Activity_Course 2 Automatidata project
lab_2023_05_16_02_18_18

May 16, 2023

0.1 Automatidata project

Course 2 - Get Started with Python

You have just started as a data professional in a fictional data analytics firm, Automatidata. Their client, the New York City Taxi and Limousine Commission (New York City TLC), has hired the Automatidata team for its reputation in helping their clients develop data-based solutions.

The team is still in the early stages of the project. Previously, you were asked to complete a project proposal by your supervisor, DeShawn Washington. You have received notice that your project proposal has been approved and that New York City TLC has given the Automatidata team access to their data. To get clear insights, New York TLC's data must be analyzed, key variables identified, and the dataset ensured it is ready for analysis.

A notebook was structured and prepared to help you in this project. Please complete the questions inside and prepare a summary for the data team.

1 Course 2 End-of-course project: Inspect and analyze data

In this activity, you will examine data provided and prepare it for analysis.

The purpose of this project is to investigate and understand the data provided.

The goal is to use a dataframe contructed within Python, perform a cursory inspection of the provided dataset and inform team members of your findings.
This activity has three parts:

Part 1: Understand the situation * How can you best prepare to understand and organize the provided taxi cab information?

Part 2: Understand the data

- Create a pandas dataframe for data learning, and future exploratory data analysis (EDA) and statistical activities.
- View and interpret the datasets data table
- Sort and interpret the data table for two variables of your choice.

Part 3: Understand the variables

- What is the min, mean, and max of your chosen variables?
- Visualize the variables

Follow the instructions and answer the following questions 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 Identify data types and relevant variables using Python

2.0.1 Exercise Instructions:

Complete the following step-by-step instructions to inspect and analyze this NYC taxi dataset.

This activity will help ensure the information is,

1. Ready to answer questions and yield insights
2. Ready for visualizations
3. Ready for future hypothesis testing and statistical methods

Follow the instructions and answer questions to complete this activity. Afterward,

1. Write a short Executive Summary using your findings.
2. Use the structured notebook provided to help you in this project. Please complete the questions inside and prepare a summary for the data team.
3. Consider the questions presented in the Course 2 PACE strategy document.
4. Compare your data insights with the provided exemplar to confirm of your approach and results.

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.

2.1 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)`

2.2 PACE: Plan

2.2.1 Task 1. Understand the situation

- How can you best prepare to understand and organize the provided taxi cab information?

Begin by exploring your dataset and consider reviewing the Data Dictionary.

2.3 PACE: Analyze

2.3.1 Task 2a. Build dataframe

Create a pandas dataframe for data learning, exploratory data analysis (EDA), and statistical activities.

Code the following,

- import pandas as pd #library exercise for buidling dataframes
- import numpy as np #numpy is imported with pandas
- import matplotlib.pyplot as plt #visualization library
- import seaborn as sns #visualization library
- df = pd.read_csv('2017_Yellow_Taxi_Trip_Data.csv')

Note: pair the data object name "df" with pandas functions to manipulate data, such as df.groupby().

[1]:
```
import pandas as pd #library exercise for buidling dataframes

import numpy as np #numpy is imported with pandas

import matplotlib.pyplot as plt #visualization library

import seaborn as sns #visualization library

df = pd.read_csv('2017_Yellow_Taxi_Trip_Data.csv')
```

2.3.2 Task 2b. Understand the data - Interpret the data table

View and interpret the datasets data table by **coding the following,**

1. df.head(10)
2. df.info()

Consider the following two questions:

Question 1: When reviewing the df.head() output, are there any data points that surprise you or are not correct?

Question 2: When reviewing the df.info output, What kind of data types are we working with?

