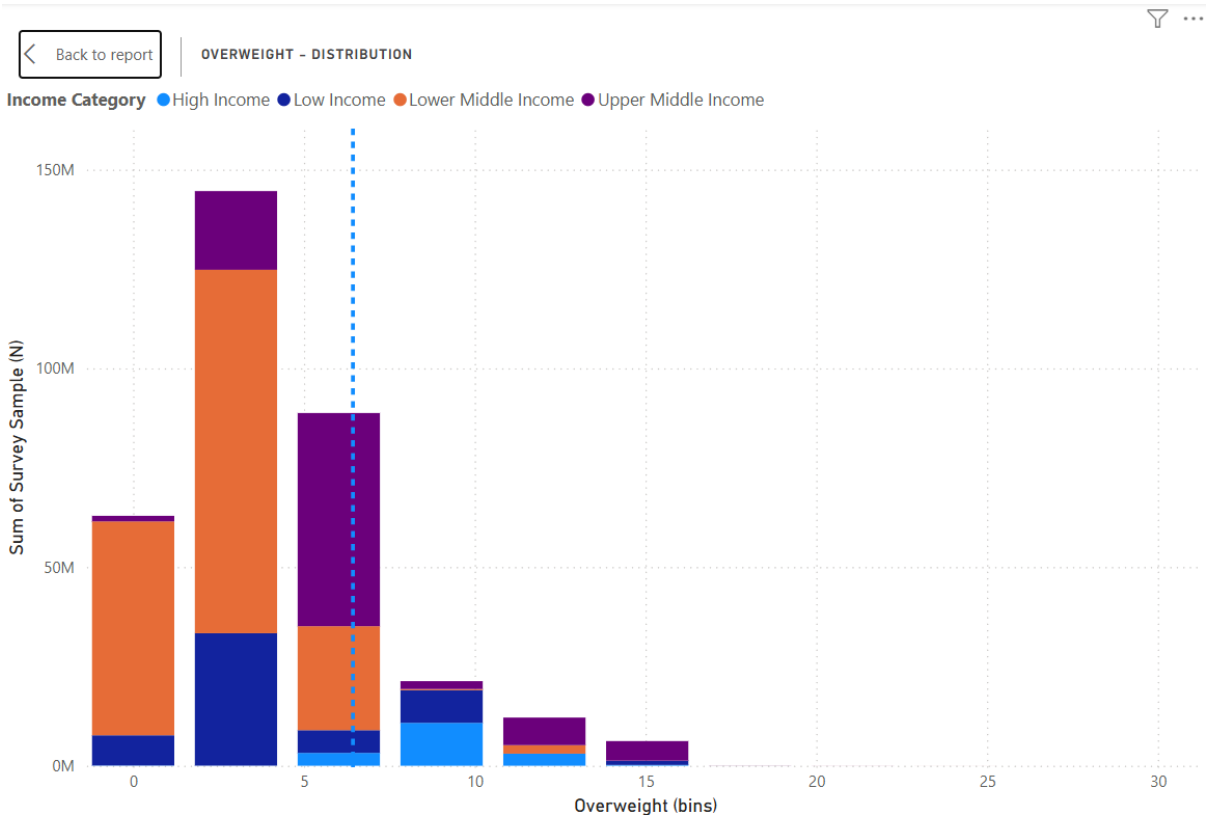
Overweight Distribution:

Description:

- The first graph shows the distribution of overweight status across different income categories.
- The x-axis represents "Overweight (bins)" with ranges to categorize overweight individuals.
- The y-axis shows the "Sum of Survey Sample (N)," which indicates the number of individuals surveyed in each category.
- The bars are color-coded based on income categories:
    - High Income: Blue
    - Low Income: Orange
    - Lower Middle Income: Light Blue
    - Upper Middle Income: Purple

Visualization:



Code:

