Google Cloud Data Analytics Lab Answers

# Test Case 1: Simple Sales Report

## Aim

To analyze sales data and visualize product-wise total sales.

## Requirements

Google Colab, Python, Pandas, Matplotlib

## Algorithm

1. Load dataset into a DataFrame.  
2. Fill missing 'Price' with the average price of the product.  
3. Calculate 'Total\_Sales = Quantity \* Price'.  
4. Identify the product with highest total sales.  
5. Plot a bar chart of total sales per product.

## Procedure

1. Import necessary libraries.  
2. Load CSV data using pandas.read\_csv().  
3. Use groupby() and transform() to fill missing prices.  
4. Create a new column 'Total\_Sales'.  
5. Use groupby() to find total sales per product.  
6. Use idxmax() to find the top product.  
7. Visualize using matplotlib.

## Program Code

import pandas as pd  
import matplotlib.pyplot as plt  
  
df = pd.read\_csv('sales.csv')  
df['Price'] = df.groupby('Product')['Price'].transform(lambda x: x.fillna(x.mean()))  
df['Total\_Sales'] = df['Quantity'] \* df['Price']  
top\_product = df.groupby('Product')['Total\_Sales'].sum().idxmax()  
sales\_per\_product = df.groupby('Product')['Total\_Sales'].sum()  
sales\_per\_product.plot(kind='bar', title='Total Sales by Product')  
plt.ylabel('Sales')  
plt.show()

## Sample Output

Top Product: 'Product A'. Bar chart showing total sales by product.

# Test Case 2: Daily Temperature Tracker

## Aim

To analyze and visualize daily temperatures by calculating average and plotting trends.

## Requirements

Google Colab, Python, Pandas, Matplotlib

## Algorithm

1. Load temperature dataset.  
2. Fill missing values with column averages.  
3. Calculate 'Average\_Temp'.  
4. Find the hottest day.  
5. Plot average temperature over time.

## Procedure

1. Import libraries.  
2. Load data and clean missing values.  
3. Calculate average temperature.  
4. Find the max temperature row.  
5. Visualize trend with a line graph.

## Program Code

import pandas as pd  
import matplotlib.pyplot as plt  
  
df = pd.read\_csv('temperature.csv')  
df['Min\_Temp'].fillna(df['Min\_Temp'].mean(), inplace=True)  
df['Max\_Temp'].fillna(df['Max\_Temp'].mean(), inplace=True)  
df['Average\_Temp'] = (df['Min\_Temp'] + df['Max\_Temp']) / 2  
hottest\_day = df.loc[df['Average\_Temp'].idxmax(), 'Date']  
plt.plot(df['Date'], df['Average\_Temp'])  
plt.xticks(rotation=45)  
plt.title('Average Temperature Over Time')  
plt.xlabel('Date')  
plt.ylabel('Average Temp')  
plt.tight\_layout()  
plt.show()

## Sample Output

Hottest Day: '2023-06-15'. Line graph of temperature trend.

# Test Case 3: COVID-19 Daily Cases

## Aim

To analyze COVID-19 cases by identifying trends and outliers.

## Requirements

Google Colab, Python, Pandas, Matplotlib

## Algorithm

1. Load dataset.  
2. Replace nulls in 'Cases' with 0.  
3. Calculate total and average cases.  
4. Identify peak day.  
5. Plot daily trend.

## Procedure

1. Import libraries.  
2. Clean and process data.  
3. Compute metrics.  
4. Visualize trends.

## Program Code

import pandas as pd  
import matplotlib.pyplot as plt  
  
df = pd.read\_csv('covid\_cases.csv')  
df['Cases'].fillna(0, inplace=True)  
total\_cases = df['Cases'].sum()  
average\_cases = df['Cases'].mean()  
peak\_day = df.loc[df['Cases'].idxmax(), 'Date']  
plt.plot(df['Date'], df['Cases'])  
plt.title('COVID-19 Daily Cases')  
plt.xlabel('Date')  
plt.ylabel('Cases')  
plt.xticks(rotation=45)  
plt.tight\_layout()  
plt.show()

## Sample Output

Total Cases: 10,500. Average Daily Cases: 350. Peak Day: '2021-05-14'.

# Test Case 4: Movie Ratings Dataset

## Aim

To analyze movie ratings and visualize top-rated movies.

## Requirements

Google Colab, Python, Pandas, Matplotlib

## Algorithm

1. Load dataset.  
2. Remove missing ratings.  
3. Compute average rating.  
4. Find top 3 movies.  
5. Plot top 5 movies.

## Procedure

1. Load and clean data.  
2. Use nlargest() to get top movies.  
3. Plot ratings.

## Program Code

import pandas as pd  
import matplotlib.pyplot as plt  
  
df = pd.read\_csv('movie\_ratings.csv')  
df = df.dropna(subset=['Viewer\_Rating'])  
avg\_rating = df['Viewer\_Rating'].mean()  
top\_movies = df.sort\_values(by='Viewer\_Rating', ascending=False).head(5)  
plt.bar(top\_movies['Movie\_Name'], top\_movies['Viewer\_Rating'])  
plt.title('Top 5 Movie Ratings')  
plt.ylabel('Rating')  
plt.xticks(rotation=45)  
plt.tight\_layout()  
plt.show()

## Sample Output

Average Rating: 4.2. Top Movies: Movie A, Movie B, Movie C...

# Test Case 5: Online Course Completion Data

## Aim

To analyze course completion rates among students.

## Requirements

Google Colab, Python, Pandas, Matplotlib

## Algorithm

1. Load data.  
2. Replace missing 'Completion\_Status' with 'No'.  
3. Count completed students.  
4. Visualize with a pie chart.

## Procedure

1. Load and clean dataset.  
2. Use value\_counts() to get counts.  
3. Plot pie chart.

## Program Code

import pandas as pd  
import matplotlib.pyplot as plt  
  
df = pd.read\_csv('course\_completion.csv')  
df['Completion\_Status'].fillna('No', inplace=True)  
status\_counts = df['Completion\_Status'].value\_counts()  
plt.pie(status\_counts, labels=status\_counts.index, autopct='%1.1f%%', startangle=140)  
plt.title('Course Completion Status')  
plt.show()

## Sample Output

Completed: 120. Not Completed: 80. Pie chart showing percentage breakdown.