FA_1 Rosales, Frances Aneth

No. 1 Problem:

GIVEN TABLE 2.14

Viewing Time Number of (minutes)	Number of (minutes) Students
300–399	14
400–499	46
500–599	58
600–699	76
700–799	68
800–899	62
900–999	48
1000–1099	22
1100–1199	6

(a) The upper limit of the fifth class

In [139…

```
classes_interval_lower = [300, 400, 500, 600, 700, 800, 900, 1000, 1100]
classes_interval_upper = [399, 499, 599, 699, 799, 899, 999, 1099, 1199]

fifth_upper_class = classes_interval_upper[4]
print("The upper limit of the fifth class:", fif_upp)
```

```
The upper limit of the fifth class: 799
```

(b) The lower limit of the eighth class

In [140…

```
classes_interval_lower = [300, 400, 500, 600, 700, 800, 900, 1000, 1100]
classes_interval_upper = [399, 499, 599, 699, 799, 899, 999, 1099, 1199]

eigth_lower_class = classes_interval_lower[7]

print("The lower limit of the eighth class:", eigth_lower_class)
```

```
The lower limit of the eighth class: 1000
```

(c) The class mark of the seventh class

In [160…

```
classes_interval_lower = [300, 400, 500, 600, 700, 800, 900, 1000, 1100]
classes_interval_upper = [399, 499, 599, 699, 799, 899, 999, 1099, 1199]


Seventh_class_lower = classes_interval_lower[6]
Seventh_class_upper = classes_interval_upper[6]

midpoint_seventh_class = (Seventh_class_upper + Seventh_class_lower) / 2

print("7th Lower Class:", Seventh_class_lower)
print("7th Upper Class:", Seventh_class_upper)
print("The class mark of the seventh class", midpoint_seventh_class)
```

```
7th Lower Class: 900
7th Upper Class: 999
The class mark of the seventh class 949.5
```

(d) The class boundaries of the last class

In [161…

```
classes_interval_lower = [300, 400, 500, 600, 700, 800, 900, 1000, 1100]
classes_interval_upper = [399, 499, 599, 699, 799, 899, 999, 1099, 1199]


last_class_lower = classes_interval_lower [8]
last_class_upper = classes_interval_upper [8]

midpoint_last_class = (last_class_upper + last_class_lower) / 2

print("Last Lower Class:", last_class_lower)
print("Last Upper Class:", last_class_upper)
print("The class boundaries of the last class", midpoint_last_class)
```

```
Last Lower Class: 1100
Last Upper Class: 1199
The class boundaries of the last class 1149.5
```

(e) The class-interval size

In [163…

```
classes_interval_lower = [300, 400, 500, 600, 700, 800, 900, 1000, 1100]
classes_interval_upper = [399, 499, 599, 699, 799, 899, 999, 1099, 1199]


interval_class_lower = (classes_interval_lower [8]-.5)#1100
interval_class_upper = (classes_interval_upper [8]+.5) #1199

interval_last_class = interval_class_upper-interval_class_lower

print("Lower Class Interval:", interval_class_lower)
print("Upper Class Interval :", interval_class_upper)
print("The class-interval size", interval_last_class)
```

```
Lower Class Interval: 1099.5
Upper Class Interval : 1199.5
The class-interval size 100.0
```

(f) The frequency of the fourth class

In [144…

```
num_students = [14, 46, 58, 76, 68, 62, 48, 22, 6]
fourth_num_students = num_students[3]

print("The frequency of the fourth class", fourth_num_students)
```

```
The frequency of the fourth class 76
```

(g) The relative frequency of the sixth class

In [145…

```
num_students = [14,46,58,76,68,62,48,22,6]

sixth_num_students = num_students[5]
total_students = sum(num_students)

rel_frequency= (sixth_num_students/total_students)*100
print("6th Class:", sixth_num_students)
print("Total No. of Students:", total_students)
print("The relative frequency of the sixth class:", rel_frequency,("%"))
```

```
6th Class: 62
Total No. of Students: 400
The relative frequency of the sixth class: 15.5 %
```

(h) The percentage of students whose weekly viewing time does not exceed 600 minutes

