Course 3 Automatidata project

Course 3 - Go Beyond the Numbers: Translate Data into Insights

You are the newest data professional in a fictional data consulting firm: Automatidata. The team is early into the project, having only just completed an initial plan of action and some early Python coding work.

Luana Rodriquez, the senior data analyst at Automatidata, is pleased with the work you have already completed and requests your assistance with some EDA and data visualization work for the New York City Taxi and Limousine Commission project (New York City TLC) to get a general understanding of what taxi ridership looks like. The management team is asking for a Python notebook showing data structuring and cleaning, as well as any matplotlib/seaborn visualizations plotted to help understand the data. At the very least, include a box plot of the ride durations and some time series plots, like a breakdown by quarter or month.

Additionally, the management team has recently asked all EDA to include Tableau visualizations. For this taxi data, create a Tableau dashboard showing a New York City map of taxi/limo trips by month. Make sure it is easy to understand to someone who isn't data savvy, and remember that the assistant director at the New York City TLC is a person with visual impairments.

A notebook was structured and prepared to help you in this project. Please complete the following questions.

Course 3 End-of-course project: Exploratory data analysis

In this activity, you will examine data provided and prepare it for analysis. You will also design a professional data visualization that tells a story, and will help data-driven decisions for business needs.

Please note that the Tableau visualization activity is optional, and will not affect your completion of the course. Completing the Tableau activity will help you practice planning out and plotting a data visualization based on a specific business need. The structure of this activity is designed to emulate the proposals you will likely be assigned in your career as a data professional. Completing this activity will help prepare you for those career moments.

The purpose of this project is to conduct exploratory data analysis on a provided data set. Your mission is to continue the investigation you began in C1 and perform further EDA on this data with the aim of learning more about the variables.

The goal is to clean data set and create a visualization.

This activity has 4 parts:

Part 1: Imports, links, and loading

Part 2: Data Exploration

- Data cleaning

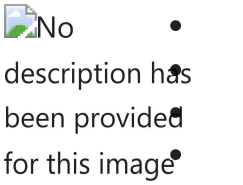
Part 3: Building visualizations

Part 4: Evaluate and share results

Follow the instructions and answer the questions below to complete the activity. Then, you will complete an Executive Summary using the questions listed on the PACE Strategy Document.

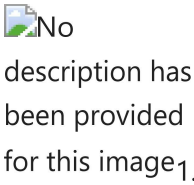
Visualize a story in Tableau and Python

PACE stages



- `[Plan](#scrollTo=psz51YkZVwtN&line=3&uniqifier=1)`
- `[Analyze](#scrollTo=mA7Mz_SnI8km&line=4&uniqifier=1)`
- `[Construct](#scrollTo=Lca9c8XON8lc&line=2&uniqifier=1)`
- `[Execute](#scrollTo=401PgchTPr4E&line=2&uniqifier=1)`

Throughout these project notebooks, you'll see references to the problem-solving framework PACE. The following notebook components are labeled with the respective PACE stage: Plan, Analyze, Construct, and Execute.



PACE: Plan

1. Identify any outliers:

    - What methods are best for identifying outliers?
    - Use numpy functions to investigate the `mean()` and `median()` of the data and understand range of data values
    - Use a boxplot to visualize the distribution of the data
    - Use histograms to visualize the distribution of the data
- How do you make the decision to keep or exclude outliers from any future models?

- There are three main options for dealing with outliers: keeping them as they are, deleting them, or reassigning them. Whether you keep outliers as they are, delete them, or reassign values is a decision that you make taking into account the nature of the outlying data and the assumptions of the model you are building. To help you make the decision, you can start with these general guidelines:
- Delete them: If you are sure the outliers are mistakes, typos, or errors and the dataset will be used for modeling or machine learning, then you are more likely to decide to delete outliers. Of the three choices, you'll use this one the least.
- Reassign them: If the dataset is small and/or the data will be used for modeling or machine learning, you are more likely to choose a path of deriving new values to replace the outlier values.
- Leave them: For a dataset that you plan to do EDA/analysis on and nothing else, or for a dataset you are preparing for a model that is resistant to outliers, it is most likely that you are going to leave them in.

Task 1. Imports, links, and loading

Go to Tableau Public The following link will help you complete this activity. Keep Tableau Public open as you proceed to the next steps.

Link to supporting materials: Public Tableau: https://public.tableau.com/s/

For EDA of the data, import the data and packages that would be most helpful, such as pandas, numpy and matplotlib.

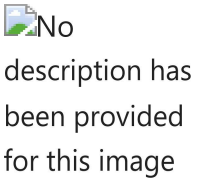
Then, import the dataset.

In [30]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import datetime as dt
import seaborn as sns
```

Note: As shown in this cell, the dataset has been automatically loaded in for you. You do not need to download the .csv file, or provide more code, in order to access the dataset and proceed with this lab. Please continue with this activity by completing the following instructions.

In [31]:
```
df=pd.read_csv('data/2017_Yellow_Taxi_Trip_Data.csv')
```

PACE: Analyze



Consider these questions in your PACE Strategy Document to reflect on the Analyze stage.

Task 2a. Data exploration and cleaning

Decide which columns are applicable

The first step is to assess your data. Check the Data Source page on Tableau Public to get a sense of the size, shape and makeup of the data set. Then answer these questions to yourself:

Given our scenario, which data columns are most applicable? Which data columns can I eliminate, knowing they won't solve our problem scenario?

Consider functions that help you understand and structure the data.

- `head()`
- `describe()`
- `info()`
- `groupby()`
- `sortby()`

Consider these questions as you work:

What do you do about missing data (if any)?

Are there data outliers?

What do the distributions of your variables tell you about the question you're asking or the problem you're trying to solve?

Find these answers later in the notebook.

Start by discovering, using head and size.

In [32]:
```
df.head(10)
```

Out[32]:

	Unnamed: 0	VendorID	tpep_pickup_datetime	tpep_dropoff_datetime	passenger_count	tr
0	24870114	2	03/25/2017 8:55:43 AM	03/25/2017 9:09:47 AM	6	
1	35634249	1	04/11/2017 2:53:28 PM	04/11/2017 3:19:58 PM	1	
2	106203690	1	12/15/2017 7:26:56 AM	12/15/2017 7:34:08 AM	1	
3	38942136	2	05/07/2017 1:17:59 PM	05/07/2017 1:48:14 PM	1	
4	30841670	2	04/15/2017 11:32:20 PM	04/15/2017 11:49:03 PM	1	
5	23345809	2	03/25/2017 8:34:11 PM	03/25/2017 8:42:11 PM	6	
6	37660487	2	05/03/2017 7:04:09 PM	05/03/2017 8:03:47 PM	1	
7	69059411	2	08/15/2017 5:41:06 PM	08/15/2017 6:03:05 PM	1	
8	8433159	2	02/04/2017 4:17:07 PM	02/04/2017 4:29:14 PM	1	
9	95294817	1	11/10/2017 3:20:29 PM	11/10/2017 3:40:55 PM	1	

In [33]:
```
df.size
```

Out[33]:
```
408582
```

Use describe...

In [34]:
```
df.describe()
```

Out[34]:

	Unnamed: 0	VendorID	passenger_count	trip_distance	RatecodeID	PULocati
count	2.269900e+04	22699.000000	22699.000000	22699.000000	22699.000000	22699.00
mean	5.675849e+07	1.556236	1.642319	2.913313	1.043394	162.41
std	3.274493e+07	0.496838	1.285231	3.653171	0.708391	66.63
min	1.212700e+04	1.000000	0.000000	0.000000	1.000000	1.00
25%	2.852056e+07	1.000000	1.000000	0.990000	1.000000	114.00
50%	5.673150e+07	2.000000	1.000000	1.610000	1.000000	162.00
75%	8.537452e+07	2.000000	2.000000	3.060000	1.000000	233.00
max	1.134863e+08	2.000000	6.000000	33.960000	99.000000	265.00

And info.

In [35]:
```
df.info()
```

