namma_yatri

July 28, 2024

1 Importing Modules/Libraries

[16]:
```
import mysql.connector
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

2 List all trips with their fare and corresponding payment method.

[113]:
```
my_db = mysql.connector.connect(
    host="localhost",
    user="root",
    password="",
    database="namma_yatri"
)

my_cursor = my_db.cursor()

query = '''
SELECT trips.tripid, trips.fare, Payment.method
FROM trips
JOIN Payment ON trips.faremethod = Payment.id;
'''

my_cursor.execute(query)
table_data = my_cursor.fetchall()

my_cursor.close()
my_db.close()

df = pd.DataFrame(table_data, columns=["Trip ID", "Total Fare", "Payment␣
 ↪Method"])

print(df)
```

```
   Trip ID  Total Fare Payment Method
0        1         776            upi
```

```
1          2        1479            upi
2          3         152    credit card
3          4         153     debit card
4          5         366            upi
..       …         …             …
978      979        1245    credit card
979      980         809           cash
980      981         695     debit card
981      982        1499            upi
982      983        1475    credit card

[983 rows x 3 columns]
```

3 Which is the most used payment method?

[104]:
```
my_db = mysql.connector.connect(
    host="localhost",
    user="root",
    passwd="",
    database="namma_yatri"
)

my_cursor = my_db.cursor()

query = '''
SELECT
    payment.method, COUNT(trips.tripid) AS trip_count
FROM
    trips
    JOIN payment ON payment.id = trips.faremethod
GROUP BY payment.method
ORDER BY trip_count DESC;
'''

my_cursor.execute(query)

table_data = my_cursor.fetchall()

my_cursor.close()
my_db.close()

df = pd.DataFrame(table_data, columns=['Payment Method', 'Total Trips'])
print(df)
```

```
  Payment Method  Total Trips
0    credit card          262
1            upi          243
```

```
2     debit card          243
3           cash          235
```

4 Find the total number of trips taken by each customer.

[106]:
```
my_db = mysql.connector.connect(
    host="localhost",
    user="root",
    passwd="",
    database="namma_yatri"
)

my_cursor = my_db.cursor()

query = '''
SELECT
    trips.custid,
    COUNT(trips.tripid) AS total_trips
FROM
    trips
GROUP BY
    trips.custid
ORDER BY
    trips.custid ASC;
'''

my_cursor.execute(query)

table_data = my_cursor.fetchall()

my_cursor.close()
my_db.close()

df = pd.DataFrame(table_data, columns=['custid', 'total_trips'])
print(df)
```

```
    custid  total_trips
0        1           13
1        2            9
2        3            5
3        4            8
4        5           10
..     …           …
94      95            6
95      96           12
96      97           10
97      98            9
```

```
98      99           11

[99 rows x 2 columns]
```

5 Which five locations had the most trips?

[107]:
```
my_db = mysql.connector.connect(
    host="localhost",
    user="root",
    passwd="",
    database="namma_yatri"
)

my_cursor = my_db.cursor()

query = '''
SELECT
    trips.loc_from,
    trips.loc_to,
    COUNT(trips.tripid) AS trip_count
FROM
    trips
GROUP BY
    trips.loc_from, trips.loc_to
ORDER BY
    trip_count DESC;
'''

my_cursor.execute(query)

table_data = my_cursor.fetchall()

my_cursor.close()
my_db.close()

df = pd.DataFrame(table_data, columns=['Loc From', 'Loc To', 'Total Trips'])
print(df.head(5))
```

```
   Loc From  Loc To  Total Trips
0        16      21            5
1        35       5            5
2        35      26            4
3        30      23            4
4        18      10            4
```

6 Which area got the highest cancellations?

[109]:
```
import mysql.connector
import pandas as pd

my_db = mysql.connector.connect(
    host="localhost",
    user="root",
    passwd="",
    database="namma_yatri"
)

my_cursor = my_db.cursor()

query = '''
SELECT
    assembly.assembly AS location,
    COUNT(trip_details.driver_not_cancelled) AS total_cancellations
FROM
    trips
LEFT JOIN
    trip_details ON trip_details.tripid = trips.tripid
JOIN
    assembly ON assembly.id = trips.loc_to
GROUP BY
    assembly.assembly
ORDER BY
    total_cancellations DESC;
'''

my_cursor.execute(query)
table_data = my_cursor.fetchall()
my_cursor.close()
my_db.close()

df = pd.DataFrame(table_data, columns=['Location', 'Total Cancellations'])
print(df)
```

	Location	Total Cancellations
0	Hoskote	37
1	Chamrajpet	36
2	Kanakapura	34
3	Vijay Nagar	33
4	Yelahanka	32
5	Dasarahalli	31
6	Shanti Nagar	31
7	Gandhi Nagar	30
8	Devanahalli	29

9	Doddaballapur	29
10	B. T. M. Layout	29
11	Sarvagnanagar	29
12	Padmanabhanagar	28
13	Govindraj Nagar	28
14	Mahadevapura	28
15	Rajarajeshwarinagar	28
16	Anekal	27
17	Bommanahalli	27
18	Ramanagaram	26
19	Krishnarajapuram	26
20	Pulakeshinagar	25
21	Chickpet	25
22	Jayanagar	25
23	Mahalakshmi Layout	25
24	Nelamangala	25
25	Rajaji Nagar	24
26	Magadi	24
27	Other Assemblies	24
28	Hebbal	24
29	Channapatna	23
30	C. V. Raman Nagar	21
31	Bangalore South	21
32	Byatarayanapura	21
33	Yeshwantpur	20
34	Shivajinagar	20
35	Malleshwaram	19
36	Basavanagudi	19

7 Find the number of trips that started from each assembly point.

[35]:
```
my_db = mysql.connector.connect(
    host="localhost",
    user="root",
    passwd="",
    database="namma_yatri"
)

my_cursor = my_db.cursor()

query = '''
SELECT
    trips.loc_from,
    COUNT(trips.tripid) AS total_trips
FROM
    trips
GROUP BY
```

```
    trips.loc_from
ORDER BY
    total_trips DESC;
'''

my_cursor.execute(query)

table_data = my_cursor.fetchall()

my_cursor.close()
my_db.close()

df = pd.DataFrame(table_data, columns=['Loc From', 'Total Trips'])
print(df)
```

	Loc From	Total Trips
0	35	39
1	18	36
2	6	33
3	20	33
4	16	32
5	12	32
6	28	31
7	9	31
8	17	31
9	25	30
10	36	30
11	31	30
12	21	29
13	3	29
14	19	28
15	13	28
16	37	27
17	14	27
18	10	27
19	24	26
20	2	26
21	15	26
22	1	26
23	11	25
24	32	24
25	29	24
26	7	24
27	33	22
28	5	22
29	22	22
30	23	21
31	4	21

```
32          8           19
33         26           19
34         27           19
35         30           17
36         34           17
```

8 List all trips with their durations.

[39]:
```
my_db = mysql.connector.connect(
    host="localhost",
    user="root",
    passwd="",
    database="namma_yatri"
)

my_cursor = my_db.cursor()

query = '''
SELECT
    count(trips.tripid) as no_of_trips,
    duration.duration
FROM
    trips
JOIN
    duration ON trips.duration = duration.id
GROUP BY duration.duration;
'''

my_cursor.execute(query)
table_data = my_cursor.fetchall()

my_cursor.close()
my_db.close()

