Activity_Course 3 Automatidata project lab

December 5, 2025

1 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 in an early stage of 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.

1.0.1 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 C2 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.

Be sure to complete this activity before moving on. The next course item will provide you with a completed exemplar to compare to your own work.

2 Visualize a story in Tableau and Python

3 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.

3.1 PACE: Plan

In this stage, consider the following questions where applicable to complete your code response: 1. Identify any outliers:

- What methods are best for identifying outliers?
- How do you make the decision to keep or exclude outliers from any future models?

Plotting a Scatter plot would be the best method to identify outliers. By investigating the outliers i.e. whether they have any impact on the data, whether they can cause a significant bias or not etc. we can decide whether to keep them for the future models.

3.1.1 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: Tableau Public: 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.

[11]:
```
# Import packages and libraries

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import datetime
```

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.

[12]:
```
df = pd.read_csv('2017_Yellow_Taxi_Trip_Data.csv')
```

3.2 PACE: Analyze

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

3.2.1 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()

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

Are there data outliers? What are they and how might you handle them?

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

Missing data should be removed as the empty values of key variables can cause vague conclusions or the Y Outputs. Yes data can have outliers which usually refers to unusual values of the key variables. They could be less or not reasonable. There are various techniques to handle outliers. These are by plotting them on visual plots, identifying them visually, and rrmoving them if requied after investigation. Distributions can tell about the shape of the data. Whether it is skewed or normal and so on. It can tell which side the distribution is skewed which can indicate effect of outliers on the distribution. This is further addressed by normalization via statistical methods.

Start by discovering, using head and size.

[3]:
```
df.head()
```

[3]:
```
   Unnamed: 0  VendorID    tpep_pickup_datetime   tpep_dropoff_datetime  \
0    24870114         2   03/25/2017 8:55:43 AM   03/25/2017 9:09:47 AM
1    35634249         1   04/11/2017 2:53:28 PM   04/11/2017 3:19:58 PM
2   106203690         1   12/15/2017 7:26:56 AM   12/15/2017 7:34:08 AM
3    38942136         2   05/07/2017 1:17:59 PM   05/07/2017 1:48:14 PM
4    30841670         2  04/15/2017 11:32:20 PM  04/15/2017 11:49:03 PM

   passenger_count  trip_distance  RatecodeID store_and_fwd_flag  \
0                6           3.34           1                  N
1                1           1.80           1                  N
2                1           1.00           1                  N
3                1           3.70           1                  N
4                1           4.37           1                  N

   PULocationID  DOLocationID  payment_type  fare_amount  extra  mta_tax  \
0           100           231             1         13.0    0.0      0.5
1           186            43             1         16.0    0.0      0.5
2           262           236             1          6.5    0.0      0.5
3           188            97             1         20.5    0.0      0.5
4             4           112             2         16.5    0.5      0.5

   tip_amount  tolls_amount  improvement_surcharge  total_amount
0        2.76           0.0                    0.3         16.56
1        4.00           0.0                    0.3         20.80
2        1.45           0.0                    0.3          8.75
3        6.39           0.0                    0.3         27.69
4        0.00           0.0                    0.3         17.80
```

[4]:
```
df.size
```

[4]:
```
408582
```

Use describe…

[5]:
```
df.describe()
```

[5]:
```
         Unnamed: 0      VendorID  passenger_count  trip_distance  \
count  2.269900e+04  22699.000000     22699.000000   22699.000000
mean   5.675849e+07      1.556236         1.642319       2.913313
std    3.274493e+07      0.496838         1.285231       3.653171
```

```
min    1.212700e+04      1.000000         0.000000       0.000000
25%    2.852056e+07      1.000000         1.000000       0.990000
50%    5.673150e+07      2.000000         1.000000       1.610000
75%    8.537452e+07      2.000000         2.000000       3.060000
max    1.134863e+08      2.000000         6.000000      33.960000

         RatecodeID  PULocationID  DOLocationID  payment_type   fare_amount  \
count  22699.000000  22699.000000  22699.000000  22699.000000  22699.000000
mean       1.043394    162.412353    161.527997      1.336887     13.026629
std        0.708391     66.633373     70.139691      0.496211     13.243791
min        1.000000      1.000000      1.000000      1.000000   -120.000000
25%        1.000000    114.000000    112.000000      1.000000      6.500000
50%        1.000000    162.000000    162.000000      1.000000      9.500000
75%        1.000000    233.000000    233.000000      2.000000     14.500000
max       99.000000    265.000000    265.000000      4.000000    999.990000

              extra       mta_tax    tip_amount  tolls_amount  \
count  22699.000000  22699.000000  22699.000000  22699.000000
mean       0.333275      0.497445      1.835781      0.312542
std        0.463097      0.039465      2.800626      1.399212
min       -1.000000     -0.500000      0.000000      0.000000
25%        0.000000      0.500000      0.000000      0.000000
50%        0.000000      0.500000      1.350000      0.000000
75%        0.500000      0.500000      2.450000      0.000000
max        4.500000      0.500000    200.000000     19.100000

       improvement_surcharge  total_amount
count           22699.000000  22699.000000
mean                0.299551     16.310502
std                 0.015673     16.097295
min                -0.300000   -120.300000
25%                 0.300000      8.750000
50%                 0.300000     11.800000
75%                 0.300000     17.800000
max                 0.300000   1200.290000
```

And info.

[6]:
```
df.info()
```

```
<class 'pandas.core.frame.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  object
 3   tpep_dropoff_datetime  22699 non-null  object
 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  object
 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), object(3)
memory usage: 3.1+ MB
```

3.2.2 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 assure 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.

3.2.3 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

Box plot, Scatter plot

3.3 PACE: Construct

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

3.3.1 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)!

3.3.2 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.

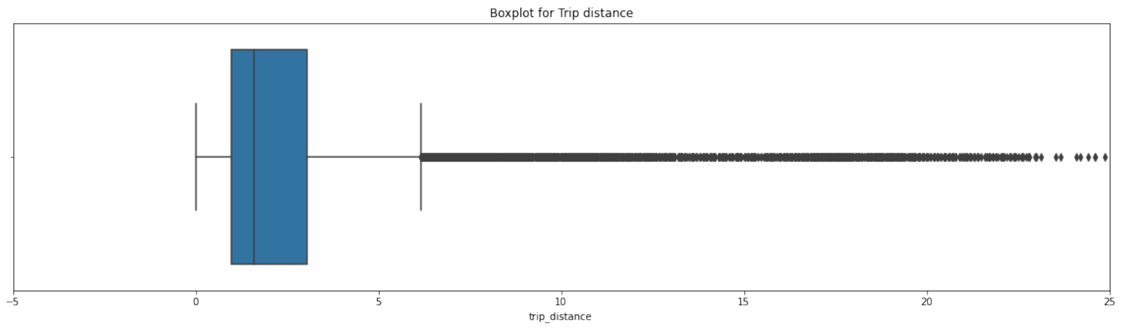
Note: Remember to convert your date columns to datetime in order to derive total trip duration.

