The Journal of Open Source Software

# Grama: A Grammar of Model Analysis

**Zachary del Rosario**[1]

**1** Visiting Professor, Olin College of Engineering

**DOI:** 10.21105/joss.02462

**Software**

- Review
- Repository
- Archive

**Editor:** Matthew Sottile

**Reviewers:**

- @BastinRobin
- @rodrigokataishi

**Submitted:** 05 July 2020
**Published:** 28 July 2020



## Summary

`Grama` is a Python package implementing a *functional grammar of model analysis* emphasizing the quantification of uncertainties. In `Grama` a *model* contains both a function mapping inputs to outputs as well as a distribution characterizing uncertainties on those inputs. This conceptual object unifies the engineer/scientist's definition of a model with that of a statistician. `Grama` provides an *implementation* of this model concept, as well as *verbs* to carry out model-building and model-analysis.

### Statement of Need

Uncertainty Quantification (UQ) is the science of analyzing uncertainty in scientific problems and using those results to inform decisions. UQ has important applications to building safety-critical engineering systems, and to making high-consequence choices based on scientific models. However, UQ is generally not taught at the undergraduate level: Many engineers leave their undergraduate training with a purely deterministic view of their discipline, which can lead to probabilistic design errors that negatively impact safety (del Rosario, Fenrich, & Iaccarino, 2020). To that end, I have developed a grammar of model analysis—`Grama`—to facilitate rapid model analysis, communication of results, and the teaching of concepts, all with quantified uncertainties. Intended users of `Grama` are scientists and engineers at the undergraduate level and upward, seeking to analyze computationally-lightweight models.

### Differentiating Attributes

Packages similar to `Grama` exist, most notably Sandia National Lab's `Dakota` (Adams, 2017) and `UQLab` (Marelli & Sudret, 2014) out of ETH Zurich. While both of these packages are mature and highly featured, `Grama` has several differentiating attributes. First, `Grama` emphasizes an explicit but flexible *model object*: this object enables sharp decomposition of a UQ problem into a model-building stage and a model-analysis stage. This logical decomposition enables simplified syntax and a significant reduction in boilerplate code. Second, `Grama` implements a functional programming syntax to emphasize operations against the model object, improving readability of code. Finally, `Grama` is designed from the ground-up as a pedagogical and communication tool. For learnability: Its *verb-prefix* syntax is meant to remind the user how functions are used based solely on their name, and the package is shipped with fill-in-the-blank Jupyter notebooks (Kluyver et al., 2016) to take advantage of the pedagogical benefits of active learning (Freeman et al., 2014). For communication: The model object and functional syntax abstract away numerical details for presentation in a notebook, while preserving tracability and reproducibility of results through the inspection of source code.

### Inspiration and Dependencies

`Grama` relies heavily on the SciKit package ecosystem for its numerical backbone (Hunter, 2007; McKinney, 2010; Pedregosa et al., 2011; van der Walt, Colbert, & Varoquaux, 2011; Virtanen et al., 2020). The functional design is heavily inspired by the `Tidyverse` (Wickham et al., 2019), while its implementation is built upon `dfply` (Katovich, 2019). Additional functionality for materials data via an optional dependency on Matminer (Ward et al., 2018).

## Acknowledgements

I acknowledge contributions from Richard W. Fenrich on the laminate plate model.

## References

Adams, B. M. (2017). *Sandia capabilities for uncertainty quantification.* Sandia National Lab.(SNL-NM), Albuquerque, NM (United States).

del Rosario, Z., Fenrich, R. W., & Iaccarino, G. (2020). When are design allowables conservative? In *AIAA SciTech 2020 Forum* (p. 0414). doi:10.2514/6.2020-0414

Freeman, S., Eddy, S. L., McDonough, M., Smith, M. K., Okoroafor, N., Jordt, H., & Wenderoth, M. P. (2014). Active learning increases student performance in science, engineering, and mathematics. *Proceedings of the National Academy of Sciences*, *111*(23), 8410–8415. doi:10.1073/pnas.1319030111

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science Engineering*, *9*(3), 90–95. doi:10.1109/MCSE.2007.55

Katovich, K. (2019). Dfply. *GitHub repository*. https://github.com/kieferk/dfply; GitHub.

Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B. E., Bussonnier, M., Frederic, J., Kelley, K., et al. (2016). Jupyter notebooks-a publishing format for reproducible computational workflows. In *ELPUB* (pp. 87–90). doi:10.3233/978-1-61499-649-1-87

Marelli, S., & Sudret, B. (2014). UQLab: A framework for uncertainty quantification in matlab. In *Vulnerability, uncertainty, and risk: Quantification, mitigation, and management* (pp. 2554–2563). doi:10.1061/9780784413609.257

McKinney. (2010). Data Structures for Statistical Computing in Python. In Stéfan van der Walt & Jarrod Millman (Eds.), *Proceedings of the 9th Python in Science Conference* (pp. 56–61). doi:10.25080/Majora-92bf1922-00a

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., et al. (2011). Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research*, *12*, 2825–2830.

van der Walt, S., Colbert, S. C., & Varoquaux, G. (2011). The numpy array: A structure for efficient numerical computation. *Computing in Science Engineering*, *13*(2), 22–30. doi:10.1109/MCSE.2011.37

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., et al. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, *17*, 261–272. doi:10.1038/s41592-019-0686-2

Ward, L., Dunn, A., Faghaninia, A., Zimmermann, N. E., Bajaj, S., Wang, Q., Montoya, J., et al. (2018). Matminer: An open source toolkit for materials data mining. *Computational Materials Science*, *152*, 60–69. doi:10.1016/j.commatsci.2018.05.018

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., Grolemund, G., et al. (2019). Welcome to the tidyverse. *Journal of Open Source Software*, *4*(43), 1686. doi:10.21105/joss.01686