Group 208 - Hugo Dupouy (GT903738077)

**Table of Contents**

[Introduction (200 words max - 5%) 2](#_Toc97457424)

[Background and Description of Problem (200 words - 5%) 2](#_Toc97457425)

[Literature review 2](#_Toc97457426)

[Main Findings (40%) 2](#_Toc97457427)

[Guidelines 2](#_Toc97457428)

[Examples 2](#_Toc97457429)

[Conclusion 2](#_Toc97457430)

[Appendix 2](#_Toc97457431)

# Introduction (200 words max - 5%)

*Programming-Oriented Problems*

*18. (1-2 members) Make me a nice library of random variate generation routines. You can use your favorite high-level language like C++, Java, Python, Matlab, or even Excel. Include in your* ***library routines for generating random variates from all of the usual discrete and continuous distributions****, e.g., Bern(p), Geom(p), Exp(\_), Normal(\_; \_2), Gamma(\_; \_), Weibull(\_; \_), etc., etc. (Just one routine per distribution is fine.) Include in your write-up an easy user's guide, complete source code, and some appropriate examples.*

It would best to keep it for the end after the conclusion.

Goal

This research project aims to build a library where users would be able to utilize discrete and continuous distributions routines for random variates in Python.

# Background and Description of Problem (200 words - 5%)

## 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, the coding language Python 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).

Within this python package, we 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 Binonial, Poisson, Uniform, Exponential, ErlanGamma, Triangular, Beta, Weibull, Cauchy, and Normal (Goldsman & Goldsman, 2020). We will be using the probability distributions from the package scipy which allow to have the statistical functions that already combine different methods (The SciPy community, 2022). Regarding the graphs, they will be generated based on Seaborn package (Waskom, 2021).

# Main Findings (40%)

Programming-Oriented Problems. These problems are often concerned with Monte Carlo analysis of a game or preparing an easy-to-use library for some simulation functionality for us.

o Describe the problem at hand, and maybe provide a small literature review.

o Develop and document your code.

o Show how to run your program(s).

o Give illustrations of what you can learn from your code (e.g., whether or not a PRN generator is any good, or whether or not a certain strategy will work better than others in blackjack). Make sure to be statistically rigorous if you’re carrying out MC experiments.

## Guidelines

## Examples

# Conclusion (200 words - 5%)

# Appendix

We will expand on the below distributions that we saw in class so far.

<https://www.datacamp.com/community/tutorials/probability-distributions-python>

<https://www.kaggle.com/hamelg/python-for-data-22-probability-distributions>

<https://cmdlinetips.com/2018/03/probability-distributions-in-python/>

### Discrete Distributions

Bernoulli(p)

Binomial(n,p)

Geometric

Negative binomial(r,p)

Poisson(λ)

### Continuous Distributions

Uniform(a,b)

Exponential(λ)

Erlangk(λ)

Gamma(α, λ)

Triangular(a, b, c)

Beta(a, b)

Weibull(a, b)

Cauchy

Normal (μ, σ2)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Distributions** | **Formula** | **Graph** | **Test** | **Moments** | **Callable** | **Running offline** | **In-Package** | **Running online** |
| ***Discrete Distributions*** |  |  |  |  |  |  |  |  |
| Bernoulli(p) | ✓ | ✓ | ✓ | ✓ |  | ✓ |  |  |
| Binomial(n,p) | ✓ | ✓ | ✓ | ✓ |  | ✓ |  |  |
| Geometric | ✓ | ✓ | ✓ | ✓ |  | ✓ |  |  |
| Negative binomial(r,p) | ✓ | ✓ | ✓ | ✓ |  | ✓ |  |  |
| Poisson(λ) | ✓ | ✓ | ✓ | ✓ |  | ✓ |  |  |
| ***Continuous Distributions*** |  |  |  |  |  |  |  |  |
| Uniform(a,b) | ✓ | ✓ | ✓ | ✓ |  | ✓ |  |  |
| Exponential(λ) | ✓ | ✓ | ✓ | ✓ |  | ✓ |  |  |
| Erlangk(λ) | ✓ | ✓ | ✓ | ✓ |  | ✓ |  |  |
| Gamma(α, λ) | ✓ | ✓ | ✓ | ✓ |  | ✓ |  |  |
| Triangular(a, b, c) | ✓ | ✓ | ✓ | ✓ |  | ✓ |  |  |
| Beta(a, b) | ✓ | ✓ | ✓ | ✓ |  | ✓ |  |  |
| Weibull(a, b) | ✓ | ✓ | ✓ | ✓ |  | ✓ |  |  |
| Cauchy | ✓ | ✓ | ✓ | ✓ |  | ✓ |  |  |
| Normal (μ, σ2) | ✓ | ✓ | ✓ | ✓ |  | ✓ |  |  |

### Library - Python package

<https://towardsdatascience.com/deep-dive-create-and-publish-your-first-python-library-f7f618719e14>

(Bienvenu, 2020)

<https://besjournals.onlinelibrary.wiley.com/doi/full/10.1111/2041-210X.13313>

# 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