[7]:
```
# Convert data columns to datetime

#no columns in the format of date
df['tpep_dropoff_datetime'] = pd.to_datetime(df['tpep_dropoff_datetime'])
df['tpep_pickup_datetime'] = pd.to_datetime(df['tpep_pickup_datetime'])
df['trip_duration'] = df['tpep_dropoff_datetime'] - df['tpep_pickup_datetime']
df['trip_duration'] = df['trip_duration'] / np.timedelta64(1, 'h')
df.head()
```

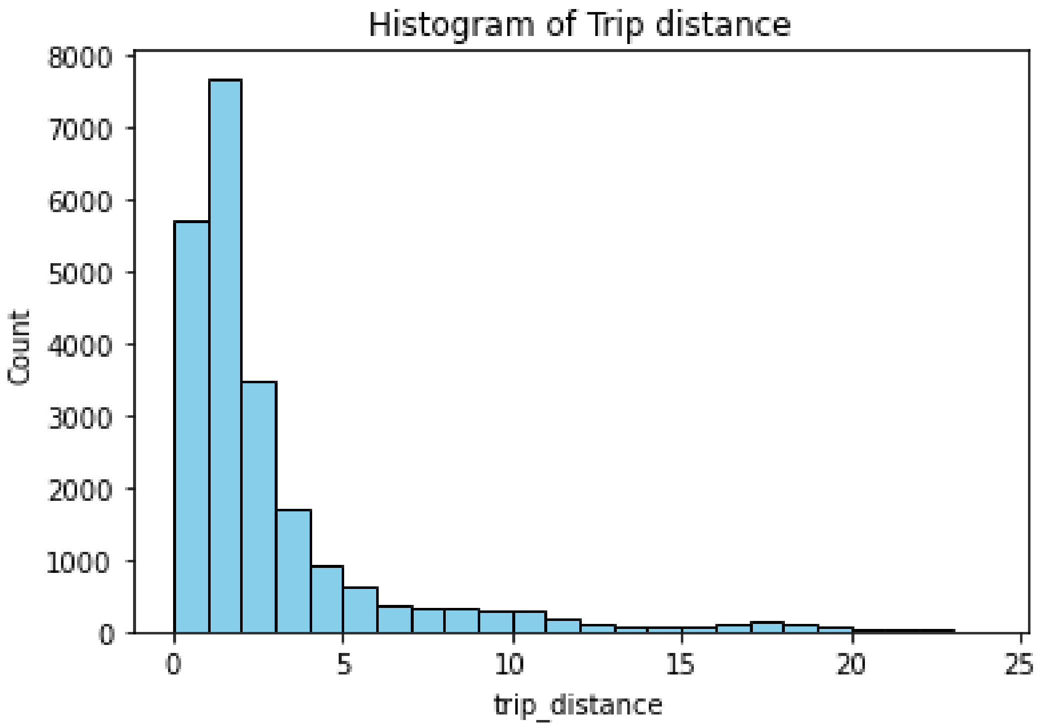
[7]:
```
   Unnamed: 0  VendorID tpep_pickup_datetime tpep_dropoff_datetime  \
0    24870114         2  2017-03-25 08:55:43   2017-03-25 09:09:47
1    35634249         1  2017-04-11 14:53:28   2017-04-11 15:19:58
2   106203690         1  2017-12-15 07:26:56   2017-12-15 07:34:08
3    38942136         2  2017-05-07 13:17:59   2017-05-07 13:48:14
4    30841670         2  2017-04-15 23:32:20   2017-04-15 23:49:03

   passenger_count  trip_distance  RatecodeID store_and_fwd_flag  \
0                6           3.34           1                  N
1                1           1.80           1                  N
2                1           1.00           1                  N
3                1           3.70           1                  N
4                1           4.37           1                  N

   PULocationID  DOLocationID  payment_type  fare_amount  extra  mta_tax  \
0           100           231             1         13.0    0.0      0.5
1           186            43             1         16.0    0.0      0.5
2           262           236             1          6.5    0.0      0.5
3           188            97             1         20.5    0.0      0.5
4             4           112             2         16.5    0.5      0.5

   tip_amount  tolls_amount  improvement_surcharge  total_amount  \
0        2.76           0.0                     0.3         16.56
1        4.00           0.0                     0.3         20.80
2        1.45           0.0                     0.3          8.75
3        6.39           0.0                     0.3         27.69
4        0.00           0.0                     0.3         17.80

   trip_duration
0       0.234444
1       0.441667
2       0.120000
3       0.504167
4       0.278611
```

trip distance

[8]:
```python
# Create box plot of trip_distance

plt.figure(figsize=(20,5))
sns.boxplot(x = df['trip_distance'])
plt.title('Boxplot for Trip distance')
plt.xlim(-5,25)

plt.show()
```



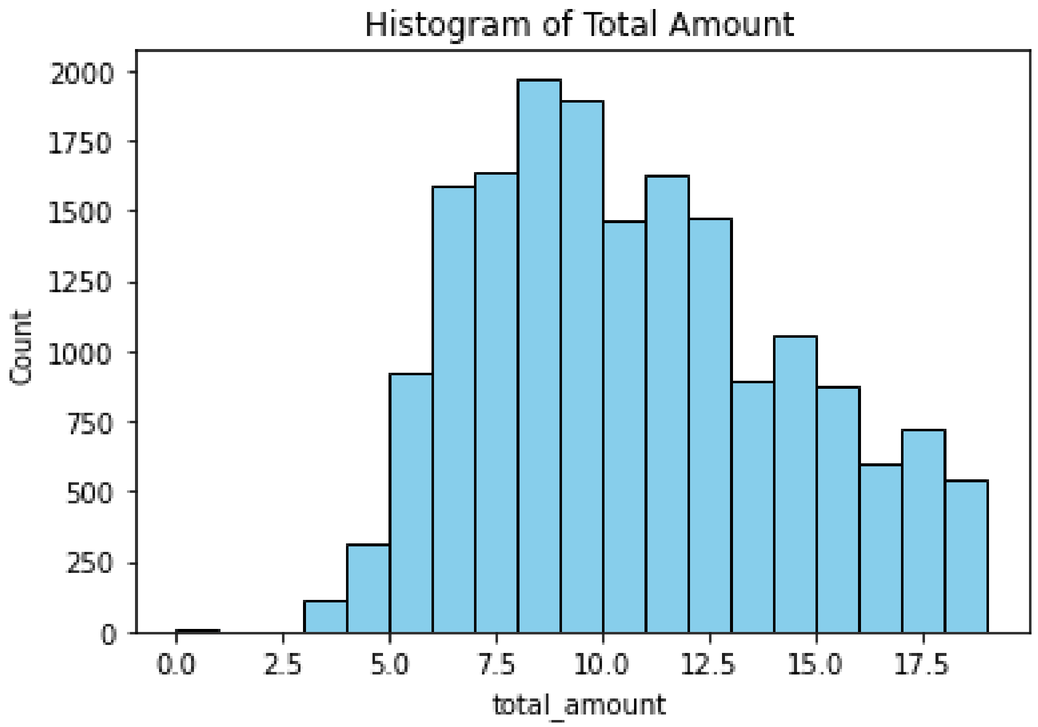
[64]:
```python
# Create histogram of trip_distance

plt.hist(df['trip_distance'], bins=range(0,25,1), color='skyblue',␣
 ↪edgecolor='black')
plt.xlabel('trip_distance')
plt.ylabel('Count')
plt.title('Histogram of Trip distance')

plt.show()
```



total amount

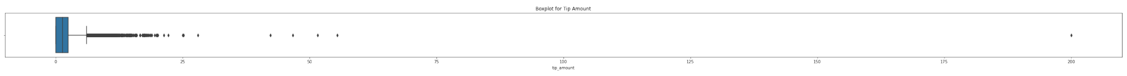
[10]:
```
# Create box plot of total_amount

plt.figure(figsize=(100,2))
sns.boxplot(x = df['total_amount'])
plt.title('Boxplot for Total Amount')
plt.xlim(-500,1300)
plt.show()
```