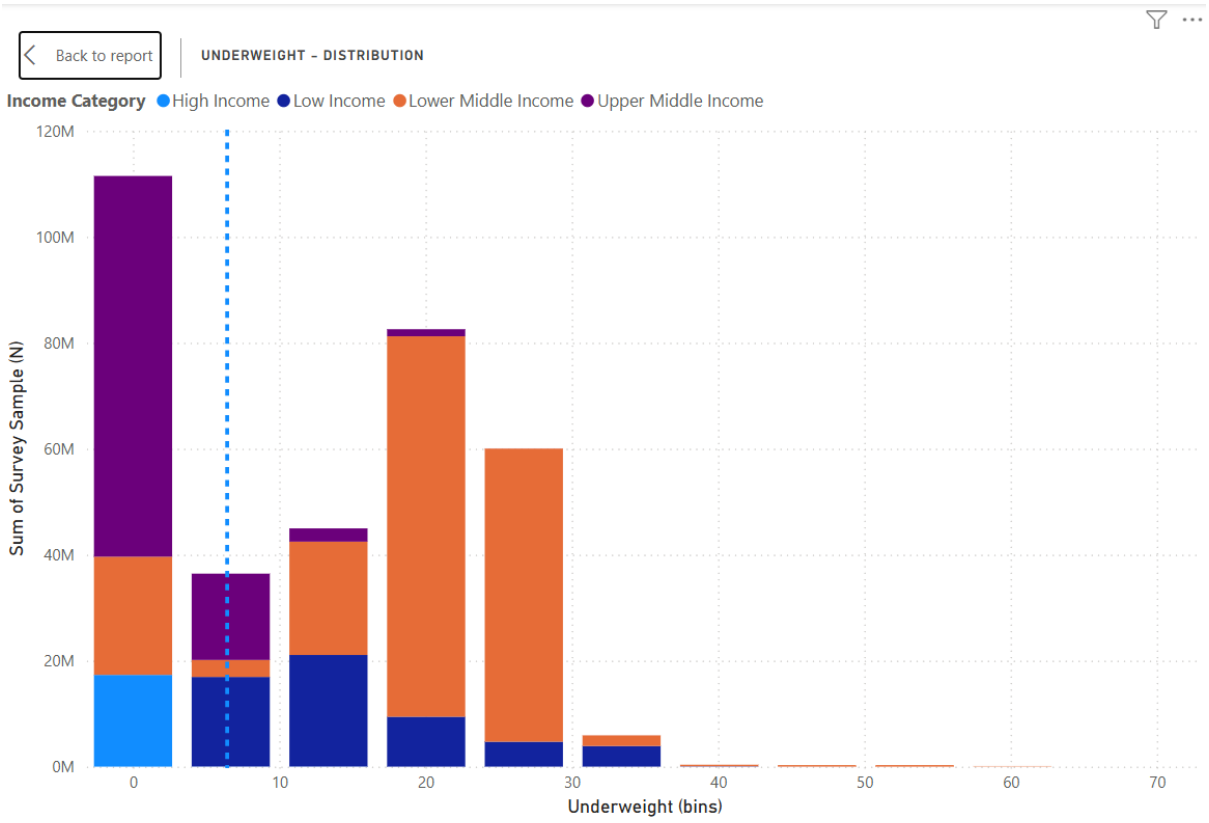
```
import matplotlib.pyplot as plt

import numpy as np


# Sample Data

bins = [0, 5, 10, 15, 20, 25, 30]

high_income = [30, 50, 10, 5, 2, 1]  # example data

low_income = [40, 90, 20, 10, 5, 2]   # example data

lower_middle_income = [10, 20, 5, 3, 1, 0]  # example data

upper_middle_income = [20, 10, 2, 1, 0, 0] # example data


# Create Bar Plot

barWidth = 0.85

r1 = np.arange(len(bins))  # positions of bars


plt.bar(r1, high_income, color='blue', edgecolor='grey', label='High Income')

plt.bar(r1, low_income, bottom=high_income, color='orange', edgecolor='grey', label='Low Income')

plt.bar(r1, lower_middle_income, bottom=np.array(high_income)+np.array(low_income),

    color='lightblue', edgecolor='grey', label='Lower Middle Income')

plt.bar(r1, upper_middle_income,
bottom=np.array(high_income)+np.array(low_income)+np.array(lower_middle_income),

    color='purple', edgecolor='grey', label='Upper Middle Income')


plt.xlabel('Overweight (bins)')

plt.ylabel('Sum of Survey Sample (N)')

plt.title('Overweight - Distribution')

plt.xticks(r1, bins)

plt.legend()

plt.show()
```

Underweight Distribution:

Description:

- The second graph illustrates the distribution of underweight status across different income categories.
- Similar to the first graph, the axes and color coding follow the same format.
- The focus is on underweight individuals represented in various income categories.

