GSoC 2025

Project Proposal

NUMFOCUS
OPEN CODE = BETTER SCIENCE

ARVIZ

ArviZ Plots - Feature Parity

Shreyansh

Mentors: Osvaldo Martin, Andy Maloney & Rohan Babbar

Google Summer of Code

Table of Contents

1. Introduction

1.1. Project Synopsis

ArviZ is undergoing a major refactoring, splitting into three sub-packages: ArviZ-base, ArviZ-stats, and ArviZ-plots. This project focuses on ensuring feature parity in ArviZ-plots with legacy ArviZ by reintroducing essential visualization tools and enhancing them with new features. The main goal is to implement and test plotting functions critical for model comparison, criticism, and sampling diagnostics.

Key features to be implemented include:

- MCSE plots
- Dark theme (“tumma”), based on Plotly dark theme with modifications
- Quantile dot plots (default to 100 dots, as an option in plot_dist)
- plot_t_stats for t-stat visualizations
- Trace plots for circular variables
- Improving Rank plots with better envelope
- Rank Distribution plots (plot_rank_dist)
- Parallel coordinate plots

These implementations will improve the usability and comprehensiveness of ArviZ-plots, aligning it with legacy ArviZ.

1.2. Relevant Work

1.2.1. In Arviz-Plots

I have experience working with the ArviZ-plots & I am familiar with its codebase. So far, I have contributed several pull requests to the repository and am actively working on additional improvements. My contributions include:

- PR 147
- PR 178
- PR 183
- PR 189
- PR 124
- PR 136
- PR 199

Currently working on Issue 204

1.2.2. Outside Arviz-plots

1. I have previously worked with sampling strategies like Markov Chain Monte Carlo (MCMC) while implementing a machine learning algorithm designed to optimize traffic distribution and reduce pollution in a specific area. This work is particularly relevant because MCMC is crucial for approximating and sampling from complex, high-dimensional posterior distributions, which is a fundamental aspect of Bayesian inference and probabilistic modeling.

   Here is a link to the project:

- Efficient Traffic Distribution Algorithm

  This experience has strengthened my understanding of MCMC methods and their practical applications, making me well-equipped to contribute to ArviZ.

2. I have completed the **Design and Analysis of Software Systems** course, which focuses on designing efficient and scalable systems. This course has given me a strong understanding of the importance of refactoring and system modularity, enabling me to make informed design decisions while implementing new functionalities in ArviZ-plots. Additionally, it introduced me to essential tools such as pylint for maintaining code quality.

   Here are some projects relevant to this course that demonstrate my experience in system design and software architecture:

   - UML Class Diagram for IMS Portal – A structured design representation of the backbone system of our institution.
   - Drawing Tool (Python) – Contributions to the design and implementation of a drawing editor.
   - Telugu Input Chrome Extension – Development of a Chrome extension for Telugu language auto-prediction.

3. I have worked extensively with Python on a project that involved **building statistical machine learning models from scratch**, including **K-Nearest Neighbors (KNN), Gaussian Mixture Models (GMM), Hidden Markov Models (HMM), Multi-Layer Perceptrons (MLP), Neural Networks, Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Optical Character Recognition (OCR)**.

   A significant aspect of this project was visualizing various performance metrics of these models using Matplotlib and other plotting tools. This experience also demonstrates my command of Python and its scientific computing ecosystem.

   Here is the link to the project:

   - PyTorch From Scratch – A foundational PyTorch-like library built from scratch for various statistical machine learning models.

4. I am an **undergraduate researcher at the Machine Learning Lab, IIIT Hyderabad**, where I am actively working on **Reinforcement Learning, Multi-Agent Systems, and Intelligent Transportation**. Through my research, I have frequently encountered **Bayesian models** and have developed **familiarity with their concepts, applications, and analysis techniques**. My experience in these areas has further strengthened my understanding of **probabilistic modeling and Bayesian inference**, which aligns well with my contributions to ArviZ-plots.

