

Corner.py: Turning the Corner

[1] The corner.py package is a software intended to aid in the display of many types of data samples, particularly high-dimensional data. Corner.py can also help in understanding relationships between various parameters, like those used in Bayesian statistics. The most recent version of corner.py is listed under the “releases” section on the corner.py GitHub as v2.2.3, which is what version I used for my analysis of this package. Corner.py uses matplotlib to visualize multidimensional data samples using a scatterplot matrix to help reveal covariances. Corner.py was originally created to display the results of Markov Chain Monte Carlo simulations.

[2] The Monte Carlo simulation was one of the most recent topics that we discussed during lecture, so it was still fresh and interesting to me. Corner.py is useful when observing convergence and spread in MCMC chains, which is how it is relevant to Monte Carlo simulation.

[3, 4] This package was created in June of 2016 by Daniel Foreman-Mackey at the University of Washington. It is still maintained by Foreman-Mackey. There is a contribution guide on the corner.py Github under CONTRIBUTING.md, where contributors are encouraged to send pull requests for necessary changes to the package. A related code involved is matplotlib, which was made in 2007. The MCMC Hammer code, which corner.py was intended to help show the simulations of, originated in 2013.

[5, 6] The installation process involved doing the standard `!pip install corner`, which was fairly straightforward and easy to accomplish. The installation process was through the “standard” pip protocol.

[8] The source code is publicly available on the corner.py GitHub repository, under the “core.py” file.

[9] While corner.py can be utilized in conjunction with other packages, such as scipy, it is not explicitly used by other packages.

[10, 11, 12] My example code for corner.py is used through Jupyter Notebook. The accompanying notebook shows corner.py being used to plot random data created through `np.random` as 1D histogram plots and 2D scatter plots. Matplotlib is used for annotating plots,

Hannah Ketner

5/9/2025

while corner.py handles the creation of figures.

```
# tweaked example
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)
```

Figure 1: The above image shows the code used for the generation of random data and the use of corner.py to plot the data as 1D histogram plots and 2D scatter plots.

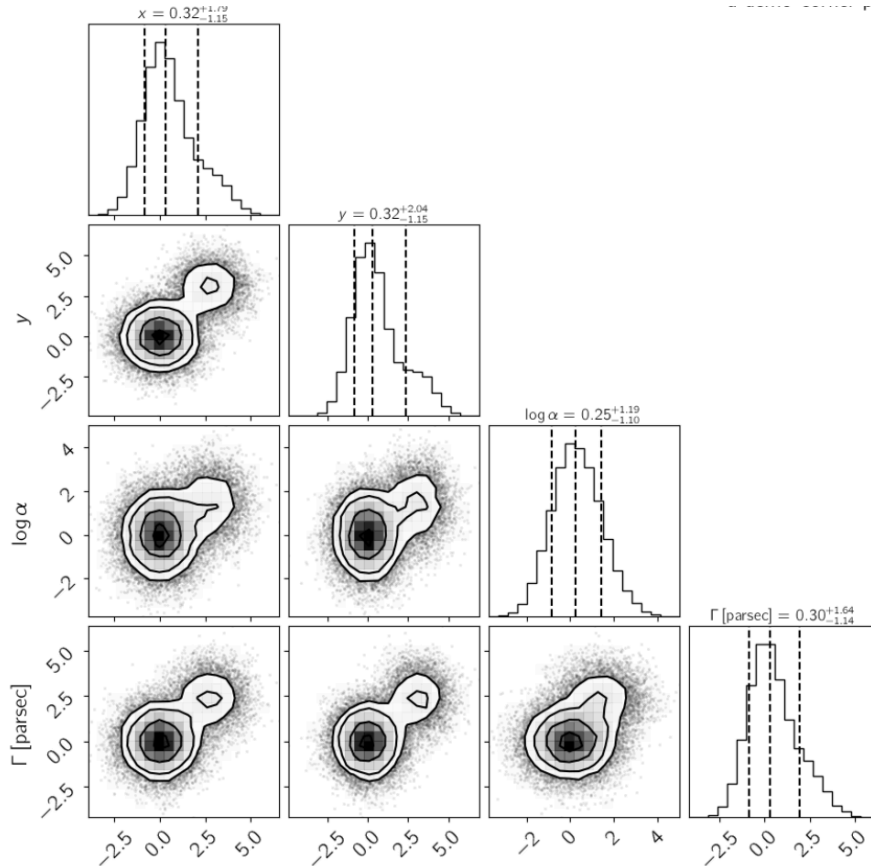


Figure 2: The above image shows the output plots created through `corner.py` of the 1D histogram plots and 2D scatter plots.

[13] The package `corner.py` is purely Python, without any need for accompanying C/C++/Fortran code.

[14] The package `corner.py` takes in data, labels, quantiles, kwargs, and titles. The code example uses `np.random` to generate random data to be plotted, meaning that the data is generated from scratch. `Corner.py` can also use datasets such as those used in Bayesian statistics.

[15] The output of the package are 1D histogram plots and 2D density plots, with optional annotations such as parameter labels and quantile lines.

[16] The code provides use with plotting, but it does not contain elements to support unit tests, regression, or benchmarking.

Hannah Ketner

5/9/2025

[17] You can determine the sensibility of the plots by printing the mean and standard deviation of the data by using `print(np.mean(data))` and `print(np.std(data))`, then seeing how closely those values are represented by the plots. Additionally, ensuring that the standard data percentiles are matched by the distribution can make the user confident that the package is working as intended.

[18] This code example relies upon numpy, particularly `np.random` for the code example to produce random data, as well as matplotlib for supporting annotations to the figure.

[19] There is a GitHub repository for the `corner.py` package that serves as documentation, as well as providing example code that could be modified. Additionally, there's a webpage `corner.readthedocs.io`, where every parameter used in `corner.py` is explained fully.

[20] The GitHub repository for `corner.py` requests that the JOSS paper is cited in this format:

```
@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] References:

Hannah Ketner

5/9/2025

- corner.py GitHub: <https://github.com/dfm/corner.py>
- corner.py ASCL: <https://ascl.net/1702.002>
- corner.py readthedocs.io: <https://corner.readthedocs.io/en/latest/>

[22] Corner.py has been used in research articles such as No Conclusive Evidence for Transits of Proxima b in Most Photometry (DOI: 10.3847/1538-3881/153/3/93), where the transits of proxima b are observed, and The Population of Long-period Transiting Exoplanets (DOI: 10.3847/0004-6256/152/6/206), which is about the population of long-period transiting exoplanets.

[23] To run the corner.py code, the installation of Latex was required, which could not easily be done through a pip install command. A TeX distribution package had to be installed to use the Latex components of the code.

[24] I have had no prior experience with this package. This project was a collaboration between me and Talya Lebson.