[2]:
```
df.head(10)
```

[2]:

	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
5	23345809	2	03/25/2017 8:34:11 PM	03/25/2017 8:42:11 PM
6	37660487	2	05/03/2017 7:04:09 PM	05/03/2017 8:03:47 PM
7	69059411	2	08/15/2017 5:41:06 PM	08/15/2017 6:03:05 PM
8	8433159	2	02/04/2017 4:17:07 PM	02/04/2017 4:29:14 PM
9	95294817	1	11/10/2017 3:20:29 PM	11/10/2017 3:40:55 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
5	6	2.30	1	N
6	1	12.83	1	N
7	1	2.98	1	N
8	1	1.20	1	N
9	1	1.60	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
5	161	236	1	9.0	0.5	0.5
6	79	241	1	47.5	1.0	0.5
7	237	114	1	16.0	1.0	0.5
8	234	249	2	9.0	0.0	0.5
9	239	237	1	13.0	0.0	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
5	2.06	0.0	0.3	12.36

```
6        9.86           0.0                    0.3         59.16
7        1.78           0.0                    0.3         19.58
8        0.00           0.0                    0.3          9.80
9        2.75           0.0                    0.3         16.55
```

[3]:
```
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
```

2.3.3 Task 2c. Understand the data - Sort by variables

Sort and interpret the data table for two variables of your choice.

Answer the following three questions:

Question 1: Sort your first variable (`trip_distance`) from maximum to minimum value, do the values seem normal?

Question 2: Sort your by your second variable (`total_amount`), are any values unusual?

Question 3: Are the resulting rows similar for both sorts? Why or why not?

[6]:
```
df_sort = df.sort_values(by=['trip_distance'],ascending=False)
df_sort
```

[6]:

	Unnamed: 0	VendorID	tpep_pickup_datetime	tpep_dropoff_datetime
9280	51810714	2	06/18/2017 11:33:25 PM	06/19/2017 12:12:38 AM
13861	40523668	2	05/19/2017 8:20:21 AM	05/19/2017 9:20:30 AM
6064	49894023	2	06/13/2017 12:30:22 PM	06/13/2017 1:37:51 PM
10291	76319330	2	09/11/2017 11:41:04 AM	09/11/2017 12:18:58 PM
29	94052446	2	11/06/2017 8:30:50 PM	11/07/2017 12:00:00 AM
…	…	…	…	…
2440	63574825	1	07/26/2017 10:26:58 PM	07/26/2017 10:26:58 PM
15916	47368116	1	06/29/2017 7:30:30 PM	06/29/2017 7:43:29 PM
1350	91619825	2	10/30/2017 8:20:29 AM	10/30/2017 8:20:38 AM
246	78660848	1	09/18/2017 8:50:53 PM	09/18/2017 8:51:03 PM
17788	58079289	1	07/08/2017 12:54:02 AM	07/08/2017 12:55:03 AM

	passenger_count	trip_distance	RatecodeID	store_and_fwd_flag
9280	2	33.96	5	N
13861	1	33.92	5	N
6064	1	32.72	3	N
10291	1	31.95	4	N
29	1	30.83	1	N
…	…	…	…	…
2440	1	0.00	1	N
15916	1	0.00	1	N
1350	1	0.00	1	N
246	1	0.00	1	N
17788	2	0.00	1	N

	PULocationID	DOLocationID	payment_type	fare_amount	extra	mta_tax
9280	132	265	2	150.00	0.0	0.0
13861	229	265	1	200.01	0.0	0.5
6064	138	1	1	107.00	0.0	0.0
10291	138	265	2	131.00	0.0	0.5
29	132	23	1	80.00	0.5	0.5
…	…	…	…	…	…	…
2440	162	264	2	5.50	0.5	0.5
15916	79	148	3	8.50	1.0	0.5
1350	193	193	1	2.50	0.0	0.5
246	145	145	2	2.50	0.5	0.5
17788	158	158	3	2.50	0.5	0.5

	tip_amount	tolls_amount	improvement_surcharge	total_amount
9280	0.00	0.00	0.3	150.30
13861	51.64	5.76	0.3	258.21
6064	55.50	16.26	0.3	179.06
10291	0.00	0.00	0.3	131.80
29	18.56	11.52	0.3	111.38
…	…	…	…	…
2440	0.00	0.00	0.3	6.80

```
15916         0.00          0.00                   0.3         10.30
1350          0.66          0.00                   0.3          3.96
246           0.00          0.00                   0.3          3.80
17788         0.00          0.00                   0.3          3.80

[22699 rows x 18 columns]
```

