**Toronto City pulse analysis**

import pandas as pd

import random

from datetime import datetime, timedelta

# generate sample data

locations = ['Downtown','Uptown','Midtown','Airport','Harbor']

data = []

for i in range(100):

pickup\_time = datetime(2023,5,1,random.randint(7,22),random.randint(0,59))

delivery\_time = pickup\_time + timedelta(minutes = random.randint(15,90))

pickup\_location = random.choice(locations)

drop\_location = random.choice(locations)

distance\_km = round(random.uniform(2,15),2)

data.append([f'ORD{i+1}',pickup\_time, delivery\_time, pickup\_location, drop\_location, distance\_km])

df = pd.DataFrame(data, columns=['order\_id', 'pickup\_time', 'delivery\_time', 'pickup\_location', 'drop\_location', 'distance\_km'])

df.to\_csv('delivery\_data.csv', index=False)

df.head()

A screenshot of a computer

AI-generated content may be incorrect.

#Use API to import data from weatherdata

import requests

api\_key = "b381a8508aef0c1d96fa2c7abdaf5e3b"

city = "Toronto"

data = "2023-05-01"

url = f"https://api.openweathermap.org/data/2.5/weather?q={city}&appid={api\_key}&units=metric"

response = requests.get(url)

weather\_data = response.json()

weather\_data

A screen shot of a computer program

AI-generated content may be incorrect.

#Load and prcess data

df = pd.read\_csv('delivery\_data.csv')

df['pickup\_time'] = pd.to\_datetime(df['pickup\_time'])

df['delivery\_time'] = pd.to\_datetime(df['delivery\_time'])

df['duration\_minutes'] = (df['delivery\_time'] - df['pickup\_time']).dt.total\_seconds() / 60

df.head()

A screenshot of a computer

AI-generated content may be incorrect.

#Visualization

import matplotlib.pyplot as plt

import seaborn as sns

# Plot delivery duration distribution

sns.histplot(df['duration\_minutes'], kde=True)

plt.title("Distribution of Delivery Times")

plt.xlabel("Minutes")

plt.show()

A diagram of a distribution of delivery times

AI-generated content may be incorrect.

#Add Weather or Traffic Tags (Simple Version)

import random

df['weather'] = [random.choice(['Clear', 'Rain', 'Cloudy']) for \_ in range(len(df))]

df['traffic'] = [random.choice(['Low', 'Medium', 'High']) for \_ in range(len(df))]

sns.boxplot(x='weather', y='duration\_minutes', data=df)

plt.title("Delivery Time by Weather")

plt.show()

A diagram of a delivery time

AI-generated content may be incorrect.

sns.boxplot(x='traffic', y='duration\_minutes', data=df)

plt.title("Delivery Time by Traffic Level")

plt.show()

A diagram of a delivery chart

AI-generated content may be incorrect.