[65]:
```
# Create histogram of total_amount

plt.hist(df['total_amount'], bins=range(0,20,1), color='skyblue',␣
 ↪edgecolor='black')
plt.xlabel('total_amount')
plt.ylabel('Count')
plt.title('Histogram of Total Amount')

plt.show()
```

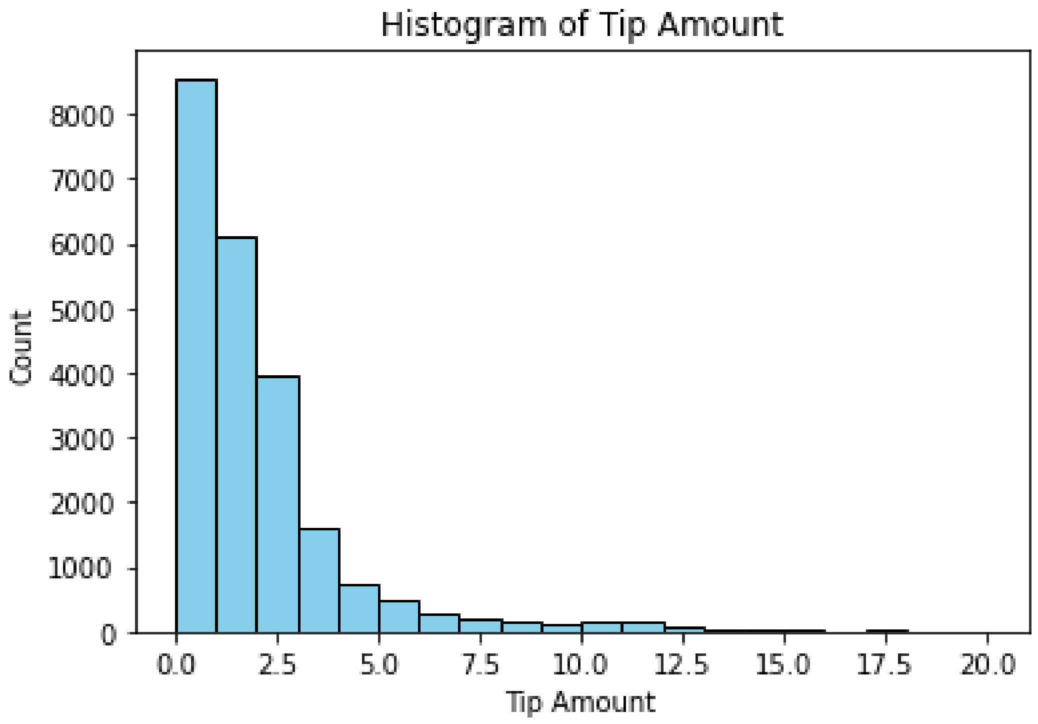


tip amount

[12]:
```
# Create box plot of tip_amount
plt.figure(figsize=(50,2))
sns.boxplot(x = df['tip_amount'])
plt.title('Boxplot for Tip Amount')
plt.show()
```

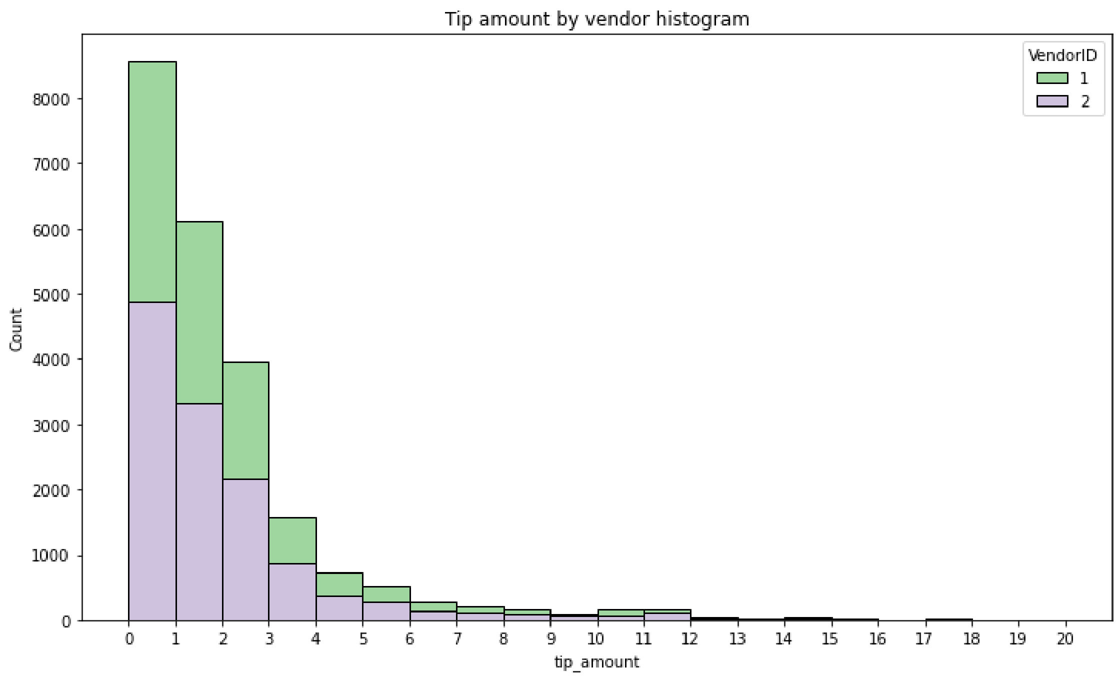


[67]:
```
# Create histogram of tip_amount
plt.hist(df['tip_amount'], bins=range(0,21,1), color='skyblue',␣
 ↪edgecolor='black')
plt.xlabel('Tip Amount')
plt.ylabel('Count')
plt.title('Histogram of Tip Amount')

plt.show()
```



