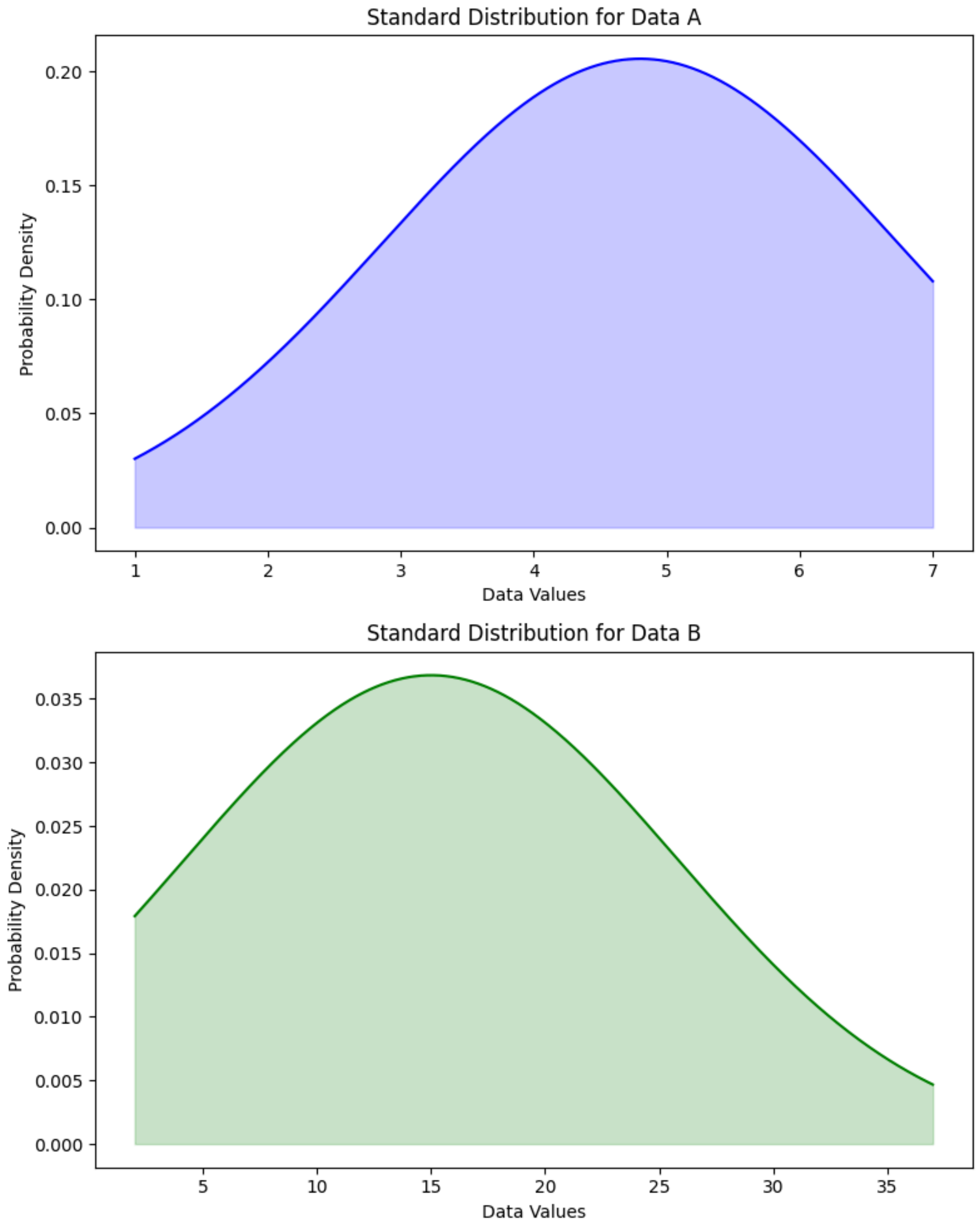
1. Find the Variance and standart deviation of the following data:

A. 1,3,3,4,5,5,6,7,7,7

B. 2,5,11,14,14,22,37

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

data_a = [1, 3, 3, 4, 5, 5, 6, 7, 7, 7]
data_b = [2, 5, 11, 14, 14, 22, 37]

variance_a = np.var(data_a, ddof=0)
std_dev_a = np.std(data_a, ddof=0)

variance_b = np.var(data_b, ddof=0)
std_dev_b = np.std(data_b, ddof=0)


x_a = np.linspace(min(data_a), max(data_a), 1000)
x_b = np.linspace(min(data_b), max(data_b), 1000)

mean_a = np.mean(data_a)
mean_b = np.mean(data_b)
print(f"Variance A:{variance_a}")
print(f"Mean A:{mean_a}")
print(f"standart deviation A:{std_dev_a}")
print(f"Variance B:{variance_b}")
print(f"Mean B:{mean_b}")
print(f"standart deviation B:{std_dev_b}")

