Corner.py: The Covariance Corner

By Talya Lebson

5/12/2025

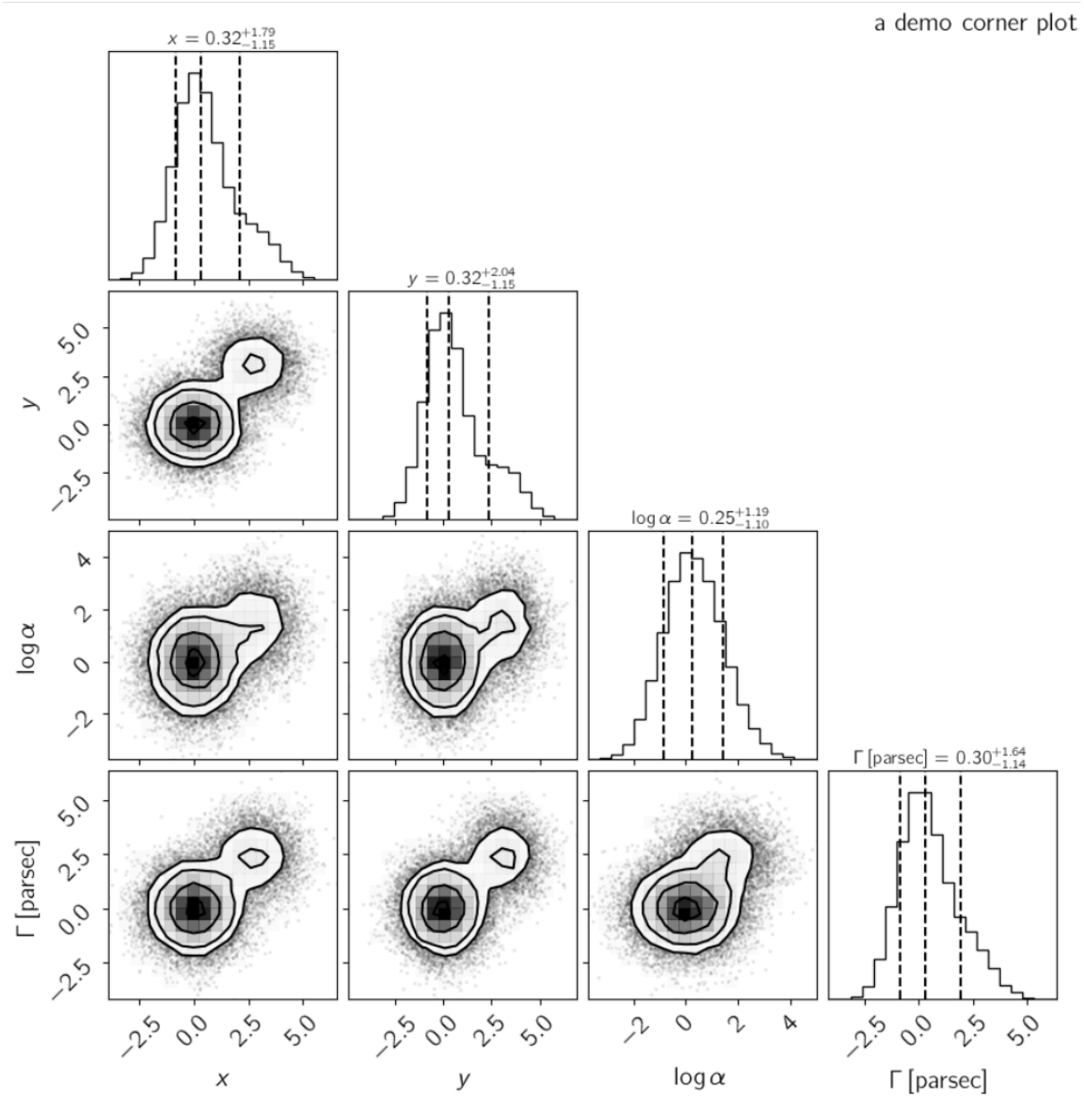
1. The name of the package is corner.py: Corner Plots with the version number 2.2.3. The basic aim is to make a scatterplot matrix for multidimensional samples to help reveal covariances using matplotlib. Corner.py was originally made to show the results of the Markov Chain Monte Carlo. A scatterplot matrix is a way to visualize many scatter plots, which shows the relationship between different variables. A covariance matrix is important for understanding how different variables change together and depend on each other.
2. I selected this package because it is related to a recent lecture about Monte Carlo's, which was interesting to me. I am also interested in learning more about covariances and how they work. I know it is important to understand how variables change together when analyzing data, so I am hoping this helps me understand that better.
3. This package was created in June 2016 by Daniel Foreman-Mackey. This code uses matplotlib, which was made in 2003. This code was originally named triangle.py, but later its name was changed to corner.py. The emcee: The MCMC Hammer was created in 2013 and is a Markov chain Monte Carlo ensemble sampler. This is often used with corner.py.
4. It is still maintained by the original authors. The most recent edit was made on two weeks ago. There are instructions on GitHub (https://github.com/dfm/corner.py/blob/main/src/corner/core.py) on how to contribute to the project. These instructions involve how to report a bug, how to request a feature, how to submit changes, and how to test the project.
5. It was very easy to install and use. I used !pip install corner to install, and it worked very easily.
6. It was installed via the "standard" pip method.
7. The source code is available in the corner.py GitHub repository. Here is the link to the repository: https://github.com/dfm/corner.py.
8. Corner.py is often used alongside the emcee package. It also appears in the ASCL. No other package was found to use corner.py within it.
9. The code works in a Jupyter notebook and also in scripts.
10. Here is an example using the code based on an example provided by the program creators on GitHub. This code visualizes Gaussian samples with multiple variables:

```
import numpy as np
```

```
from matplotlib import rcParams
import corner

rcParams["font.size"] = 16
rcParams["font.family"] = "sans-serif"
rcParams["font.sans-serif"] = ["Computer Modern Sans"]
rcParams["text.usetex"] = True
rcParams["text.latex.preamble"] = r"\usepackage{cmbright}"

np.random.seed(99)

# Set up the parameters of the problem.
ndim, nsamples = 4, 30000

# Generate some fake data.
data1 = np.random.randn(ndim * 4 * nsamples // 5).reshape(
    [4 * nsamples // 5, ndim]
)
data2 = 4 * np.random.rand(ndim)[None, :] + np.random.randn(
    ndim * nsamples // 5
).reshape([nsamples // 5, ndim])
data = np.vstack([data1, data2])

# Plot it.
figure = corner.corner(
    data,
    labels=[
        r"$x$",
```

```
        r"$y$",
        r"$\log \alpha$",
        r"$\Gamma \, [\mathrm{parsec}]$",
    ],
    quantiles=[0.16, 0.5, 0.84],
    show_titles=True,
    title_kwargs={"fontsize": 12},
)
figure.gca().annotate(
    "a demo corner plot",
    xy=(1.0, 1.0),
    xycoords="figure fraction",
    xytext=(-20, -10),
    textcoords="offset points",
    ha="right",
    va="top",
)
figure.savefig("demo.png", dpi=300)
```

