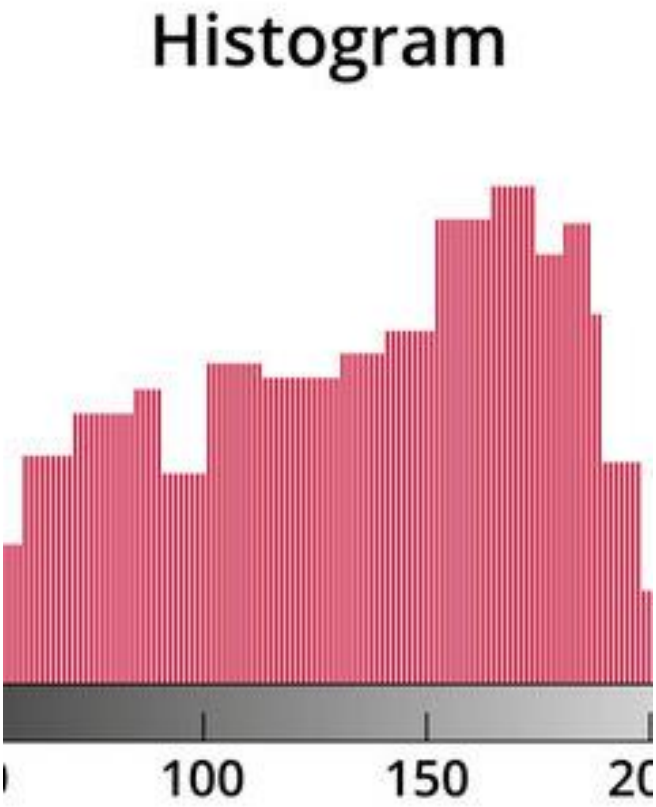
EE230: Probability and Random Processes



ASSIGNMENT 2

Submitted To: Hanumant Singh Shekhawat

Mukul Goel
220108037

Introduction

- **The objective of this assignment is to write a Python code to check whether the histogram or its integration is more suitable for data generated from the uniform distribution with different bin sizes.**

Observation

The plots display histograms with different bin sizes (10, 20, 50) and their corresponding cumulative distribution functions (CDFs) for a dataset generated from a uniform distribution.

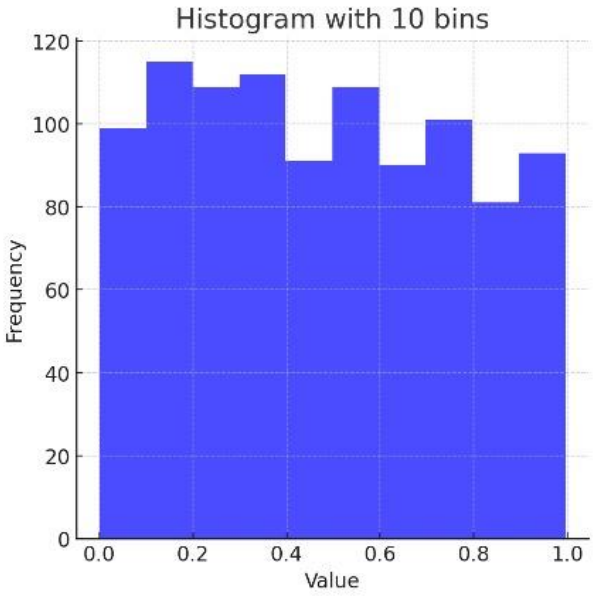
By comparing the histograms and CDFs, we can observe how the choice of bin size affects the representation of the data distribution. Smaller bin sizes provide more granularity, while larger bin sizes offer a smoother overview. The CDFs are less sensitive to the choice of bin size, as they tend to smooth out fluctuations and show the overall trend of the data accumulation.

