ES111-Introduction to Statistics and Probability

CEP Report

Implementation of Central Limit Theorem on a Dataset

Submitted to:
Mr. Fahad Zulfiqar

Prepared by:
Anas Raza Aslam
u2021095@giki.edu.pk

Objective

The purpose of this task is to analyse the distribution of z-scores for different sample sizes using a given dataset of **weights**. The goal is to assess how the sample size affects the z-score distribution and visualize the results using histograms. The task involved importing the necessary libraries, loading the dataset, generating random samples, calculating z-scores, and plotting the distributions.

Libraries Used

The following libraries and frameworks were utilized in this project:

- Pandas
- NumPy
- Matplotlib

Importing Data

The pandas library is used to import the dataset from a CSV file called "**weights_data.csv**". The dataset contained a column named "**weights**" with 1000 entries.

```
# Importing Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Importing weights file
weights_data = pd.read_csv("weights_data.csv")
```

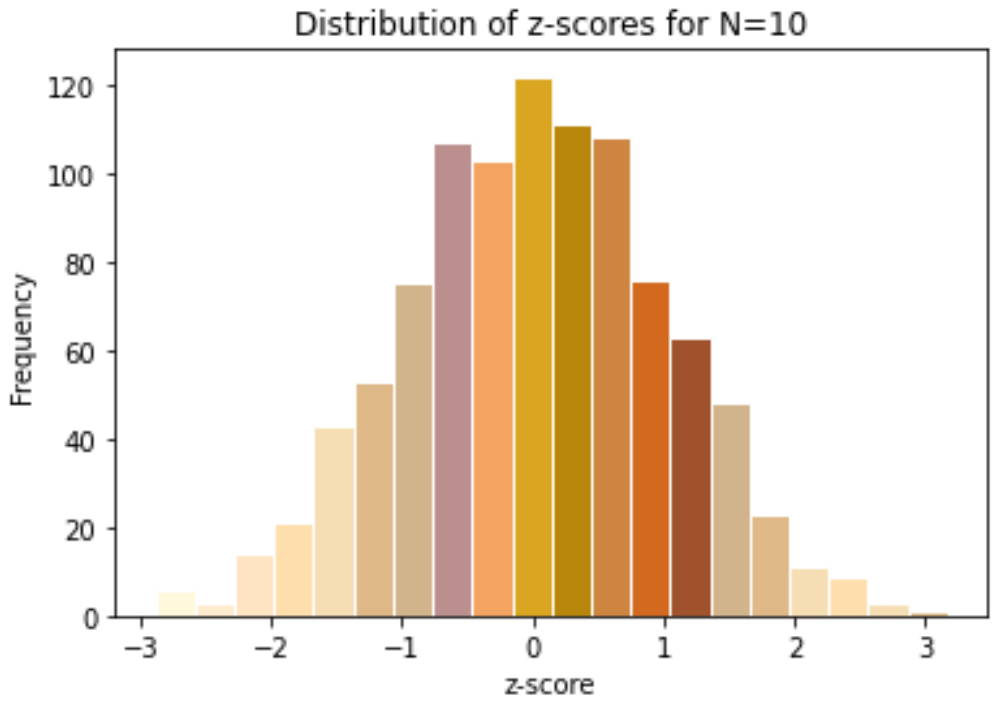
Taking Sample for N = 10

In this step, a random sample of size N = 10 is taken from the "weights" column of the dataset. The formula for calculating z-score is applied to each sample. The process is repeated 1000 times, and the calculated z-scores are stored in a list.

Plotting the Distribution

A histogram is generated using matplotlib to visualize the distribution of the z-scores calculated in the previous step. The histogram has 20 bins and white edges. Each bin is filled with a unique colour. The plot is labelled with appropriate titles and axis labels.

```
# Ploting the results
N, bins, patches = plt.hist(z_scores, bins=20, color= 'green', edgecolor = 'white')
colors = ['#FFF8DC', '#FFEBCD', '#FFE4C4', '#FFDEAD', '#F5DEB3', '#DEB887', '#D2B48C', '#BC8F8F', '#F4A460', '#DAA520',
for i in range(len(N)):
    c = colors[i]
    patches[i].set_fc(c)
plt.title("Distribution of z-scores for N=10")
plt.xlabel("z-score")
plt.ylabel("Frequency")
plt.show()
```



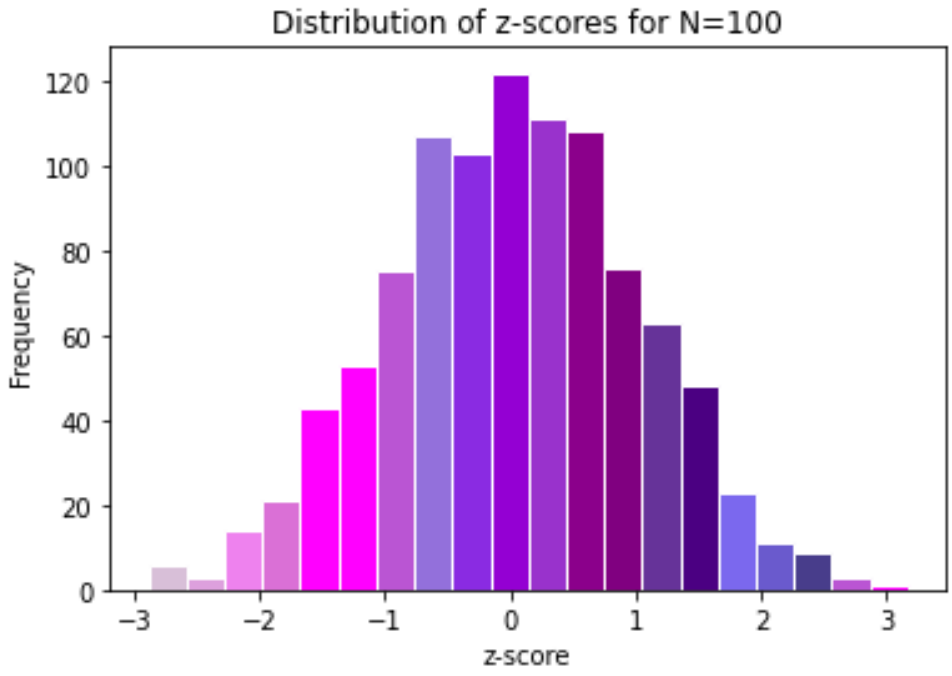
Taking Sample for N = 100

Similarly, a random sample of size N = 100 (without replacement) is taken from the "weights" column of the dataset. The z-scores are calculated for each sample using the same formula as before. The process is repeated 1000 times, and the z-scores are stored in a new list.

Plotting the Distribution

Another histogram is generated to visualize the distribution of the z-scores obtained for N = 100 samples. This histogram also has 20 bins, white edges, and a colour scheme distinct from the previous plot. Each bin is filled with a unique colour. The plot is labelled with appropriate titles and axis labels.

```
# Ploting the results
N, bins, patches = plt.hist(z_scores, bins=20, edgecolor = 'white')
colors = ['#D8BFD8', '#DDA0DD', '#EE82EE', '#DA70D6', '#FF00FF', '#FF00FF', '#BA55D3', '#9370DB', '#8A2BE2', '#9400D3',
for i in range(len(N)):
    c = colors[i]
    patches[i].set_fc(c)
plt.title("Distribution of z-scores for N=100")
plt.xlabel("z-score")
plt.ylabel("Frequency")
plt.show()
```



Note: Colours used in graphs are only for better and attractive visuals. They do not specify anything in the dataset.