11. The package produces figures along with matplotlib. Corner itself uses matplotlib and generates plots with built-in customization.
12. Here is a figure created in corner.py:



This is a figure of a corner plot with randomly generated Gaussian data with 4 dimensions. The histograms on the diagonal show the marginal distribution for each variable, and the off-diagonal circles show the covariance between the parameters. The elliptical contours show the correlation strength.

13. The package is pure Python that depends on numpy and matplotlib.
14. The package takes in data, labels, quantiles, show_title=True, and title_kwargs. The example from the GitHub used np.random to generate random data that used parameters.

15. The output of the package is a figure plot of the covariance between parameters and the marginal distribution for each variable.
16. The code provides plotting but does not contain any unit tests, regression, or benchmarking.
17. The plot appears to be a valid representation of a covariance matrix. Additionally, it is widely used in peer review work.
18. The package uses numpy to produce random data, and from matplotlib it imports rcParams, which is used to annotate the figures.
19. The package provides a GitHub repository, which includes examples of how to use the code. On top of that, there is a website https://corner.readthedocs.io/en/latest/ that explains all the parameters used in corner.py. This was sufficient for me to understand this package.
20. They provide a preferred citation: @article{corner,

    doi = {10.21105/joss.00024},

    url = {https://doi.org/10.21105/joss.00024},

    year = {2016},

    month = {jun},

    publisher = {The Open Journal},

    volume = {1},

    number = {2},

    pages = {24},

    author = {Daniel Foreman-Mackey},

    title = {corner.py: Scatterplot matrices in Python},

    journal = {The Journal of Open Source Software}

    }

21. No other references were used in my report.
22. Corner.py has been used in research articles such as The Population of Long-period Transiting Exoplanets (DOI: 10.3847/0004-6256/152/6/206) which is about the period of long-term transitioning exoplanets, and No Conclusive Evidence for Transits of Proxima b in MOST Photometry (DOI: 10.3847/1538-3881/153/3/93) which is about observing transits of proxima B.

23. I had to install LaTeX on my computer to use their example, but other than that, the class was sufficient for me to use this package.
24. I had no other prior experience with this data. I worked with Hannah Ketner on this project.