df = pd.DataFrame(table_data, columns=['Total Trips', 'Duration'])
print(df)
```

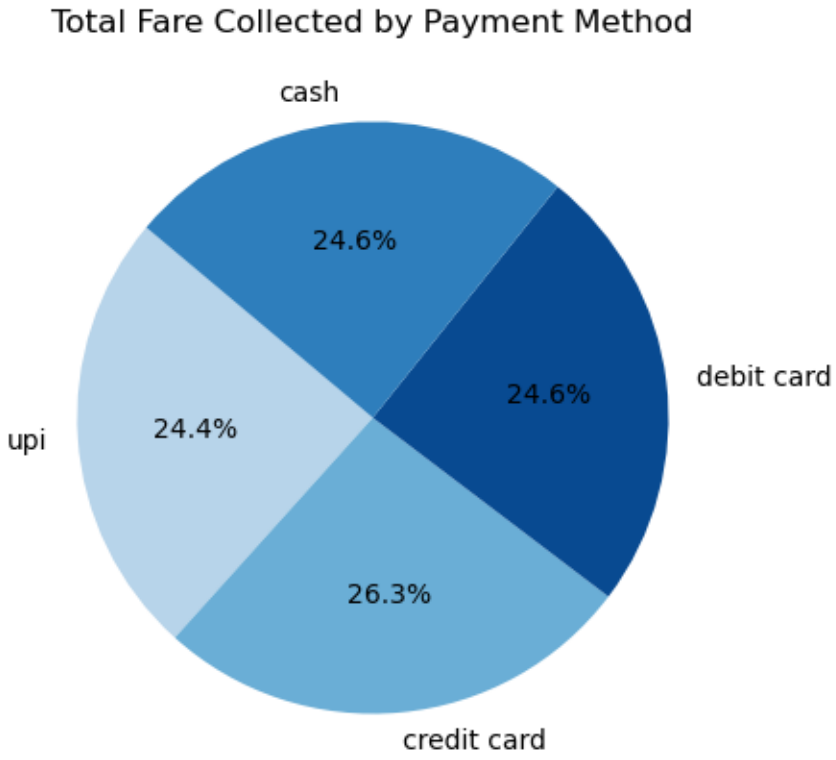
```
    Total Trips Duration
0            35    19-20
1            39    14-15
2            32    23-24
3            36      3-4
4            39      1-2
5            52    13-14
6            44    18-19
7            48    11-12
8            40    15-16
```

```
9             39      7-8
10            48    22-23
11            41      2-3
12            53      0-1
13            34    16-17
14            42      5-6
15            48    17-18
16            43     9-10
17            33      4-5
18            48      6-7
19            39    10-11
20            45    12-13
21            33      8-9
22            40    21-22
23            32    20-21
```

9 Total Fare Collected by Payment Method

[93]:
```
import matplotlib.cm as cm

my_db = mysql.connector.connect(
    host="localhost",
    user="root",
    passwd="",
    database="namma_yatri"
)

my_cursor = my_db.cursor()

query = '''
SELECT
    payment.method AS payment_method,
    SUM(trips.fare) AS total_fare
FROM
    trips
JOIN
    payment ON trips.faremethod = payment.id
GROUP BY
    payment.method;
'''

my_cursor.execute(query)
table_data = my_cursor.fetchall()
my_cursor.close()
my_db.close()

df = pd.DataFrame(table_data, columns=['Payment Method', 'Total Fare'])
```

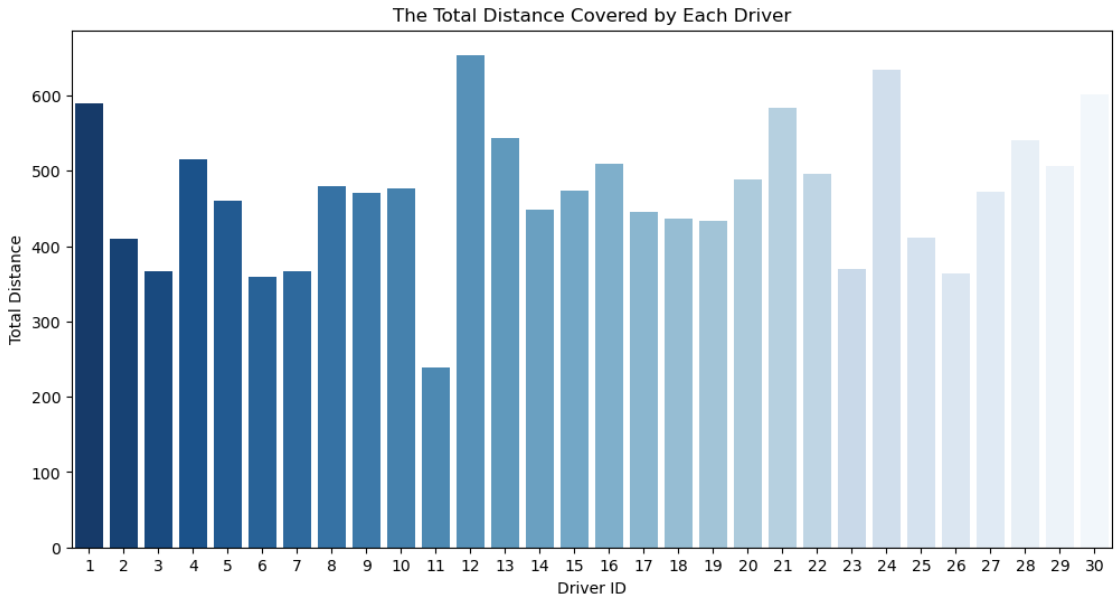
```
plt.figure(figsize=(5, 5))
cmap = cm.Blues
colors = cmap([0.3, 0.5, 0.9, 0.7])