1.3. Technical Skills

1.3.1. Languages

C, C++, Go, x86 Assembly, Bash, Python, R, Javascript

1.3.2. Theory

Probability and Statistics, Machine and Data Learning, Statistical methods in AI, Design and Analysis of Software Systems, Data Structures and Algorithms, Algorithm Analysis and Design, Operating Systems, Computer Systems Organization

1.3.3. Tools

Linux, Docker, gdb, LaTeX, Typst

1.4. Bio

1.4.1. About

I'm a 3rd year undergraduate student pursuing my B.Tech in Computer Science at IIIT-H, and also an undergraduate researcher in Machine Learning Lab at IIIT-H. I am deeply interested in probabilistic modeling, sampling techniques and Bayesian inferences.

1.4.2. Why choose me

- I have thoroughly studied the key concepts underlying this project and have a clear understanding of the goals of each component, the implementation methods being used to achieve them, and the specific requirements involved. I am familiar with Bayesian modeling and sampling methods, and I am actively deepening my knowledge in this area through continuous learning.
- I just went through paper "Graphical test for discrete uniformity and its applications in goodness-of-fit evaluation and multiple sample comparison". Therefore, I understand the proposed way to ensure correct envelope for comparison of multiple samples in rank plot.
- I am well-versed in Python and plotting libraries like Matplotlib, Plotly and Bokeh. I have built many personal projects using Python.
- I aim to heavily test and document all the functionality that I will be implementing.
- I am familiar with the codebase and have made contributions to this project.

I chose this project as it is the perfect way to utilize my skill-set, while learning a lot more at the same time.

1.5. Contact Information

Name	Shreyansh
College	International Institute of Information Technology, Hyderabad, India
Degree Program	B.Tech in Computer Science
Time Zone	GMT +5:30
Links	Github - Linkedin
Emails	saurabhshreyansh868@gmail.com shreyansh.bhanu@research.iiit.ac.in
Phone Number	+91 9801917506

2. Benefits to Community

This project plays a crucial role in ensuring that ArviZ remains a powerful and user-friendly tool for Bayesian data analysis. By achieving feature parity in the new modular structure, we enhance the maintainability, extensibility, and efficiency of the library. The key benefits to the community include:

- **Improved Usability**: The reintroduction of essential visualization tools and diagnostics will provide researchers and practitioners with intuitive and informative plots, improving their workflow
- **Better Maintainability**: The modular refactoring makes ArviZ more structured and easier to contribute to, fostering a more active open-source community.
- **Enhanced Model Diagnostics**: By refining model comparison and sampling diagnostic tools, users can perform more reliable and interpretable Bayesian inference.
- **Increased Accessibility**: Improved documentation and visualization options will make it easier for new users to adopt ArviZ and integrate it into their Bayesian workflows.
- **Scalability and Future Growth**: The restructuring ensures that ArviZ can continue evolving with emerging probabilistic modeling techniques, making it future-proof for growing needs in Bayesian statistics.

This project will ultimately strengthen ArviZ's role as a go-to library for Bayesian model visualization and diagnostics, benefiting researchers, data scientists, and the broader statistical community.

3. Deliverables

3.1. Primary Goals

I aim to implement these goals during the course of GSoC this year.

3.1.1. Add MCSE

Visualize MCSE to quantify the variability in posterior estimates due to finite sampling, improving uncertainty assessment in Bayesian inference

3.1.2. Add Dark theme ("tumma")

Introduce a dark theme to enhance readability and usability of Bayesian diagnostic plots in low-light environments

3.1.3. Add Quantile dot plots

Represent distributions using discrete quantile dot plots, providing an intuitive alternative to KDE, ECDF, and histograms for distribution visualizations.

3.1.4. Add Plot_t_stats

Implement t-statistic plots to evaluate discrepancies between observed and simulated test statistics, aiding Bayesian model criticism.

3.1.5. Trace-plots for circular variables

Extend trace plots to handle circular variables (e.g., angles), ensuring accurate visualization of posterior samples in models with periodic parameters.

