The Journal of Open Source Software

itrm: Interactive Terminal Utilities

David Woodburn [1]

1 Air Force Institute of Technology, Dayton, Ohio, United States of America

DOI: 10.21105/joss.07417

Software

- Review
- Repository
- Archive

Editor: Marcel Stimberg

Reviewers:

- @domoritz
- @DevangThakkar

Submitted: 02 October 2024
Published: 22 November 2024

Summary

The `itrm` library provides a terminal-based solution for interactively analyzing data. While it does have facilities for progress bars, heat maps, bar charts, and tables, its primary feature is interactive data plots. It will render as a dot matrix of Unicode Braille characters (or ASCII text if needed) your (x,y) data. Using a vertical cursor, you can inspect data points, zoom in on a region, or scrub through the data. With simple 2 to 3-character keybindings, you can run a variety of transforms on the data: absolute value, autocorrelation, Allan variance, differentiation, integration, Fourier transform, smoothing, detrending, trimming, etc. You can leave behind a ghost of the cursor and then analyze the data between the ghost position and current position of the cursor. It can plot multiple data sets simultaneously with automatic color differentiation, and it allows the user to focus on just one data set at a time, even hiding all others if desired.

This library is written in pure Python, and aside from some builtin modules has only NumPy as a dependency. It is optimized to run smoothly on data sets of up to at least a million points. And, it has been built to run on all platforms (Linux, macOS, Windows) and any fully-implemented terminal (i.e., not Jupyter notebooks). It will even run on Windows' Console and VS Code's builtin terminal.

This library can be installed with `pip install itrm` and apart from builtin libraries has only Numpy as a dependency. Examples using this library can be found here.

Statement of need

Many developers, engineers, and scientists spend significant time working within terminal environments, where switching contexts to generate visual plots can be both time-consuming and annoying. Traditional plotting tools, while effective for producing polished final figures, are often insufficient for quickly inspecting and understanding data. Curious to see what the frequency content of the data is? Want to filter out the high-frequency noise so you can see a pattern more clearly? Interested in the derivative of the data? All these will typically require you go back to your source and write several lines of code before you can see the results. This challenge is amplified when working remotely with a server, where visualizing data often requires saving it to a file, transferring it to a local machine, and writing scripts to generate plots—a tedious and repetitive process. A more efficient solution is needed to streamline data visualization and interaction directly within the terminal.

There are already several terminal-based plotting tools available, several of them written for Python. The following table summarizes a non-exhaustive list of these tools. The `Line plots` column refers to whether the library provides for plots of (x,y) data. The `Y-axis` column refers to whether the y value can be read off by the user. The `Resolution` column refers to the character set used to print to the terminal: Braille (4 by 2 matrix of dots) would be `high`, half blocks (2 by 1 Unicode drawing characters) would be `half`, and asterisks (`*`) would be `low`.

Name	Author	Language	Line plots	Y-axis	Resolution
`asciichartpy`	Kroitor (2020)	Python	yes	yes	low
`bashplotlib`	Lamp (2019)	Python	yes	no	low
`drawille`	Tauber (2020)	Python	no	no	high
`pipeplot`	Chistyakov (2020)	Python	no	no	low
`plotext`	Piccolomo (2022)	Python	yes	limited	high
`plotille`	Ippen (2022)	Python	yes	limited	high
`pysparklines`	Whaley (2021)	Python	no	no	high
`termgraph`	Kazmierczak (2021)	Python	no	yes	high
`terminalplot`	Kressibucher (2023)	Python	yes	no	low
`termplot`	Powell (2018)	Python	no	no	low
`termplotlib`	Schlömer (2021)	Python	yes	limited	low
`uniplot`	Stetter (2024)	Python	yes	limited	half
`gnuplot`	Williams et al. (2024)	C	yes	limited	low
`termeter`	Sasaki (2015)	Go	yes	limited	high
`ttyplot`	Sawicki (2024)	C	no	limited	low
`youplot`	Tools (2024)	Ruby	yes	limited	high

None of these tools is interactive in the sense of providing data point inspection, view zooming and panning, point-to-point comparisons, etc. And, certainly none of them provide the ability to process the data with common transforms on the fly. Only a few of them provide sub-character resolution through the use of Braille characters.

References

Chistyakov, I. (2020). Pipeplot. In *GitHub repository*. GitHub. https://github.com/MyGodIsHe/pipeplot

Ippen, T. (2022). Plotille. In *GitHub repository*. GitHub. https://github.com/tammoippen/plotille

Kazmierczak, M. (2021). Termgraph. In *GitHub repository*. GitHub. https://github.com/mkaz/termgraph

Kressibucher, M. (2023). Terminalplot. In *GitHub repository*. GitHub. https://github.com/kressi/terminalplot/

Kroitor, I. (2020). Asciichartpy. In *GitHub repository*. GitHub. https://github.com/kroitor/asciichart

Lamp, G. (2019). Bashplotlib. In *GitHub repository*. GitHub. https://github.com/glamp/bashplotlib/tree/master

Piccolomo, S. (2022). Plotext. In *GitHub repository*. GitHub. https://github.com/piccolomo/plotext

Powell, W. (2018). Termplot. In *GitHub repository*. GitHub. https://github.com/iswdp/termplot

Sasaki, A. (2015). Termeter. In *GitHub repository*. GitHub. https://github.com/atsaki/termeter

Sawicki, A. (2024). Ttyplot. In *GitHub repository*. GitHub. https://github.com/tenox7/ttyplot

Schlömer, N. (2021). Termplotlib. In *GitHub repository*. GitHub. https://github.com/nschloe/termplotlib

Stetter, O. (2024). Uniplot. In *GitHub repository*. GitHub. https://github.com/olavolav/

uniplot

Tauber, A. (2020). Drawille. In *GitHub repository*. GitHub. https://github.com/asciimoo/drawille

Tools, R. D. (2024). YouPlot. In *GitHub repository*. GitHub. https://github.com/red-data-tools/YouPlot

Whaley, B. (2021). Pysparklines. In *GitHub repository*. GitHub. https://github.com/RedKrieg/pysparklines

Williams, T., Kelley, C., Lang, R., Kotz, D., Campbell, J., Elber, G., Woo, A., & others. (2024). *Gnuplot*. http://www.gnuplot.info/