GSOC 2017 – Numfocus (Matplotlib)

W M D Kanchana N Ranasinghe

1 Categorical Color

1.1 Abstract

Categorical information (non-numerical like gender, favorite food, etc.) is quite common in datasets that we analyze. However Matplotlib support for plotting data is available mostly for numerical data, with few direct APIs that can automatically detect categorical data and support it. This project attempts to build an API for plotting heat-maps and other scalarMappables using categorical data. It will also work on developing normalization and color-map APIs supporting categorical data as these will be dependencies for the main API.

Currently, users wanting to plot categorical data with color support in Matplotlib must manually map their categorical data to numbers, creating non-existent links between classes and requiring a lot of additional effort. The proposed APIs will automatically analyze categorical data and plot heat-maps or provide color-map/norm without no additional user interference being required.

Intensity	**Involves**	**Mentors**
Intermediate	Python, Data science	[@tacaswell][@story645]

1.2 Technical Details

With the widespread presence and usage of large datasets of varied nature, the analysis of categorical data has received much attention. Most machine learning, big-data analysis, and data-mining research and projects require the datasets available to be visualized at various stages of work. A variety of plotting techniques are utilized for this matter, and Matplotlib plays a prominent role as an essential tool for many researchers. Despite its diverse functionality with regard to numerical data visualization, Matplotlib is quite restricted in the domain of categorical data analysis. In most cases, users are compelled to manually map their categorical data to numerical data, thus introducing unnecessary relationships and patterns to the datasets, creating redundant data, and being withheld from the simplicity and ease of usage inherent to Matplotlib.

This work hopes to tackle this issue to a certain extent by creating APIs to automatically identify categorical data and support them. The Categorical Color Project will include two main tasks:

1) working on heat-map and other scalarMappables APIs
2) working on normalization and color-map APIs.

A main aim of this project is to integrate unit support into these APIs without subjecting them to any direct changes. Matplotlib currently has these frameworks set-up with functionality for numerical data. This functionality will be extended to support categorical data. The work of this project will be done solely using Python. Libraries used will include Numpy and Matplotlib. There will be no new (unused in Matplotlib previously) third-party libraries used in the course of this project.

1.3 Proposed Solution

Categorical data mostly includes pandas categorical data types alongside numpy structured arrays and python dictionaries. Currently, APIs for plotting heat-maps and obtaining color-map/norm of a dataset directly support only numerical data due to their limited units understanding. All plotting requires a norm function to map data to relevant colors. Currently, it requires data to be continuous as well. We hope to overcome this through a direct integration of categorical normalization with the discrete data color-maps currently supported. Changes to the units conversions occurring when analyzing categorical data will also be looked into. These will provide the basic background for the norm and color-map APIs. Our final API will make it possible for users to easily obtain norm/color-map for categorical data directly as below.

```
data = getCategoricalDataFromPandas()
cmap, norm = categorical_colors_API(data)
```

With regards to heat-map generation we hope to incorporate similar units conversion knowledge to the existing APIs to support categorical data. In addition, upon integration with categorical norm and color-map APIs, the imshow function will be able to identify categorical data automatically and display adequate plots in a manner similar to how it analyses numerical data.

```
data = getCategoricalDataFromPandas()
fig, ax = matplotlib.pyplot.subplots()
ax.imshow(data)
```

Also this categorical support will be extended to support colorbars and legends for the imshow function with regards to categorical data. This will take a similar approach.

1.4 Schedule of Deliverables

1.4.1 May 1st - May 28th, Community Bonding Period

- Working on understanding units support extension for categorical data
- Further understanding core Matplotlib internals and their interconnections
- Self-implementation of previous work done to support categorical color for better understanding
- Setting up of project blog covering all major technical details
- Working on MEPs in Matplotlib Developers' Guide, and Matplotlib issues related to normalization, color-map, and color-bar code.

1.4.2 May 29th - June 3rd

- extending units support (categorical data) for heat-maps

This is the starting point for the project. The units framework provides support for custom instances to be converted to values compatible with Matplotlib. The custom instances discussed are those of classes that can convert themselves to arrays despite lacking the array interface. Matplotlib currently supports this for certain plots (bar charts/line plots/scatter plots) drawn from categorical data. In this stage, we focus on heat-maps (imshow function) as we attempt to allow categorical data to be converted to Matplotlib compatible values, generating necessary locators and ticks for this task. This work will be built off existing development discussed in 7383.

1.4.3 June 5th - June 9th

- review/document units support extension
- tests/bug fixing units support extension

1.4.4 June 12th - June 16th

- framework outline for norm functionality extensions
- norm basic functionality including tests and documentation

Based on the units support, the next step is normalization support for categorical data. With regards to the normalization function, in a Matplotlib sense, it encompasses a function mapping different data values to different colors (or color levels). Currently, the different data values to be mapped from are expected to be numerical. Also it requires data to be continuous. This limitation will be worked around to enable support for categorical data. Also during the course of the project focus will be given to the currently available discrete normalization methods (matplotlib.colors.BoundaryNorm) which handle data in a different sense which may be used as a foundation for this additional unit support.

