DATA SCIENCE FOR ECONOMISTS

ECON 220 LAB

Jafet Baca-Obando

Week 4, Probability and Functions in Python – 09/19/2025

Outline

- Empirical probability
- Theoretical probability
- Creating functions in Python

A simple function

- This function's name is roll_dice.
- Running this function returns a random integer between 1 and 6 (i.e., 1, 2, 3, 4, 5, or 6)

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

# Implement the function once
roll_dice()
```

✓ 0.0s Python

```
1
```

Theoretical probability

- Assuming the die is fair, the theoretical probability will always be 1/6.
- That is, there is one in six change of rolling a specific number, say rolling a three.

```python
theoretical_prob = 1/6
```

✓ 0.0s Python

An involved function

- This function takes two arguments:
  - num_tosses: the number of experiments
  - target_number: target result (any number from 1 to 6)

```
def simulate_rolls(num_tosses, target_number):
    count = 0
    for _ in range(num_tosses):
        roll = roll_dice()
        print(f"The dice fell on: {roll}")
        if roll == target_number:
            count += 1
    return count / num_tosses
```

Running the involved function

```python
# Simulate 1000 dice rolls and calculate the empirical probability of getting a 3
num_tosses = 1000
target_number = 3
empirical_prob = simulate_rolls(num_tosses, target_number)
```

✓ 0.0s Python

```
The dice fell on: 4
The dice fell on: 5
The dice fell on: 2
The dice fell on: 5
The dice fell on: 4
```

Empirical probability

- After n number of tosses, the probability of rolling 3 is ____.
- This result varies 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

- If the dice is fair, the probability of rolling 3 is ____.
- This is ____ regardless of the number of tosses, n, because there are six sides, each equally likely to happen.

The law of large numbers

- Now, we want to get empirical probabilities over the number of trials. Specifically, we're going to get an average of the probability of rolling a specific number over the number of trials.

```python
def convergence_rolls(num_tosses, target_number):
    empirical_probs = [] # Initialize an empty list to store empirical probabilities
    count = 0

    for i in range(1, num_tosses + 1):
        roll = roll_dice()
        if roll == target_number:
            count += 1
        empirical_probs.append(count / i) # Append empirical probability at this trial

    return empirical_probs
```

The law of large numbers

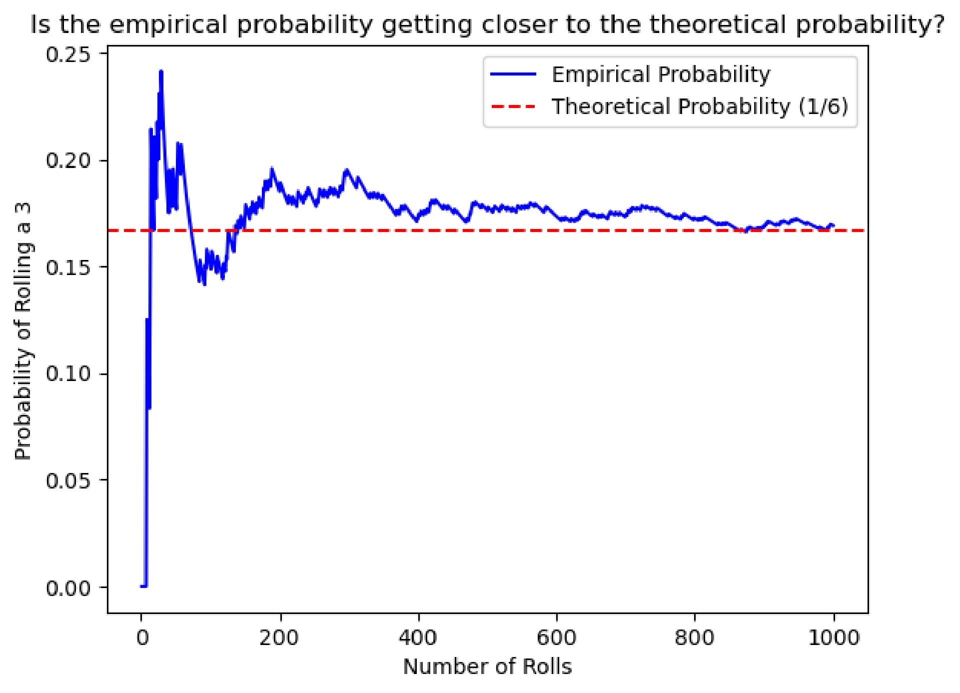
```python
# Simulate 1000 dice rolls and track empirical probabilities
num_tosses = 1000
target_number = 3
empirical_probs = convergence_rolls(num_tosses, target_number)

empirical_probs
```

Python

```python
plt.figure()
plt.plot(range(1, num_tosses + 1), empirical_probs, label='Empirical Probability', color='blue')
plt.axhline(y=theoretical_prob, color='red', linestyle='--', label='Theoretical Probability (1/6)')
plt.xlabel('Number of Rolls')
plt.ylabel('Probability of Rolling a 3')
plt.title('Is the empirical probability getting closer to the theoretical probability?')
plt.legend()
plt.show()
```

Python

Empirical and theoretical probabilities



Recap

- We distinguished between theoretical probability and empirical probability.
- We learned how to create our own functions in Python.
- We visualized the law of large numbers.

To-do list

- **Complete Data Exercise 1**
  - Upload Jupyter notebook (.ipynb file) and HTML file on **September 20**.
- Complete Data Exercise 2
  - Upload Jupyter notebook (.ipynb file) and HTML file on **September 27**.