In [146…

```
total_students_class600 = 0
num_students = [14,46,58,76,68,62,48,22,6]


for i in range(3):
    total_students_class600 += num_students[i]

total_students = sum(num_students)

percent_wkly_600 =(total_students_class600/total_students)*100

print("600 below Class:", total_students_class600)
print("Total No. of Students:", total_students)
print("The percentage of students whose weekly viewing time does not exceed 600
```

```
600 below Class: 118
Total No. of Students: 400
The percentage of students whose weekly viewing time does not exceed 600 minutes:
29.5 %
```

(i) The percentage of students with viewing times greater than or equal to 900 minutes

In [147…

```
total_students_class900 = 0

for i in range(7):
    total_students_class900 += num_students[i]

total_students = sum(num_students)
percent_wkly_900 =(total_students_class900/total_students)*100

print("Greater than or equal to 900 Class:", total_students_class900)
print("Total No. of Students:", total_students)
print("The percentage of students with viewing times greater than or equal to 90
```

```
Greater than or equal to 900 Class: 372
Total No. of Students: 400
The percentage of students with viewing times greater than or equal to 900 minute
s: 93.0 %
```

(j) The percentage of students whose viewing times are at least 500 minutes but less than 1000 minutes

In [148…

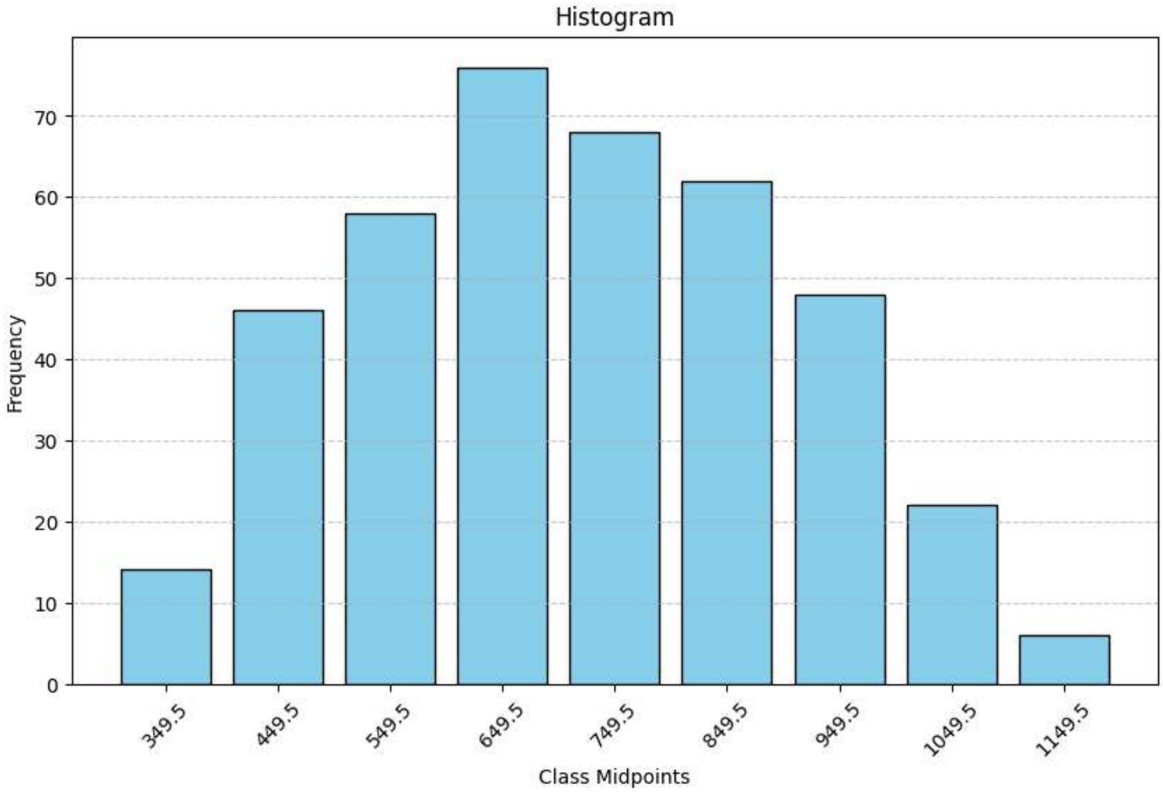
```
total_students_class_interval = 0
num_students = [14,46,58,76,68,62,48,22,6]

for i in range(3, 8):
    total_students_class_interval += num_students[i]

total_students = sum(num_students)
percent_wkly_interval =(total_students_class_interval/total_students)*100

print("Least 500 but less than 1000 Class:", total_students_class_interval)
print("Total No. of Students:", total_students)
print("The percentage of students with viewing times greater than or equal to 90
Least 500 but less than 1000 Class: 276
Total No. of Students: 400
The percentage of students with viewing times greater than or equal to 900 minute
s: 69.0 %
```