3.1.6. Rank-plots

Implement `plot_rank_dist` and implement paper 'Graphical test for discrete uniformity and its applications in goodness-of-fit evaluation and multiple sample comparison' to take comparison of multiple samples into consideration.

3.1.7. Parallel plot

Introduce parallel coordinate plots to analyze dependencies and interactions between multiple parameters in high-dimensional Bayesian models. This would help in special diagnosis of HMC (Hamiltonian Markov Chain)

3.1.8. Testing

I will be implementing unit tests for all the functionalities mentioned above.

3.1.9. Documentations

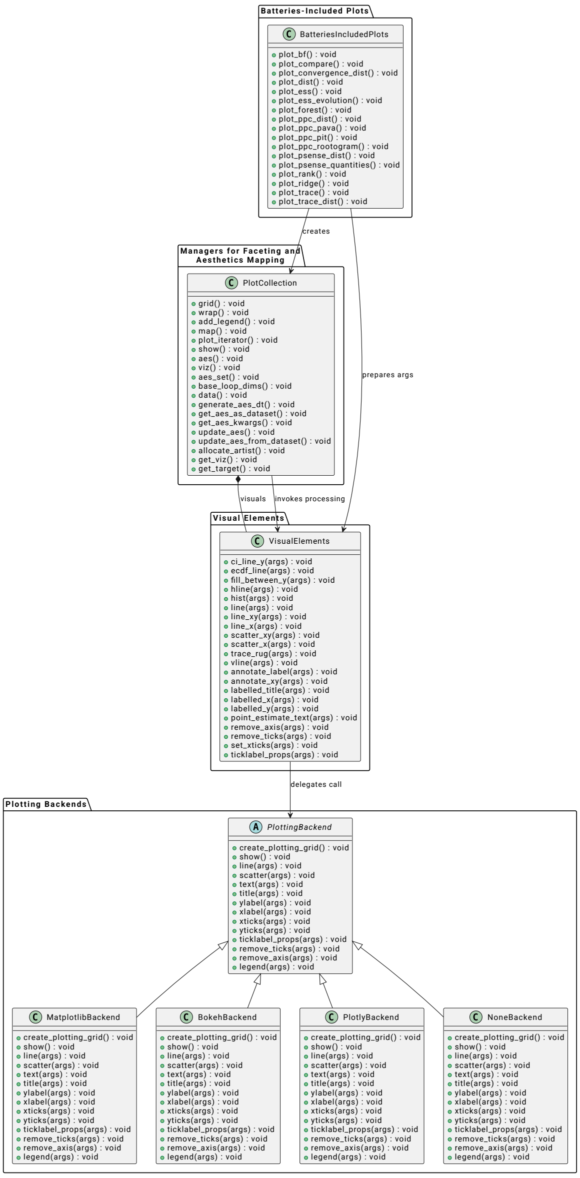
I will be doing documentations also for the features which I will be implementing.

3.2. Stretch Goals

If time permits, I would like to add more features to Arviz plots too which are/were not part of legacy version of Arviz

4. Project Details

4.1. Codebase



4.2. Add MCSE

4.2.1. Goal:

Goal is to have a diagnostic method which quantifies the uncertainty in estimates obtained from MCMC samples due to finite sampling. It will indicate the variability of statistics such as mean or quantiles due to sampling error.

4.2.2. Method description:

We would often want to know the error we are making while approximating the posterior, and we would like to have this error below some desired precision value in our results. So the information about this error is given by MCSE, it also takes into account the auto-correlation of samples similar to ESS.

4.2.3. Implementation:

- Complete PR 79 or start from scratch.
- Utilize ArviZ's statistical functions (arviz-stats) to compute MCSE.
- Add plot_mcse in batteries-included plots module of ArviZ-plots
- Ensure compatibility with xarray and multiple backends (Matplotlib, Plotly).

