Data Description

Q1:

BEGIN  
  
1. Initialize a list representing trust levels with repeated values:  
 - 'A' appears 4 times  
 - 'M' appears 28 times  
 - 'H' appears 6 times  
 - 'S' appears 2 times  
  
2. Convert the list to a data series  
  
3. Calculate frequency of each category in the series  
  
4. Calculate relative frequency by dividing frequency by total number of responses  
  
5. Convert relative frequency to percentage  
  
6. Define the order of categories as: ['M', 'H', 'A', 'S']  
  
7. Create a table with columns:  
 - Category (unnamed column)  
 - Frequency  
 - Relative Frequency  
 - Relative Frequency (%)  
  
8. Print the table  
  
9. Create a 2x2 grid layout for plotting (4 subplots)  
  
10. Display the table as text in subplot (1,0):  
 - Turn off axis  
 - Display text in monospace font  
  
11. Plot bar chart in subplot (0,0):  
 - X-axis: Category  
 - Y-axis: Frequency  
 - Add gridlines on Y-axis  
 - Annotate each bar with frequency value  
  
12. Plot pie chart in subplot (0,1):  
 - Use category frequencies  
 - Show percentage on each slice  
 - Remove y-axis label for clarity  
  
13. Leave subplot (1,1) empty (axis turned off)  
  
14. Adjust layout to prevent overlapping  
  
15. Display the plots  
  
END

Q2:

BEGIN  
1. Initialize two lists:  
 - List of grades: ['F', 'D', 'D+', 'C', 'C+', 'B', 'B+', 'A']  
 - Corresponding number of students: [8, 10, 15, 40, 25, 10, 7, 5]  
  
2. Create a data table (DataFrame) with columns:  
 - "Grade"  
 - "Number of Students"  
  
3. Calculate total number of students.  
  
4. For each grade:  
 - Calculate relative frequency = number of students / total number of students  
 - Convert relative frequency to percentage  
  
5. Add two new columns to the table:  
 - "Relative Frequency"  
 - "Relative Frequency (%)"  
6. Print the data table  
7. Create a 2x2 grid layout for plotting (4 subplots)  
8. Plot bar chart in subplot (0,0):  
 - X-axis: Grade  
 - Y-axis: Number of Students  
 - Add gridlines on Y-axis  
 - Annotate each bar with number of students  
  
9. Plot pie chart in subplot (0,1):  
 - Use grade as labels  
 - Show percentage on each slice  
 - Equal aspect ratio for a perfect circle  
  
10. In subplot (1,0), display the table:  
 - Convert data table to string  
 - Turn off axis  
 - Display text in monospace font  
  
11. Leave subplot (1,1) empty (axis turned off)  
  
12. Adjust layout to prevent overlapping  
  
13. Display the plots

END

Q3:

BEGIN  
  
1. Define class boundaries (bins) for the data:  
 - [21, 27.5, 34, 40.5, 47, 53.5, 60]  
  
2. Define class interval labels corresponding to bins:  
 - ["21–27.5", "27.5–34", "34–40.5", "40.5–47", "47–53.5", "53.5–60"]  
  
3. Define frequency values for each class:  
 - [6, 6, 6, 6, 6, 6]  
  
4. Calculate midpoints for each class interval using bin boundaries  
  
5. Calculate total frequency  
  
6. For each class:  
 - Compute relative frequency = frequency / total  
 - Compute percentage frequency = relative frequency \* 100  
  
7. Compute cumulative frequency (running total of frequencies)  
  
8. Create and print a table containing:  
 - Class Interval, Midpoint, Frequency  
  
9. Define and print summary statistics:  
 - Maximum value, Minimum value, Class Width  
  
10. Extend midpoints and frequencies to include 0 values at the start and end for polygon plotting  
  
11. Create a 1x3 layout for visualizations (3 subplots)  
  
12. Plot Histogram (subplot 0):  
 - X-axis: Class Interval  
 - Y-axis: Frequency  
 - Use bars to represent frequency  
 - Add gridlines  
  
13. Plot Frequency Polygon (subplot 1):  
 - X-axis: Midpoint  
 - Y-axis: Frequency  
 - Connect points with lines and markers  
 - Add gridlines  
  
14. Plot Frequency Curve (subplot 2):  
 - Use spline interpolation to smooth the curve  
 - X-axis: Midpoint  
 - Y-axis: Smoothed Frequency  
 - Add gridlines  
  
15. Adjust layout for readability  
  
16. Display the plots  
  
END

Q4:

BEGIN

1. Define class intervals and corresponding frequency values:

- Class Intervals: ['55 - 60', '60 - 65', ..., '90 - 95']

- Frequencies: [8, 10, 15, 20, 12, 8, 4, 2]

2. Extract lower and upper bounds from each class interval string

3. Calculate the midpoint for each class interval:

- Midpoint = (Lower Bound + Upper Bound) / 2

4. Calculate Modified Cumulative Frequency (C.F):

- First value is the first frequency

- Each subsequent value is the current frequency + previous frequency

5. Create a table with columns:

- Class Interval, Midpoint, Frequency, C.F

6. Create a 2x3 subplot layout with an overall title "Statistical Charts"

7. Plot Histogram in subplot [0, 0]:

- X-axis: Class Interval

- Y-axis: Frequency

- Add text labels above each bar to show frequency

8. Plot Frequency Polygon in subplot [0, 1]:

- Add 0-frequency at the beginning and end for continuity

- Plot midpoints vs frequencies with markers

9. Plot Frequency Curve in subplot [0, 2]:

- Use spline interpolation to smooth the line

- Plot smooth curve using new x-values and interpolated y-values

10. Plot Frequency Ogive in subplot [1, 0]:

- Plot both frequency and cumulative frequency

- Add markers and a legend for clarity

11. Display the table in subplot [1, 1]:

- Convert DataFrame to formatted text

- Turn off axis

- Show table text using monospaced font

12. Leave subplot [1, 2] blank:

- Axis turned off

13. Adjust layout spacing and display all plots

END

Q5:

BEGIN

1. Define class intervals and corresponding frequency values:

- Class Intervals: ['20 - 30', '30 - 40', ..., '80 - 90']

- Frequencies: [8, 12, 25, 32, 22, 18, 5]

2. Extract lower and upper bounds from each class interval string

3. Calculate the midpoint for each class:

- Midpoint = (Lower Bound + Upper Bound) / 2

4. Calculate:

- Relative Frequency = Frequency / Total Frequency

- Relative Frequency (%) = Relative Frequency \* 100

5. Create a data table with columns:

- Class Interval, Midpoint, Frequency, Relative Frequency, Relative Frequency (%)

6. Set up a 2x2 subplot layout with an overall title "Statistical Charts"

7. Plot Histogram in subplot [0, 0]:

- X-axis: Class Interval

- Y-axis: Frequency

- Plot bars and annotate with values above each bar

8. Plot Frequency Polygon in subplot [0, 1]:

- Extend midpoints and frequencies with 0s at both ends

- Connect points with lines and markers

9. Plot Frequency Curve in subplot [1, 0]:

- Use spline interpolation for a smooth curve

- X-axis: Class of degree

- Y-axis: Frequency

10. Display data table as text in subplot [1, 1]:

- Convert DataFrame to formatted string

- Turn off axis and render text in monospaced font

11. Adjust layout spacing and display all plots

END

Descriptive Statistics

Q1:

BEGIN

1. Define a list of numeric data values

2. Convert the list into a pandas Series

3. Calculate the following descriptive statistics:

- Sample Size (n)

- Mean

- Median

- Mode (as a list)

- Minimum and Maximum

- Range = Maximum - Minimum

- First Quartile (Q1)

- Second Quartile (Q2/Median)

- Third Quartile (Q3)

- Interquartile Range (IQR) = Q3 - Q1

- Variance

- Standard Deviation

- Geometric Mean

- Harmonic Mean

- Coefficient of Variation (CV %) = (Standard Deviation / Mean) \* 100

4. Store all statistics in a dictionary

5. Convert the dictionary to a table (DataFrame) with columns:

- "Statistic"

