Assignment

Dataset: Sales Dataset

Libraries Used: Numpy, Pandas

Problem Statements and Solutions

1. Find the total sales amount.

```
sales['Sales Amount'].sum()
```

2. Calculate the average sales amount.

```
sales['Sales Amount'].mean()
```

3. Find the highest sales amount recorded.

```
sales['Sales Amount'].max()
```

4. Identify the lowest sales amount recorded.

```
sales['Sales Amount'].min()
```

5. Count the number of sales transactions.

```
sales.shape[0]
```

6. Find the total sales amount for each product.

```
sales.groupby('Product Name')['Sales Amount'].sum()
```

7. Identify the product with the highest sales.

```
sales.groupby('Product Name')['Sales Amount'].sum().idxmax()
```

8. Calculate the number of sales per region.

```
sales['Region'].value_counts()
```

9. Find the total sales amount per region.

```
sales.groupby('Region')['Sales Amount'].sum()
```

10. Determine the month with the highest total sales.

```
sales.groupby(sales['Date'].dt.month)['Sales Amount'].sum().idxmax()
```

11. Find the average sales amount per month.

```
sales.groupby(sales['Date'].dt.month)['Sales Amount'].mean()
```

12. Check how many unique products were sold.

```
sales['Product Name'].nunique()
```

13. List all the unique regions where sales happened.

```
sales['Region'].unique()
```

14. Find total sales amount for a specific product (e.g., 'Laptop').

```
sales[sales['Product Name'] == 'Laptop']['Sales Amount'].sum()
```

15. Filter all sales above Rs.10,000.

```
sales[sales['Sales Amount'] > 10000]
```

16. Calculate the percentage contribution of each product to total sales.

```
sales.groupby('Product Name')['Sales Amount'].sum() / sales['Sales Amount'].sum() * 100
```

17. Identify top 5 highest sales transactions.

```
sales.nlargest(5, 'Sales Amount')
```

18. Find the median sales amount.

```
sales['Sales Amount'].median()
```

19. Check if there are any missing values in the dataset.

```
sales.isnull().sum()
```

20. Replace missing sales amount values with the mean sales amount.

```
sales['Sales Amount'].fillna(sales['Sales Amount'].mean(), inplace=True)
```

Notes:

- Adjust the column names as per the actual dataset.
- Minimum fields required: Product Name, Region, Sales Amount, and Date.
- NumPy functions like np.sum(), np.mean() can also be used where appropriate.