4.3. Add Dark theme ("tumma")

4.3.1. Goal:

Goal is to add dark theme to enable users to opt for it according to their preference. The theme selection framework is already implemented and there are already many other themes implemented to be opted, but dark theme is still missing so this project aims to implement it. Since other themes are already named in Catalan, Spanish, English etc. So the suggested name for this is "Tumma" which means dark in Finnish language.

4.3.2. Implementation:

- Base the theme on Plotly's dark mode or some other dark mode
- Use the "variat" palette with modifications to implement dark theme
- Since theme selection framework is already implemented, so the focus will be on Implementing dark theme only

4.4. Add Quantile dot plots

4.4.1. Goal:

Goal is to have another method to visualize distributions. Other methods like kde, ecdf and histograms are already implemented in all three backends. So we would add `Quantile Dot Plot` distribution visualization method to it.

4.4.2. Method description:

A Quantile Dot Plot displays the distribution of a sample in terms of its quantiles. It makes the reading of median or other quantiles easier. While choosing number of dots is subjective, we would keep default no. of dots as 100.

4.4.3. Implementation:

- Default to 100 dots per distribution.
- Implement as an option similar to kde, ecdf and hist (kind="dot").
- Ensure consistency with other distribution representations (kde, ecdf, hist).

- Efficiently handle varying sample sizes using xarray.
- Reference link (https://python.arviz.org/en/stable/examples/plot_dot.html)

4.5. Add Plot_t_stats

4.5.1. Goal:

Goal is to implement a separate method called `plot_t_stats` to compute test statistics computed from observations and predictive distributions. It will exclude p-values and u-values. Similar to plot_bpv_tstat, but focuses only on t-statistics (excludes p-values and u-values).

4.5.2. Method description:

For test statistics (which can be mean, median, std or any quantile etc), we would be computing

$$p(T(y)^* < T(y) \mid y)$$

, where T is any of the above mentioned test statistics. This way we would implement t-statistic plots to evaluate discrepancies between observed and simulated test statistics, aiding Bayesian model criticism.

4.5.3. Implementation:

- Extract relevant code from plot_bpv_tstat.
- Remove unrelated computations (p-value, u-value).
- Refactor to align with ArviZ-plots modularization.
- Ensure usability for model criticism tasks

4.6. Trace-plots for circular variables

4.6.1. Goal:

Currently the `plot_trace` implementation doesn't handle circular variables. So the goal is to improve present `plot_trace` to handle it.

4.6.2. Method description:

To handle circular variables we can implement circular histograms or annular wedge diagrams to better visualize the kde or histograms of these circular variables. For this we would need to add these new circular histograms or annular wedge methods in all three backends. The user will be provide information about circular variables by passing the name of circular variable through a new parameter of function named `circ_var_name`.

4.6.3. Implementation:

- Use plot_trace_circ as a reference.
- Adapt plot_trace to support circular variables.
- Normalize angular values (0 to 2π).
- Implement optional circular wrapping for visualization clarity.

4.7. rank-plots Enhancement and plot_rank_dist Implementation

4.7.1. Goal:

Goal is to implementing plot_rank_dist. Also Currently the envelope doesn't take comparison of multiple samples into consideration. The recent paper 'Graphical test for discrete uniformity and its applications in goodness-of-fit evaluation and multiple sample comparison' of 2021 discusses this, I have gone through it and will be implementing it.

4.7.2. Method description:

The paper mentioned above proposes new simulation- and optimization-based methods to obtain simultaneous confidence bands for the whole empirical cumulative distribution function (ECDF) of the PIT values under the assumption of uniformity. It further extends the simulation and optimization methods to determine simultaneous confidence bands for testing whether multiple samples come from the same underlying distribution. This multiple sample comparison test is useful as a complementary diagnostic in multi-chain Markov chain Monte Carlo (MCMC) convergence diagnostics.

4.7.3. Implementation:

- Implement plot_rank_dist, similar to plot_trace_dist.
- Ensure correct envelope for comparison of multiple samples (reference paper: https://link.springer.com/article/10.1007/s11222-022-10090-6)
- Additionally if time permits and if the requirement says so then I will be implementing plot_trace_rank and plot_trace_rank_dist as well. Which will be combination of rank_plots, trace_plots and distribution_plots which are already implemented.

4.8. Parallel plot

4.8.1. Goal:

The goal is to implement parallel plot for special diagnosis of HMC. It aims to visualize relationships between multiple parameters using parallel coordinate plots.

4.8.2. Method description:

- For general MCMC methods, simple comparisons between chains are often the best we can do.
- But if we focus on Hamiltonian Monte Carlo (HMC), we can gain much deeper insights into how well our model is performing.
- HMC works best when the shape of the probability distribution is smooth.
- If the geometry is too complex or has sharp curves, HMC struggles. In such cases, the leapfrog integrator starts to diverge from the expected trajectory.
- Hence we can visualize this divergence using parallel plot.
- Parallel plots display relationships between different parameters and can reveal if divergent samples follow a distinct pattern.
- If HMC is working well, parallel coordinate plots will not show a clear pattern of divergences.
- If HMC is struggling, divergences will be clustered, indicating a region of high curvature that obstructs exploration.
- We would do color coding to show divergent samples and non-divergent samples.
- (source: https://arxiv.org/pdf/1709.01449)

4.8.3. Implementation:

- Base implementation on existing example.
- Support visualization for all the backends.

5. Project Schedule

The proposed schedule has been made keeping in mind the GSoC timeline.

5.1. Community Bonding Period (May 8th - June 1st)

During this period I aim to,

- (**May 8th - May 12th**) : Go through the available PRs related to our project (eg: PR 79 for MCSE) and previous implementations in legacy ArviZ repo.
- (**May 13th - May 15th**): Figure out what are the components missing. Then decide to work either on those available codes or to start from scratch.
- (**May 16th - June 1st**): Add MCSE

5.2. First Coding Period (June 2nd - July 13th)

I will aim to complete the following project components during this period,

- (**June 2nd - June 12th**): Add Dark theme 'tumma'
- (**June 13th - June 19th**): Add quantile dot
- (**June 20th - July 2nd**): Add plot_t_stats
- (**July 3rd - July 13th**): Trace plots for circular variables

5.3. Midterm Evaluation (July 14th - July 17th)

During this period I aim to,

- (**July 14th - July 17th**): Go through documentations and tests of functionalities added and polish them further.

5.4. Second Coding Period (July 18th - Aug 24th)

I aim to complete the following components of project during this period,

- (**July 18th - Aug 10th**): rank-plots enhancement (includes Paper implementation which can take little more time, so gave extra time here)
- (**Aug 11th - Aug 20th**): parallel plots
- (**Aug 21th - Aug 24th**): Buffer time

5.5. Final Submission (Aug 25th - Sep 1st)

Buffer time,

- (**Aug 25th - Sep 1st**): Buffer time to handle backlogs (if there exists), documentations and test polishing

5.6. Extension

If required and seems necessary from Mentor's side, then the timeline can be extended as well.