2.21

(a) a histogram

In [157…

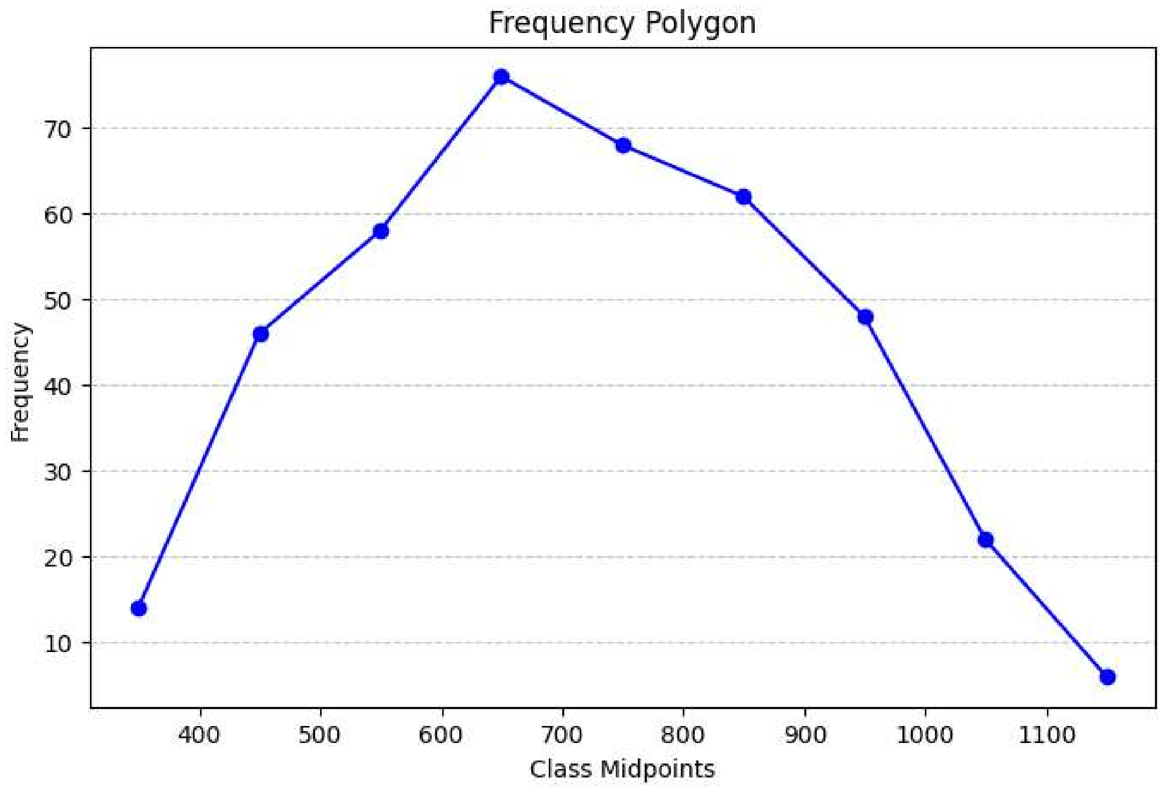
```python
import matplotlib.pyplot as plt

# Class interval boundaries
classes_interval_lower = [300, 400, 500, 600, 700, 800, 900, 1000, 1100]
classes_interval_upper = [399, 499, 599, 699, 799, 899, 999, 1099, 1199]

num_students = [14, 46, 58, 76, 68, 62, 48, 22, 6]
class_midpoints = [0] * 9
for i in range(9):
    class_midpoints[i] = ((classes_interval_lower[i] + classes_interval_upper[i]


plt.figure(figsize=(10, 6))
plt.bar(class_midpoints, num_students, width=80, color='skyblue', edgecolor='bla
plt.xlabel("Class Midpoints")
plt.ylabel("Frequency")
plt.title("Histogram")
plt.xticks(class_midpoints, rotation=45)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```



(b) Frequency Polygon

In [158…

```
plt.figure(figsize=(8, 5))
plt.plot(class_midpoints, num_students, marker='o', linestyle='-', color='blue')
plt.xlabel("Class Midpoints")
plt.ylabel("Frequency")
plt.title("Frequency Polygon")
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```



2.22

(a) a relative-frequency distribution,

In [180…

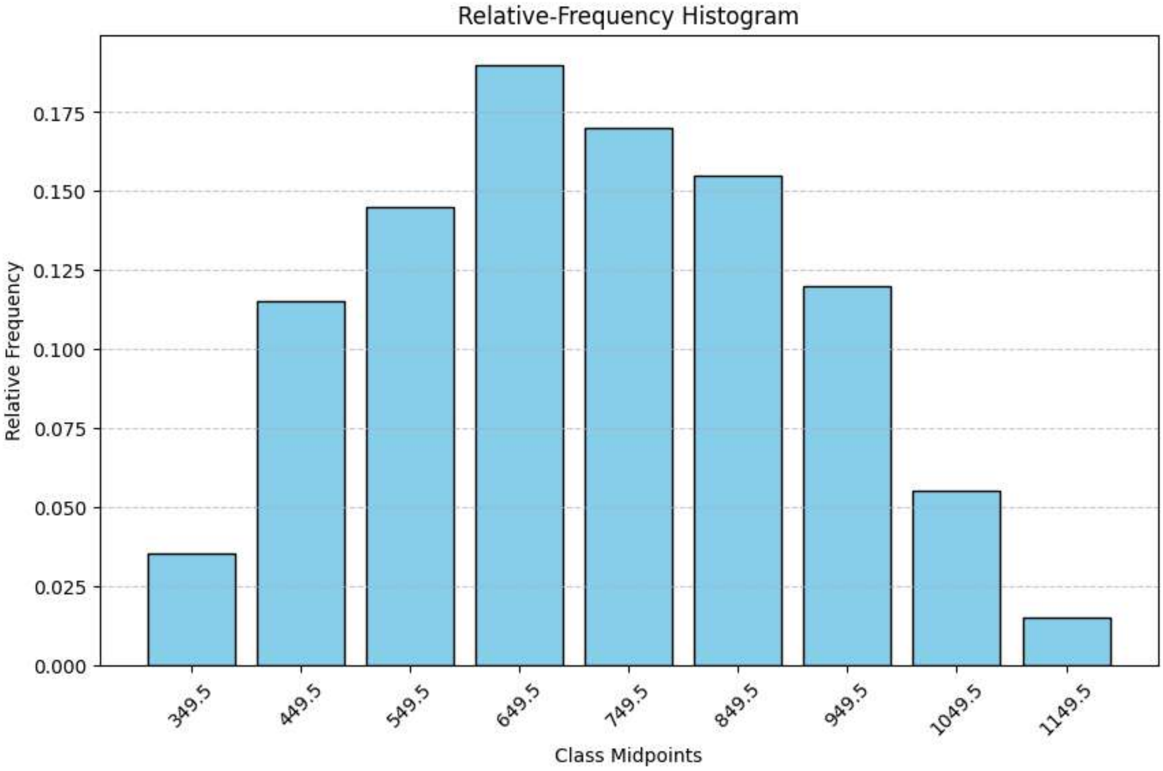
```
classes_interval_lower = [300, 400, 500, 600, 700, 800, 900, 1000, 1100]
classes_interval_upper = [399, 499, 599, 699, 799, 899, 999, 1099, 1199]

num_students = [14, 46, 58, 76, 68, 62, 48, 22, 6]

total_data_points = sum(num_students)

fre_each = [0] * 9

for i in range(9):
    fre_each[i] = num_students[i] / total_data_points

for i in range(9):
    print(f"Relative Frequency {fre_each[i]:.4f}: {(fre_each[i])*100:.2f}%")
```

```
Relative Frequency 0.0350: 3.50%
Relative Frequency 0.1150: 11.50%
Relative Frequency 0.1450: 14.50%
Relative Frequency 0.1900: 19.00%
Relative Frequency 0.1700: 17.00%
Relative Frequency 0.1550: 15.50%
Relative Frequency 0.1200: 12.00%
Relative Frequency 0.0550: 5.50%
Relative Frequency 0.0150: 1.50%
```

