Data Intake Report

This report is generated from the exploratory data analysis notebook (project.ipynb). It documents the datasets used, the steps taken to clean and prepare the data, and early findings.

Code:

import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
import seaborn as sns  
from datetime import datetime

Code:

cab\_data = pd.read\_csv("Cab\_Data.csv")  
customer = pd.read\_csv("Customer\_ID.csv")  
transaction = pd.read\_csv("Transaction\_ID.csv")  
city = pd.read\_csv("City.csv")

Code:

# Check shape and columns  
for df, name in zip([cab\_data, customer, transaction, city],   
 ["Cab\_Data", "Customer\_ID", "Transaction\_ID", "City"]):  
 print(f"\n{name} Info:")  
 display(df.info())  
 display(df.head())

Code:

print("Duplicates before removal:")  
print(cab\_data.duplicated().sum(), customer.duplicated().sum(), transaction.duplicated().sum(), city.duplicated().sum())  
  
# Remove if any  
cab\_data.drop\_duplicates(inplace=True)  
customer.drop\_duplicates(inplace=True)  
transaction.drop\_duplicates(inplace=True)  
city.drop\_duplicates(inplace=True)

Code:

# Fixing Excel serial date format  
master\_data['Date of Travel'] = pd.to\_datetime(master\_data['Date of Travel'], origin='1899-12-30', unit='D')

Code:

print(master\_data['Date of Travel'].head())

Code:

# Merge cab and transaction  
cab\_txn = pd.merge(cab\_data, transaction, on='Transaction ID', how='inner')

Code:

# Merge with customer demographics  
cab\_txn\_cust = pd.merge(cab\_txn, customer, on='Customer ID', how='inner')

Code:

# Merge with city data  
master\_data = pd.merge(cab\_txn\_cust, city, on='City', how='left')

Code:

# Convert date column  
master\_data['Date of Travel'] = pd.to\_datetime(master\_data['Date of Travel'], origin='1899-12-30', unit='D')

Code:

# Profit and Margin  
master\_data['Profit'] = master\_data['Price Charged'] - master\_data['Cost of Trip']  
master\_data['Profit Margin (%)'] = (master\_data['Profit'] / master\_data['Cost of Trip']) \* 100

Code:

# Date-based features  
master\_data['Year'] = master\_data['Date of Travel'].dt.year  
master\_data['Month'] = master\_data['Date of Travel'].dt.month  
master\_data['Weekday'] = master\_data['Date of Travel'].dt.day\_name()

Code:

print(master\_data.describe(include='all'))

Code:

company\_group = master\_data.groupby('Company')  
company\_summary = company\_group[['Price Charged', 'Cost of Trip', 'Profit']].sum()  
company\_summary['Profit Margin (%)'] = (company\_summary['Profit'] / company\_summary['Cost of Trip']) \* 100  
print(company\_summary)

Code:

monthly\_trend = master\_data.groupby(['Year', 'Month', 'Company'])['Price Charged'].sum().reset\_index()  
  
plt.figure(figsize=(12, 6))  
sns.lineplot(data=monthly\_trend, x='Month', y='Price Charged', hue='Company', style='Year', markers=True)  
plt.title("Monthly Revenue Trend by Company")  
plt.show()

Code:

top\_cities = master\_data.groupby('City')['Profit'].sum().sort\_values(ascending=False).head(10)  
top\_cities.plot(kind='barh', figsize=(10, 6), title='Top 10 Cities by Total Profit')  
plt.gca().invert\_yaxis()  
plt.show()

Code:

# Age Distribution  
sns.histplot(master\_data['Age'], kde=True)  
plt.title("Customer Age Distribution")  
plt.show()

Code:

# Gender vs Spend  
sns.boxplot(x='Gender', y='Price Charged', data=master\_data)  
plt.title("Spending by Gender")  
plt.show()

Code:

# Calculate profit  
master\_data['Profit'] = master\_data['Price Charged'] - master\_data['Cost of Trip']

Code:

# Extract date features  
master\_data['Year'] = master\_data['Date of Travel'].dt.year  
master\_data['Month'] = master\_data['Date of Travel'].dt.month  
master\_data['DayOfWeek'] = master\_data['Date of Travel'].dt.dayofweek  
master\_data['Week'] = master\_data['Date of Travel'].dt.isocalendar().week

Code:

import seaborn as sns  
sns.countplot(data=master\_data, x='Company')

Code:

monthly\_revenue = master\_data.groupby(['Year', 'Month', 'Company'])['Price Charged'].sum().reset\_index()  
monthly\_revenue['Date'] = pd.to\_datetime(monthly\_revenue[['Year', 'Month']].assign(DAY=1))  
  
sns.lineplot(data=monthly\_revenue, x='Date', y='Price Charged', hue='Company')

Code:

sns.barplot(data=master\_data, x='Company', y='Profit')

Code:

print(master\_data.columns)

Code:

# Remove leading/trailing spaces and unify casing  
master\_data.columns = master\_data.columns.str.strip().str.title()

Code:

# Check column names again  
print(master\_data.columns)

Code:

import seaborn as sns  
import matplotlib.pyplot as plt  
  
sns.boxplot(data=master\_data, x='Company', y='Income (Usd/Month)')  
plt.title("Income (Usd/Month)")  
plt.xticks(rotation=45)  
plt.tight\_layout()  
plt.show()

Code:

# Show a few rows  
print(master\_data.head())

Code:

from scipy.stats import ttest\_ind  
  
a\_profit = master\_data[master\_data['Company'] == 'Pink Cab']['Profit']  
b\_profit = master\_data[master\_data['Company'] == 'Yellow Cab']['Profit']  
  
t\_stat, p\_val = ttest\_ind(a\_profit, b\_profit, equal\_var=False)  
print("T-test result:", t\_stat, "P-value:", p\_val)

Code: