

NYC TLC Project Part 1

November 5, 2024

NYC TLC Project

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.

To get clear insights, New York TLC's data must be analyzed, key variables identified, and the dataset ensured it is ready for analysis.

This project 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

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

The goal is to use a dataframe constructed 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 * Prepare to understand and organize the provided taxi cab dataset and information.

Part 2: Understand the data

- Create a pandas dataframe for data learning, future exploratory data analysis (EDA), and statistical activities.
- Compile summary information about the data to inform next steps.

Part 3: Understand the variables

- Use insights from your examination of the summary data to guide deeper investigation into specific variables.

1 Identify data types and relevant variables using Python

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

```
import pandas as pd
# Load dataset into dataframe
```

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

done

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

```
[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
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

```
[4]: 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
```

```
[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

****Task 2c. Understand the data - Investigate the variables**

Sort and interpret the data table for two variables: `trip_distance` and `total_amount`.

Answer the following three questions:

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

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

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

Question 1: The values align with our earlier data discovery, where we noticed that the longest rides are approximately 33 miles.

Question 2: Yes, the first two values are significantly higher than the others.

Question 3: The most expensive rides are not necessarily the longest ones.

```
[8]: df.sort_values(by="trip_distance", ascending=False)
```

```
[8]:      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]

```
[9]: df.sort_values(by="total_amount", ascending=False).head(20)
```

```
[9]:      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
6064       49894023         2  06/13/2017 12:30:22 PM  06/13/2017 1:37:51 PM
16379     101198443         2  11/30/2017 10:41:11 AM  11/30/2017 11:31:45 AM
3582      111653084         1  01/01/2017 11:53:01 PM  01/01/2017 11:53:42 PM
11269      51920669         1  06/19/2017 12:51:17 AM  06/19/2017 12:52:12 AM
9280       51810714         2  06/18/2017 11:33:25 PM  06/19/2017 12:12:38 AM
1928       51087145         1  06/16/2017 6:30:08 PM  06/16/2017 7:18:50 PM
10291      76319330         2  09/11/2017 11:41:04 AM  09/11/2017 12:18:58 PM
6708       91660295         2  10/30/2017 11:23:46 AM  10/30/2017 11:23:49 AM
11608     107690629         2  12/19/2017 5:00:56 PM  12/19/2017 6:41:56 PM
908        25075013         2  03/27/2017 1:01:38 PM  03/27/2017 1:38:44 PM
7281      111091850         2  01/01/2017 3:02:53 AM  01/01/2017 3:03:02 AM
18130      90375786         1  10/26/2017 2:45:01 PM  10/26/2017 4:12:49 PM
13621      93330154         1  11/04/2017 1:32:14 PM  11/04/2017 2:18:50 PM
13359       3055315         1  01/12/2017 7:19:36 AM  01/12/2017 7:19:56 AM
29        94052446         2  11/06/2017 8:30:50 PM  11/07/2017 12:00:00 AM
```

	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	
6064	1	32.72	3	N	
16379	1	25.50	5	N	
3582	1	7.30	5	N	
11269	2	0.00	5	N	
9280	2	33.96	5	N	
1928	2	12.50	5	N	
10291	1	31.95	4	N	
6708	1	0.32	5	N	

11608	2	23.00	3	N
908	2	26.12	4	N
7281	1	0.00	5	N
18130	1	30.50	1	N
13621	2	19.80	5	N
13359	1	0.00	5	N
29	1	30.83	1	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	
6064	138	1	1	107.00	0.0	0.0	
16379	132	265	2	140.00	0.0	0.5	
3582	1	1	1	152.00	0.0	0.0	
11269	265	265	1	120.00	0.0	0.0	
9280	132	265	2	150.00	0.0	0.0	
1928	211	265	1	120.00	0.0	0.0	
10291	138	265	2	131.00	0.0	0.5	
6708	264	83	1	100.00	0.0	0.5	
11608	151	1	1	99.50	1.0	0.0	
908	138	265	1	100.00	0.0	0.5	
7281	265	265	1	100.00	0.0	0.5	
18130	132	220	1	90.50	0.0	0.5	
13621	265	230	1	105.00	0.0	0.0	
13359	1	1	1	75.00	0.0	0.0	
29	132	23	1	80.00	0.5	0.5	

	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
6064	55.50	16.26	0.3	179.06
16379	0.00	16.26	0.3	157.06
3582	0.00	0.00	0.3	152.30
11269	20.00	11.52	0.3	151.82
9280	0.00	0.00	0.3	150.30
1928	5.00	12.50	0.3	137.80
10291	0.00	0.00	0.3	131.80
6708	25.20	0.00	0.3	126.00
11608	10.00	12.50	0.3	123.30
908	15.00	5.76	0.3	121.56
7281	20.16	0.00	0.3	120.96

18130	19.85	8.16	0.3	119.31
13621	8.00	2.64	0.3	115.94
13359	18.65	18.00	0.3	111.95
29	18.56	11.52	0.3	111.38