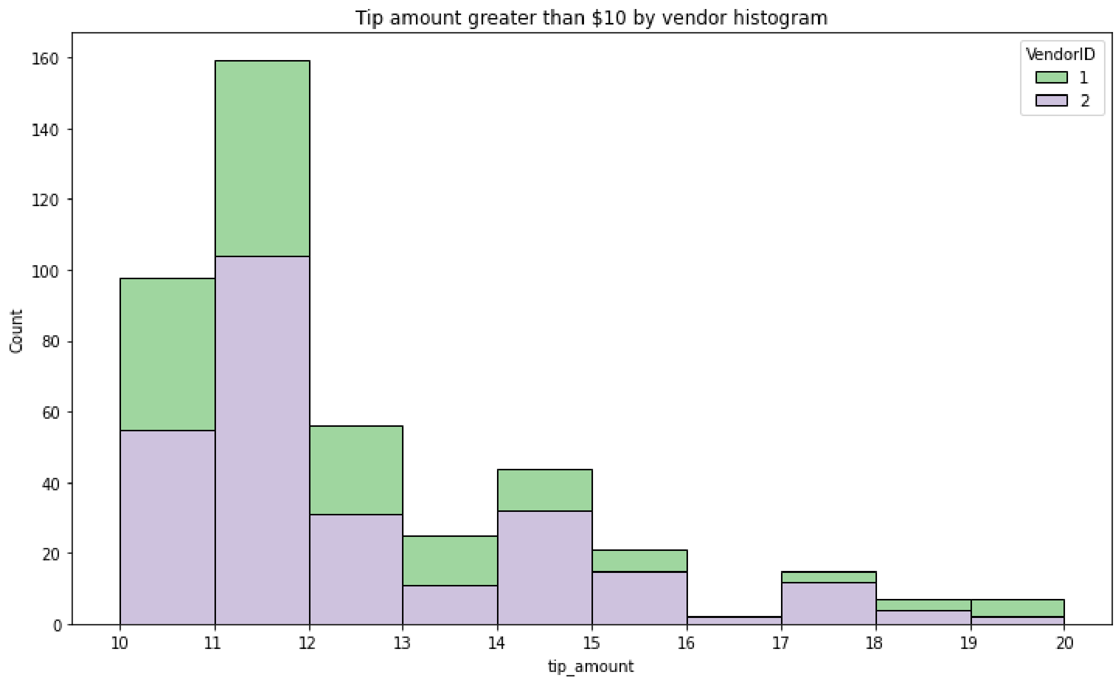
tip_amount by vendor

[68]:
```
# Create histogram of tip_amount by vendor

plt.figure(figsize=(12,7))
ax = sns.histplot(data=df, x='tip_amount', bins=range(0,21,1),
                  hue='VendorID',
                  multiple='stack',
                  palette='Accent')
ax.set_xticks(range(0,21,1))
ax.set_xticklabels(range(0,21,1))
plt.title('Tip amount by vendor histogram');
```



Next, zoom in on the upper end of the range of tips to check whether vendor one gets noticeably more of the most generous tips.

[69]:
```python
# Create histogram of tip_amount by vendor for tips > $10
df_tips_10_bigger = df[df['tip_amount']>10]
plt.figure(figsize=(12,7))
ax = sns.histplot(data=df_tips_10_bigger, x='tip_amount', bins=range(10,21,1),
                  hue='VendorID',
                  multiple='stack',
                  palette='Accent')
ax.set_xticks(range(10,21,1))
ax.set_xticklabels(range(10,21,1))
plt.title('Tip amount greater than $10 by vendor histogram');
```



Mean tips by passenger count

Examine the unique values in the `passenger_count` column.

[16]:
```
df['passenger_count'].unique()
```

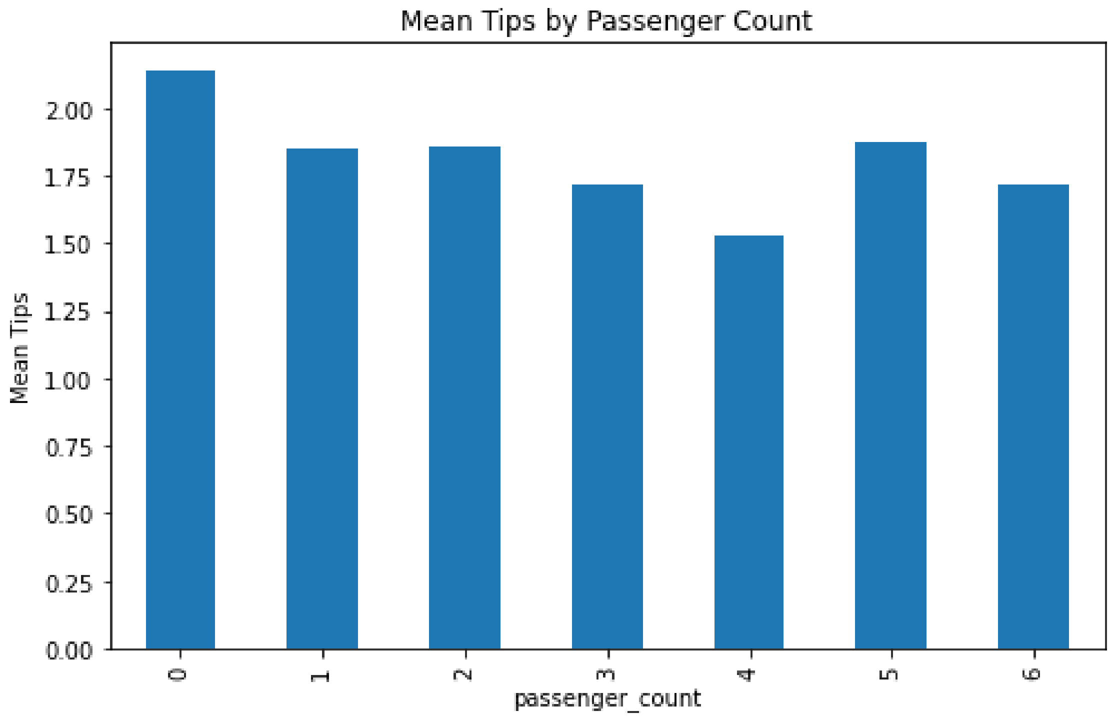
[16]:
```
array([6, 1, 2, 4, 5, 3, 0])
```

[17]:
```
# Calculate mean tips by passenger_count
mean_tip_by_pcount = df.groupby('passenger_count')['tip_amount'].mean()
mean_tip_by_pcount
```

[17]:
```
passenger_count
0    2.135758
1    1.848920
2    1.856378
3    1.716768
4    1.530264
5    1.873185
6    1.720260
Name: tip_amount, dtype: float64
```

[18]:
```
# Create bar plot for mean tips by passenger count
mean_tip_by_pcount.plot(kind = 'bar', figsize = (8,5))
plt.title('Mean Tips by Passenger Count')
plt.ylabel('Mean Tips')
```

[18]:
```
Text(0, 0.5, 'Mean Tips')
```



Create month and day columns

[19]:
```
# Create a month column
df['Month'] = df['tpep_pickup_datetime'].dt.month

# 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.

[28]:
```
# Get total number of rides for each month
trips_by_month = df.groupby('Month').size()
trips_by_month
```

[28]:
```
Month
1      1997
2      1769
3      2049
4      2019
5      2013
6      1964
```