```
<class 'pandas.DataFrame'>
RangeIndex: 22699 entries, 0 to 22698
Data columns (total 18 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Unnamed: 0             22699 non-null  int64
 1   VendorID               22699 non-null  int64
 2   tpep_pickup_datetime   22699 non-null  str
 3   tpep_dropoff_datetime  22699 non-null  str
 4   passenger_count        22699 non-null  int64
 5   trip_distance          22699 non-null  float64
 6   RatecodeID             22699 non-null  int64
 7   store_and_fwd_flag     22699 non-null  str
 8   PULocationID           22699 non-null  int64
 9   DOLocationID           22699 non-null  int64
 10  payment_type           22699 non-null  int64
 11  fare_amount            22699 non-null  float64
 12  extra                  22699 non-null  float64
 13  mta_tax                22699 non-null  float64
 14  tip_amount             22699 non-null  float64
 15  tolls_amount           22699 non-null  float64
 16  improvement_surcharge  22699 non-null  float64
 17  total_amount           22699 non-null  float64
dtypes: float64(8), int64(7), str(3)
memory usage: 3.1 MB
```

Task 2b. Assess whether dimensions and measures are correct

On the data source page in Tableau, double check the data types for the applicable columns you selected on the previous step. Pay close attention to the dimensions and measures to ensure they are correct.

In Python, consider the data types of the columns. *Consider:* Do they make sense?

Review the link provided in the previous activity instructions to create the required Tableau visualization.

Task 2c. Select visualization type(s)

Select data visualization types that will help you understand and explain the data.