Visualization:



Code:

Sample Data for Underweight

2underweight_high_income = [20, 30, 10, 5, 1, 0]

3underweight_low_income = [50, 100, 40, 20, 5, 1]

```
4underweight_lower_middle_income = [30, 20, 5, 2, 0, 0]

5underweight_upper_middle_income = [10, 10, 2, 1, 0, 0]

6

7plt.bar(r1, underweight_high_income, color='blue', edgecolor='grey', label='High Income')

8plt.bar(r1, underweight_low_income, bottom=underweight_high_income, color='orange',
edgecolor='grey', label='Low Income')

9plt.bar(r1, underweight_lower_middle_income,
bottom=np.array(underweight_high_income)+np.array(underweight_low_income),

10      color='lightblue', edgecolor='grey', label='Lower Middle Income')

11plt.bar(r1, underweight_upper_middle_income,
bottom=np.array(underweight_high_income)+np.array(underweight_low_income)

12      +np.array(underweight_lower_middle_income), color='purple', edgecolor='grey',
label='Upper Middle Income')

13

14plt.xlabel('Underweight (bins)')

15plt.ylabel('Sum of Survey Sample (N)')

16plt.title('Underweight - Distribution')

17plt.xticks(r1, bins)

18plt.legend()

19plt.show()
```

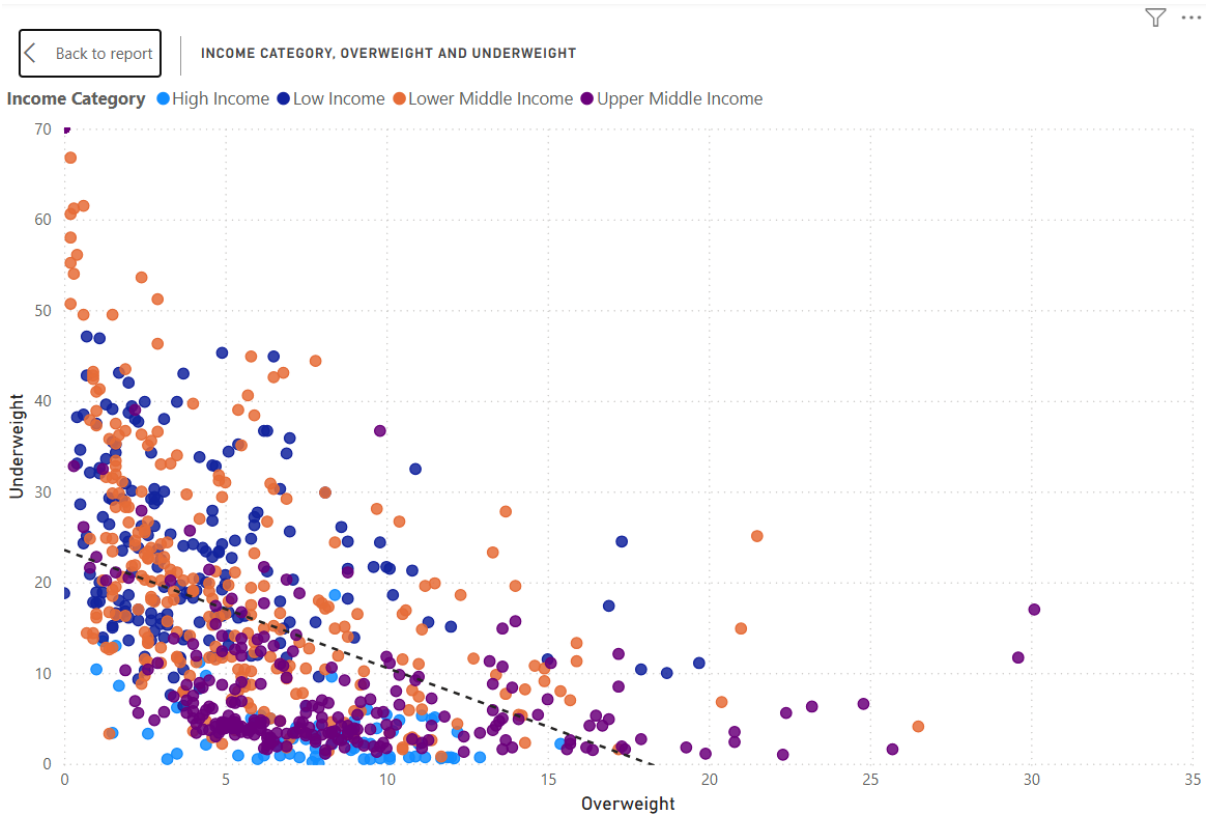
Relationship Between Overweight and Underweight:

Description:

- The third graph is a scatter plot showing the relationship between overweight and underweight status across different income categories.
- Each point represents an individual or an aggregated sample within a specific income category.
- The income categories are color-coded similarly to the previous graphs.

- The trend indicates that as overweight measurements increase, underweight measurements tend to decrease.

Visualization:



Code:

```
# Sample data for scatter plot

2overweight_mm = np.random.randint(0, 30, size=200)

3underweight_mm = np.random.randint(0, 70, size=200)

4income_category = np.random.choice(['High', 'Low', 'Lower Middle', 'Upper Middle'], size=200)

5

6# Define colors for each income category

7colors = {

8    'High': 'blue',

9    'Low': 'orange',

10    'Lower Middle': 'lightblue',
```

```
    'Upper Middle': 'purple'
}

plt.figure(figsize=(10, 6))
for category in colors.keys():
    plt.scatter(overweight_mm[income_category == category],
                underweight_mm[income_category == category],
                label=category, color=colors[category])

plt.xlabel('Overweight')
plt.ylabel('Underweight')
plt.title('Income Category, Overweight and Underweight')
plt.legend()
plt.savefig('scatter_plot.png')  # Save plot
plt.show()
```

Survey Sample Distribution by Income Category:

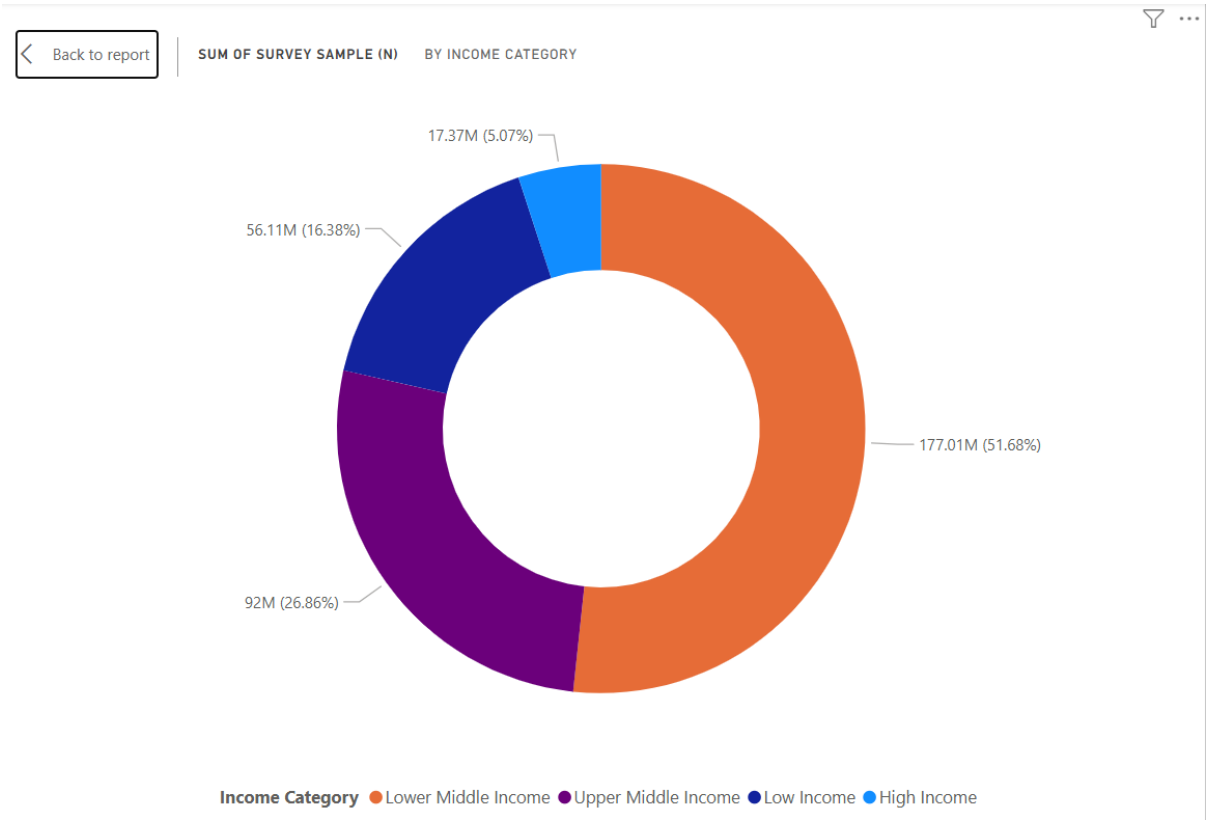
- **Type of Graph**: Donut Chart
- **Purpose**: Displays the distribution of survey samples across different income categories.

Income Categories:

- **Lower Middle Income**:
    - **Value**: 177.01M
    - **Percentage**: 51.68%
- **Upper Middle Income**:
    - **Value**: 56.11M
    - **Percentage**: 16.38%
- **Low Income**:
    - **Value**: 92M
    - **Percentage**: 26.86%

- **High Income**:
    - **Value**: 17.37M
    - **Percentage**: 5.07%

Visualization:



Code:

```
import matplotlib.pyplot as plt

2

3# Data for the doughnut chart

4labels = ['Lower Middle Income', 'Upper Middle Income', 'Low Income', 'High Income']

5sizes = [177.01, 56.11, 92, 17.37]  # in millions

6colors = ['#e59733', '#9f4e93', '#4b77be', '#5bc0de']

7explode = (0.1, 0, 0, 0)  # explode Lower Middle Income

8

9# Create a pie chart
```

```
plt.figure(figsize=(8, 8))
plt.pie(sizes, explode=explode, labels=labels, colors=colors,
        autopct='%1.2f%%', startangle=90, shadow=True)

# Draw center circle for doughnut
centre_circle = plt.Circle((0, 0), 0.70, fc='white')
fig = plt.gcf()
fig.gca().add_artist(centre_circle)

plt.axis('equal')  # Equal aspect ratio ensures that the pie is drawn as a circle.
plt.title('Sum of Survey Sample (N) by Income Category')
plt.show()
```

LDC, LIFD, and Stunting Metrics by Income Category:

- **Type of Graph**: Stacked Area Chart
- **Purpose**: Displays trends of various metrics (LDC, LIFD, LLDC/SID2, and Average of Stunting) across different income categories.