```
7     1697
8     1724
9     1734
10    2027
11    1843
12    1863
dtype: int64
```

Reorder the results to put the months in calendar order.

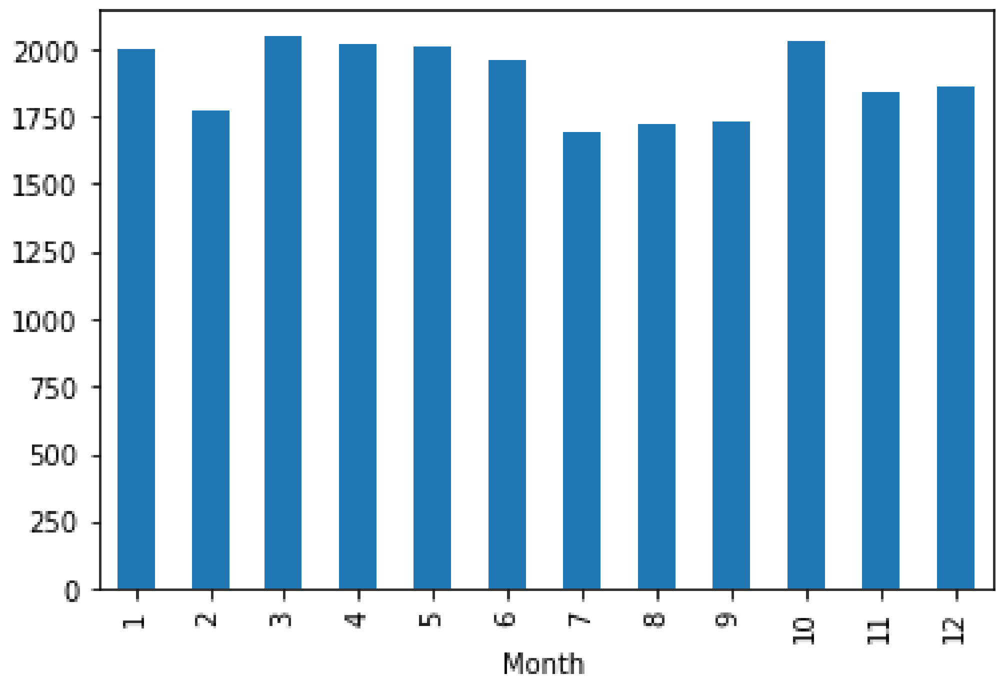
```
[32]: # Reorder the monthly ride list so months go in orde
      # Not required if months are of numerical type
```

```
[30]: # Show the index
      trips_by_month.reset_index()
```

```
[30]:     Month     0
      0       1  1997
      1       2  1769
      2       3  2049
      3       4  2019
      4       5  2013
      5       6  1964
      6       7  1697
      7       8  1724
      8       9  1734
      9      10  2027
      10     11  1843
      11     12  1863
```

```
[33]: # Create a bar plot of total rides per month
      trips_by_month.plot.bar()
```

```
[33]: <matplotlib.axes._subplots.AxesSubplot at 0x727873c22d50>
```



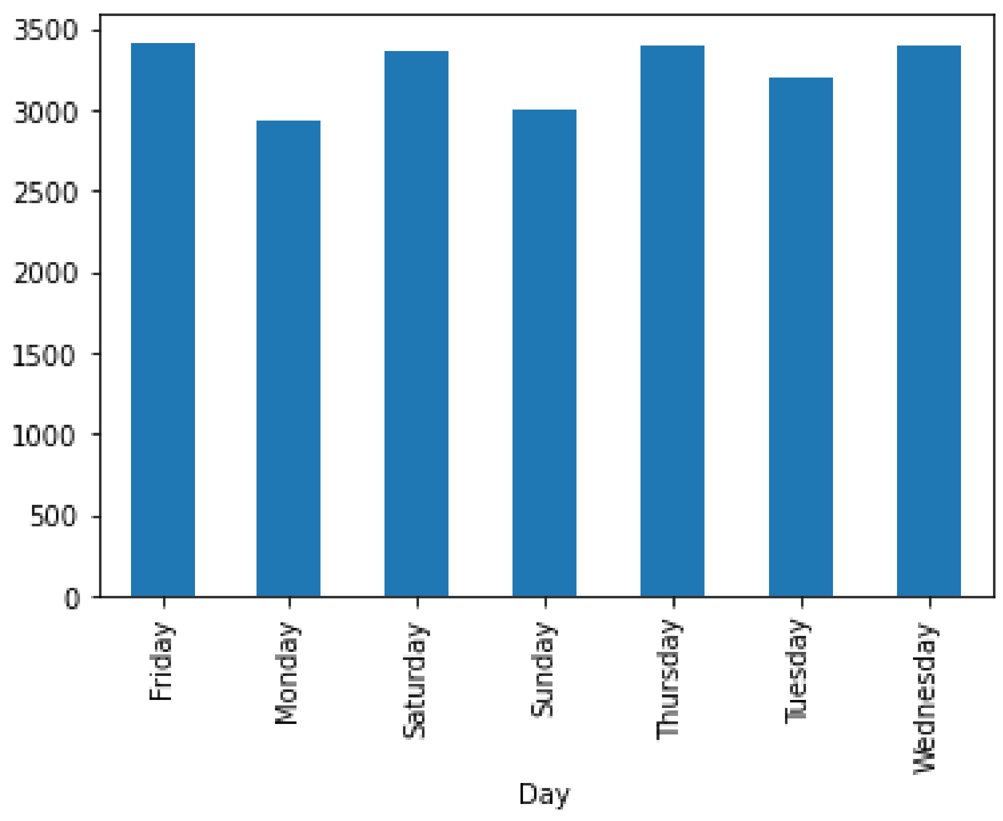
Plot total ride count by day

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

```
[34]: # Repeat the above process, this time for rides by day
      trips_by_day = df.groupby('Day').size()
```

```
[35]: # Create bar plot for ride count by day
      trips_by_day.plot.bar()
```

```
[35]: <matplotlib.axes._subplots.AxesSubplot at 0x727873b3ac90>
```



Plot total revenue by day of the week

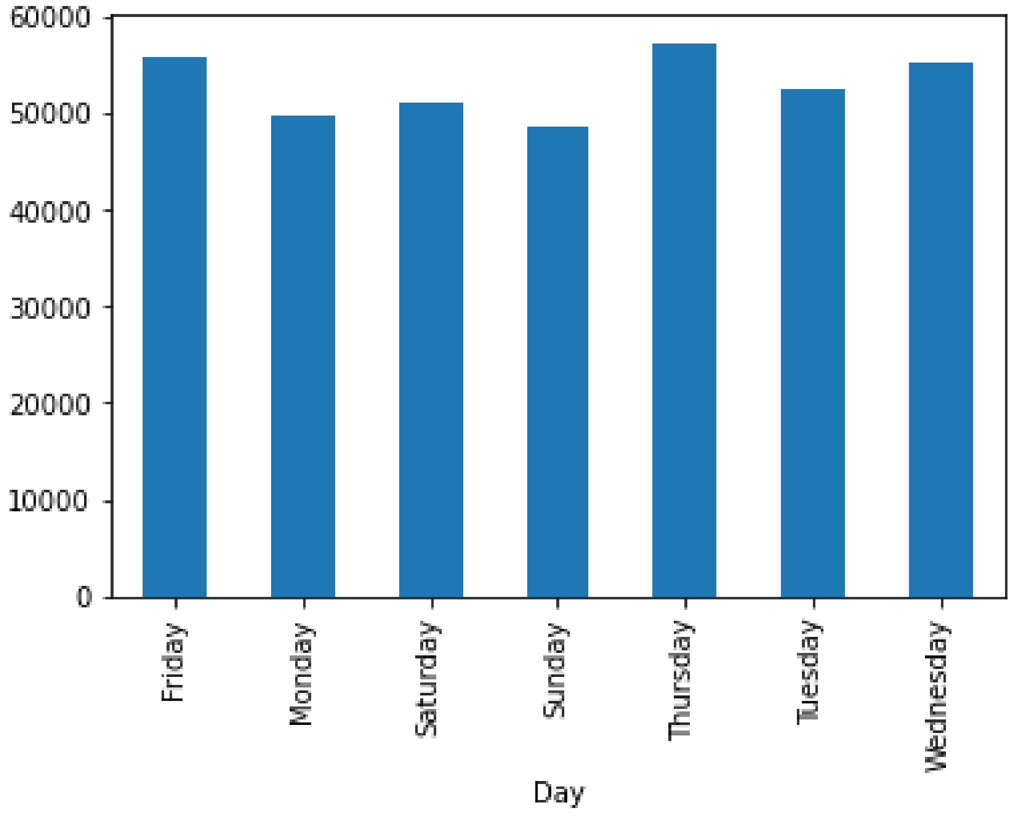
Repeat the above process, but now calculate the total revenue by day of the week.

[36]:
```
# Repeat the process, this time for total revenue by day
total_revenue_by_day = df.groupby('Day')['total_amount'].sum()
total_revenue_by_day
```

[36]:
```
Day
Friday       55818.74
Monday       49574.37
Saturday     51195.40
Sunday       48624.06
Thursday     57181.91
Tuesday      52527.14
Wednesday    55310.47
Name: total_amount, dtype: float64
```

[37]:
```
# Create bar plot of total revenue by day
total_revenue_by_day.plot.bar()
```

[37]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x727873b9add0>
```