Now that you know which data columns you'll use, it is time to decide which data visualization makes the most sense for EDA of the TLC dataset. What type of data visualization(s) would be most helpful?

- Line graph
- Bar chart
- Box plot
- Histogram
- Heat map

- Scatter plot
- A geographic map

PACE: Construct

Consider these questions in your PACE Strategy Document to reflect on the Construct stage.

Task 3. Data visualization

You've assessed your data, and decided on which data variables are most applicable. It's time to plot your visualization(s)!

Boxplots

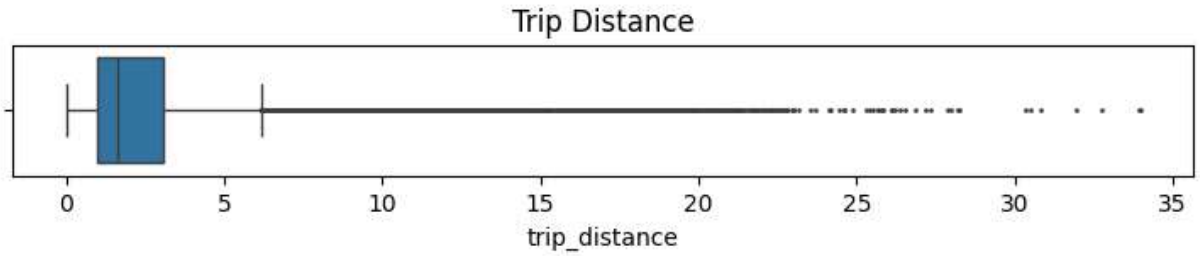
Perform a check for outliers on relevant columns such as trip distance and trip duration. Remember, some of the best ways to identify the presence of outliers in data are box plots and histograms.

Note: Remember to convert your date columns to datetime in order to derive total trip duration.

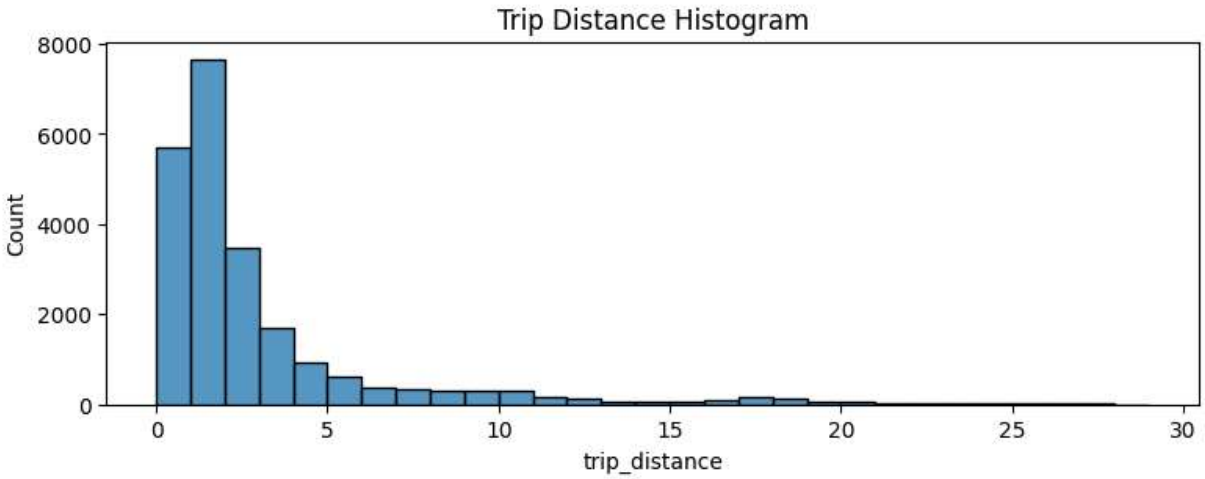
In [36]:
```
# Convert data columns to datetime
df['tpep_pickup_datetime']=pd.to_datetime(df['tpep_pickup_datetime'])
df['tpep_dropoff_datetime']=pd.to_datetime(df['tpep_dropoff_datetime'])
```

trip_distance

In [37]:
```
# Create box plot of trip_distance
plt.figure(figsize=(9,1))
plt.title('Trip Distance')
sns.boxplot(data=None, x=df['trip_distance'], fliersize=1);
```

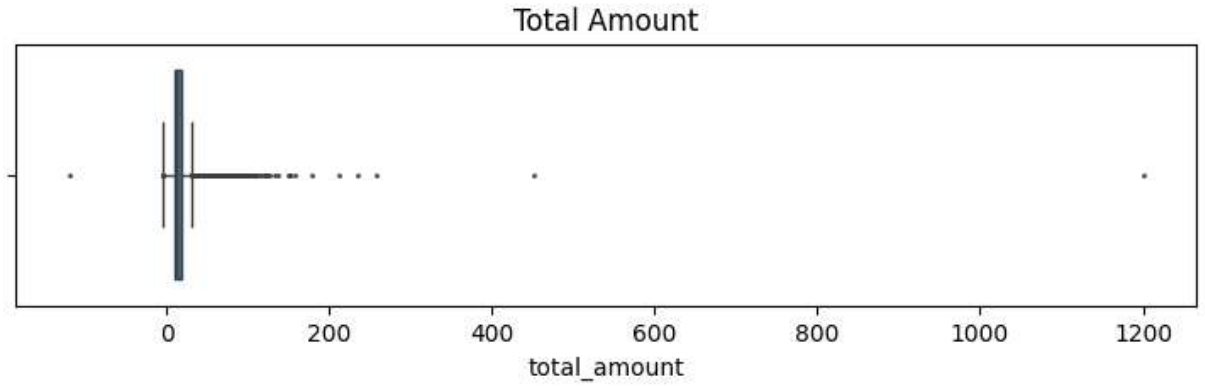


In [38]:
```
# Create histogram of trip_distance
plt.figure(figsize=(9,3))
sns.histplot(df['trip_distance'], bins=range(0,30,1))
plt.title('Trip Distance Histogram');
```

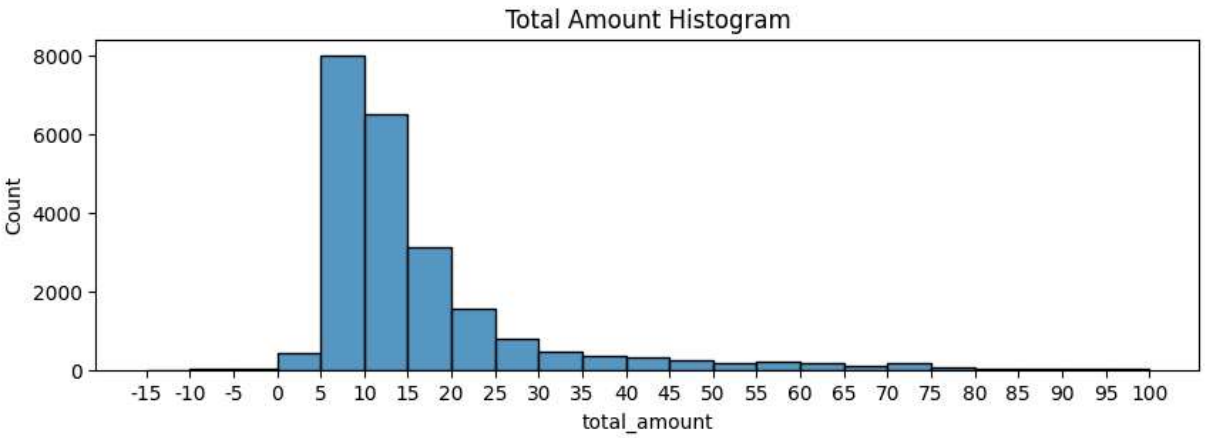


total_amount

In [39]:
```
# Create box plot of total_amount
plt.figure(figsize=(9,2))
plt.title('Total Amount')
sns.boxplot(x=df['total_amount'], fliersize=1);
```

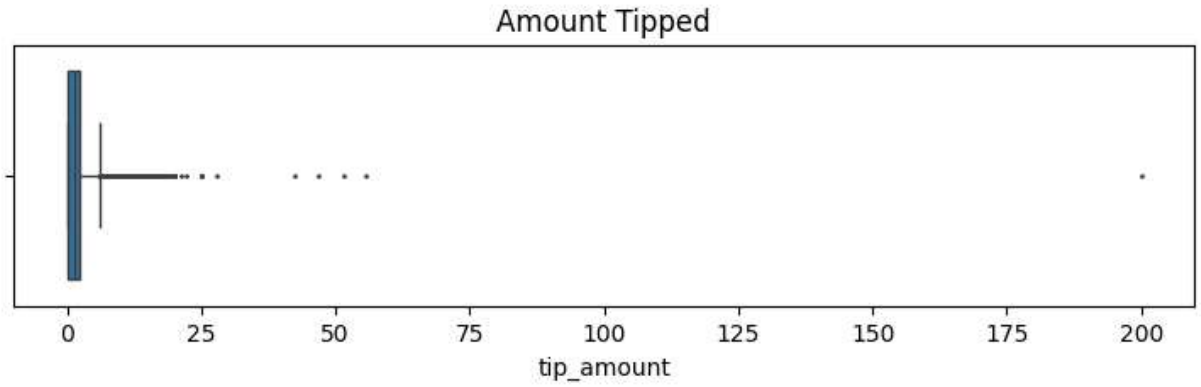


In [40]:
```
# Create histogram of total_amount
plt.figure(figsize=(10,3))
ax = sns.histplot(df['total_amount'], bins=range(-15,105,5))
ax.set_xticks(range(-15,105,5))
ax.set_xticklabels(range(-15,105,5))
plt.title('Total Amount Histogram');
```

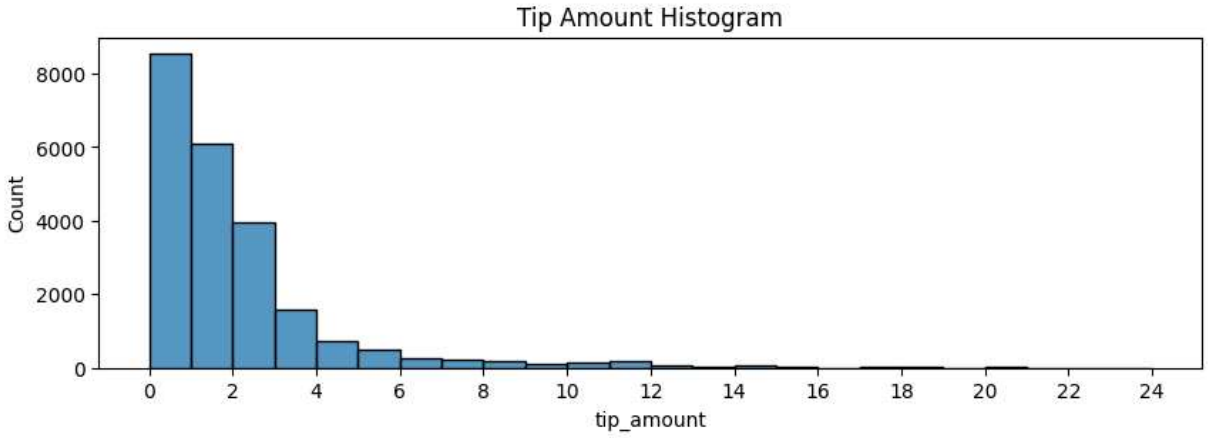


tip_amount

In [41]:
```
# Create box plot of tip_amount
plt.figure(figsize=(9,2))
plt.title('Amount Tipped')
sns.boxplot(x=df['tip_amount'], fliersize=1);
```

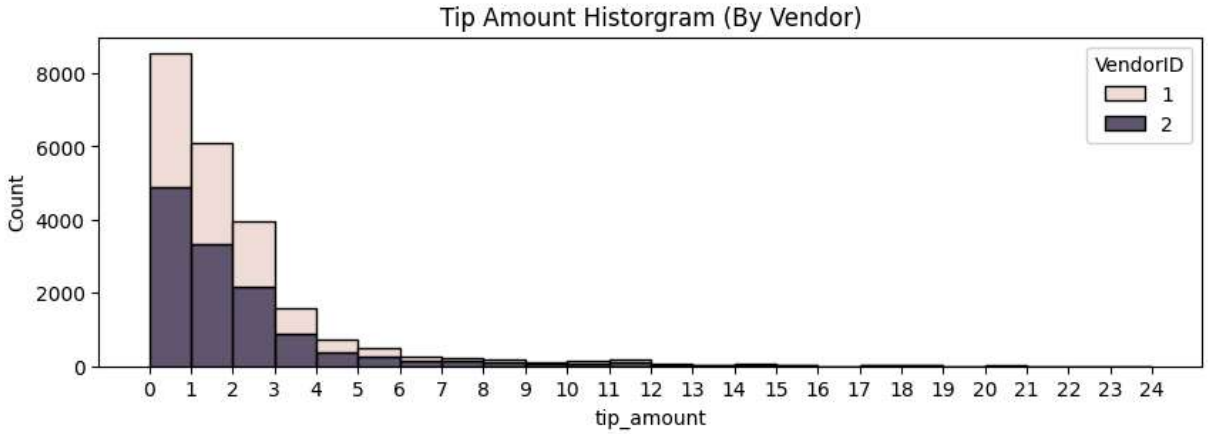


In [42]:
```
# Create histogram of tip_amount
plt.figure(figsize=(10,3))
ax = sns.histplot(df['tip_amount'], bins=range(0,25,1))
ax.set_xticks(range(0,25,2))
ax.set_xticklabels(range(0,25,2))
plt.title('Tip Amount Histogram');
```

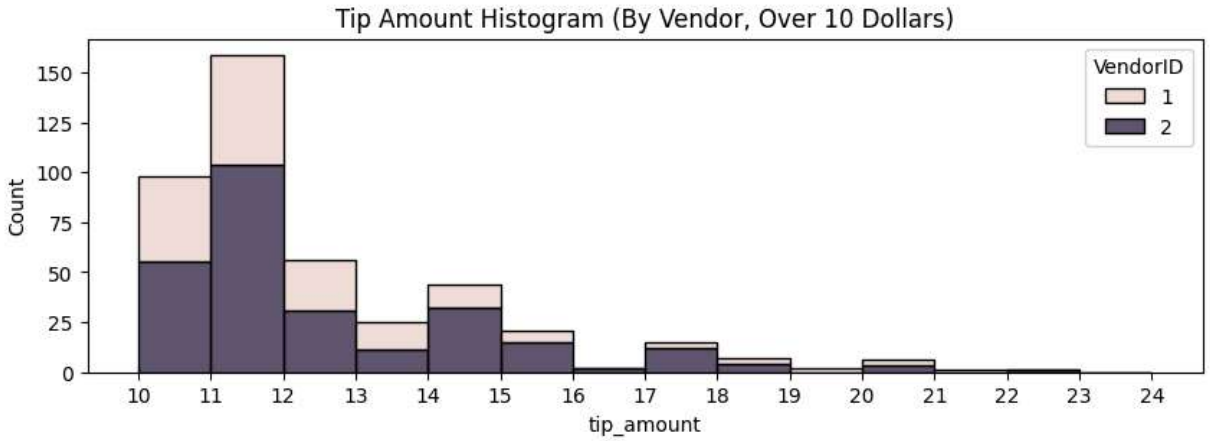


tip_amount by vendor

In [43]:
```
# Create histogram of tip_amount by vendor
plt.figure(figsize=(10,3))
ax = sns.histplot(data=df, x='tip_amount', bins=range(0,25,1), hue='VendorID', mult
ax.set_xticks(range(0,25,1))
ax.set_xticklabels(range(0,25,1))
plt.title('Tip Amount Historgram (By Vendor)');
```



In [44]:
```
# Create histogram of tip_amount by vendor for tips > $10
tips_ten_plus = df[df['tip_amount'] > 10]
plt.figure(figsize=(10,3))
ax = sns.histplot(data=tips_ten_plus, x='tip_amount', bins=range(10,25,1), hue='Ven
ax.set_xticks(range(10,25,1))
ax.set_xticklabels(range(10,25,1))
plt.title('Tip Amount Histogram (By Vendor, Over 10 Dollars)');
```



Mean tips by passenger count

Examine the unique values in the `passenger_count` column.

In [45]:
```
df['passenger_count'].value_counts()
```

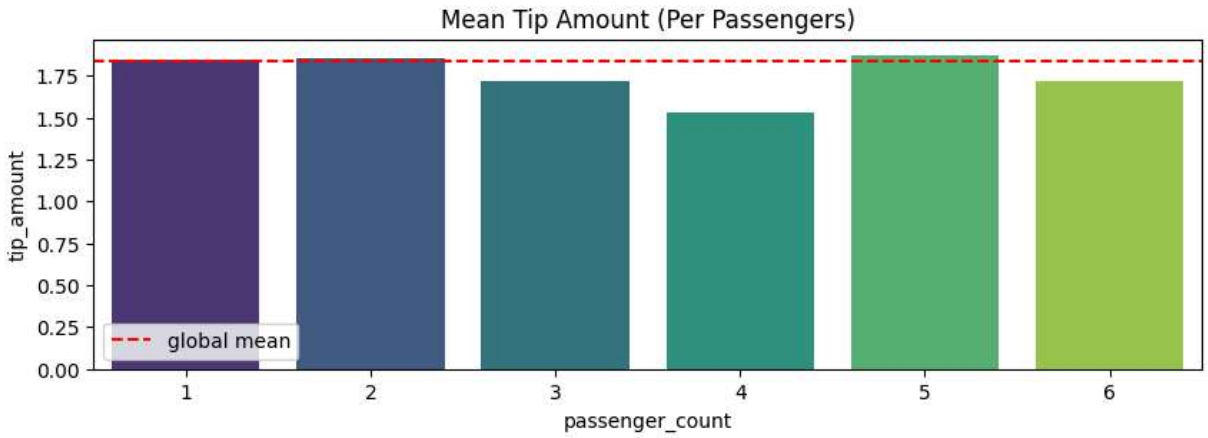
Out[45]:
```
passenger_count
1    16117
2     3305
5     1143
3      953
6      693
4      455
0       33
Name: count, dtype: int64
```

In [46]:
```
# Calculate mean tips by passenger_count
mean_tips_passenger_count = df.groupby(['passenger_count'])[['tip_amount']].mean()
mean_tips_passenger_count
```

Out[46]:

passenger_count	tip_amount
0	2.135758
1	1.848920
2	1.856378
3	1.716768
4	1.530264
5	1.873185
6	1.720260

In [47]:
```
# Create bar plot for mean tips by passenger count
data = mean_tips_passenger_count.tail(-1)
plt.figure(figsize=(10,3))
current_palette = sns.color_palette("viridis", n_colors=len(data))
ax = sns.barplot(
    x=data.index,
    y=data['tip_amount'],
    hue=data.index,
    palette=list(current_palette),  # Explicitly cast to list to avoid warnings
    legend=False
)
ax.axhline(df['tip_amount'].mean(), ls='--', color='red', label='global mean')
ax.legend()
plt.title('Mean Tip Amount (Per Passengers)');
```



Create month and day columns

In [48]:
```
# Create a month column
df['month'] = df['tpep_pickup_datetime'].dt.month_name()
# Create a day column
df['day'] = df['tpep_pickup_datetime'].dt.day_name()
```

Plot total ride count by month

Begin by calculating total ride count by month.

In [49]:
```
# Get total number of rides for each month
monthly_rides = df['month'].value_counts()
monthly_rides
```

Out[49]:
```
month
March        2049
October      2027
April        2019
May          2013
January      1997
June         1964
December     1863
November     1843
February     1769
September    1734
August       1724
July         1697
Name: count, dtype: int64
```

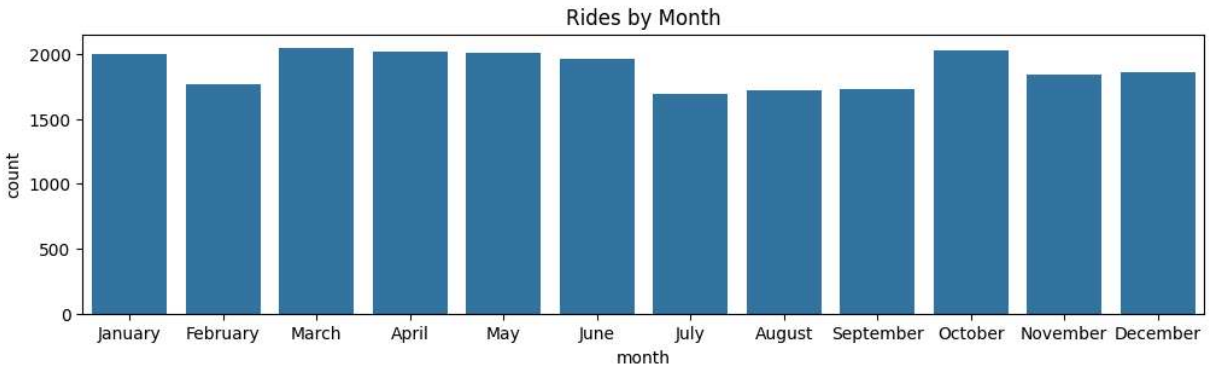
In [50]:
```
# Reorder the monthly ride list so months go in order
month_order = ['January', 'February', 'March', 'April', 'May', 'June', 'July',
         'August', 'September', 'October', 'November', 'December']

