**Programming Probability Distributions**

**Introduction**

It's time to get more experience programming in Python. Probability distributions involve algebraic functions and visualizing those functions.

You'll start out by making a few basic functions to calculate and visualize continuous uniform probability distributions.

Then, you'll program a more complex distribution: a non-uniform discrete probability distribution.

Next, you'll use those skills to explore how a self-driving car, or really any robot, might represent its location probabilities in a discrete, 1-D world. Then you'll expand into probabilities in a 2-D world. You're going to implement the 2-D world using something called a **class**, which is a sort of programming template. You'll learn more about classes in the exercises.

Finally, you'll become familiar with a probability distribution that is especially important to self-driving cars: the normal distribution.

**Skills**

Here is a summary of the skills you'll work on in these excercises:

**Python Skills**

* Writing a function in python
* Initializing 1-Dimensional and 2-Dimensional arrays
* Creating visualizations
* Classes in Python

**Probability Skills**

* Calculating probabilities for a continuous uniform distribution
* Calculating probabilities for a discrete non-uniform distribution
* Location probabilities on a 1-D grid
* Location probabilities on a 2-D grid

**Visualizing a Piece-Wise Continuous Probability Distribution**

Now it's time for a challenge!

After Sebastian introduced a uniform continuous probability distribution, he then showed you a piece-wise continuous function representing the probability of when someone was born. You are going to code a generic piece-wise continuous probability density function in Python.

Your result is going to look something like this:

A screenshot of a cell phone

Description automatically generated

**Instructions**

Write a function that, given a list of x-axis intervals, relative probabilities and a total probability, calculates the height of each bar. Remember that the sum of the area for all bars should be the total probability.

Here is an example input and output based on the above visualization:

* a vehicle accident is 5 times more likely from 5am t o 10am versus midnight to 5am.
* a vehicle accident is 3 times more likely from 10am to 4pm versus midnight to 5am.
* a vehicle accident is 6 times more likely from 4pm to 9pm versus midnight to 5am.
* a vehicle accident is 1/2 as likely from 9pm to midnight versus midnight to 5am.
* The probability of getting in an accident on any given day is .05

The inputs would look like this.

For the hours, you can use 24 hour time hour\_intervals = [0, 5, 10, 16, 21, 24]

relative\_probabilities = [1, 5, 3, 6, 0.5]

total\_probability = 0.05

The output would be the height of each bar:

[0.0006451612903225806,

0.0032258064516129032,

0.0016129032258064516,

0.003870967741935484,

0.0005376344086021505]

**Hints**

* Summing the area of all the bars equals the total probability, which in this case is 0.05.
* The relative probabilities and total probability can be used to find the exact area of each bar. If the area of the first bar is A, then the area of the second bar is 5A, the third bar is 3A, etc.
* Once you know the area of each bar, you can divide each area by its width to calculate the bar height.

But the function should be generic. It should be able to receive an arbitrary list of numbers with the relative probabilities of each interval and any value for total probability. Before trying to write the program, it might be helpful to calculate the results you expect with a pen and paper. That will help you work through the logic of the programming.

You'll find the exercise in the next part of the lesson.

# Conclusion

In this lesson you've programmed a few discrete probability distribution including a continuous uniform distribution, a piece-wise uniform distribution, and a discrete distribution representing a mapped grid.

In the next section, you'll gain experience working with a very important continuous distribution called the **Gaussian distribution**. A Gaussian distribution is oftentimes used to represent uncertainty in sensor measurements or locations. Continue on to the next lesson to learn more.