- "Value"

6. Print each statistic:

- If the value is a list (like mode), print all elements

- If the value is numeric, print with 2 decimal places

7. Create a 1x1 plot to display the statistics table as text

8. Turn off the axis and render the table text using monospace font

9. Adjust layout and display the plot

END

Q2:

BEGIN

1. Define four datasets:

- Income Data

- Rating Data

- Age Data

- Education Data

2. Convert each dataset into a pandas Series

3. For each dataset, calculate the following:

- Sample Size (n)

- Mean

- Median

- Mode (if no clear mode, use "No Mode")

- Geometric Mean

- Harmonic Mean

- Minimum and Maximum

- Range = Maximum - Minimum

- First Quartile (Q1)

- Second Quartile (Q2)

- Third Quartile (Q3)

- Inter-Quartile Range (IQR) = Q3 - Q1

- Variance

- Standard Deviation

- Coefficient of Variation (CV%) = (Standard Deviation / Mean) \* 100

4. Store the results of each dataset in separate dictionaries:

- stats\_dicti for Income

- stats\_dictr for Rating

- stats\_dicta for Age

- stats\_dicte for Education

5. Print formatted descriptive statistics for each dataset:

- For each statistic, print name and value (2 decimal places for floats)

6. Calculate the correlation coefficient between Income and Education using:

- Pearson correlation formula via numpy

7. Print the correlation value

8. Perform simple linear regression:

- Use Education as independent variable (x)

- Use Income as dependent variable (y)

- Calculate slope and intercept of the regression line using `np.polyfit`

9. Print slope and intercept of the regression line

END

Q3:

BEGIN

1. Define two datasets:

- First Midterm scores

- Final Midterm scores

2. Calculate the Pearson Correlation Coefficient between:

- Final Midterm and First Midterm scores

3. Perform linear regression using:

- Dependent variable (Y): First Midterm

- Independent variable (X): Final Midterm

- Get slope (b), intercept (a), r-value, p-value, and standard error

4. Choose an X-value for prediction (e.g., X = 24)

5. Compute the predicted Y-value using the regression line:

- y\_pred = a + b \* X\_value

6. Format and prepare a results summary string showing:

- Correlation Coefficient

- Slope

- Intercept

- Regression equation

- Prediction result

7. Create a 1x2 subplot layout:

- First subplot: Scatter Plot of student scores

- Plot First Midterm and Final Midterm scores

- Label axes and add legend

- Enable grid

- Second subplot: Display formatted statistical results

- Turn off axis

- Show formatted results string in monospace font

8. Adjust layout and display the plots

END

Some Probability Distributions

BEGIN

Q1: Binomial Distribution

1. Define parameters for binomial distribution:

- Number of trials (n) = 4

- Probability of success (p) = 0.35

2. Calculate and display individual binomial probabilities:

- P(X = 0)

- P(X = 1)

- P(X = 2)

- P(X = 3)

- P(X = 4)

3. Calculate cumulative probabilities (P(X ≤ k)):

- P(X ≤ 2)

- P(X ≤ 3)

4. Calculate upper tail probabilities (P(X ≥ k)):

- P(X ≥ 2) = 1 - P(X ≤ 1)

- P(X ≥ 3) = 1 - P(X ≤ 2)

5. Compute expected value, variance, and standard deviation:

- E(X) = n \* p

- V(X) = n \* p \* (1 - p)

- S(X) = sqrt(V(X))

Q2: Standard Normal Distribution (Z ~ N(0,1))

6. Compute and display standard normal probabilities:

- P(Z < 0.67)

- P(Z < 1.96)

- P(-1.23 < Z < 2.30) = P(Z < 2.30) - P(Z < -1.23)

Q3: Normal Distribution with μ = 12, σ = 4

7. Define parameters:

- Mean (μ) = 12

- Standard Deviation (σ) = 4

8. Compute probabilities for the normal distribution:

- P(X > 9) = 1 - P(X ≤ 9)

- P(X < 10)

- P(8 < X < 14) = P(X < 14) - P(X < 8)

END