monthly_rides = monthly_rides.reindex(index=month_order)
monthly_rides
```

Out[50]:
```
month
January      1997
February     1769
March        2049
April        2019
May          2013
June         1964
July         1697
August       1724
September    1734
October      2027
November     1843
December     1863
Name: count, dtype: int64
```

In [51]:
```
# Show the index
monthly_rides.index
```

Out[51]:
```
Index(['January', 'February', 'March', 'April', 'May', 'June', 'July',
       'August', 'September', 'October', 'November', 'December'],
      dtype='str', name='month')
```

In [52]:
```
# Create a bar plot of total rides per month
plt.figure(figsize=(12,3))
ax = sns.barplot(x=monthly_rides.index, y=monthly_rides)
plt.title('Rides by Month');
```



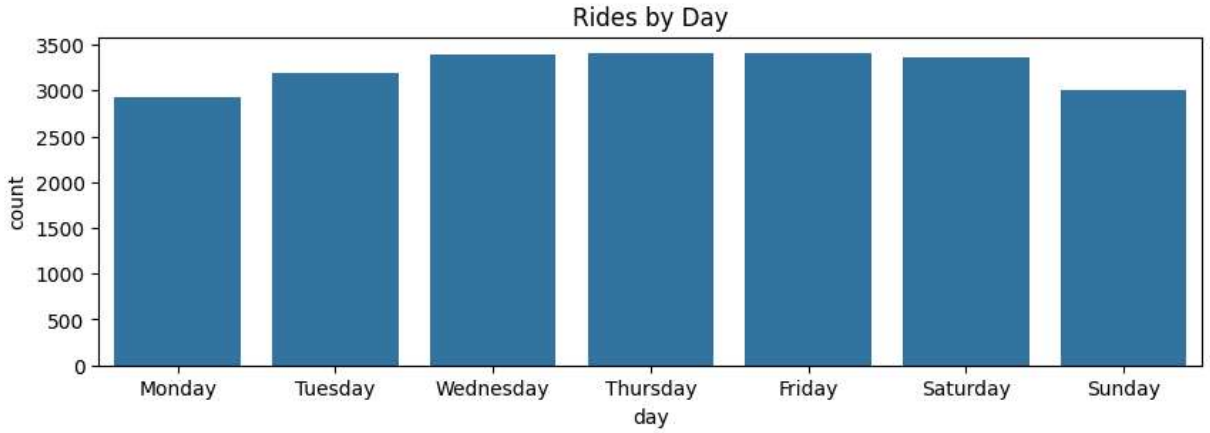
Plot total ride count by day

Repeat the above process, but now calculate the total rides by day of the week.

In [53]:
```
# Repeat the above process, this time for rides by day
daily_rides = df['day'].value_counts()
day_order = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'S
daily_rides = daily_rides.reindex(index=day_order)
daily_rides
```

Out[53]:
```
day
Monday       2931
Tuesday      3198
Wednesday    3390
Thursday     3402
Friday       3413
Saturday     3367
Sunday       2998
Name: count, dtype: int64
```

In [54]:
```
# Create bar plot for ride count by day
plt.figure(figsize=(10,3))
ax = sns.barplot(x=daily_rides.index, y=daily_rides)
plt.title('Rides by Day');
```



Plot total revenue by day of the week

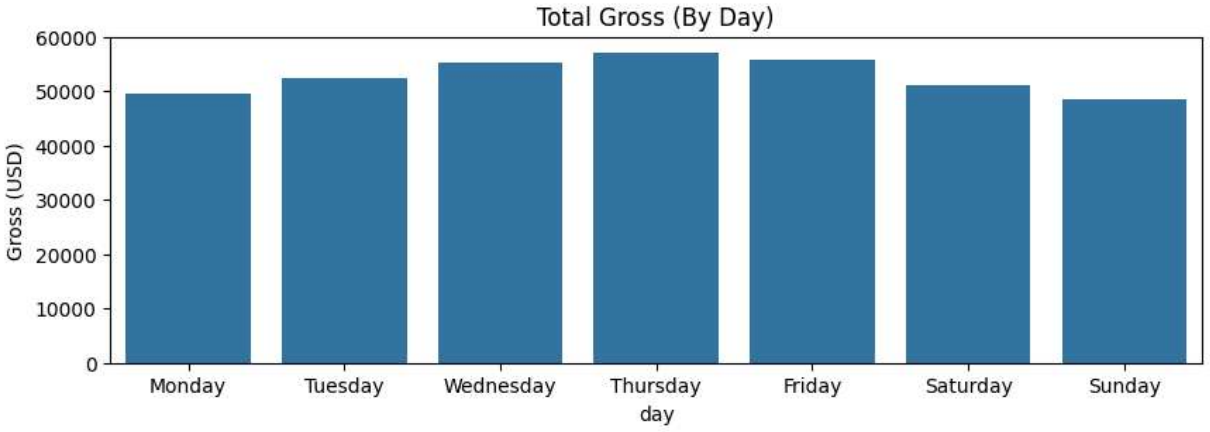
Repeat the above process, but now calculate the total revenue by day of the week.

In [55]:
```
# Repeat the process, this time for total revenue by day
day_order = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'S
total_amount_by_day = df.groupby('day')[['total_amount']].sum()
total_amount_by_day = total_amount_by_day.reindex(index=day_order)
total_amount_by_day
```

Out[55]:

day	total_amount
Monday	49574.37
Tuesday	52527.14
Wednesday	55310.47
Thursday	57181.91
Friday	55818.74
Saturday	51195.40
Sunday	48624.06

In [56]:
```
# Create bar plot of total revenue by day
plt.figure(figsize=(10,3))
ax = sns.barplot(x=total_amount_by_day.index, y=total_amount_by_day['total_amount']
ax.set_ylabel('Gross (USD)')
plt.title('Total Gross (By Day)');
```



Plot total revenue by month

In [57]:
```
# Repeat the process, this time for total revenue by month
total_amount_by_month = df.groupby('month')[['total_amount']].sum()
total_amount_by_month = total_amount_by_month.reindex(index=month_order)
total_amount_by_month
```

Out[57]:

month	total_amount
January	31735.25
February	28937.89
March	33085.89
April	32012.54
May	33828.58
June	32920.52
July	26617.64
August	27759.56
September	28206.38
October	33065.83
November	30800.44
December	31261.57

In [58]:
```
# Create a bar plot of total revenue by month
plt.figure(figsize=(10,3))
ax = sns.barplot(x=total_amount_by_month.index, y=total_amount_by_month['total_amou
plt.title('Total revenue by month');
```

You can create a scatterplot in Tableau Public, which can be easier to manipulate and present. If you'd like step by step instructions, you can review the following link. Those instructions create a scatterplot showing the relationship between total_amount and trip_distance. Consider adding the Tableau visualization to your executive summary, and adding key insights from your findings on those two variables.

Tableau visualization guidelines

Plot mean trip distance by drop-off location

In [59]:
```
# Get number of unique drop-off location IDs
df['DOLocationID'].nunique()
```

Out[59]: 216

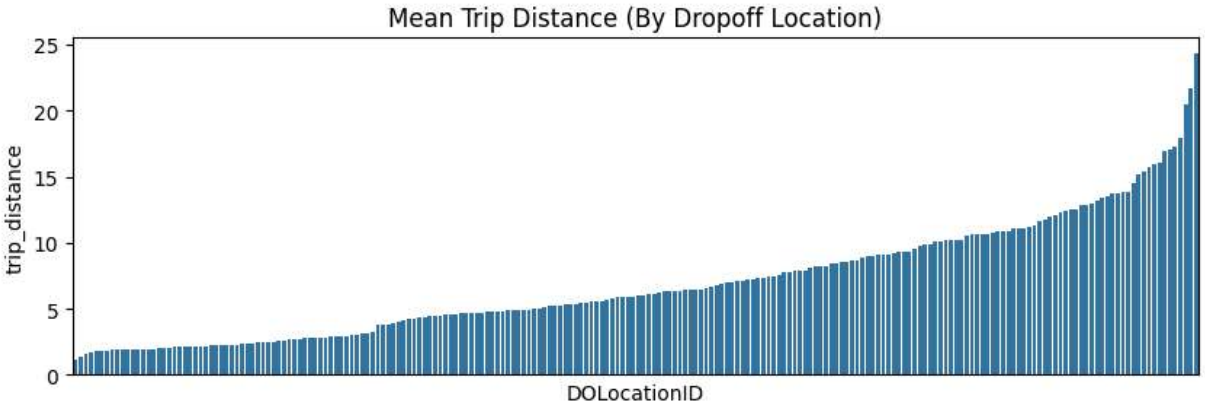
In [60]:
```
# Calculate the mean trip distance for each drop-off location
distance_dropoff = df.groupby('DOLocationID')[['trip_distance']].mean()