pdf_a = (1 / (std_dev_a * np.sqrt(2 * np.pi))) * np.exp(-0.5 * ((x_a - mean_a) / std_dev_a) ** 2)
pdf_b = (1 / (std_dev_b * np.sqrt(2 * np.pi))) * np.exp(-0.5 * ((x_b - mean_b) / std_dev_b) ** 2)

fig, ax = plt.subplots(2, 1, figsize=(8, 10))

ax[0].plot(x_a, pdf_a, label='Data A', color='blue')
ax[0].fill_between(x_a, pdf_a, color='blue', alpha=0.2)
ax[0].set_title('Standard Distribution for Data A')
ax[0].set_xlabel('Data Values')
ax[0].set_ylabel('Probability Density')

ax[1].plot(x_b, pdf_b, label='Data B', color='green')
ax[1].fill_between(x_b, pdf_b, color='green', alpha=0.2)
ax[1].set_title('Standard Distribution for Data B')
ax[1].set_xlabel('Data Values')
ax[1].set_ylabel('Probability Density')

plt.tight_layout()
plt.show()
```

```
Variance A:3.7600000000000007
Mean A:4.8
standart deviation A:1.9390719429665317
Variance B:117.14285714285714
Mean B:15.0
standart deviation B:10.82325538564332
```



2. The price of milk in shops are as follow

49 44 41 52 47 43

A. Find mean and standart deviation of the prices of milk

B. The prices of sugar in shops have an average proof of 52p and a standart deviation of 3.9. Make two valid comparisons between the prices of milk and sugar

```
import numpy as np

# Milk prices
milk_prices = np.array([49, 44, 41, 52, 47, 43])

# Mean and standard deviation of milk prices
mean_milk = np.mean(milk_prices)
std_milk = np.std(milk_prices)

# Sugar prices
mean_sugar = 52
std_sugar = 3.9

# Output the results
print("A")
print("Mean price of milk:", mean_milk)
print("Standard deviation of milk prices:", std_milk)

print("B")
print("Dalam hal ini dapat dibandingkan bahwa harga susu rata-rata adalah 46p dengan standar deviasi 3,74 sedangkan gula dengan")
print("harga 52p dengan standar deviasi 3.9")
print("Jadi, bisa disimpulkan bahwa standar deviasi susu relatif kecil terhadap rata ratanya sehingga membuktikan bahwa persebaran")
print("data tersebut memusat, sebaliknya dikarenakan standar deviasi gula relatif besar terhadap rata ratanya sehingga membuktikan")
print("bahwa persebaran data tersebut menyebar")
```

```
A
Mean price of milk: 46.0
Standard deviation of milk prices: 3.7416573867739413
B
Dalam hal ini dapat dibandingkan bahwa harga susu rata-rata adalah 46p dengan standar deviasi 3,74 sedangkan gula dengan
harga 52p dengan standar deviasi 3.9
Jadi, bisa disimpulkan bahwa standar deviasi susu relatif kecil terhadap rata ratanya sehingga membuktikan bahwa persebaran
data tersebut memusat, sebaliknya dikarenakan standar deviasi gula relatif besar terhadap rata ratanya sehingga membuktikan
bahwa persebaran data tersebut menyebar
```

3. The prices (in pounds) of 6 two-bedroom flats in Glasgow are as follows

85000 98000 140000 110000 120000

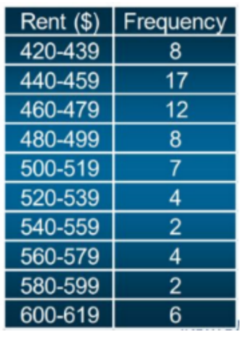
A. Calculate the mean and standard deviation of the prices of the flats.

B. The mean price for a two - bedroom flat in Edinburgh is £128000 and the standard deviation is £2600. Make two valid comparisons about the prices of flats in Glasgow and Edinburgh

```
import numpy as np
glasgow_prices = np.array([85000, 98000, 140000, 110000, 120000])
mean_glasgow = np.mean(glasgow_prices)
std_dev_glasgow = np.std(glasgow_prices, ddof=1)
mean_edinburgh = 128000
std_dev_edinburgh = 2600
print(f"A.Mean harga flat di Glasgow: £{mean_glasgow:.2f}")
print(f"\tDeviasi standar harga flat di Glasgow: £{std_dev_glasgow:.2f}")
print("\nB. Perbandingan:")
print(f"\t1. Harga rata-rata flat di Glasgow (£{mean_glasgow:.2f}) lebih rendah daripada di Edinburgh (£{mean_edinburgh:.2f}).")
print(f"\t2. Deviasi standar harga flat di Glasgow (£{std_dev_glasgow:.2f}) lebih tinggi daripada di Edinburgh (£{std_dev_edinburgh:.2f}).")
```