plt.pie(df['Total Fare'], labels=df['Payment Method'], colors=colors,␣
 ↪autopct='%1.1f%%', startangle=140)
plt.title('Total Fare Collected by Payment Method')
plt.show()
```



10 Calculate the total distance covered by each driver.

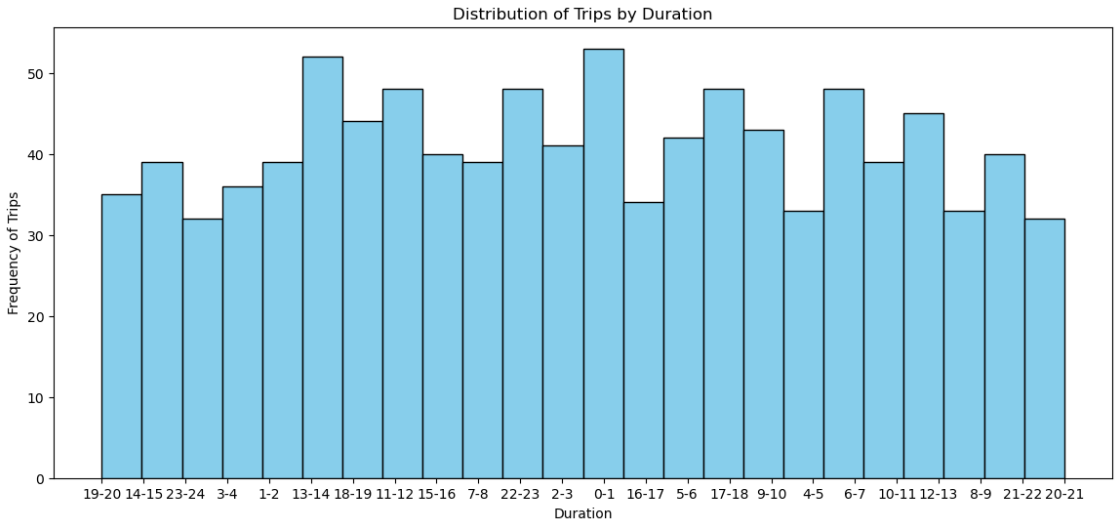
[89]:
```
my_db = mysql.connector.connect(
    host="localhost",
    user="root",
    passwd="",
    database="namma_yatri"
)

my_cursor = my_db.cursor()
```

```
query = '''
SELECT
    trips.driverid,
    SUM(trips.distance) AS total_distance
FROM
    trips
GROUP BY
    trips.driverid;'''

my_cursor.execute(query)

table_data = my_cursor.fetchall()

my_cursor.close()
my_db.close()

df = pd.DataFrame(table_data, columns=['driverid', 'total_distance'])

plt.figure(figsize=(12,6))
bar_plot = sns.barplot(x='driverid', y='total_distance', data=df,␣
 ↪palette='Blues_r')

