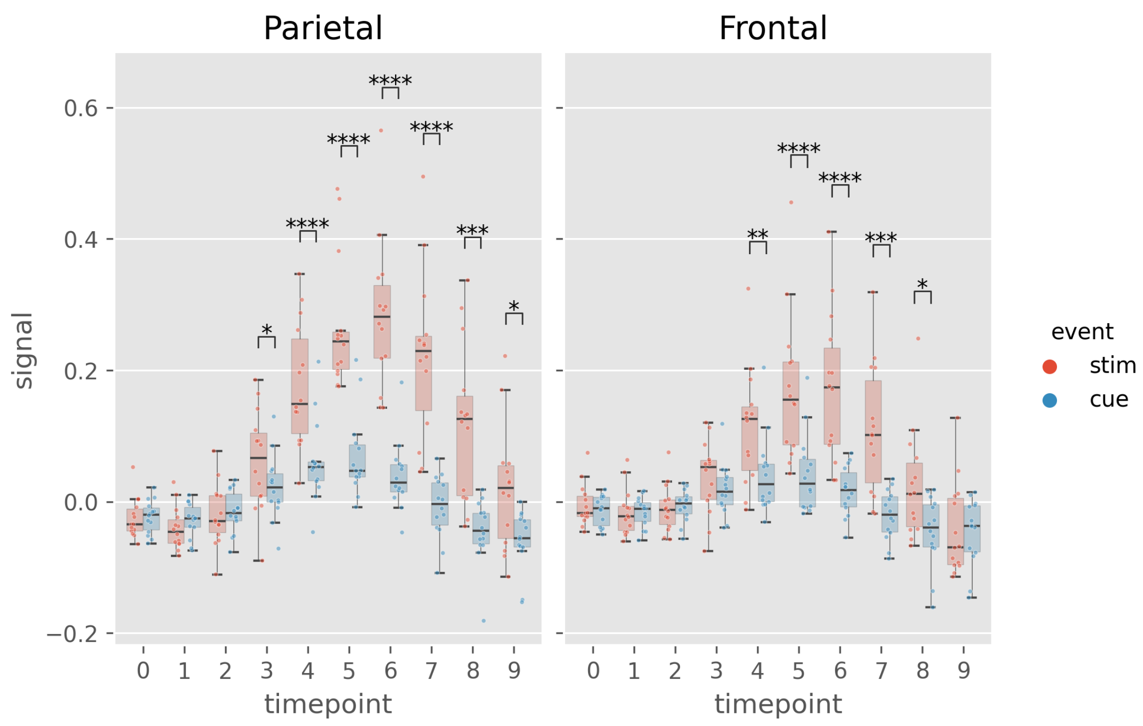

Figure 1: Example figure of `plotastic` (version 0.1). Image style was set by `plt.style.use('ggplot')`

Table 2: Results from `DA.check_sphericity()`. `plotastic` assesses sphericity after grouping the data by all grouping dimensions (hue, row, col). For example, `DA.check_sphericity()` grouped the 'fmri' dataset by "region" (col) and "event" (hue), performing four subsequent sphericity tests for four datasets.

'region', 'event'	spher	W	chi2	dof	pval	group count	n per group
'frontal', 'cue'	True	3.26e+20	-462.7	44	1	10	[14]
'frontal', 'stim'	True	2.45e+17	-392.2	44	1	10	[14]
'parietal', 'cue'	True	1.20e+20	-452.9	44	1	10	[14]
'parietal', 'stim'	True	2.44e+13	-301.9	44	1	10	[14]

Table 3: Results of `DA.omnibus_rm_anova()`. `plotastic` performs one two-factor RM-ANOVA per axis (grouping the data by row and col dimensions) using x and hue as the within-factors. For this example, `DA.omnibus_rm_anova()` grouped the 'fmri' dataset by "region" (col), performing two subsequent two-factor RM-ANOVAs. Within-factors are "timepoint" (x) and "event" (hue). For conciseness, GG-Correction and effect sizes are not shown.