```
A.Mean harga flat di Glasgow: £110600.00
        Deviasi standar harga flat di Glasgow: £21019.04

B. Perbandingan:
        1. Harga rata-rata flat di Glasgow (£110600.00) lebih rendah daripada di Edinburgh (£128000.00).
        2. Deviasi standar harga flat di Glasgow (£21019.04) lebih tinggi daripada di Edinburgh (£2600.00).
```

4. Given below is the previous sample of monthly rents for 70 efficiency apartments, presented here as grouped data in the form of a frequency distribution. Calculate the mean of the grouped data and compare it to the actual sample mean



Rent ($)	Frequency
420-439	8
440-459	17
460-479	12
480-499	8
500-519	7
520-539	4
540-559	2
560-579	4
580-599	2
600-619	6

```
midpoints = [429.5, 449.5, 469.5, 489.5, 509.5, 529.5, 549.5, 569.5, 589.5, 609.5]
frequencies = [8, 17, 12, 8, 7, 4, 2, 4, 2, 6]


fi_xi_sum = sum(f * x for f, x in zip(frequencies, midpoints))

total_frequency = sum(frequencies)
mean = fi_xi_sum / total_frequency
print(f"A.Rata-rata populasi:\n\t{mean}")


midpoints_sample = [429.5, 449.5, 469.5, 489.5, 509.5, 529.5, 549.5, 569.5, 589.5, 609.5]
frequencies_sample = [4, 10, 10, 6, 5, 3, 2, 4, 2, 4]

fi_xi_sum_sample = sum(f * x for f, x in zip(frequencies_sample, midpoints_sample))

total_frequency_sample = sum(frequencies_sample)
mean = fi_xi_sum_sample / total_frequency_sample
print(f"B.Rata-rata sample:\n\t{mean}\n\n")

print('''Jadi, dapat disimpulkan pembandingan sebuah data dengan rata rata sampel maupun rata rata populasi sebagai berikut :
Jika kita menghitung rata-rata menggunakan seluruh populasi, maka kita mendapatkan rata-rata yang benar-benar mewakili semua data (rata-rata populasi) dalam hal ini pasti hasilnya lebih akurat.
Namun, jika kita hanya menggunakan sebagian data (sampel), maka rata-rata sampel hanya merupakan perkiraan dari rata-rata populasi. Jika ukuran sampel cukup besar dan representatif, maka rata-rata sar
''')
```

```
A.Rata-rata populasi:
        493.2142857142857
A.Rata-rata sample:
        499.5
```