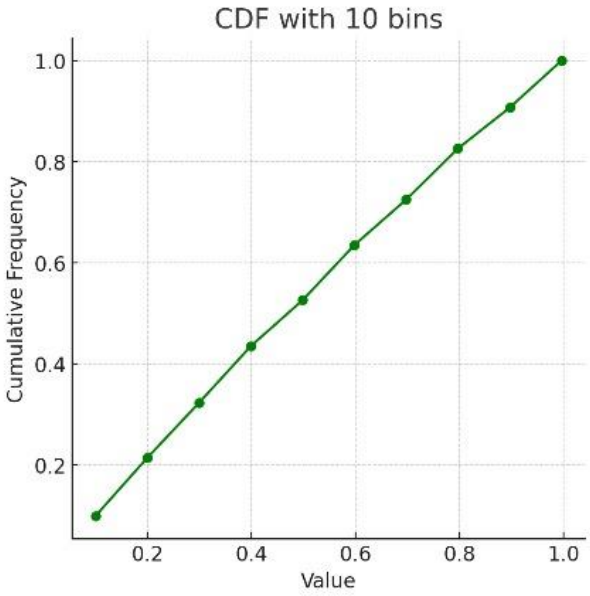
The suitability of using a histogram or its integration (CDF) often depends on the specific analysis you wish to perform. If you need to understand the frequency distribution in different intervals of your data, a histogram is more suitable. If you're interested in the proportion of data below a certain value, the CDF is more appropriate.

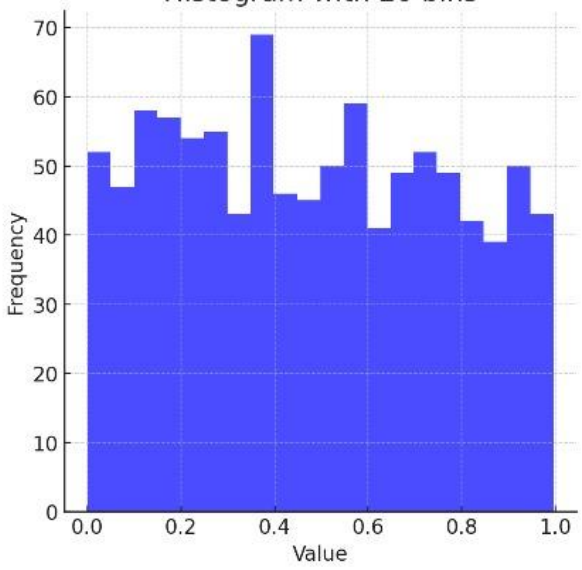
Python Code

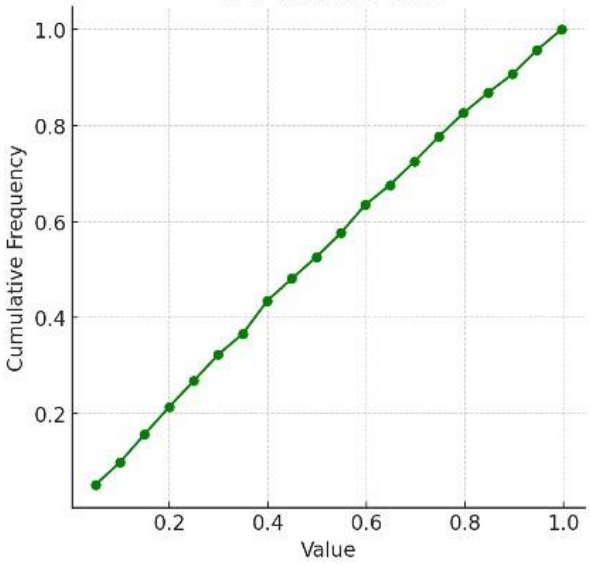
The following *Python code* will create the dataset, plot the histogram, calculate the CDF, and display the results for comparison.

