

E-Commerce Data Analysis Project

Python Code for Google Colab

```
# =====
# E-Commerce Data Analysis Project
# Python File for Google Colab
# Reads Excel & CSV
# =====
import pandas as pd
import matplotlib.pyplot as plt
# =====
# STEP 1: Upload files in Colab
# =====
from google.colab import files
files.upload()
# =====
# STEP 2: Load Dataset Files
# =====
customers = pd.read_excel("customers.xlsx")
orders = pd.read_excel("orders.xlsx")
order_items = pd.read_excel("order_items.xlsx")
products = pd.read_excel("products.xlsx")
payments = pd.read_excel("payments.xlsx")
sellers = pd.read_excel("sellers.xlsx")
geolocation = pd.read_csv("geolocation.csv")
print("■ All files loaded successfully")
# =====
# BASIC PROBLEMS
# =====
# 1. Unique customer cities
print("Unique customer cities:", customers['customer_city'].nunique())
# 2. Orders placed in 2017
orders['order_purchase_timestamp'] = pd.to_datetime(
    orders['order_purchase_timestamp']
)
orders_2017 = orders[orders['order_purchase_timestamp'].dt.year == 2017]
print("Orders in 2017:", len(orders_2017))
# 3. Total sales per category
merged = order_items.merge(products, on='product_id', how='left')
category_sales = merged.groupby('product_category')['price'].sum()
print(category_sales)
# =====
# INTERMEDIATE PROBLEMS
# =====
# Orders per month in 2018
orders_2018 = orders[orders['order_purchase_timestamp'].dt.year == 2018]
orders_2018['month'] = orders_2018['order_purchase_timestamp'].dt.month
monthly_orders = orders_2018.groupby('month').size()
plt.figure()
plt.plot(monthly_orders.index, monthly_orders.values)
plt.xlabel("Month")
plt.ylabel("Number of Orders")
plt.title("Orders per Month in 2018")
plt.show()
# =====
# ADVANCED PROBLEMS
# =====
# Correlation between product price & purchase count
price_analysis = order_items.groupby('product_id').agg(
    purchase_count=('product_id', 'count'),
    avg_price=('price', 'mean')
)
correlation = price_analysis['purchase_count'].corr(
    price_analysis['avg_price']
)
print("Correlation between price and purchase count:", correlation)
# Top 10 sellers by revenue
seller_revenue = (
    order_items.groupby('seller_id')['price']
    .sum()
```

```
        .sort_values(ascending=False)
        .head(10)
    )
    plt.figure()
    seller_revenue.plot(kind='bar')
    plt.xlabel("Seller ID")
    plt.ylabel("Revenue")
    plt.title("Top 10 Sellers by Revenue")
    plt.show()
    print("■ Analysis Completed Successfully")
```