'region'	Source	SS	ddof1	ddof2	MS	F	p-unc	stars
'parietal'	timepoint	1.583	9	117	0.175	26.20	3.40e-24	****
'parietal'	event	0.770	1	13	0.770	85.31	4.48e-07	****
'parietal'	timepoint * event	0.623	9	117	0.069	29.54	3.26e-26	****
'frontal'	timepoint	0.686	9	117	0.076	15.98	8.28e-17	****
'frontal'	event	0.240	1	13	0.240	23.44	3.21e-4	***
'frontal'	timepoint * event	0.242	9	117	0.026	13.031	3.23e-14	****

Overview

The functionality of `plotastic` revolves around a seamless integration of statistical analysis and plotting, leveraging the capabilities of `pingouin`, `seaborn`, `matplotlib` and `statannotations` (Vallat, 2018; Waskom, 2021; Hunter, 2007; Charlier et al., 2022). It utilizes long-format `pandas` `DataFrames` as its primary input, aligning with the conventions of `seaborn` and ensuring compatibility with existing data structures (Wickham, 2014; Team, 2020; McKinney, 2010).

`plotastic` was inspired by `seaborn` using the same set of intuitive and consistent parameters (y, x, hue, row, col) found in each of its plotting functions (Waskom, 2021). These parameters intuitively delineate the data dimensions plotted, yielding 'facetted' subplots, each presenting y against x. This allows for rapid and insightful exploration of multidimensional relationships. `plotastic` extends this principle to statistical analysis by storing these `seaborn` parameters (referred to as dimensions) in a `DataAnalysis` object and intelligently passing them to statistical functions of the `pingouin` library. This approach is based on the impression that most decisions during statistical analysis can be derived from how the user decides to arrange the data in a plot. This approach also prevents code repetition and streamlines statistical analysis. For example, the subject keyword is specified only once during `DataAnalysis` initialisation, and `plotastic` selects the appropriate paired or unpaired version of the test. Using `pingouin` alone requires the user to manually pick the correct test and to repeatedly specify the subject keyword in each testing function.

In essence, `plotastic` translates plotting parameters into their statistical counterparts. This translation minimizes user input and also ensures a coherent and logical connection between plotting and statistical analysis. The goal is to allow the user to focus on choosing the correct statistical test (e.g. parametric vs. non-parametric) and worry less about specific implementations.

At its core, `plotastic` employs iterators to systematically group data based on various dimensions, aligning the analysis with the distinct requirements of tests and plots. Normality testing is performed on each individual sample, which is achieved by splitting the data by all grouping dimensions and also the x-axis (hue, row, col, x). Sphericity and homoscedasticity testing is performed on a complete sampleset listed on the x-axis, which is achieved by splitting the data by all grouping dimensions (hue, row, col) (Tab. 2). For omnibus and posthoc analyses, data is grouped by the row and col dimensions in parallel to the `matplotlib` axes, before performing one two-factor analysis per axis using x and hue as the within/between-factors. (Tab. 3).

`DataAnalysis` visualizes data through predefined plotting functions designed for drawing multilayered plots. A notable emphasis within `plotastic` is placed on showcasing individual datapoints alongside aggregated means or medians. In detail, each plotting function initializes a `matplotlib` figure and axes using `plt.subplots()` while returning a `DataAnalysis` object for method chaining. Axes are populated by seaborn plotting functions (e.g., sns.boxplot()), leveraging automated aggregation and error bar displays. Keyword arguments are passed to these seaborn functions, ensuring the same degree of customization. Users can further customize plots by chaining DataAnalysis methods or by applying common matplotlib code to override plotastic settings. Figures are exported using plt.savefig().

plotastic also focuses on annotating statistical information within plots, seamlessly incorporating p-values from pairwise comparisons using statannotations (Charlier et al., 2022). This integration simplifies the interface and enables options for pair selection in multidimensional plots, enhancing both user experience and interpretability.

For statistics, plotastic integrates with the pingouin library to support classical assumption and hypothesis testing, covering parametric/non-parametric and paired/non-paired variants. Assumptions such as normality, homoscedasticity, and sphericity are tested. Omnibus tests include two-factor RM-ANOVA, ANOVA, Friedman, and Kruskal-Wallis. Posthoc tests are implemented through pingouin.pairwise_tests(), offering (paired) t-tests, Wilcoxon, and Mann-Whitney-U.

To sum up, plotastic stands as a unified and user-friendly solution catering to the needs of researchers and data scientists, seamlessly integrating statistical analysis with the power of plotting in Python. It streamlines the workflow, translates seaborn parameters into statistical terms, and supports extensive customization options for both analysis and visualization.

Discussion

Is plotastic tested? Coverage? Does it cover every feature? What is not covered

The full code of an example analysis is shown in **??**.

Although the awareness for multidimensional data is increasing (Dunn et al., 2017), intuitive tools for the analysis of such data are still lacking.

Is plotastic USABLE for biologists? - Yes but use is limited by minimal knowledge of Python - However, that is subject to change as Python is becoming more popular in biology and AI assisted coding decreased the barrier to entry significantly. Tools like github copilot are able to generate code, fix bugs and suggest improvements. This is a game changer for biologists that are not familiar with programming. - Furthermore, installing and using plotastic for biologists is overestimated. These steps re needed: - Install anaconda from the internet - Open the terminal - Type `pip install plotastic` - Check Rea

- lack of GUI? Jupyter Notebooks include buttons to save a figure, and can be extended with features like sorting displayed tables (?) are compatible with packages with GUIs, like. However, GUIs do not promise an easier use, in fact, given the text-based approach of codes, this makes code very compatible with Large Language Models, offering a powerful debugging platform. In fact, tools like vscode offer GitHub copilot for free for academics, integrating with the interface and displaying suggestions for code completion.

A GUI on the other hand is hard to design, maintain, is prone to errors and when users face an error, there is no options for workarounds

The evaluation of plotastic within this thesis reflects its potential to address key challenges in the field of data analysis. The software integrates a comprehensive suite of statistical tests, such as ANOVA and t-tests, designed for adaptability and ease of use, leveraging the functionalities of pingouin.

In the context of the reproducibility crisis in scientific research, plotastic offers noteworthy contributions, though it is not positioned as a universal remedy. The tool's unique approach to integrating statistical analysis with visual representation establishes a new paradigm, promoting methodological transparency. By mandating that statistical analyses accompany relevant graphical outputs, plotastic ensures that analyses are not only conducted with proper scientific rigor but also documented in a manner that facilitates replication, provided the user possesses proficiency in Python.

Peng (2011) claims that exploration of the data and the analysis code may be sufficient to verify the quality of the scientific claims.

Mesirov (2010) called out for "simple, intuitive ways to both capture and embed our computational work directly into our papers". In this thesis, combining Jupyter Notebooks with plotastic could represent an intriguing solution to solve that problem.

Statistical or Data illiteracy is a critical issue that's well researched for both clinicians and biomedical researchers (Lakhlifi et al., 2023; Federer et al., 2016). plotastic removes a lot of need for statistical knowledge, yet still performs complex statistical analyses, if the measurements were correctly performed and the results are correctly interpreted. Using unpaired non-parametric tests, plotastic provides statistically sound for a wide array of study types without the need for the user to understand the underlying statistical principles. This is a significant advantage for biologists, who often lack the necessary statistical training to perform these analyses manually.

Usability is a critical attribute of analytical software, particularly as researchers confront increasingly complex datasets. While the developer's intimate familiarity with plotastic may bias perceptions of its ease of use, it is recognized that novices may initially encounter challenges. Nevertheless, plotastic is distinguished by its user-friendly interface, enabling users with minimal statistical training to perform sophisticated analyses by intuitively mapping plotting concepts to statistical operations.

The transition to a new analytical framework, especially one that incorporates coding, presents a learning curve. However, the advantages of plotastic in terms of analytical clarity, speed, and depth are anticipated to outweigh these initial challenges. Support mechanisms, such as assistance from advanced AI like ChatGPT, are available to mitigate these hurdles, supporting users across varying levels of expertise.

