Jason Waseq
CSE 107
Lab 6

This project aims to demonstrate the Central Limit Theorem (CLT) by simulating the rolling of a loaded die and observing how the normalized sum of these rolls converges to a standard normal distribution. Specifically, you'll define a loaded die with custom probabilities for each face, then repeatedly roll this die n=500 times, calculating a value Zn for each set of rolls. By repeating this process 20,000 times, you'll estimate the Cumulative Distribution Function (CDF) of Zn. Finally, you'll present these estimated CDF values in a table formatted like a standard normal distribution table and compare it to the actual standard normal table to illustrate the CLT.

Code in Python:

```
import numpy as np
from scipy.stats import norm

# Define the number of trials and the value of n
num_trials = 20000
n = 500

# Define the range of z values
z_values = np.arange(0.00, 3.50, 0.01)

# Function to simulate rolling a loaded die
def roll_loaded_die(probabilities):
    return np.random.choice(range(1, 7), p=probabilities)

# Function to calculate Zn
def calculate_Zn(rolls, mu, sigma):
    Sn = np.sum(rolls)
    Zn = (Sn - n * mu) / (sigma * np.sqrt(n))
    return Zn

def main():
    # Define the probabilities for the loaded die
    probabilities = [0.1, 0.15, 0.2, 0.25, 0.15, 0.15]

    # Verify that the probabilities sum to 1
    if sum(probabilities) != 1:
        raise ValueError("Probabilities must sum to 1")

    # Calculate the mean and variance of the loaded die
    mu = np.sum([(i + 1) * p for i, p in enumerate(probabilities)])
```

```
    variance = np.sum([((i + 1) - mu)**2 * p for i, p in enumerate(probabilities)])
    sigma = np.sqrt(variance)

    # Initialize a list to store the results of the trials
    Zn_values = []

    # Run the simulation
    for _ in range(num_trials):
        # Roll the loaded die n times
        rolls = [roll_loaded_die(probabilities) for _ in range(n)]

        # Calculate Zn
        Zn = calculate_Zn(rolls, mu, sigma)

        # Append Zn to the list of results
        Zn_values.append(Zn)

    # Calculate the experimental CDF
    experimental_cdf = []
    for z in z_values:
        # Calculate the relative frequency of Zn <= z
        relative_frequency = np.sum(Zn_values <= z) / num_trials
        experimental_cdf.append(relative_frequency)

    # Reshape the experimental CDF into a table
    experimental_cdf_table = np.reshape(experimental_cdf, (35, 10))

    # Print the experimental CDF table
    print("Experimental CDF")
    print("          .00       .01       .02       .03       .04       .05       .06       .07       .08       .09")
    print("     ----------------------------------------------------------------------------------------------------")
    for i in range(35):
        row_label = i / 10
        print(f"  {row_label:.1f} | ", end="")
        for j in range(10):
            print(f"{experimental_cdf_table[i][j]:.4f}  ", end="")
        print()

if __name__ == "__main__":
    main()
```

Output:

Experimental CDF

	.00	.01	.02	.03	.04	.05	.06	.07	.08	.09
0.0	0.5028	0.5028	0.5028	0.5151	0.5151	0.5151	0.5265	0.5265	0.5265	0.5370
0.1	0.5370	0.5370	0.5485	0.5485	0.5485	0.5594	0.5594	0.5594	0.5714	0.5714
0.2	0.5714	0.5824	0.5824	0.5824	0.5936	0.5936	0.5936	0.6056	0.6056	0.6056
0.3	0.6176	0.6176	0.6176	0.6294	0.6294	0.6294	0.6399	0.6399	0.6399	0.6508
0.4	0.6508	0.6508	0.6612	0.6612	0.6703	0.6703	0.6703	0.6818	0.6818	0.6818
0.5	0.6922	0.6922	0.6922	0.7020	0.7020	0.7020	0.7117	0.7117	0.7117	0.7221
0.6	0.7221	0.7221	0.7319	0.7319	0.7319	0.7422	0.7422	0.7422	0.7518	0.7518
0.7	0.7518	0.7616	0.7616	0.7616	0.7708	0.7708	0.7708	0.7796	0.7796	0.7796
0.8	0.7876	0.7876	0.7876	0.7965	0.7965	0.7965	0.8049	0.8049	0.8118	0.8118
0.9	0.8118	0.8204	0.8204	0.8204	0.8277	0.8277	0.8277	0.8353	0.8353	0.8353
1.0	0.8433	0.8433	0.8433	0.8510	0.8510	0.8510	0.8581	0.8581	0.8581	0.8649
1.1	0.8649	0.8649	0.8710	0.8710	0.8710	0.8776	0.8776	0.8776	0.8840	0.8840
1.2	0.8840	0.8893	0.8893	0.8893	0.8943	0.8943	0.8943	0.8991	0.8991	0.9049
1.3	0.9049	0.9049	0.9101	0.9101	0.9101	0.9150	0.9150	0.9150	0.9195	0.9195
1.4	0.9195	0.9236	0.9236	0.9236	0.9274	0.9274	0.9274	0.9315	0.9315	0.9315
1.5	0.9364	0.9364	0.9364	0.9404	0.9404	0.9404	0.9433	0.9433	0.9433	0.9464
1.6	0.9464	0.9464	0.9490	0.9490	0.9490	0.9523	0.9523	0.9523	0.9548	0.9548
1.7	0.9548	0.9577	0.9577	0.9602	0.9602	0.9602	0.9627	0.9627	0.9627	0.9646
1.8	0.9646	0.9646	0.9667	0.9667	0.9667	0.9685	0.9685	0.9685	0.9701	0.9701
1.9	0.9701	0.9717	0.9717	0.9717	0.9732	0.9732	0.9732	0.9750	0.9750	0.9750
2.0	0.9771	0.9771	0.9771	0.9789	0.9789	0.9789	0.9800	0.9800	0.9800	0.9816
2.1	0.9816	0.9816	0.9829	0.9829	0.9841	0.9841	0.9841	0.9856	0.9856	0.9856
2.2	0.9867	0.9867	0.9867	0.9877	0.9877	0.9877	0.9886	0.9886	0.9886	0.9891
2.3	0.9891	0.9891	0.9901	0.9901	0.9901	0.9909	0.9909	0.9909	0.9918	0.9918
2.4	0.9918	0.9925	0.9925	0.9925	0.9931	0.9931	0.9931	0.9937	0.9937	0.9937
2.5	0.9941	0.9941	0.9941	0.9946	0.9946	0.9946	0.9951	0.9951	0.9955	0.9955
2.6	0.9955	0.9959	0.9959	0.9959	0.9961	0.9961	0.9961	0.9964	0.9964	0.9964
2.7	0.9967	0.9967	0.9967	0.9968	0.9968	0.9968	0.9970	0.9970	0.9970	0.9972
2.8	0.9972	0.9972	0.9974	0.9974	0.9974	0.9977	0.9977	0.9977	0.9978	0.9978
2.9	0.9978	0.9980	0.9980	0.9980	0.9980	0.9980	0.9980	0.9981	0.9981	0.9984
3.0	0.9984	0.9984	0.9986	0.9986	0.9986	0.9987	0.9987	0.9987	0.9987	0.9987
3.1	0.9987	0.9988	0.9988	0.9988	0.9990	0.9990	0.9990	0.9992	0.9992	0.9992
3.2	0.9994	0.9994	0.9994	0.9994	0.9994	0.9994	0.9995	0.9995	0.9995	0.9995
3.3	0.9995	0.9995	0.9996	0.9996	0.9996	0.9996	0.9996	0.9996	0.9997	0.9997
3.4	0.9997	0.9997	0.9997	0.9997	0.9997	0.9997	0.9997	0.9997	0.9997	0.9997

Conjecture:

The experiment demonstrates the Central Limit Theorem (CLT) by simulating the rolling of a loaded die n times and calculating the normalized sum Zn. Despite the loaded die having a non-normal, discrete distribution, after many trials, the experimental CDF of Zn closely approximates the standard normal CDF, as predicted by the CLT. This highlights the theorem's generality: regardless of the underlying distribution of the individual random variables (the die rolls), the distribution of their normalized sum converges to a standard normal distribution as n increases.