plt.title('The Total Distance Covered by Each Driver')
plt.xlabel('Driver ID')
plt.ylabel('Total Distance')
plt.show()
```



11 Which Duration has more Trips?

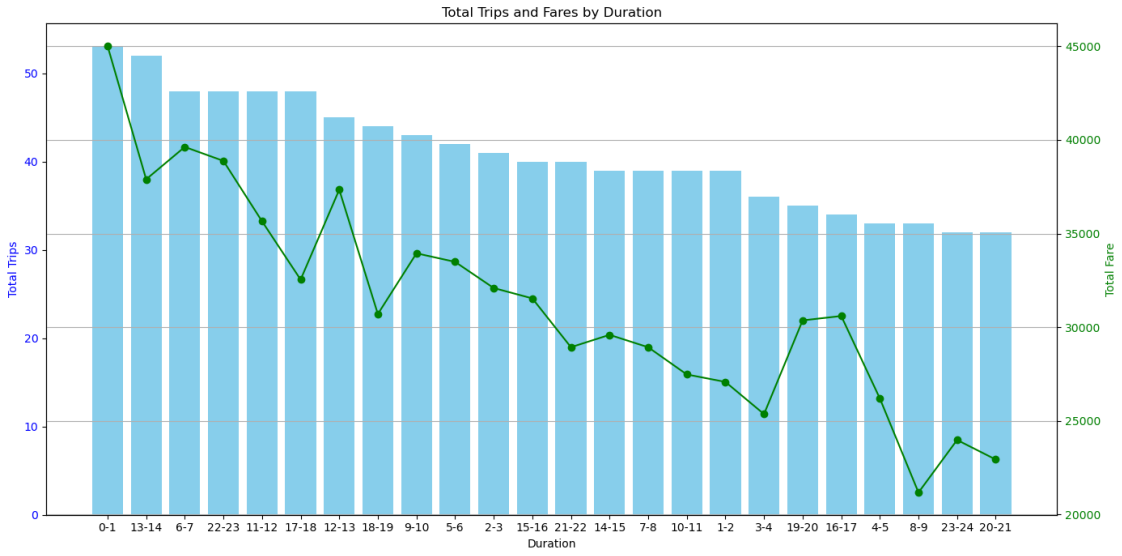
[111]:
```
my_db = mysql.connector.connect(
    host="localhost",
    user="root",
    passwd="",
    database="namma_yatri"
)

my_cursor = my_db.cursor()

query = '''
SELECT
    duration.duration,
    COUNT(trips.tripid) AS total_trips
FROM
    trips
JOIN
    duration ON trips.duration = duration.id
GROUP BY
    duration.duration;
'''

my_cursor.execute(query)
table_data = my_cursor.fetchall()
my_cursor.close()
my_db.close()

df = pd.DataFrame(table_data, columns=['Duration', 'Total Trips'])

plt.figure(figsize=(14,6))
plt.hist(df['Duration'], weights=df['Total Trips'], bins=len(df['Duration']),␣
 ↪color='skyblue', edgecolor='black')
plt.xlabel('Duration')
plt.ylabel('Frequency of Trips')
plt.title('Distribution of Trips by Duration')
plt.show()
```



12 Which duration got the highest trips and fares?

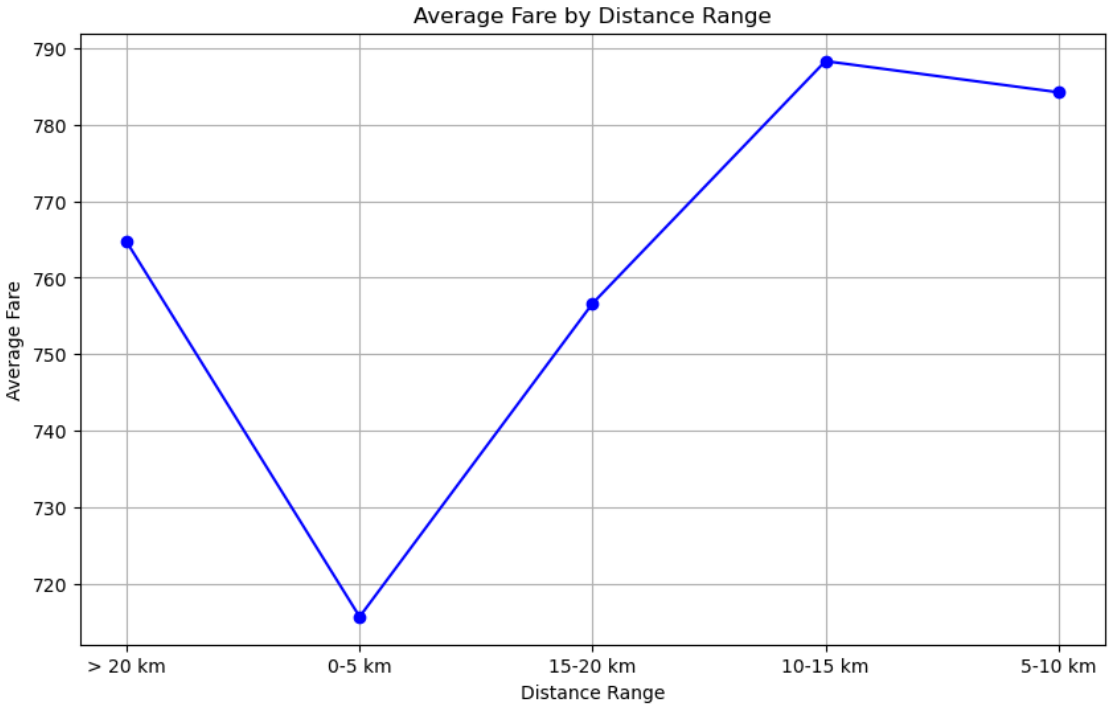
[66]:
```
my_db = mysql.connector.connect(
    host="localhost",
    user="root",
    passwd="",
    database="namma_yatri"
)