In conclusion, plotastic is posited as a valuable tool in the landscape of scientific research, offering a means to enhance the reproducibility and efficiency of data analysis. Its development ethos emphasizes simplifying complex analytical tasks, thereby contributing to the broader goal of fostering transparent and reproducible research practices.

DO we apply the principles of Semi-Automation to the software?

what features are missing? - Bivariate analysis - Filer to help save the output? - StatResults: System to suggest the correct test, based on the data

References

Armstrong, R. A. (2014, September). When to use the Bonferroni correction. *Ophthalmic & Physiological Optics: The Journal of the British College of Ophthalmic Opticians (Optometrists)*, *34*(5), 502–508. doi: 10.1111/opo.12131

Baker, M. (2016, May). 1,500 scientists lift the lid on reproducibility. *Nature*, *533*(7604), 452–454. Retrieved 2024-04-22, from `https://www.nature.com/articles/533452a` doi: 10.1038/533452a

Begley, C. G., & Ioannidis, J. P. A. (2015, January). Reproducibility in science: Improving the standard for basic and preclinical research. *Circulation Research*, *116*(1), 116–126. doi: 10.1161/CIRCRESAHA.114.303819

Bustin, S. A. (2014, December). The reproducibility of biomedical research: Sleepers awake! *Biomolecular Detection and Quantification*, *2*, 35–42. Retrieved 2024-03-18, from `https://www.sciencedirect.com/science/article/pii/S2214753515000030` doi: 10.1016/j.bdq.2015.01.002

Charlier, F., Weber, M., Izak, D., Harkin, E., Magnus, M., Lalli, J., … Repplinger, S. (2022, October). *Trevismd/statannotations: V0.5.* Zenodo. Retrieved 2023-11-16, from `https://zenodo.org/record/7213391` doi: 10.5281/ZENODO.7213391

Dunn, W., Burgun, A., Krebs, M.-O., & Rance, B. (2017, November). Exploring and visualizing multidimensional data in translational research platforms. *Briefings in Bioinformatics*, *18*(6), 1044–1056. Retrieved 2024-04-23, from `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5862238/` doi: 10.1093/bib/bbw080

Federer, L. M., Lu, Y.-L., & Joubert, D. J. (2016, January). Data literacy training needs of biomedical researchers. *Journal of the Medical Library Association : JMLA*, *104*(1), 52–57. Retrieved 2024-04-24, from `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4722643/` doi: 10.3163/1536-5050.104.1.008

Gomez-Cabrero, D., Abugessaisa, I., Maier, D., Teschendorff, A., Merkenschlager, M., Gisel, A., … Tegnér, J. (2014, March). Data integration in the era of omics: Current and future challenges. *BMC Systems Biology*, *8*(2), I1. Retrieved 2024-03-18, from `https://doi.org/10.1186/1752-0509-8-S2-I1` doi: 10.1186/1752-0509-8-S2-I1

Gómez-López, G., Dopazo, J., Cigudosa, J. C., Valencia, A., & Al-Shahrour, F. (2019, May). Precision medicine needs pioneering clinical bioinformaticians. *Briefings in Bioinformatics*, *20*(3), 752–766. doi: 10.1093/bib/bbx144

Goodman, S. N., Fanelli, D., & Ioannidis, J. P. A. (2016, June). What does research reproducibility mean? *Science Translational Medicine*, *8*(341), 341ps12-341ps12. Retrieved 2024-03-18, from `https://www.science.org/doi/10.1126/scitranslmed.aaf5027` doi: 10.1126/scitranslmed.aaf5027

Gosselin, R.-D. (2021, February). Insufficient transparency of statistical reporting in preclinical research: A scoping review. *Scientific Reports*, *11*(1), 3335. Retrieved 2024-03-11, from `https://www.nature.com/articles/s41598-021-83006-5` doi: 10.1038/s41598-021-83006-5

Hunter, J. D. (2007, May). Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*, *9*(3), 90–95. Retrieved 2023-11-15, from `https://ieeexplore.ieee.org/document/4160265` doi: 10.1109/MCSE.2007.55

Incerti, D., Thom, H., Baio, G., & Jansen, J. P. (2019, May). R You Still Using Excel? The Advantages of Modern Software Tools for Health Technology Assessment. *Value in Health*, *22*(5), 575–579. Retrieved 2024-03-11, from `https://www.sciencedirect.com/science/article/pii/S1098301519300506` doi: 10.1016/j.jval.2019.01.003

Krzywinski, M., & Savig, E. (2013, July). Multidimensional data. *Nature methods*, *10*(7), 595. Retrieved 2024-04-22, from `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6092027/`

Lakhlifi, C., Lejeune, F.-X., Rouault, M., Khamassi, M., & Rohaut, B. (2023, April). Illusion of knowledge in statistics among clinicians: Evaluating the alignment between objective accuracy and subjective confidence, an online survey. *Cognitive Research: Principles and Implications*, *8*, 23. Retrieved 2024-04-24, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10118231/ doi: 10.1186/s41235-023-00474-1

Leek, J. T., & Peng, R. D. (2015, April). Statistics: P values are just the tip of the iceberg. *Nature*, *520*(7549), 612–612. Retrieved 2024-04-22, from https://www.nature.com/articles/520612a doi: 10.1038/520612a

McKay, B. S., Irving, P. E., Skumatz, C. M., & Burke, J. M. (1997, November). Cell-cell adhesion molecules and the development of an epithelial phenotype in cultured human retinal pigment epithelial cells. *Experimental Eye Research*, *65*(5), 661–671. doi: 10.1006/exer.1997.0374

McKinney, W. (2010, January). Data Structures for Statistical Computing in Python. In (pp. 56–61). doi: 10.25080/Majora-92bf1922-00a

Mesirov, J. P. (2010, January). Accessible Reproducible Research. *Science*, *327*(5964), 415–416. Retrieved 2024-04-22, from https://www.science.org/doi/10.1126/science.1179653 doi: 10.1126/science.1179653

Peng, R. D. (2011, December). Reproducible Research in Computational Science. *Science*, *334*(6060), 1226–1227. Retrieved 2024-03-18, from https://www.science.org/doi/10.1126/science.1213847 doi: 10.1126/science.1213847

Rebl, H., Finke, B., Schroeder, K., & Nebe, J. B. (2010, October). Time-dependent metabolic activity and adhesion of human osteoblast-like cells on sensor chips with a plasma polymer nanolayer. *The International Journal of Artificial Organs*, *33*(10), 738–748.

Team, T. P. D. (2020, February). *Pandas-dev/pandas: Pandas.* Zenodo. Retrieved from https://doi.org/10.5281/zenodo.3509134 doi: 10.5281/zenodo.3509134

Vallat, R. (2018, November). Pingouin: Statistics in Python. *Journal of Open Source Software*, *3*(31), 1026. Retrieved 2023-05-29, from https://joss.theoj.org/papers/10.21105/joss.01026 doi: 10.21105/joss.01026

Waskom, M. L. (2021, April). Seaborn: Statistical data visualization. *Journal of Open Source Software*, *6*(60), 3021. Retrieved 2023-03-26, from https://joss.theoj.org/papers/10.21105/joss.03021 doi: 10.21105/joss.03021

Wickham, H. (2014, September). Tidy Data. *Journal of Statistical Software*, *59*, 1–23. Retrieved 2023-11-15, from https://doi.org/10.18637/jss.v059.i10 doi: 10.18637/jss.v059.i10

Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G., Axton, M., Baak, A., ... Mons, B. (2016, March). The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data*, *3*(1), 160018. Retrieved 2024-03-18, from https://www.nature.com/articles/sdata201618 doi: 10.1038/sdata.2016.18

Appendices

A Supplementary Data & Methods

A.1 Figures

A.2 Tables

A.3 Materials & Methods

B Documentation of `plotastic`