# Sort the results in descending order by mean trip distance
distance_dropoff = distance_dropoff.sort_values(by='trip_distance')
distance_dropoff
```

Out[60]:

DOLocationID	trip_distance
207	1.200000
193	1.390556
237	1.555494
234	1.727806
137	1.818852
...	...
51	17.310000
11	17.945000
210	20.500000
29	21.650000
23	24.275000

216 rows × 1 columns

In [61]:
```
# Create a bar plot of mean trip distances by drop-off location in ascending order
plt.figure(figsize=(10,3))
ax = sns.barplot(x=distance_dropoff.index,
                 y=distance_dropoff['trip_distance'],
                 order=distance_dropoff.index)
ax.set_xticklabels([])
ax.set_xticks([])
plt.title('Mean Trip Distance (By Dropoff Location)');
```



Histogram of rides by drop-off location

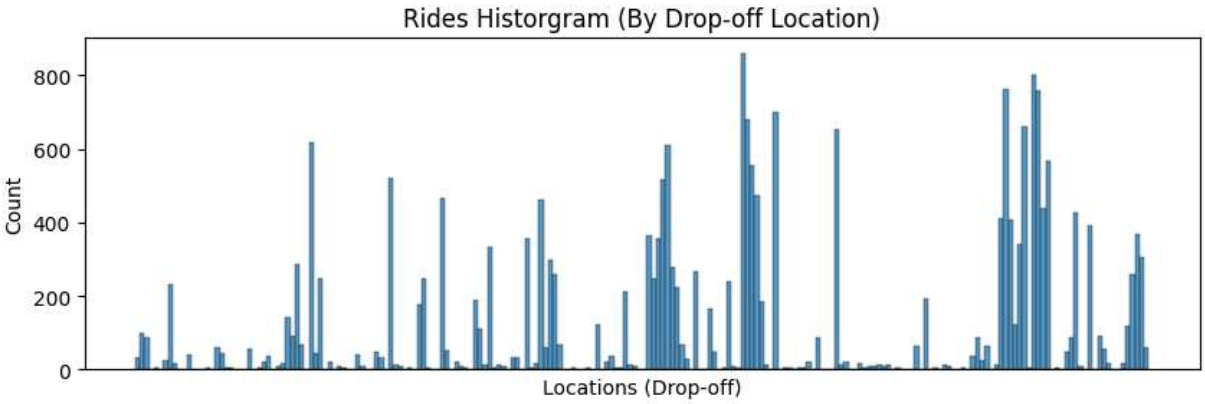
First, check whether the drop-off locations IDs are consecutively numbered. For instance, does it go 1, 2, 3, 4..., or are some numbers missing (e.g., 1, 3, 4...). If numbers aren't all consecutive, the histogram will look like some locations have very few or no rides when in reality there's no bar because there's no location.

There are many ways to do this.

In [62]:
```
# Check if all drop-off locations are consecutively numbered
df['DOLocationID'].max() - len(set(df['DOLocationID']))
```

Out[62]:
```
np.int64(49)
```

In [63]:
```
plt.figure(figsize=(10,3))
sorted_dropoffs = df['DOLocationID'].sort_values()
sorted_dropoffs = sorted_dropoffs.astype('str')
sns.histplot(sorted_dropoffs, bins=range(0, df['DOLocationID'].max(), 1))
plt.xticks([])
plt.xlabel('Locations (Drop-off)')
plt.title('Rides Historgram (By Drop-off Location)');
```



PACE: Execute

Consider the PACE Strategy Document to reflect on the Execute stage.

Task 4a. Results and evaluation

Having built visualizations in Tableau and in Python, what have you learned about the dataset? What other questions have your visualizations uncovered that you should pursue?

Pro tip: Put yourself in your client's perspective. What would they want to know?

Use the following code fields to pursue any additional EDA based on the visualizations you've already plotted. Also use the space to make sure your visualizations are clean, easily understandable, and accessible.

Ask yourself: Did you consider color, contrast, emphasis, and labeling?

[Learners: insert your response here]

I learned that most Wednesday - Friday contain the highest revenue for rides, and most rides occur with less than a 5 mile distance from pick-up to drop-off.

My questions are what other connections can be made about the data, such as are tips higher on the weekends even though rides are less abundant.

The client might ask about durations of trips, that was not apart of the data but it can be easily procured.

In [64]:
```
df['trip_duration'] = (df['tpep_dropoff_datetime']-df['tpep_pickup_datetime'])
```

In [65]:
```
df.head(10)
```

Out[65]:

	Unnamed: 0	VendorID	tpep_pickup_datetime	tpep_dropoff_datetime	passenger_count	tı
0	24870114	2	2017-03-25 08:55:43	2017-03-25 09:09:47	6	
1	35634249	1	2017-04-11 14:53:28	2017-04-11 15:19:58	1	
2	106203690	1	2017-12-15 07:26:56	2017-12-15 07:34:08	1	
3	38942136	2	2017-05-07 13:17:59	2017-05-07 13:48:14	1	
4	30841670	2	2017-04-15 23:32:20	2017-04-15 23:49:03	1	
5	23345809	2	2017-03-25 20:34:11	2017-03-25 20:42:11	6	
6	37660487	2	2017-05-03 19:04:09	2017-05-03 20:03:47	1	
7	69059411	2	2017-08-15 17:41:06	2017-08-15 18:03:05	1	
8	8433159	2	2017-02-04 16:17:07	2017-02-04 16:29:14	1	
9	95294817	1	2017-11-10 15:20:29	2017-11-10 15:40:55	1	

10 rows × 21 columns

Task 4b. Conclusion

Make it professional and presentable

You have visualized the data you need to share with the director now. Remember, the goal of a data visualization is for an audience member to glean the information on the chart in mere seconds.

Questions to ask yourself for reflection: Why is it important to conduct Exploratory Data Analysis? Why are the data visualizations provided in this notebook useful?

EDA is important because EDA helps a data professional to understand the data, visualize the outliers, clean null or missing values, and prepare the data for future representation.

Visualizations helped me understand the way the data forms, how groups interact with eachother, and the min/max attributes related to outliers.

You've now completed professional data visualizations according to a business need. Well done!

Congratulations! You've completed this lab. However, you may not notice a green check mark next to this item on Coursera's platform. Please continue your progress regardless of the check mark. Just click on the "save" icon at the top of this notebook to ensure your work has been logged.