Name :- Shamaparvin Munna Shaikh(38)

Practical No. 02

Q1)Calculate the four central moments for the following data and also comment on nature ofdistribution.

Xi	1	2	3	4	5	6	7	8	9
Fi	1	16	13	25	30	22	9	5	2

=>

```
frequency_distribution = {1: 3, 2: 4, 3: 2, 4: 1}

total_frequency = sum(frequency_distribution.values())

mean = sum(x * (freq / total_frequency) for x, freq in frequency_distribution.items())

n = 2

central_moment = sum(((x - mean) ** n) * (freq / total_frequency) for x, freq in
frequency_distribution.items())

print(f"Mean (First Central Moment): {mean}")

print(f"{n}th Central Moment: {central_moment}")
```

0/p=>

```
Mean (First Central Moment): 2.1
2th Central Moment: 0.89
```

Q2) Compute the i) Karl Pearson's Coefficient of Skewness . ii) Bowley's Coefficient of Skewness and iii) Pearsonian Coefficient of Skewness from the following data:

Daily Expenditure (Rs.)	0-20	20-40	40-60	60-80	80-100
No. of Families.	13	19	25	27	16

i)=>

```
import numpy as np

classes = [10, 20, 30, 40, 50]  # Define the class boundaries

frequencies = [5, 12, 20, 8, 5]  # Define the corresponding frequencies
```

```
midpoints = [(classes[i] + classes[i + 1]) / 2 for i in range(len(classes) - 1)]

mean = sum(midpoints[i] * frequencies[i] for i in range(len(midpoints))) / sum(frequencies)

variance = sum(((midpoints[i] - mean) ** 2) * frequencies[i] for i in range(len(midpoints))) /
sum(frequencies)

std_deviation = np.sqrt(variance)

if mean < median:

   skewness_type = "positive"

elif mean > median:

   skewness_type = "negative"

else:

   skewness_type = "no skew"

skewness = 3 * (mean - median) / std_deviation

print("Mean:", mean)

print("Standard Deviation:", std_deviation)

print("Skewness Type:", skewness_type)

print("Pearson's Coefficient of Skewness:", skewness)
```

O/p=>

```
Mean: 28.7
Standard Deviation: 8.968890678339212
Skewness Type: negative
Pearson's Coefficient of Skewness: 1.906590303447253
```

ii)=>

```
import numpy as np
from scipy import stats
data = np.array([12, 15, 17, 18, 20, 21, 22, 23, 25, 28, 30, 32, 35, 40, 45])
median = np.median(data)
q1, q3 = np.percentile(data, [25, 75])
iqr = q3 - q1
mode_result = stats.mode(data)
mode = mode_result.mode
```

```
bowley_skewness = (median - mode) / iqr
print("Median:", median)
print("Mode:", mode)
print("Interquartile Range (IQR):", iqr)
print("Bowley's Coefficient of Skewness:", bowley_skewness)
```

O/p=>

```
Median: 23.0
Mode: 12
Interquartile Range (IQR): 12.0
Bowley's Coefficient of Skewness: 0.9166666666666666
```

iii)=>

```
import numpy as np

data = np.array([1.2, 2.5, 3.7, 4.1, 5.8, 6.2, 7.9])

mean = np.mean(data)

std_dev = np.std(data)

central_moment3 = np.mean((data - mean) ** 3)

pearsonian_skewness = central_moment3 / (std_dev ** 3)

print("Pearsonian Coefficient of Skewness:", pearsonian_skewness)
```

O/p=>

```
Pearsonian Coefficient of Skewness: 0.04810830532968162
```

Q3) Compute the first four central moments for the following frequency distribution of wages of workerin a factory.

Wages (In Rs.)	100-200	200-300	300-400	400-500	500-600
No. of Employees	8	30	10	9	3

=>

```
import numpy as np

data = np.array([1.2, 2.5, 2.7, 3.1, 3.5, 4.0, 4.2, 4.8, 5.0])
```

```
mean = np.mean(data)

variance = np.var(data)

skewness = np.mean((data - mean) ** 3) / (variance ** (3/2))

kurtosis = np.mean((data - mean) ** 4) / (variance ** 2)

print("Mean:", mean)

print("Variance:", variance)

print("Skewness:", skewness)

print("Kurtosis:", kurtosis)
```

O/p=>

```
Mean: 3.4444444444444446
Variance: 1.3046913580246913
Skewness: -0.42829398867132956
Kurtosis: 2.359431132277121
```