ECON 220 Lab
(Week 4)

Probability and Functions in Python

Justin S. Eloriaga

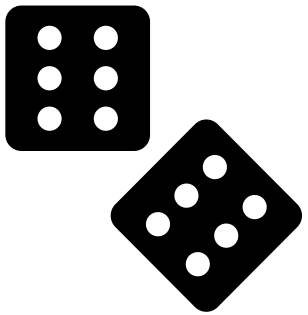
Outline

- Empirical vs. Theoretical Probability
- Making Functions in Python

A Simple Function

```
def roll_die():
    return random.randint(1, 6)

# Example usage
roll_die()
```



The function's name is **roll_die**. Running this will return us a **random** number, to be specific, a **random integer** between **1 to 6** (i.e. 1,2,3,4,5, and 6)

The Obvious

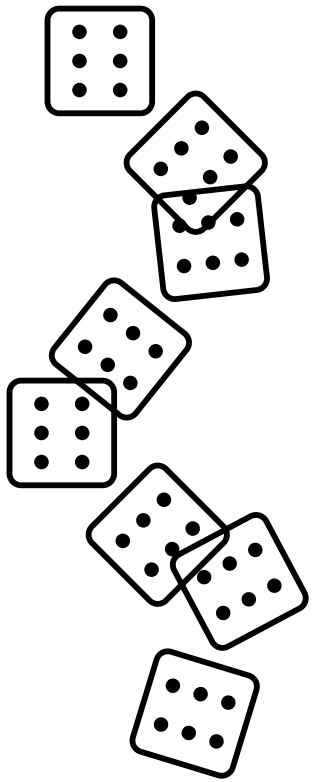
```
theoretical_prob = 1/6
```

Assuming the die was fair, the theoretical probability will always be 1/6.

That is, there is a one in six chance of rolling a specific number, say rolling a three.

A More Complicated Function

```python
# Simulating multiple rolls
def simulate_rolls(n_trials, target_number):
    count = 0
    for _ in range(n_trials):
        roll = roll_die()
        print(f"The die fell on {roll}")  # Debugging statement
        if roll == target_number:
            count += 1  # Correct counting
    return count / n_trials  # Empirical probability
```



The function takes ***two arguments*** to run. (1) n_trials which is the number of experiments. (2) target_number which is the target result (i.e. any number from 1 to 6)

Running the Complicated Function

```
# Running the simulation
n_trials = 1000  # Adjust for testing
target_number = 3  # We're checking probability of rolling a 3

empirical_prob = simulate_rolls(n_trials, target_number)
```

It is good practice to declare an object that the function takes so it is easier to keep track of changes.

i.e. Running simulate_rolls(1000,3) will give you the same thing, but it's easier to change things when you make an object like this.

Empirical Probability vs Theoretical Probability

- Empirical Probability
  - After ***n*** number of trials, the probability of rolling 3 is _____.
  - This is going to very by ***n***!
    - The smaller the ***n***, the less likely it is to be near the theoretical probability.
    - The larger the ***n***, the more likely it is going to be near the theoretical probability **(LAW OF LARGE NUMBERS)**
- Theoretical Probability
  - Assuming the die was ***fair***, the probability of rolling 3 is _____.
  - Clearly, this is 1/6 regardless of ***n*** since there are six sides to the die, assuming fairness, are equally likely.

Illustrating the Law of Large Numbers

```python
def convergence_rolls(n_trials, target_number):
    empirical_probs = []  # Store empirical probabilities over time
    count = 0  # Counter for occurrences of target_number

    for i in range(1, n_trials + 1):
        roll = roll_die()
        if roll == target_number:
            count += 1
        empirical_probs.append(count / i)  # Empirical probability at this trial

    return empirical_probs  # Ensure it returns a list of length n_trials
```

Similar to what we did before! Make a function. Then, we want to basically get empirical probabilities over the number of trials! We are going to get an average of the probability of rolling a specific number over the amount of trials!

Illustrating the Law of Large Numbers

```
# Running the simulation
n_trials = 1000  # Number of trials
target_number = 3  # Checking probability of rolling a 3

empirical_probs = convergence_rolls(n_trials, target_number)
```

```
plt.figure()
plt.plot(range(1, n_trials + 1), empirical_probs, label="Empirical Probability")
plt.axhline(y=theoretical_prob, color='r', linestyle='--', label="Theoretical Probability (1/6)")
plt.xlabel("Number of Rolls")
plt.ylabel("Probability of Rolling a 3")
plt.title(f"Empirical vs. Theoretical Probability of Rolling a {target_number}")
plt.legend()
plt.show()
```

Fancy Graph

Note: The result WILL VARY from person to person. Why?

Because… we used **random** remember? ☺

Clearly, as the number of rolls gets larger and larger, we see that the empirical probability converges to the theoretical probability