```
Jadi, dapat disimpulkan pembandingan sebuah data dengan rata rata sampel maupun rata rata populasi sebagai berikut :
Jika kita menghitung rata-rata menggunakan seluruh populasi, maka kita mendapatkan rata-rata yang benar-benar mewakili semua data (rata-rata populasi) dalam hal ini pasti hasilnya lebih akurat.
Namun, jika kita hanya menggunakan sebagian data (sampel), maka rata-rata sampel hanya merupakan perkiraan dari rata-rata populasi. Jika ukuran sampel cukup besar dan representatif, maka rata-rat
```

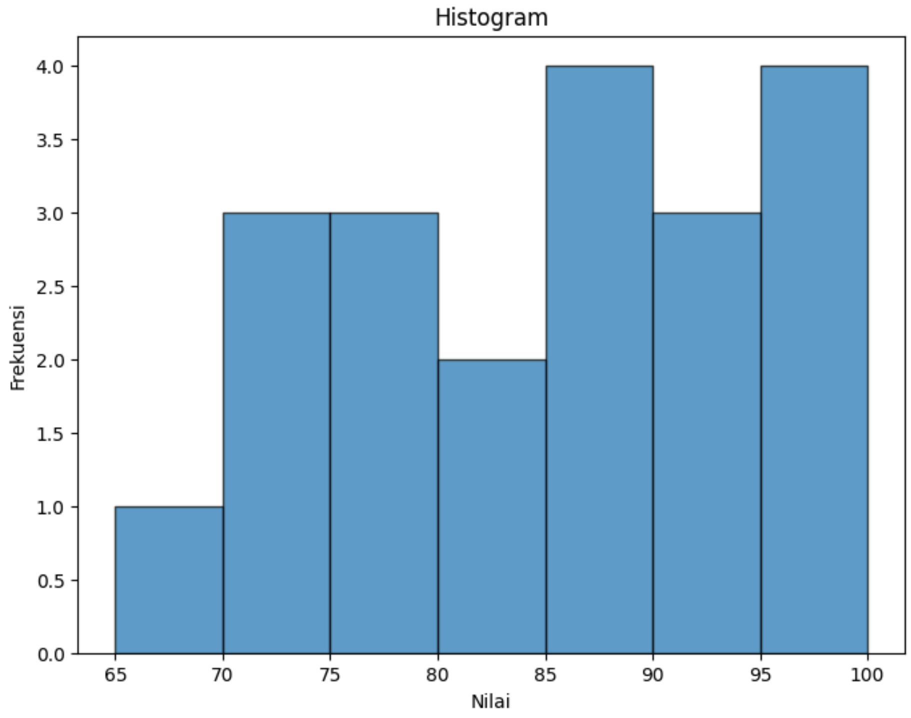
5. You are given the following dataset of 20 values

65, 70, 75, 80, 85, 90, 95, 100, 70, 85, 80, 75, 90, 85, 100, 95, 85, 90, 70, 75

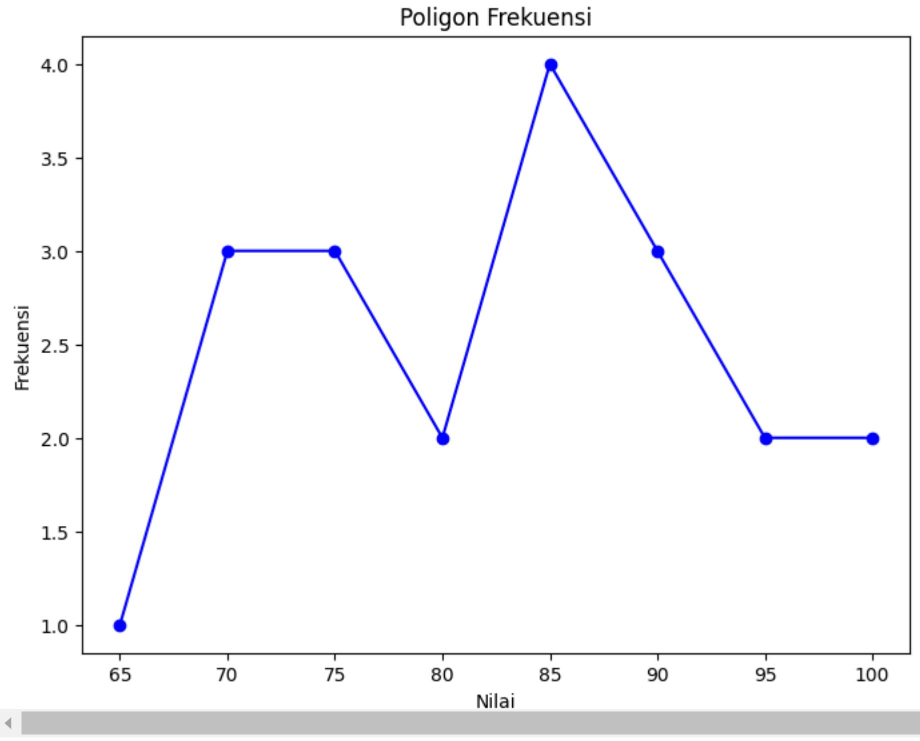
Make the frequency distribution table, histogram, and frequency polygon

```python
import pandas as pd

data = [65, 70, 75, 80, 85, 90, 95, 100, 70, 85, 80, 75, 90, 85, 100, 95, 85, 90, 70, 75]

tabel = pd.Series(data).value_counts().sort_index()
print(f"Tabel Persebaran data\n\n{tabel}")


import matplotlib.pyplot as plt

data = [65, 70, 75, 80, 85, 90, 95, 100, 70, 85, 80, 75, 90, 85, 100, 95, 85, 90, 70, 75]

plt.figure(figsize=(8, 6))
plt.hist(data, bins=7, edgecolor='black', alpha=0.7)
plt.title('Histogram')
plt.xlabel('Nilai')
plt.ylabel('Frekuensi')
plt.grid(False)
plt.show()


import matplotlib.pyplot as plt

data = [65, 70, 75, 80, 85, 90, 95, 100, 70, 85, 80, 75, 90, 85, 100, 95, 85, 90, 70, 75]

freq_table= pd.Series(data).value_counts().sort_index()

plt.figure(figsize=(8, 6))
plt.plot(freq_table.index, freq_table.values, marker='o', linestyle='-', color='b')
plt.title('Poligon Frekuensi')
plt.xlabel('Nilai')
plt.ylabel('Frekuensi')
```

```
Tabel Persebaran data

65     1
70     3
75     3
80     2
85     4
90     3
95     2
100    2
Name: count, dtype: int64
```



```
Text(0, 0.5, 'Frekuensi')
```


Poligon Frekuensi
Frekuensi
Nilai