**Optional Exercise: Deeper Analysis of NYC Taxi Data**

**Objective**

Students will extend the base script to explore additional insights from the NYC Yellow Taxi dataset hosted on S3. This reinforces skills in working with large datasets, time-based grouping, and summarization.

**Task 1: Most Frequent Drop-off Locations**

Update your script to:

* Count the most frequent drop-off locations using the DOLocationID column
* Display the top 10

**Hint:**

print("Top 10 Drop-off Locations:")

print(df['DOLocationID'].value\_counts().head(10))

**Task 2: Analyze Fare Amount by Trip Distance**

Group trips by trip distance buckets and calculate the average fare for each bucket.

**Steps:**

* Create buckets: 0–1 mile, 1–2 miles, 2–5 miles, 5–10 miles, 10+ miles
* Plot or print the average fare per bucket

**Hint:**

bins = [0, 1, 2, 5, 10, float('inf')]

labels = ['0-1', '1-2', '2-5', '5-10', '10+']

df['distance\_group'] = pd.cut(df['trip\_distance'], bins=bins, labels=labels)

print("Average Fare by Distance Group:")

print(df.groupby('distance\_group')['fare\_amount'].mean())

**Task 3: Time-Based Pickup Patterns**

Explore how pickups vary by:

* Hour of day
* Day of week

**Steps:**

* Extract hour and day of week from tpep\_pickup\_datetime
* Count trips per hour/day and visualize or tabulate the results

**Hint:**

df['weekday'] = df['tpep\_pickup\_datetime'].dt.day\_name()

print("Trips per Day of Week:")

print(df['weekday'].value\_counts())

print("Trips per Hour:")

print(df['hour'].value\_counts().sort\_index())

**Task 4 (Optional Advanced): Calculate Average Speed**

Compute average speed (in mph) using trip\_distance and trip duration.

**Hint:**

df['tpep\_dropoff\_datetime'] = pd.to\_datetime(df['tpep\_dropoff\_datetime'])

df['duration\_minutes'] = (df['tpep\_dropoff\_datetime'] - df['tpep\_pickup\_datetime']).dt.total\_seconds() / 60

df['average\_speed\_mph'] = df['trip\_distance'] / (df['duration\_minutes'] / 60)

print("Average Speed Stats:")

print(df['average\_speed\_mph'].describe())

**Submission (Optional)**

Have students:

* Save their updated Python code to a .py file
* Include comments explaining each task
* Present their results (e.g., in a Teams video or brief written rep