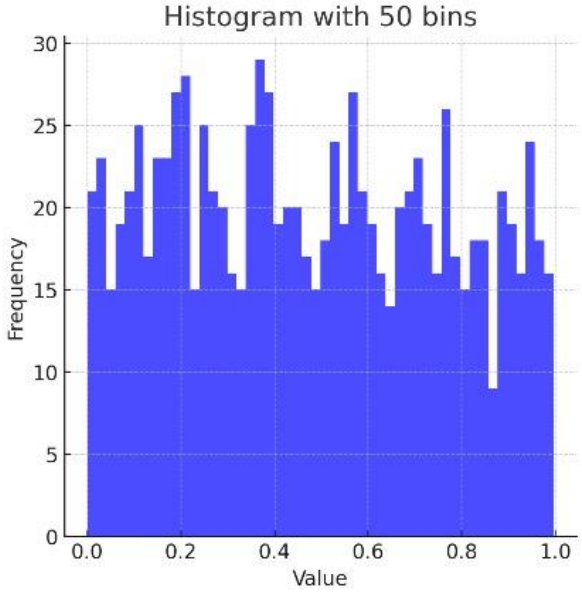
```
import numpy as np
import matplotlib.pyplot as plt

# Generate a uniform distribution dataset
data = np.random.uniform(0, 1, 1000)

# Function to plot the histogram and the CDF
def plot_histogram_and_cdf(data, bin_sizes):
    fig, axes = plt.subplots(len(bin_sizes), 2, figsize=(10, 5 * len(bin_sizes)))

    for i, bins in enumerate(bin_sizes):
        # Plot histogram
        axes[i, 0].hist(data, bins=bins, color='blue', alpha=0.7)
        axes[i, 0].set_title(f'Histogram with {bins} bins')
        axes[i, 0].set_xlabel('Value')
        axes[i, 0].set_ylabel('Frequency')

        # Calculate and plot CDF
        count, bin_edges = np.histogram(data, bins=bins)
        cdf = np.cumsum(count) / sum(count)
        axes[i, 1].plot(bin_edges[1:], cdf, color='green', marker='o', linestyle='-')
        axes[i, 1].set_title(f'CDF with {bins} bins')
        axes[i, 1].set_xlabel('Value')
        axes[i, 1].set_ylabel('Cumulative Frequency')

    plt.tight_layout()
    plt.show()

# Different bin sizes for comparison
bin_sizes = [10, 20, 50]

# Plot and compare histograms and their CDFs with different bin sizes
plot_histogram_and_cdf(data, bin_sizes)
```

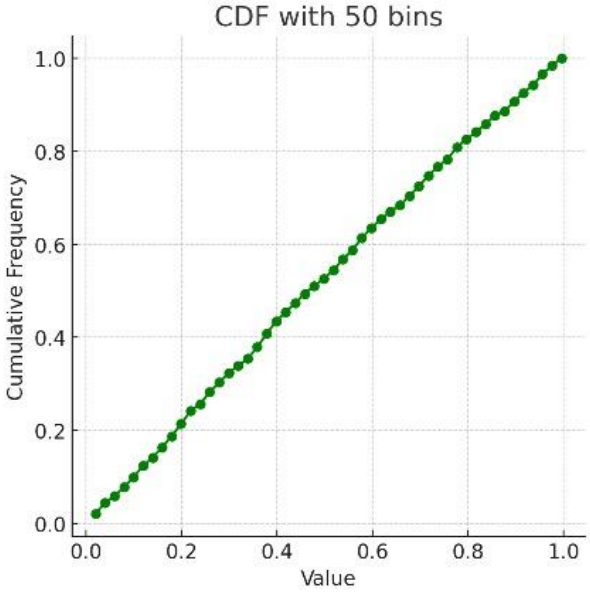

Histogram with 10 bins
Frequency
Value
0
20
40
60
80
100
120
0.0
0.2
0.4
0.6
0.8
1.0


CDF with 10 bins
Cumulative Frequency
Value
0.2
0.4
0.6
0.8
1.0


Histogram with 20 bins
Frequency
Value
0
10
20
30
40
50
60
70
0.0
0.2
0.4
0.6
0.8
1.0


CDF with 20 bins
Cumulative Frequency
Value
0.2
0.4
0.6
0.8
1.0


Histogram with 50 bins
Frequency
Value
0
5
10
15
20
25
30
0.0
0.2
0.4
0.6
0.8
1.0


CDF with 50 bins
Cumulative Frequency
Value
0.0
0.2
0.4
0.6
0.8
1.0