

Sales Data Analysis – Documentation Guideline

Project Overview:

The Sales Data Analysis Project demonstrates how to use Python and the pandas library to analyse a structured dataset of product sales.

The goals of the project are:

- Load and explore sales data from a CSV file.
- Clean the dataset by handling missing values and removing duplicates.
- Perform analysis to calculate key metrics such as total revenue, best-selling product, and regional sales distribution.
- Generate a formatted report with insights and visual documentation.

This project strengthens skills in data analysis, data cleaning, and report generation using Python.

Setup Instructions:

- **Install Python**
 - Download and install Python 3.8 or higher from [python.org](https://www.python.org/downloads/).
 - Verify installation:
Bash python --version
- **Install VS Code**
 - Download and install [Visual Studio Code](https://code.visualstudio.com/).
 - Open VS Code after installation.
- **Create a Project Folder**

Make a new folder for your project.
- **Open Folder in VS Code**

In VS Code, go to File → Open Folder and select sales_data_analysis.
- **Add Files**

Inside the folder, create the following files:

 - Sales_analysis.py → main Python program
 - Analysis_report.md → documentation
 - Sales_data.csv → dataset
 - screenshot.png → screenshot of program output
- **Run the program**

```
python sales_analysis.py
```

Code Structure:

- sales_analysis.py → Loads, cleans, and analyses the dataset.
- sales_data.csv → Raw dataset with product sales information.
- analysis_report.md → Output report summarising findings.
- requirements.txt → Lists dependencies for reproducibility.
- screenshots → Contains images showing program execution and dataset preview.

Visual Documentation:

Preview of the dataset:

First 5 rows of the dataset:

| | Date | Product | Quantity | Price | Customer_ID | Region | Total_Sales |
|---|------------|------------|----------|-------|-------------|--------|-------------|
| 0 | 2024-01-01 | Phone | 7 | 37300 | CUST001 | East | 261100 |
| 1 | 2024-01-02 | Headphones | 4 | 15406 | CUST002 | North | 61624 |
| 2 | 2024-01-03 | Phone | 2 | 21746 | CUST003 | West | 43492 |
| 3 | 2024-01-04 | Headphones | 1 | 30895 | CUST004 | East | 30895 |
| 4 | 2024-01-05 | Laptop | 8 | 39835 | CUST005 | North | 318680 |

Datatypes of the dataset:

```
Data Types:
Date          object
Product       object
Quantity      int64
Price         int64
Customer_ID   object
Region        object
Total_Sales   int64
dtype: object
```

Check NULL values from the dataset:

```
Null Values Check:  
Date          0  
Product       0  
Quantity      0  
Price         0  
Customer_ID   0  
Region        0  
Total_Sales   0  
dtype: int64
```

Check duplicate values from the dataset:

```
Duplicate Rows Count:  
0
```

Output of the metrics:

```
Total Revenue: ₹12,365,048.00  
Best-Selling Product: Laptop (₹3,889,210.00)  
  
Regional Sales Distribution:  
Region  
East      2519639  
North     3983635  
South     3737852  
West      2123922  
Name: Total_Sales, dtype: int64
```

Technical Details:

- **Algorithm:**
 1. Load the dataset using pandas.
 2. Explore data (head(), dtypes, null check, duplicate check).
 3. Clean data by dropping duplicates and handling missing values.

4. Aggregate sales metrics using groupby() and sum().
 5. Save results into a Markdown report.
- **Data Structures:**
 1. pandas DataFrame for tabular data.
 2. Series objects for grouped calculations.
 - **Architecture:**
 1. Single-script pipeline (sales_analysis.py) with modular steps for exploration, cleaning, and analysis.
 2. Output report generated in Markdown format for easy readability.

Testing Evidence:

- **Valid Data:** Input: Complete dataset → Output: Correct total revenue and product rankings.

```
Total Revenue: ₹12,365,048.00
Best-Selling Product: Laptop (₹3,889,210.00)

Regional Sales Distribution:
Region
East      2519639
North     3983635
South     3737852
West      2123922
Name: Total_Sales, dtype: int64
```

- **Missing Values:** Input: Dataset with NaN values → Output: Null values detected and dropped.

```
Null Values Check:
Date           0
Product        0
Quantity       0
Price          0
Customer_ID    0
Region         0
Total_Sales    0
dtype: int64
```

- **Duplicate Rows:** Input: Dataset with duplicate entries → Output: Duplicates counted and removed.

```
Duplicate Rows Count:  
0
```

: