# Analyzing Data Distributions: Histogram Creation and Normality Testing via Kolmogorov-Smirnov Method

To derive a histogram from a dataset and check for normality using the Kolmogorov-Smirnov (K-S) method, we can follow these steps:

**Step 1: Choose and Create the Dataset**

Let's create a synthetic dataset of 30-40 points. For simplicity, we can generate a dataset that is normally distributed.

```
import numpy as np
import matplotlib.pyplot as plt
 #Generate a dataset
np.random.seed(42)
#for reproducibility
data = np.random.normal(loc=50, scale=10, size=40)   #mean=50, std=10, n=40
```

**Step 2: Create the Histogram**

Next, we will create a histogram of the dataset.

```
#Create a histogram
plt.hist(data, bins=10, alpha=0.7, color='blue', edgecolor='black')
plt.title('Histogram of Normally Distributed Data')
plt.xlabel('Value')
plt.ylabel('Frequency')
plt.grid(axis='y', alpha=0.75)
plt.show()
```

**Step 3: Check Normality Using the Kolmogorov-Smirnov Method**

To check for normality, we will use the Kolmogorov-Smirnov test, which compares the sample distribution with a specified distribution, in this case, the normal distribution.

```
from scipy import stats
 #Perform the Kolmogorov-Smirnov test
```

```
ks_statistic, p_value = stats.kstest(data, 'norm', args=(np.mean(data), np.std(data)))
 #Display the results
print (f'K-S Statistic: {ks_statistic}')
print (f'P-value: {p_value}')
 if p_value > 0.05:
    print ("Fail to reject the null hypothesis - data is normally distributed.")
else:
    print ("Reject the null hypothesis - data is not normally distributed.")
```

**Interpretation of Results:**

- The histogram provides a visual representation of the data distribution. For a normally distributed dataset, we expect the histogram to be roughly bell-shaped and symmetric around the mean. However, since our dataset contains only 40 points, the shape of the histogram may deviate slightly from this ideal due to sampling variability. Small sample sizes can lead to irregularities in the histogram, and the curve may not perfectly resemble the classic bell shape of a normal distribution. This is a common occurrence with small datasets, as random fluctuations are more prominent. As the sample size increases, these deviations tend to diminish, and the histogram more closely approximates a bell curve.
- The K-S test results will help determine if the data significantly deviates from a normal distribution. If the p-value is greater than 0.05, we fail to reject the null hypothesis, indicating that the data can be considered normally distributed.

**Complete Code Example:**

Here’s the complete code to generate the histogram and perform the K-S test:

```
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats
 #Generate a dataset
np.random.seed(42)
 #for reproducibility
data = np.random.normal(loc=50, scale=10, size=40)
```

```
#mean=50, std=10, n=40
 #Create a histogram
plt.hist(data, bins=10, alpha=0.7, color='blue', edgecolor='black')
plt.title('Histogram of Normally Distributed Data')
plt.xlabel('Value')
plt.ylabel('Frequency')
plt.grid(axis='y', alpha=0.75)
plt.show()
 #Perform the Kolmogorov-Smirnov test
ks_statistic, p_value = stats.kstest(data, 'norm', args=(np.mean(data), np.std(data)))
 #Display the results
print (f'K-S Statistic: {ks_statistic}')
print (f'P-value: {p_value}')
 if p_value > 0.05:
    print ("Fail to reject the null hypothesis - data is normally distributed.")
else:
    print ("Reject the null hypothesis - data is not normally distributed.")
```

**Conclusion:**

This process will allow you to visualize your dataset's distribution and statistically assess its normality. Adjust the normal distribution parameters in the dataset generation step to explore different scenarios as needed.

**References:**


- Montgomery. D.C., & Runger, G.C. Applied Statistics and Probability for Engineers.
- Conover, W.J. Practical Nonparametric Statistics.
- Python.