[7]:
```
df_sort = df.sort_values(by=['total_amount'], ascending=False)
df_sort
```

[7]:
```
       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
20312   107558404         2   12/19/2017 9:40:46 AM   12/19/2017 9:40:55 AM
13861    40523668         2   05/19/2017 8:20:21 AM   05/19/2017 9:20:30 AM
12511   107108848         2   12/17/2017 6:24:24 PM   12/17/2017 6:24:42 PM
15474    55538852         2   06/06/2017 8:55:01 PM   06/06/2017 8:55:06 PM
…            …       …                      …                      …
11204    58395501         2   07/09/2017 7:20:59 AM   07/09/2017 7:23:50 AM
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
20698    14668209         2  02/24/2017 12:38:17 AM  02/24/2017 12:42:05 AM
12944    29059760         2  04/08/2017 12:00:16 AM  04/08/2017 11:15:57 PM

       passenger_count  trip_distance  RatecodeID store_and_fwd_flag  \
8476                 1           2.60           5                  N
20312                2           0.00           5                  N
13861                1          33.92           5                  N
12511                1           0.00           5                  N
15474                1           0.00           5                  N
…                  …              …           …                  …
11204                1           0.64           1                  N
14714                5           0.40           1                  N
17602                1           0.46           1                  N
20698                1           0.70           1                  N
12944                1           0.17           5                  N

       PULocationID  DOLocationID  payment_type  fare_amount  extra  mta_tax  \
8476            226           226             1       999.99    0.0      0.0
20312           265           265             2       450.00    0.0      0.0
13861           229           265             1       200.01    0.0      0.5
12511           265           265             1       175.00    0.0      0.0
15474           265           265             1       200.00    0.0      0.5
…               …             …               …           …    …        …
11204            50            48             3        -4.50    0.0     -0.5
14714           164           161             4        -4.00   -0.5     -0.5
17602            87            45             4        -4.00   -1.0     -0.5
20698            65            25             4        -4.50   -0.5     -0.5
```

```
12944           138           138              4      -120.00    0.0       0.0

       tip_amount  tolls_amount  improvement_surcharge  total_amount
8476       200.00          0.00                    0.3       1200.29
20312        0.00          0.00                    0.3        450.30
13861       51.64          5.76                    0.3        258.21
12511       46.69         11.75                    0.3        233.74
15474       11.00          0.00                    0.3        211.80
…             …             …                      …           …
11204        0.00          0.00                   -0.3         -5.30
14714        0.00          0.00                   -0.3         -5.30
17602        0.00          0.00                   -0.3         -5.80
20698        0.00          0.00                   -0.3         -5.80
12944        0.00          0.00                   -0.3       -120.30

[22699 rows x 18 columns]
```

2.4 PACE: Construct

2.4.1 Task 3a. Understand the data - View Descriptive statistics

Instructions: Using pandas df.describe(), what is the min, mean, and max of your chosen variables?

Code and perform the following,

- df.describe()
- **Question 1:** What is the min, mean, and max of your first variable?
- **Question 2:** What is the min, mean, and max of second variable?
- **Question 3:** Are the values easily readable? Would could be done to make them more easily readable?

[8]:
```
#==> ENTER YOUR CODE OR RESPONSE HERE
df.describe()
#Question 1: what is the min, mean and max of your first variable ?
# Distance: mean, 2.91, min, 0, 33.96.

#Question 2: what is the min, mean, and max of your second variable?
# Cost: mean, 16.31; min, -120.3; max, 1200.29.
```

[8]:
```
         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
```

2.4.2 Task 3b. Visualize your variables

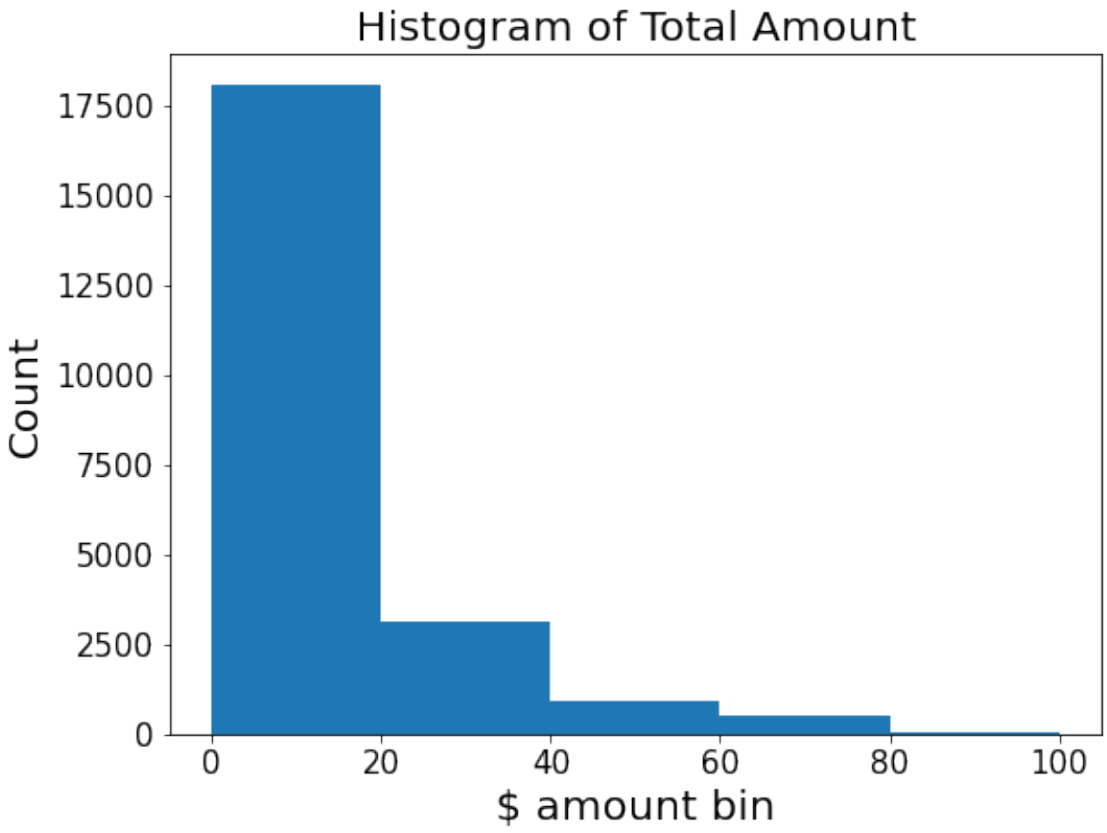
Instructions: Create a histogram for each of the two variables. Act and reflect on the following steps:

1. Histogram of your first variable (`total_amount`)
2. Histogram of your second variable (`trip_distance`)
3. Are your variables numerical (did the code work)?

[13]:
```
#==> ENTER YOUR CODE OR RESPONSE TO VISUALIZE THE FIRST VARIABLE HERE
plt.figure(figsize=(8,6))
plt.xticks(fontsize=14); plt.yticks(fontsize=14)
```

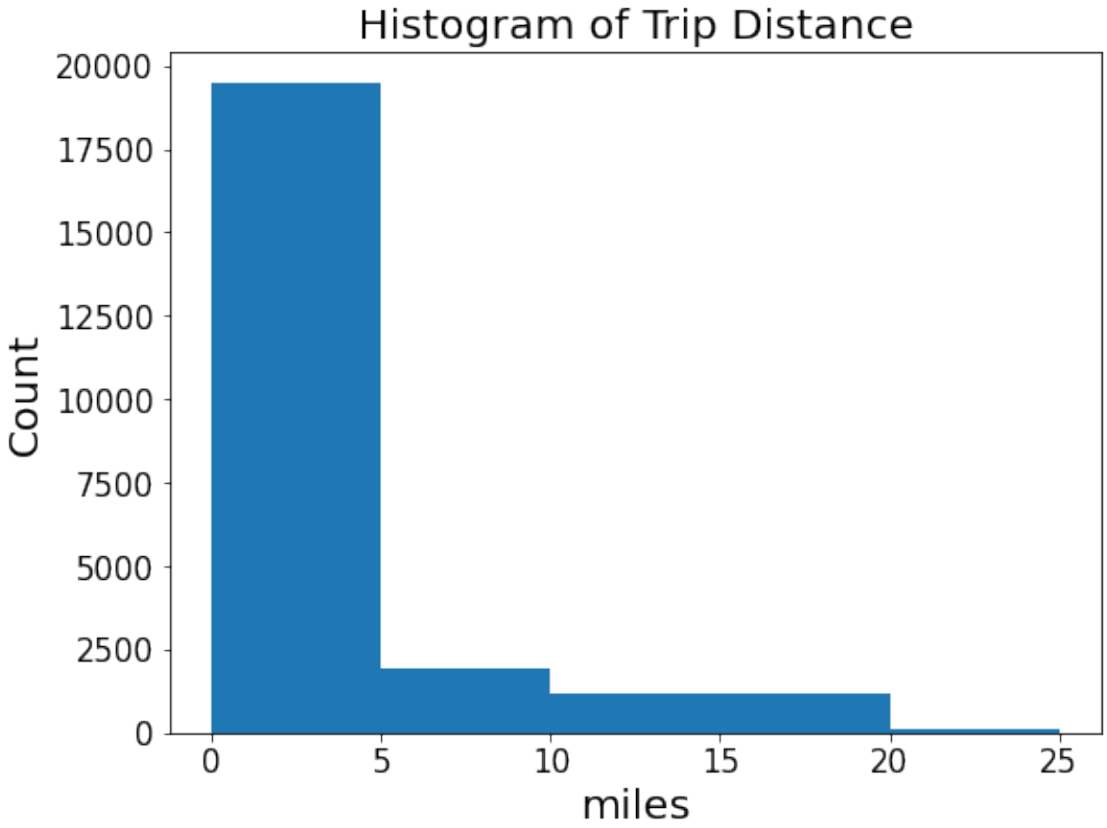
```
df = df.sort_values(by='total_amount')
plt.hist(df['total_amount'],bins=[0,20,40,60,80,100])
plt.title('Histogram of Total Amount',fontsize=20)
plt.xticks(fontsize=15); plt.yticks(fontsize=15)
plt.xlabel('$ amount bin',fontsize=20)
plt.ylabel('Count', fontsize=20)
```

[13]: Text(0, 0.5, 'Count')



[16]:
```
plt.figure(figsize=(8,6))
plt.xticks(fontsize=14); plt.yticks(fontsize=14)
df = df.sort_values(by='trip_distance')
plt.hist(df['trip_distance'],bins=[0,5,10,20,25])
plt.title('Histogram of Trip Distance',fontsize=20)
plt.xticks(fontsize=15); plt.yticks(fontsize=15)
plt.xlabel('miles',fontsize=20)
plt.ylabel('Count', fontsize=20)
```

[16]: Text(0, 0.5, 'Count')


Histogram of Trip Distance
Count
20000
17500
15000
12500
10000
7500
5000
2500
0
0
5
10
15
20
25
miles