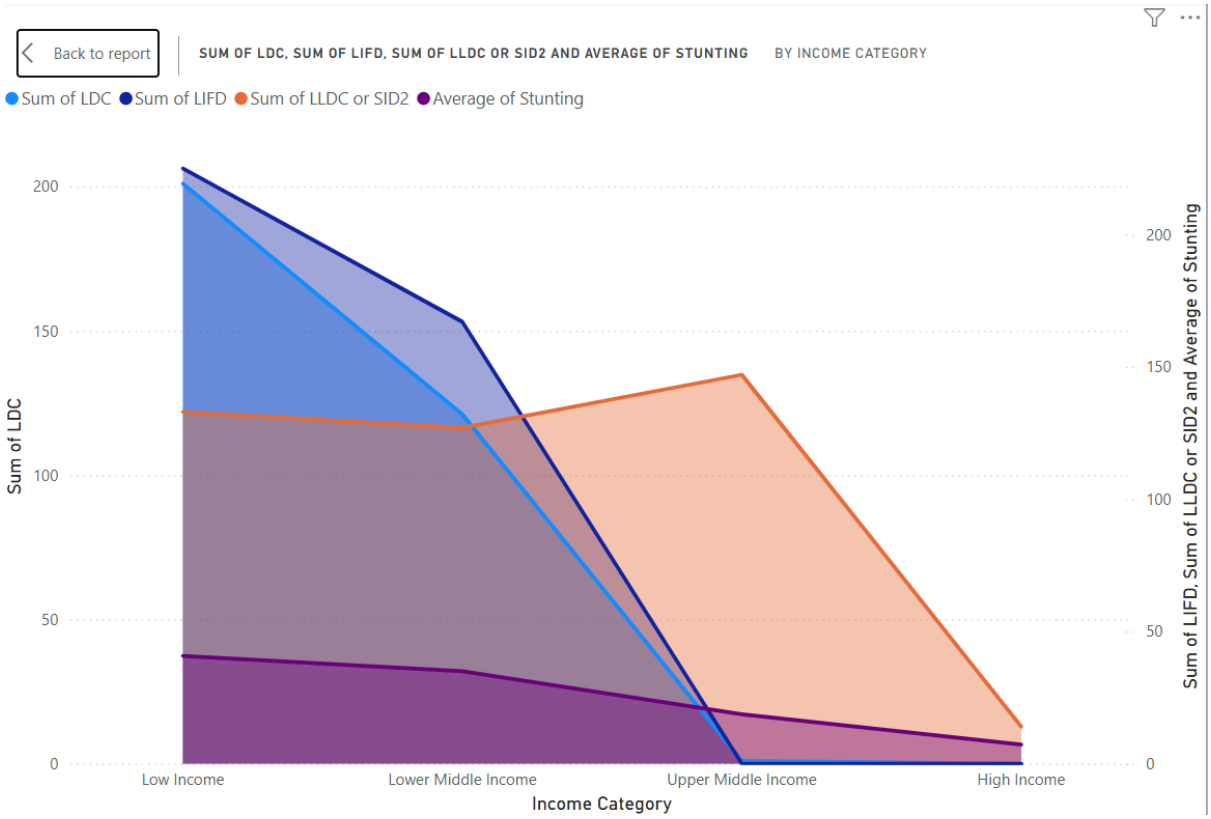
Income Categories:

- **Low Income**: Highest values, particularly for LDC metric.
- **Lower Middle Income**: Moderate levels for all metrics.
- **Upper Middle Income**: Decreasing trend for LDC and LIFD metrics.
- **High Income**: Lowest overall values in all categories, with metrics trailing off towards zero.

Color Representations:

- **LDC**: Blue
- **LIFD**: Orange
- **LLDC or SID2**: Purple
- **Average of Stunting**: Represents the last part of the range in a layered fashion.

Visualization:



Code:

```
import numpy as np

import matplotlib.pyplot as plt


# Data for the stacked area chart

income_categories = ['Low Income', 'Lower Middle Income', 'Upper Middle Income', 'High Income']

ldc = [200, 160, 100, 80]

lifd = [150, 120, 90, 70]

lldc_or_sid2 = [100, 80, 40, 30]

average_stunting = [50, 60, 30, 20]


x = np.arange(len(income_categories))


plt.figure(figsize=(10, 6))


# Create stacked area chart
```

```
plt.fill_between(x, ldc, color="#5bc0de", label='Sum of LDC', alpha=0.6)

plt.fill_between(x, ldc, ldc + lifd, color="#4b77be", label='Sum of LIFD', alpha=0.6)

plt.fill_between(x, ldc + lifd, ldc + lifd + lldc_or_sid2, color="#9f4e93", label='Sum of LLDC or SID2', alpha=0.6)

plt.plot(x, ldc + lifd + lldc_or_sid2, color="#e59733", label='Average of Stunting', linewidth=2)


# Add labels and title

plt.xticks(x, income_categories)

plt.ylabel('Sum of LDC, LIFD, LLDC/SID2, and Average of Stunting')

plt.title('Sum of LDC, LIFD, LLDC/SID2, and Average of Stunting by Income Category')

plt.legend()

plt.grid(True)


plt.show()
```

More Visualizations :