my_cursor = my_db.cursor()

query = '''
SELECT
    duration.duration,
    COUNT(trips.tripid) AS total_trips,
    SUM(trips.fare) AS total_fare
FROM
    trips
JOIN
    duration ON duration.id = trips.duration
GROUP BY
    duration.duration
ORDER BY
    total_trips DESC, total_fare DESC;
'''

my_cursor.execute(query)
table_data = my_cursor.fetchall()
```

```
my_cursor.close()
my_db.close()

df = pd.DataFrame(table_data, columns=['Duration', 'Total Trips', 'Total Fare'])

fig, ax1 = plt.subplots(figsize=(14, 7))

ax1.bar(df['Duration'], df['Total Trips'], color='skyblue', label='Total Trips')
ax1.set_xlabel('Duration')
ax1.set_ylabel('Total Trips', color='blue')
ax1.tick_params(axis='y', labelcolor='blue')

ax2 = ax1.twinx()
ax2.plot(df['Duration'], df['Total Fare'], 'g-o', label='Total Fare')
ax2.set_ylabel('Total Fare', color='green')
ax2.tick_params(axis='y', labelcolor='green')

plt.title('Total Trips and Fares by Duration')
fig.tight_layout()
plt.grid(True)
plt.show()
```



13 Calculate the average fare for different distance ranges (e.g., 0-5 km, 5-10 km).

[40]:
```
my_db = mysql.connector.connect(
    host="localhost",
    user="root",
    passwd="",
    database="namma_yatri"
)

my_cursor = my_db.cursor()

query = '''
SELECT
    CASE
        WHEN distance <= 5 THEN '0-5 km'
        WHEN distance <= 10 THEN '5-10 km'
        WHEN distance <= 15 THEN '10-15 km'
        WHEN distance <= 20 THEN '15-20 km'
        ELSE '> 20 km'
    END AS distance_range,
    AVG(fare) AS average_fare
FROM
    trips
GROUP BY
    distance_range;
'''

my_cursor.execute(query)

table_data = my_cursor.fetchall()

my_cursor.close()
my_db.close()

df = pd.DataFrame(table_data, columns=['distance_range', 'average_fare'])

plt.figure(figsize=(10, 6))
plt.plot(df['distance_range'], df['average_fare'], marker='o', linestyle='-',␣
 ↪color='blue')

plt.title('Average Fare by Distance Range')
plt.xlabel('Distance Range')
plt.ylabel('Average Fare')
plt.grid(True)
plt.show()
```


Average Fare by Distance Range
790
780
770
760
750
740
730
720
Average Fare
> 20 km
0-5 km
15-20 km
10-15 km
5-10 km
Distance Range