(b) a relative frequency histogram

In [151…

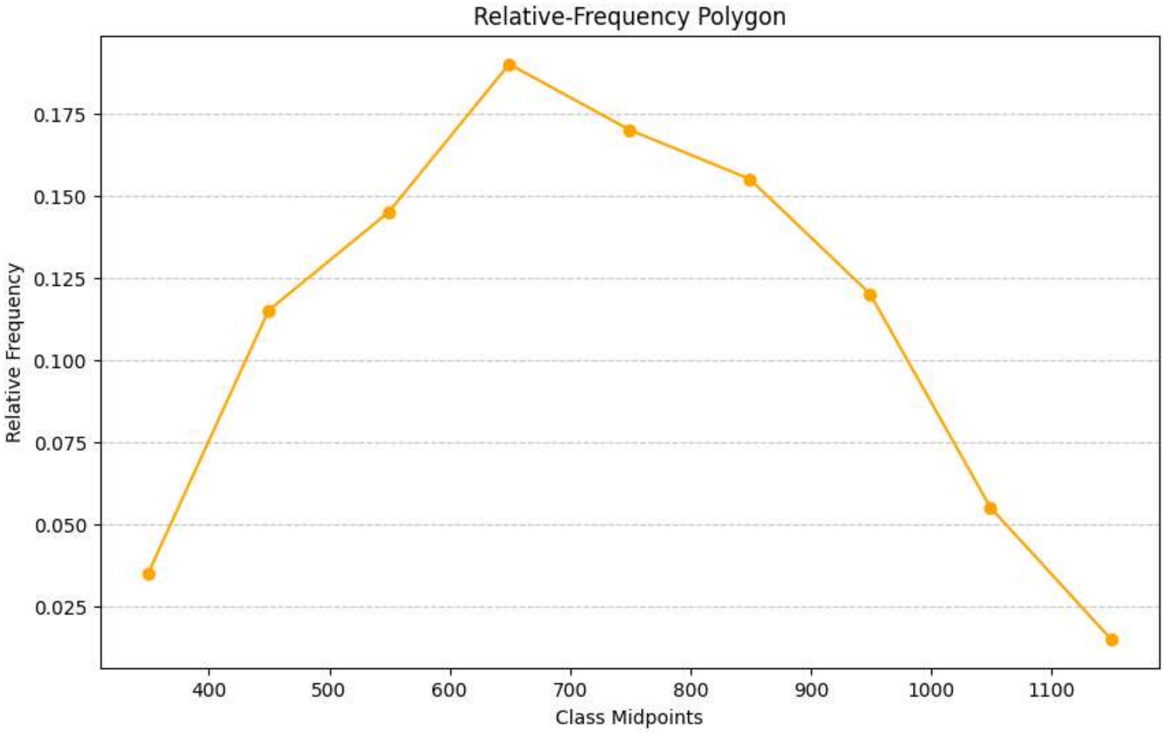
```
import matplotlib.pyplot as plt
plt.figure(figsize=(10, 6))
plt.bar(class_midpoints, fre_each, width=80, color='skyblue', edgecolor='black')
plt.xlabel("Class Midpoints")
plt.ylabel("Relative Frequency")
plt.title("Relative-Frequency Histogram")
plt.xticks(class_midpoints, rotation=45)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```



c. Create the relative-frequency polygon

In [152…

```
plt.figure(figsize=(10, 6))
plt.plot(class_midpoints, fre_each, marker='o', linestyle='-', color='orange')
plt.xlabel("Class Midpoints")
plt.ylabel("Relative Frequency")
plt.title("Relative-Frequency Polygon")
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```


Relative-Frequency Polygon
Relative Frequency
0.025
0.050
0.075
0.100
0.125
0.150
0.175
400
500
600
700
800
900
1000
1100
Class Midpoints