1.4.5 June 19th - June 23th, End of Phase 1

- framework outline for cmap functionality extensions
- cmap basic functionality including tests and documentation

Currently, the process of using matplotlib.pyplot for heat-maps and other scalarMappables includes the creation of a color-map followed by normalization of all available data to fit this color-map. Color-maps plotting functionality is threefold: sequential, divergent and qualitative. Considering the intrinsic discreteness of categorical data, the extension of the color-maps API will focus on its qualitative plotting ability. However the fundamental behavior of the color-maps API will not require any changes for handling categorical data as our approach simply involves the integration of unit support. Therefore it is possible to build this extended support for categorical data while keeping the public API intact. The same applies to work done on norm API.

1.4.6 June 26 - June 30th, Begin of Phase 2

- finalization of categorical norm and cmap APIs
- testing/bug-fixing

1.4.7 July 3rd - July 7th

- imshow basic functionality (extending plots interface) including tests and documentation
- integration of units support

The previously setup units support for heat-maps will be linked to the imshow function for further development. The work of this project will stem from existing development done for categorical color support in Matplotlib issues 6889 and 6934. The starting point for this stage will be extending the imshow function (matplotlib.axes._axes.imshow) to support categorical data by introducing necessary units support. This mainly focuses on providing it unit knowledge with regards to non-numerical data followed by extending the plots interface (matplotlib.pyplot) to accommodate categorical data.

1.4.8 July 10th - July 14th

- color-bar support for imshow

The color bar is mapped from numerical data for heat-maps currently, creating boundaries for each class. Due to discrete nature of categorical data, work may stem from ListedColormaps, and that will be used as a foundation for work. The boundaries required for each distinct stage will be extended to support non-numerical values. Work will be done to overcome the current necessity for continuity of data to be

mapped from (categorical data is discrete). This will focus on the non-numerical values of categorical data being directly mapped to each distinct color level with the developed units integration.

1.4.9 July 17th - July 21th, End of Phase 2

- legend support for imshow

This is an additional improvement attempted. Since heat-maps do not currently have a direct method for implementation of a legend, this will focus on extending the legend support of other scalarMappables. Also the categorical support for legend here will mostly be for the purpose of serving as a foundation for legend support for other scalarMappables.

1.4.10 July 24th - July 28th, Begin of Phase 3

- testing/bug-fixing of heatmaps API (imshow)
- framework outline for categorical color support of other scalarMappables

Extension of current work (support for categorical data) to other scalarMappables like matshow, pcolor, pcolormesh, scatter, etc. will be done here. The scalarMappables class will be used as the starting point to implement this as all instances stem from here. Support for color-bars and legends for these other scalarMappables will also be given focus. The work here will not be direct extensions of heat-maps and may require different approaches to extend the units support for these scalarMappables.

1.4.11 July 31st - August 4th

- basic functionality of other scalarMappables

The work here will be done on the scalarMappables class and its underlying objects. This will ensure that all support for categorical data integrated flows down to each derived type of scalarMappables.

1.4.12 August 7th - August 11th

- color-bar and legend support for other scalarMappables

Work similar to what is done for heat-maps will be extended to the scalarMappables class to support color-bars and legends for categorical data.

1.4.13 August 14th - August 18th

- overall functionality of other scalarMappables with tests and documentation
- testing and debugging of other scalarMappables code

1.4.14 August 21st - August 25th, Final Week

- review of all documentation

1.4.15 August 28th - August 29th, Submit final work

- code submission
- discussion of possible future developments on blog

1.5 Future works

- Improving the APIs built by considering different approaches for categorical data integration

1.6 Development Experience

I have been involved in programming work in python, MATLAB, Lua, Java, C and Micro C over the past three years. Some of my work is on github. With regards to open source development, I have worked on a couple of issues on matplotlib and scikitlearn over the past few weeks. In addition, I was able to get involved in this natural language processing related project for CLTK on Pali Language (mainly because I am familiar with the language). The links of work done are below.

https://github.com/matplotlib/matplotlib/pull/8371#event-1015235492

https://github.com/matplotlib/matplotlib/pull/8357

https://github.com/scikit-learn/scikit-learn/pull/8558#pullrequestreview-26737369

https://github.com/kahnchana/Pali-NLP

1.7 Other Experiences

I have been involved in some machine learning and computer vision related research work, especially regarding activity recognition in videos. I have also worked on some data mining projects related to machine learning. Currently I am involved in some work related to dense trajectories. I have also been involved in numerous mathematics related work since an early age, having also represented my country at the International Mathematical Olympiad while in high school.

1.8 Why this project?

I am quite interested in the areas of machine learning, computer vision, and robotics. As part of a machine learning research group at my university, I have constantly used Matplotlib for various purposes, including visualizing of certain categorical data. So I feel that I should try to get involved in contributing to the community as well. Also I feel a lot familiar with this project and find it interesting to work on mainly due to its relevance to data science.

1.9 Appendix

1.9.1 About Me

I am a second-year undergraduate student at the University of Moratuwa, Sri Lanka studying Electronics and Telecommunication Engineering. I have been using python for the last three years for development and research work and am quite familiar with Numpy and Matplotlib libraries. Having taken multiple courses in numerical analysis, calculus, programming, machine learning, and data-science, I am knowledgeable on the areas related to this project as well.

1.9.2 Contact Details

W M D Kanchana N Ranasinghe

kahnchana@gmail.com / 150507H@uom.lk

kahnchana@github.com

1.10 Availability

I will be having summer holidays from the first week of June till the first week of September so I will be able to commit completely towards GSoC during that period.

- Time zone: Sri Lanka Standard Time (SLST) - UTC +05:30
- Hours per week: 35 – 40 hours (except for first week of June – roughly 30 hours)