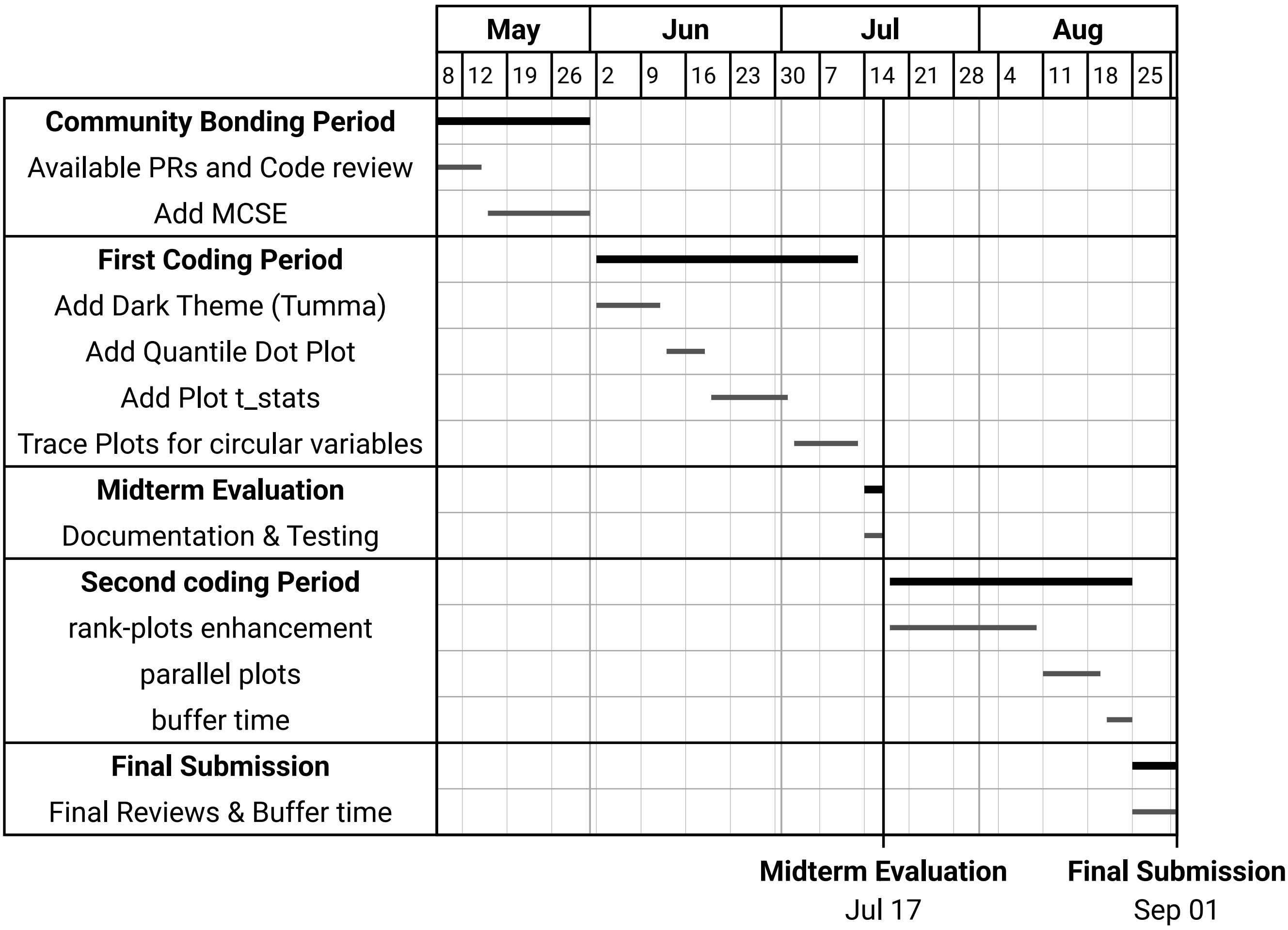
5.7. Availability

I will be available throughout the mentioned periods. Around last June or start of July I will be travelling for 4 days. So I might be unavailable for those 4 days since I won't have access to my laptop, otherwise I will be there for full GSoC period.

5.8. Plans Beyond GSoC

I plan to work further in the space of Bayesian Modelling & analysis of Bayesian Models after this project. I would love to be able to continue to be a part of ArviZ after GSoC as well.

6. Gantt Chart


May
Jun
Jul
Aug
Community Bonding Period
Available PRs and Code review
Add MCSE
First Coding Period
Add Dark Theme (Tumma)
Add Quantile Dot Plot
Add Plot t_stats
Trace Plots for circular variables
Midterm Evaluation
Documentation & Testing
Second coding Period
rank-plots enhancement
parallel plots
buffer time
Final Submission
Final Reviews & Buffer time
Midterm Evaluation
Jul 17
Final Submission
Sep 01