```
[10]: df.sort_values(by="total_amount", ascending=False).tail(20)
```

```
[10]:
```

	Unnamed: 0	VendorID	tpep_pickup_datetime	tpep_dropoff_datetime	\
14283	37675840	1	05/03/2017 7:44:28 PM	05/03/2017 7:44:38 PM	
19067	58713019	1	07/10/2017 2:40:09 PM	07/10/2017 2:40:59 PM	
10506	26005024	2	03/30/2017 3:14:26 AM	03/30/2017 3:14:28 AM	
5722	49670364	2	06/12/2017 12:08:55 PM	06/12/2017 12:08:57 PM	
4402	108016954	2	12/20/2017 4:06:53 PM	12/20/2017 4:47:50 PM	
22566	19022898	2	03/07/2017 2:24:47 AM	03/07/2017 2:24:50 AM	
1646	57337183	2	07/05/2017 11:02:23 AM	07/05/2017 11:03:00 AM	
18565	43859760	2	05/22/2017 3:51:20 PM	05/22/2017 3:52:22 PM	
314	105454287	2	12/13/2017 2:02:39 AM	12/13/2017 2:03:08 AM	
5758	833948	2	01/03/2017 8:15:23 PM	01/03/2017 8:15:39 PM	
5448	28459983	2	04/06/2017 12:50:26 PM	04/06/2017 12:52:39 PM	
4423	97329905	2	11/16/2017 8:13:30 PM	11/16/2017 8:14:50 PM	
10281	55302347	2	06/05/2017 5:34:25 PM	06/05/2017 5:36:29 PM	
8204	91187947	2	10/28/2017 8:39:36 PM	10/28/2017 8:41:59 PM	
20317	75926915	2	09/09/2017 10:59:51 PM	09/09/2017 11:02:06 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	\
14283	1	0.00	5	N	
19067	1	0.10	5	N	
10506	1	0.00	1	N	
5722	1	0.00	1	N	
4402	1	7.06	1	N	
22566	1	0.00	1	N	
1646	1	0.04	1	N	
18565	1	0.10	1	N	
314	6	0.12	1	N	
5758	1	0.02	1	N	
5448	1	0.25	1	N	
4423	2	0.06	1	N	
10281	2	0.00	1	N	
8204	1	0.41	1	N	
20317	1	0.24	1	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	\
14283	146	146	3	0.01	0.0	0.0	
19067	261	13	3	0.00	0.0	0.0	
10506	264	193	1	0.00	0.0	0.0	
5722	264	193	1	0.00	0.0	0.0	
4402	263	169	2	0.00	0.0	0.0	
22566	264	193	1	0.00	0.0	0.0	
1646	79	79	3	-2.50	0.0	-0.5	
18565	230	163	3	-3.00	0.0	-0.5	
314	161	161	3	-2.50	-0.5	-0.5	
5758	170	170	3	-2.50	-0.5	-0.5	
5448	90	68	3	-3.50	0.0	-0.5	
4423	237	237	4	-3.00	-0.5	-0.5	
10281	238	238	4	-2.50	-1.0	-0.5	
8204	236	237	3	-3.50	-0.5	-0.5	
20317	116	116	4	-3.50	-0.5	-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
14283	0.0	0.0	0.3	0.31
19067	0.0	0.0	0.3	0.30
10506	0.0	0.0	0.0	0.00
5722	0.0	0.0	0.0	0.00
4402	0.0	0.0	0.0	0.00
22566	0.0	0.0	0.0	0.00
1646	0.0	0.0	-0.3	-3.30
18565	0.0	0.0	-0.3	-3.80
314	0.0	0.0	-0.3	-3.80
5758	0.0	0.0	-0.3	-3.80
5448	0.0	0.0	-0.3	-4.30
4423	0.0	0.0	-0.3	-4.30
10281	0.0	0.0	-0.3	-4.30
8204	0.0	0.0	-0.3	-4.80
20317	0.0	0.0	-0.3	-4.80
11204	0.0	0.0	-0.3	-5.30
14714	0.0	0.0	-0.3	-5.30
17602	0.0	0.0	-0.3	-5.80
20698	0.0	0.0	-0.3	-5.80
12944	0.0	0.0	-0.3	-120.30

```
[11]: # How many of each payment type are represented in the data?
df["payment_type"].value_counts()
```

```
[11]: 1    15265
      2     7267
      3      121
      4       46
      Name: payment_type, dtype: int64
```

According to the data dictionary, the payment method was encoded as follows:

1 = Credit card
2 = Cash
3 = No charge
4 = Dispute
5 = Unknown
6 = Voided trip

```
[17]: # What is the average tip for trips paid for with credit card?
mask = df["payment_type"]==1
print(df[mask]["tip_amount"].mean())

# What is the average tip for trips paid for with cash?
mask1 = df["payment_type"]==2
print(df[mask1]["tip_amount"].mean())
```

```
2.7298001965279934
0.0
```

```
[18]: # How many times is each vendor ID represented in the data?
df["VendorID"].value_counts()
```

```
[18]: 2    12626
      1    10073
      Name: VendorID, dtype: int64
```

```
[20]: # What is the mean total amount for each vendor?
mask3 = df["VendorID"] == 1
print(df[mask3]["total_amount"].mean())

mask4 = df["VendorID"] == 2
print(df[mask4]["total_amount"].mean())
```

```
16.298118733246966
16.32038175193886
```

```
[23]: # Filter the data for credit card payments only
credit_card_df = df[df["payment_type"]==1]
```

```
credit_card_df
```

```
# Filter the credit-card-only data for passenger count only  
credit_card_df["passenger_count"].value_counts()
```

```
[23]: 1    10977  
      2     2168  
      5      775  
      3      600  
      6      451  
      4      267  
      0       27  
      Name: passenger_count, dtype: int64
```

```
[24]: # Calculate the average tip amount for each passenger count (credit card  
      → payments only)  
      credit_card_df.groupby("passenger_count").mean()[["tip_amount"]]
```

```
[24]:
```

	tip_amount
passenger_count	
0	2.610370
1	2.714681
2	2.829949
3	2.726800
4	2.607753
5	2.762645
6	2.643326