A Concrete Introduction to Probability (using Julia)

Originally authored for Python by Peter Norvig

Adapted for Julia by Dimitar Vanguelov

...

In 1814, Pierre-Simon Laplace **wrote**:

> Probability theory is nothing but common sense reduced to calculation. ... [Probability] is thus simply a fraction whose numerator is the number of favorable cases and whose denominator is the number of all the cases possible ... when nothing leads us to expect that any one of these cases should occur more than any other.

Laplace nailed it. To untangle a probability problem, all you have to do is define exactly what the cases are, and careful count the favorable and total cases. Let's be clear on our vocabulary words:

- **Trial**: A single occurrence with an outcome that is uncertain until we observe it. *For example, rolling a single die.*
- **Outcome**: A possible result of a trial; one particular state of the world. What Laplace calls a **case**. *For example: 4.*
- **Sample Space**: The set of all possible outcomes for the trial. For example, {1, 2, 3, 4, 5, 6}.

- **Event**: A subset of outcomes that together have some property we are interested in. *For example, the event "even die roll" is the set of outcomes {2, 4, 6}.*
- **Probability**: As Laplace said, the probability of an event with respect to a sample space is the "number of favorable cases" (outcomes from the sample space that are in the event) divided by the "number of all the cases" in the sample space (assuming "nothing leads us to expect that any one of these cases should occur more than any other"). Since this is a proper fraction, probability will always be a number between 0 (representing an impossible event) and 1 (representing a certain event). *For example, the probability of an even die roll is 3/6 = 1/2.*

This notebook will explore these concepts in a concrete way using Python code. The code is meant to be succint and explicit, and fast enough to handle sample spaces with millions of outcomes. If you need to handle trillions, you'll want a more efficient implementation. Peter Norvig also has **another notebook** that covers paradoxes in Probability Theory.

P is for Probability

The code below implements Laplace's quote directly: *Probability is thus simply a fraction whose numerator is the number of favorable cases and whose denominator is the number of all the cases possible.*

```
Main.workspace96.P
```

```
"""
The probability of an event, given a sample space.
"""
function P(event::Set, space::Set)

    # favioriable = outcomes that are in the event and in the sample space
    favorable = intersect(event, space)

    # the number of cases is the length, or size, of a set
    num = length(favorable)
    den = length(space)

    # this returns the result as a Rational
    return num // den
end
```

Read more about Rational numbers in Julia ***here***

```
Enter cell code...
```