```
# NYC Taxi Trip Data Analysis using PySpark

# 1. Import Libraries
from pyspark.sql import SparkSession
from pyspark.sql.functions import col, hour, avg, count, when

# 2. Initialize Spark Session
spark = SparkSession.builder \
    .appName("NYC Taxi Big Data Analysis") \
    .getOrCreate()

# 3. Load Dataset

df = spark.read.csv("yellow-tripdata-2023-01.csv", header=True, inferSchema=True)

# 4. Display Sample Data
print("Sample Records:")
df.show(5)
df.printSchema()

# 5. Data Cleaning
df_clean = df.dropna(subset=[
    "tpep_pickup_datetime", "tpep_dropoff_datetime",
    "passenger_count", "trip_distance", "total_amount"
])

df_clean = df_clean.filter(
    (col("passenger_count") > 0) &
    (col("trip_distance") > 0) &
    (col("total_amount") > 0)
)

# 6. Feature Engineering
df_clean = df_clean.withColumn("pickup_hour", hour(col("tpep_pickup_datetime")))

# 7. Analysis 1: Peak Hours for Pickup
print("Most Common Pickup Hours:")
peak_hours = df_clean.groupBy("pickup_hour") \
    .agg(count("*").alias("trip_count")) \
    .orderBy("trip_count", ascending=False)
peak_hours.show()

# 8. Analysis 2: Average Fare Based on Distance Category
df_clean = df_clean.withColumn("distance_range",
    when(col("trip_distance") <= 2, "Short") \
    .when((col("trip_distance") > 2) & (col("trip_distance") <= 5), "Medium") \
    .otherwise("Long")
)

print("Average Fare by Trip Distance Category:")
avg_fare = df_clean.groupBy("distance_range") \
    .agg(avg("total_amount").alias("avg_fare")) \
    .orderBy("distance_range")
avg_fare.show()

# 9. Analysis 3: Total Trips by Passenger Count
print("Trip Count by Number of Passengers:")
trip_by_passengers = df_clean.groupBy("passenger_count") \
    .agg(count("*").alias("trip_count")) \
    .orderBy("trip_count", ascending=False)
trip_by_passengers.show()
```

```
Sample Records:
+--------+--------------------+---------------------+---------------+-------------+----------+------------------+------------+--------
|VendorID|tpep_pickup_datetime|tpep_dropoff_datetime|passenger_count|trip_distance|RatecodeID|store_and_fwd_flag|PULocationID|DOLocati
+--------+--------------------+---------------------+---------------+-------------+----------+------------------+------------+--------
|       2| 2023-01-01 00:32:10|  2023-01-01 00:40:36|              1|         0.97|         1|                 N|         161|
|       2| 2023-01-01 00:55:08|  2023-01-01 01:01:27|              1|          1.1|         1|                 N|          43|
|       2| 2023-01-01 00:25:04|  2023-01-01 00:37:49|              1|         2.51|         1|                 N|          48|
|       1| 2023-01-01 00:03:48|  2023-01-01 00:13:25|              0|          1.9|         1|                 N|         138|
|       2| 2023-01-01 00:10:29|  2023-01-01 00:21:19|              1|         1.43|         1|                 N|         107|
```

```
+--------+--------------------+--------------------+---------------+-------------+----------+------------------+------------+--------
only showing top 5 rows

root
 |-- VendorID: integer (nullable = true)
 |-- tpep_pickup_datetime: timestamp (nullable = true)
 |-- tpep_dropoff_datetime: timestamp (nullable = true)
 |-- passenger_count: integer (nullable = true)
 |-- trip_distance: double (nullable = true)
 |-- RatecodeID: integer (nullable = true)
 |-- store_and_fwd_flag: string (nullable = true)
 |-- PULocationID: integer (nullable = true)
 |-- DOLocationID: integer (nullable = true)
 |-- payment_type: integer (nullable = true)
 |-- fare_amount: double (nullable = true)
 |-- extra: double (nullable = true)
 |-- mta_tax: double (nullable = true)
 |-- tip_amount: double (nullable = true)
 |-- tolls_amount: double (nullable = true)
 |-- improvement_surcharge: double (nullable = true)
 |-- total_amount: double (nullable = true)
 |-- congestion_surcharge: double (nullable = true)
 |-- airport_fee: double (nullable = true)

Most Common Pickup Hours:
+-----------+----------+
|pickup_hour|trip_count|
+-----------+----------+
|         18|     75421|
|         17|     73298|
|         15|     68985|
|         16|     68388|
|         14|     67535|
|         19|     65935|
|         13|     61897|
|         12|     59010|
|         20|     56105|
|         11|     54066|
|         10|     50684|
|         21|     48622|
|          9|     45580|
|         22|     42367|
|          8|     40558|
|         23|     32113|
|          7|     30569|
|          0|     27115|
|          1|     19009|
|          6|     15525|
```