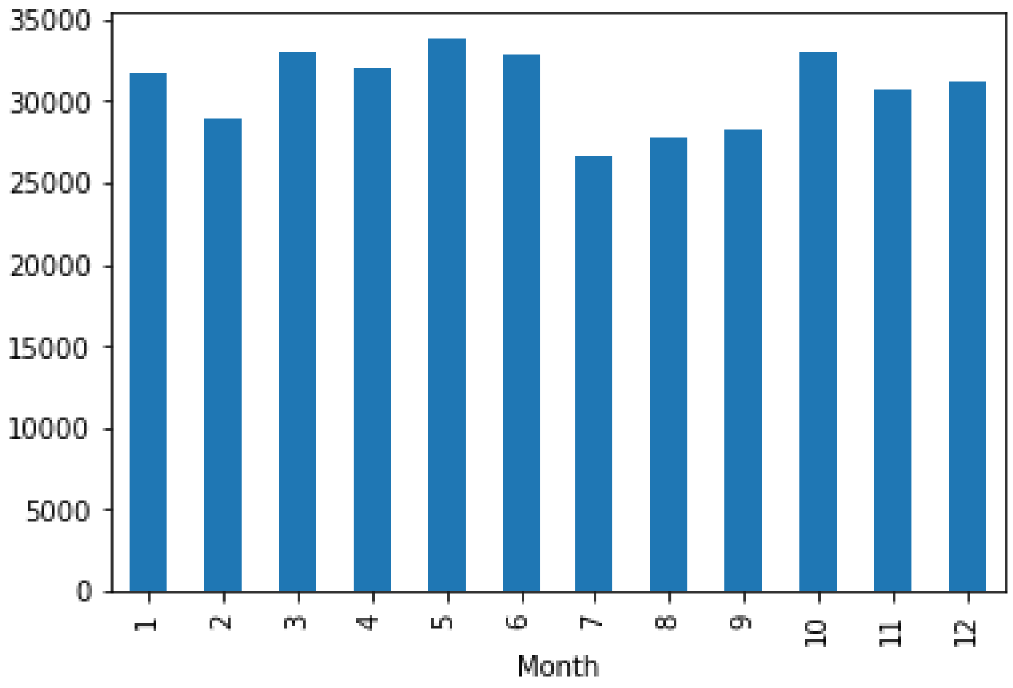


Plot total revenue by month

```
[38]: # Repeat the process, this time for total revenue by month
      total_revenue_by_month = df.groupby('Month')['total_amount'].sum()
```

```
[39]: # Create a bar plot of total revenue by month
      total_revenue_by_month.plot.bar()
```

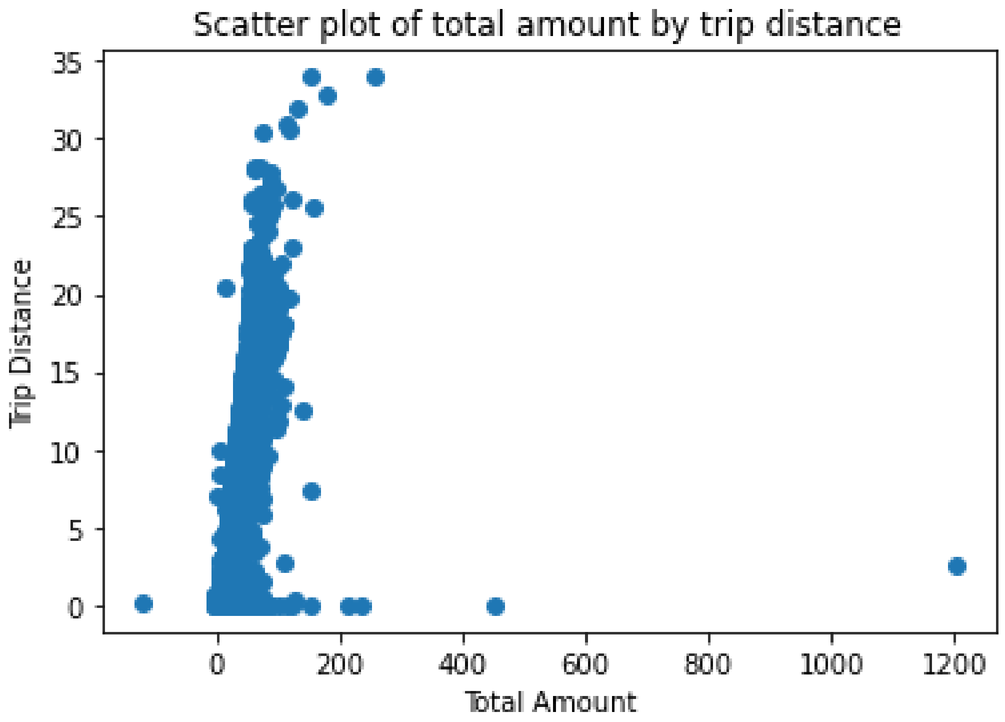
```
[39]: <matplotlib.axes._subplots.AxesSubplot at 0x7278738a0d90>
```



Scatter plot

```
[10]: plt.scatter(df['total_amount'], df['trip_distance'])
      plt.xlabel('Total Amount')
      plt.ylabel('Trip Distance')
      plt.title('Scatter plot of total amount by trip distance')
```

```
[10]: Text(0.5, 1.0, 'Scatter plot of total amount by trip distance')
```



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

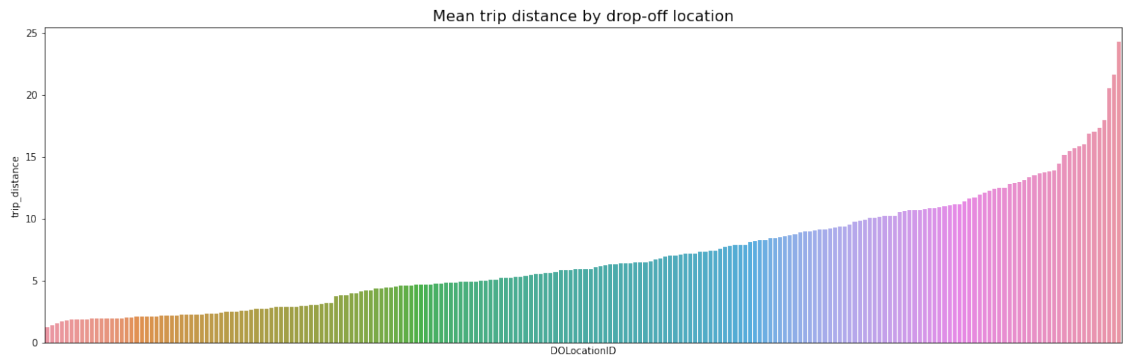
```
[40]: # Get number of unique drop-off location IDs
      df['DOLocationID'].nunique()
```

```
[40]: 216
```

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

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

[41]:
```
              trip_distance
DOLocationID
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 x 1 columns]
```

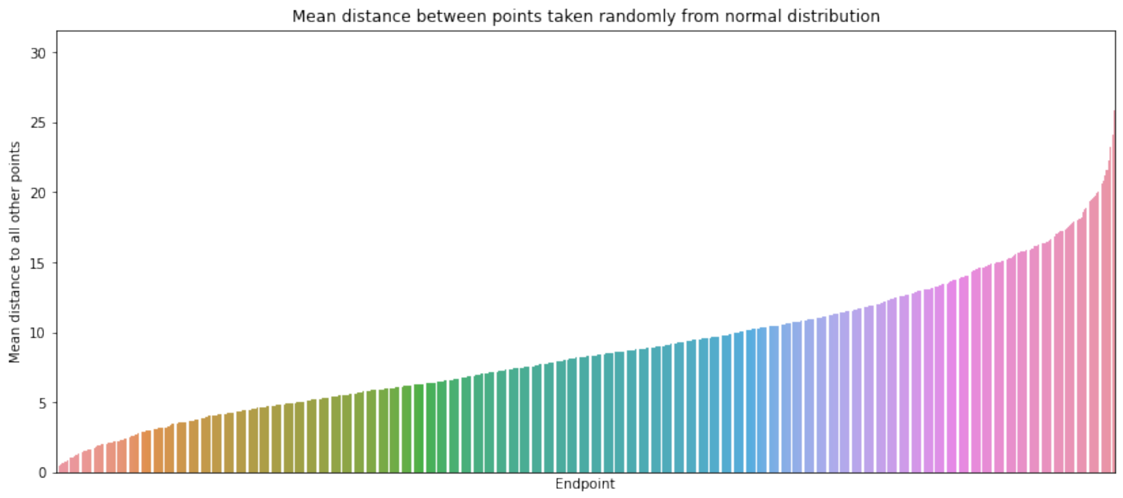
[42]:
```python
# Create a bar plot of mean trip distances by drop-off location in ascending␣
↪order by distance
plt.figure(figsize=(20,6))
ax = sns.barplot(x=mean_distance_by_dlocation.index,
                 y=mean_distance_by_dlocation['trip_distance'],
                 order=mean_distance_by_dlocation.index)
ax.set_xticklabels([])
ax.set_xticks([])
plt.title('Mean trip distance by drop-off location', fontsize=16);
```



3.4 BONUS CONTENT

To confirm your conclusion, consider the following experiment: 1. Create a sample of coordinates from a normal distribution—in this case 1,500 pairs of points from a normal distribution with a mean of 10 and a standard deviation of 5 2. Calculate the distance between each pair of coordinates 3. Group the coordinates by endpoint and calculate the mean distance between that endpoint and all other points it was paired with 4. Plot the mean distance for each unique endpoint

[47]:
```
#BONUS CONTENT

#1. Generate random points on a 2D plane from a normal distribution
sample = np.round(np.random.normal(10, 5, (3000, 2)), 1)
midway = int(len(sample)/2)
start = sample[:midway]
end = sample[midway:]

# 2. Calculate Euclidean distances between points in first half and second half␣
 ↪of array
distance = (start - end) ** 2
distance = distance.sum(axis=-1)
distance = np.sqrt(distance)

# 3. Group the coordinates by "drop-off location", compute mean distance
sample_df = pd.DataFrame({'start': [tuple(x) for x in start.tolist()],
                   'end': [tuple(x) for x in end.tolist()],
                   'distance': distance})
data = sample_df[['end', 'distance']].groupby('end').mean()
data = data.sort_values(by='distance')

# 4. Plot the mean distance between each endpoint ("drop-off location") and all␣
 ↪points it connected to
plt.figure(figsize=(14,6))
ax = sns.barplot(x=data.index,
                 y=data['distance'],
                 order=data.index)
ax.set_xticklabels([])
ax.set_xticks([])
ax.set_xlabel('Endpoint')
ax.set_ylabel('Mean distance to all other points')
ax.set_title('Mean distance between points taken randomly from normal␣
 ↪distribution');
```

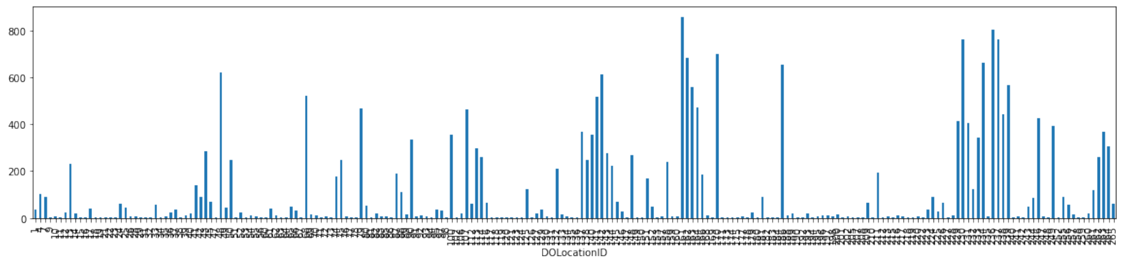


Histogram of rides by drop-off location

First, check to 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.

[73]:
```
# Check if all drop-off locations are consecutively numbered
plt.figure(figsize=(20,4))
rides_by_DOloc = df.groupby('DOLocationID').size()
rides_by_DOloc.plot.bar()
```

[73]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x72786c4e4a10>
```

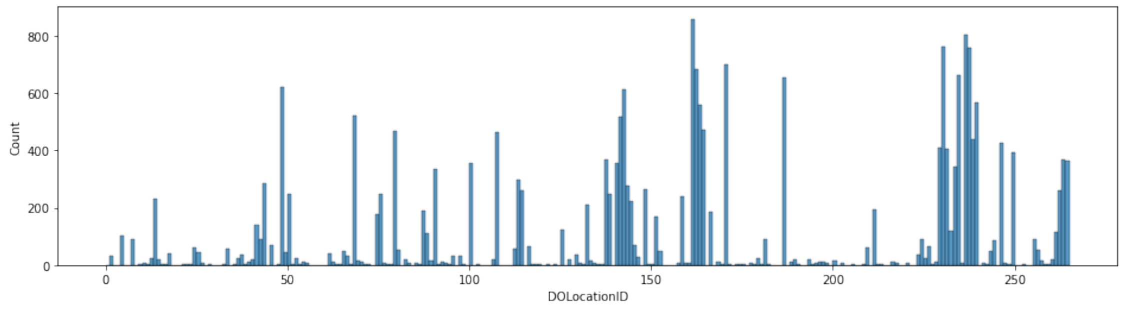


To eliminate the spaces in the historgram that these missing numbers would create, sort the unique drop-off location values, then convert them to strings. This will make the histplot function display all bars directly next to each other.

[78]:
```
plt.figure(figsize=(16,4))
# DOLocationID column is numeric, so sort in ascending order
df.sort_values(by = 'DOLocationID')
```

```
# Convert to string
df['DOLocationID'].astype(str)

# Plot
sns.histplot(data = df['DOLocationID'], bins=range(0, df['DOLocationID'].
↪max()+1, 1))
```

[78]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x72786c05d790>
```



3.5 PACE: Execute

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

3.5.1 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?

