Project Milestone Progress Report

Project #18 - Group 208 - Hugo Dupouy (GT903738077)

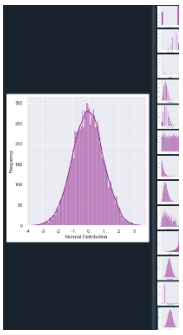
Project Progress

We have planned to make a package of Python's discrete and continuous routines for this project. Below, we have listed the distributions we did and where we are. We defined the goal, did the literature review, and did 40% of the main findings in terms of progress. We will keep the conclusion and introduction for the end. In the main part, we generated the code for each distribution based on Scipy Stats and generated the graphs through Seaborn. Here is an example below:

```
## Uniform

from scipy.stats import uniform
data_uniform = uniform.rvs(loc=start, scale=width, size=n)

graph = sns.displot(data_uniform,
                    bins=100,
                    kde=True,
                    color='purple')
graph.set(xlabel='Uniform Distribution ', ylabel='Frequency')
```



We still need to make each of them into a function and then a class to be callable. The call of the function will ask for the inputs (starts, width, and sample size of the random variate based on the example above) and then automatically generate the moments (".stats"). Below is the progress for each of the distributions:

Distributions	Formula	Graph	Test	Moments	Callable	Running offline	In-Package	Running online
Discrete Distributions						*Graph at least*		
Bernoulli(p)	✓	✓	✓			✓		
Binomial(n,p)	✓	✓	✓			✓		
Geometric	✓	✓	✓			✓		
Negative binomial(r,p)	✓	✓	✓			✓		
Poisson(λ)	✓	✓	✓			✓		
Continuous Distributions								
Uniform(a,b)	✓	✓	✓			✓		
Exponential(λ)	✓	✓	✓			✓		
$Erlang_k(\lambda)$	✓	✓	✓			✓		
Gamma(α, λ)	✓	✓	✓			✓		
Triangular(a, b, c)	✓	✓	✓			✓		
Beta(a, b)	✓	✓	✓			✓		
Weibull(a, b)	✓	✓	✓			✓		
Cauchy	✓	✓	✓			✓		
Normal (μ, σ^2)	✓	✓	✓	✓		✓		

Afterward, we will still need to make it all into a python library, write the guide, and some examples. Timewise, we firmly believe that the report will be delivered on time, and there is more available time now that midterms 1 are over. We will follow an article from Joffrey Bienvenu (2020), who provide a complete guide.

Literature review

This section will discuss the purpose of this project in creating a library based on Python through related research conducted before hands.

Created two decades ago, Python's coding language has provided high-level data structures to make interactive, interpreted, and oriented objects (Dhruv, Patel, & Doshi, 2021). The success of Python lies in its uncomplicated syntax while remaining a robust and formidable programming language in the world of data science (Dhruv, Patel, & Doshi, 2021). One of the main objectives of this project is to create an open-source package, meaning to make all the codes available in the hope of "open science" (Daele, Hoey, & Nopens, 2015). Furthermore, it would allow individuals to make future contributions to the original code published by selecting their contribution part, their level of assistance, and the ease of integrating the contributions to the project (Heron, Hanson, & Ricketts, 2013).

This python package will create a base for some probability and statistical systems of distribution to generate random variates (Goldsman & Goldsman, 2020). Based on Goldsman's class, we will implement the following distributions: Bernoulli, Binomial, Geometric, Negative Binomial, Poisson, Uniform, Exponential, ErlanGamma, Triangular, Beta, Weibull, Cauchy, and Normal (Goldsman & Goldsman, 2020). Furthermore, we will be using the probability distributions from the package Scipy, which allows statistical functions that combine different methods (The SciPy community, 2022). Finally, the graphs will be generated based on the Seaborn package (Waskom, 2021).

Concerns

At some point in the project, we got some concerns about applying the distribution. Will it need to be applied by defining the inputs (example: Triangular(a,b,c))? Would it need to be applied to a dataset? Here the guideline does not mention it, so we are sticking here to inputs from the user. In this project, we think that the real challenge is the code to be posted in a library. Moreover, one of the extra goals would be to post that library to learn that process for future similar cases.

Then, regarding Erlang, we are not sure to put a single distribution as it is a particular case of the gamma distribution and the documentation of Scipy mentioned that it would be inside it already.

Please kindly let us know if this is wrong here to adapt for the rest of the project. We are really happy about this project as we (I am alone but used to employing 'we' in essays) are learning something completely new about the library and the object-oriented class programming in Python (taking an extra class in DataCamp for this one).

References

Bienvenu, J. (2020). *Deep dive: Create and publish your first Python library.* Retrieved from https://towardsdatascience.com/deep-dive-create-and-publish-your-first-python-library-f7f618719e14

Daele, T. V., Hoey, S. V., & Nopens, I. (2015). pyIDEAS: an Open Source Python Package. *Computer Aided Chemical Engineering, 37*(1), pp. 569-574. doi:10.1016/B978-0-444-63578-5.50090-6

Dhruv, A. J., Patel, R., & Doshi, N. (2021). Python: The Most Advanced Programming Language for Computer. *Proceedings of the International Conference on Culture Heritage, Education, Sustainable Tourism, and Innovation Technologies*, pp. 292-299. doi:10.5220/0010307902920299

Goldsman, D., & Goldsman, P. (2020). *A First Course in Probability and Statistics.* USA. doi:201127.210818

Heron, M., Hanson, V. L., & Ricketts, I. (2013). Open source and accessibility: advantages. *Journal of Interaction Science, 1*(2), pp. 1-10. doi:10.1186/2194-0827-1-2

The SciPy community. (2022). *Statistical functions (scipy.stats).* Retrieved from https://docs.scipy.org/doc/scipy/reference/stats.html

Waskom, M. (2021). *Seaborn: statistical data visualization.* Retrieved from https://seaborn.pydata.org