I have learned..Rides are highest on Thursday, tip amount between 11-12 was highest for vendor 1 and most for vendor 2.

My other questions are..why total amount was zero for a ride?

My client would likely want to know..Seasonal considerations can help clients to draw conculsions for their ride data i.e. how ride counts are affected in winter season if so.

[72]:
```
df['Trip duration unit'] = 'hours'
df.head()
```

```
[72]:    Unnamed: 0  VendorID tpep_pickup_datetime tpep_dropoff_datetime  \
      0    24870114         2  2017-03-25 08:55:43   2017-03-25 09:09:47
      1    35634249         1  2017-04-11 14:53:28   2017-04-11 15:19:58
      2   106203690         1  2017-12-15 07:26:56   2017-12-15 07:34:08
      3    38942136         2  2017-05-07 13:17:59   2017-05-07 13:48:14
      4    30841670         2  2017-04-15 23:32:20   2017-04-15 23:49:03

         passenger_count  trip_distance  RatecodeID store_and_fwd_flag  \
      0                6           3.34           1                  N
      1                1           1.80           1                  N
      2                1           1.00           1                  N
      3                1           3.70           1                  N
      4                1           4.37           1                  N

         PULocationID  DOLocationID  …  extra  mta_tax  tip_amount  tolls_amount  \
      0           100           231  …    0.0      0.5        2.76           0.0
      1           186            43  …    0.0      0.5        4.00           0.0
      2           262           236  …    0.0      0.5        1.45           0.0
      3           188            97  …    0.0      0.5        6.39           0.0
      4             4           112  …    0.5      0.5        0.00           0.0

         improvement_surcharge  total_amount  trip_duration  Month       Day  \
      0                    0.3         16.56       0.234444      3  Saturday
      1                    0.3         20.80       0.441667      4   Tuesday
      2                    0.3          8.75       0.120000     12    Friday
      3                    0.3         27.69       0.504167      5    Sunday
      4                    0.3         17.80       0.278611      4  Saturday

         Trip duration unit
      0              hours
      1              hours
      2              hours
      3              hours
      4              hours

      [5 rows x 22 columns]
```

```
[14]: outlier_tip_amt = df[df['tip_amount']>150]
      outlier_tip_amt
```

```
[14]:       Unnamed: 0  VendorID   tpep_pickup_datetime  tpep_dropoff_datetime  \
      8476    11157412         1  02/06/2017 5:50:10 AM  02/06/2017 5:51:08 AM

            passenger_count  trip_distance  RatecodeID store_and_fwd_flag  \
      8476                1            2.6           5                  N

            PULocationID  DOLocationID  payment_type  fare_amount  extra  mta_tax  \
      8476          226           226              1       999.99    0.0      0.0

            tip_amount  tolls_amount  improvement_surcharge  total_amount
      8476       200.0           0.0                    0.3       1200.29
```

[15]:
```
outlier_total_amount = df[df['total_amount']<0]
outlier_total_amount
```

[15]:
```
       Unnamed: 0  VendorID    tpep_pickup_datetime   tpep_dropoff_datetime  \
314     105454287         2   12/13/2017 2:02:39 AM   12/13/2017 2:03:08 AM
1646     57337183         2  07/05/2017 11:02:23 AM  07/05/2017 11:03:00 AM
4423     97329905         2   11/16/2017 8:13:30 PM   11/16/2017 8:14:50 PM
5448     28459983         2  04/06/2017 12:50:26 PM  04/06/2017 12:52:39 PM
5758       833948         2   01/03/2017 8:15:23 PM   01/03/2017 8:15:39 PM
8204     91187947         2   10/28/2017 8:39:36 PM   10/28/2017 8:41:59 PM
10281    55302347         2   06/05/2017 5:34:25 PM   06/05/2017 5:36:29 PM
11204    58395501         2   07/09/2017 7:20:59 AM   07/09/2017 7:23:50 AM
12944    29059760         2  04/08/2017 12:00:16 AM  04/08/2017 11:15:57 PM
14714   109276092         2  12/24/2017 10:37:58 PM  12/24/2017 10:41:08 PM
17602    24690146         2   03/24/2017 7:31:13 PM   03/24/2017 7:34:49 PM
18565    43859760         2   05/22/2017 3:51:20 PM   05/22/2017 3:52:22 PM
20317    75926915         2  09/09/2017 10:59:51 PM  09/09/2017 11:02:06 PM
20698    14668209         2  02/24/2017 12:38:17 AM  02/24/2017 12:42:05 AM

       passenger_count  trip_distance  RatecodeID store_and_fwd_flag  \
314                  6           0.12           1                  N
1646                 1           0.04           1                  N
4423                 2           0.06           1                  N
5448                 1           0.25           1                  N
5758                 1           0.02           1                  N
8204                 1           0.41           1                  N
10281                2           0.00           1                  N
11204                1           0.64           1                  N
12944                1           0.17           5                  N
14714                5           0.40           1                  N
17602                1           0.46           1                  N
18565                1           0.10           1                  N
20317                1           0.24           1                  N
20698                1           0.70           1                  N

       PULocationID  DOLocationID  payment_type  fare_amount  extra  mta_tax  \
314             161           161             3         -2.5   -0.5     -0.5
1646             79            79             3         -2.5    0.0     -0.5
4423            237           237             4         -3.0   -0.5     -0.5
5448             90            68             3         -3.5    0.0     -0.5
5758            170           170             3         -2.5   -0.5     -0.5
8204            236           237             3         -3.5   -0.5     -0.5
```

```
10281           238           238             4        -2.5   -1.0      -0.5
11204            50            48             3        -4.5    0.0      -0.5
12944           138           138             4      -120.0    0.0       0.0
14714           164           161             4        -4.0   -0.5      -0.5
17602            87            45             4        -4.0   -1.0      -0.5
18565           230           163             3        -3.0    0.0      -0.5
20317           116           116             4        -3.5   -0.5      -0.5
20698            65            25             4        -4.5   -0.5      -0.5

       tip_amount  tolls_amount  improvement_surcharge  total_amount
314           0.0           0.0                   -0.3          -3.8
1646          0.0           0.0                   -0.3          -3.3
4423          0.0           0.0                   -0.3          -4.3
5448          0.0           0.0                   -0.3          -4.3
5758          0.0           0.0                   -0.3          -3.8
8204          0.0           0.0                   -0.3          -4.8
10281         0.0           0.0                   -0.3          -4.3
11204         0.0           0.0                   -0.3          -5.3
12944         0.0           0.0                   -0.3        -120.3
14714         0.0           0.0                   -0.3          -5.3
17602         0.0           0.0                   -0.3          -5.8
18565         0.0           0.0                   -0.3          -3.8
20317         0.0           0.0                   -0.3          -4.8
20698         0.0           0.0                   -0.3          -5.8
```

3.5.2 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 … Get insights about the data, question and investigate the outliers, clean and modify data whenever required.

Visualizations helped me understand .. some 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.