Improving Yellow Taxi Revenue

Jiakai Ni

Background

As more and more people choose For-Hire Vehicle (“FHV”) such as Uber as their transport vehicle, the taxi market has been supressed significantly. It is urgent for taxi drivers to take corresponding actions to sustain their income in a more efficient way.

Aim

This report aims to help yellow taxi drivers (which can pick-up and drop-off anywhere) increase their revenue based on TLC Trip Record Data. Ignoring the cost (include gas and reasonable time taken for seeking passengers etc), we hope to maximize the driver's income per unit of time (i.e. revenue). Although FHV seems to take up a large proportion of the market, the two datasets of yellow taxi and FHV are too different to produce a comparative analysis. Hence the analysis will focus on a better passenger-seeking strategy for yellow taxis drivers.

Dataset Selection

After 2016, specific coordinates are replaced with LocationID. However, passengers from the same region do not necessarily bring the same revenue for drivers. Besides not all location is allowed for pick-up or drop-off. Hence, I chose the travel records of yellow taxis in January 2016 (Taxi & Limousine Commission, 2016) as exact locations can provide sufficient information about which pick-up location would potentially achieve higher revenue. This dataset includes ten million instances and 19 attributes, which should contain enough number of instances and provide more accurate information about drive record in short period.

The weather condition may also affect revenue, so weather data in the same period is collected (Weather Underground, 2016) and reconstruct into table format (see appendix A block 9, start from page 12) , (Since the weather is only recorded once at the 51 minutes of each hour, we assume the weather condition within that hour is the same. Note that the weather data at 1:51 on January 21 is missing. Based on the data before and after, it is speculated that the time period should be no precipitation, which is marked as ‘remain’.

Sampling

Due to the large amount of data, it might be hard to fit or plot all the data at once. Therefore, several sub-datasets for Trip record will be generated. Each one of them contain 1,000,000 instances for the original dataset which are chosen randomly. If sub dataset is used, results for each sub-dataset will be compared to ensure they produce similar conclusion.

Assumption

Since there are too many factors in reality that may affect the income of yellow taxi drivers, the following analysis is based on the following assumptions.

1. The number of passengers taking other types of transportation will not affect the total number of people taking yellow taxis.
2. The number of people who needs to take yellow taxi at same circumstance (including same time period, weather, etc) is similar.
3. Ignoring the cost (include gas and reasonable time taken for seeking passengers etc)
4. Seasonal factors such as seasons and day of the week have no effect on revenue except time of day (hours).
5. Based on the attribute provided on the TLC Trip Record Data, the revenue is defined as:
$$\frac{fare\ amout + tip\ amont}{drop\ off\ time - pick\ up\ time}$$
6. in unit USD/min.
7. There is no additional payment from the passenger.
8. Yellow taxi drivers work 8 hours continues per day.

Data pre-processing

Data wrangling

1. Add weather information into a new attribute according to pick up time. To simplify the problem, this attribute will only consider precipitation or not.
2. Since only time of day (hours) is considered, then date part information is not important. So keep time (hors) part and delete the rest part in this variables.
3. Since we are interested in revenue, so attribute income and revenue are added.
4. Value 99 appears in the attribute ‘RatecodeID’, but we don’t know what 99 refers to so delete rows have 99 in attribute ‘RatecodeID’. Besides, majority value of this ‘RatecodeID’ is 1 and

there is not enough sample for other categories. So reconstruct this attribute to as 1=standard rate and 2= other.

5. Since there isn't enough sample for payment type 3,4 and 5. Reconstruct it as 1=credit card, 2=other. For some numerical variables that do not represent numbers (i.e. payment type), set the attributes as type 'category'. It's important to notice that the gap between 23:00 to 0:00 is also an hour. Hence time attribute should also be 'category' type.

Data cleaning

Since all data except the number of passengers are automatically collected, there is no filling errors, which means that the data is reliable. However, some abnormal data may be resulted by system errors are noticed, which is why data cleaning is needed. Below is the method.

1. Delete the column with null (if there exist any).
2. Several reasonable boundaries for continuous attributes are provided based on real-life experience and distributions of dataset. See the interval and distribution of data in the table and graph below (detect whether start coordinates is the same as end coordinates is time consuming, so set lower bound of duration to 0.25 and tip distance to 0.01, which should have same effect).

Attribute	Lower bound	Upper bound	Attribute	Lower bound	Upper bound
Duration (min)	0.25	150	Extra (USD)	0	∞
Total amount (USD)	0	100	MTA tax (USD)	0	∞
Fare amount (USD)	0	80	Tip amount (USD)	0	∞
Trip distance (mile)	0.01	30	Improvement surcharge (USD)	0	∞
latitude	40.5	41	Revenue (USD/min)	0	5
longitude	-74.5	-73			

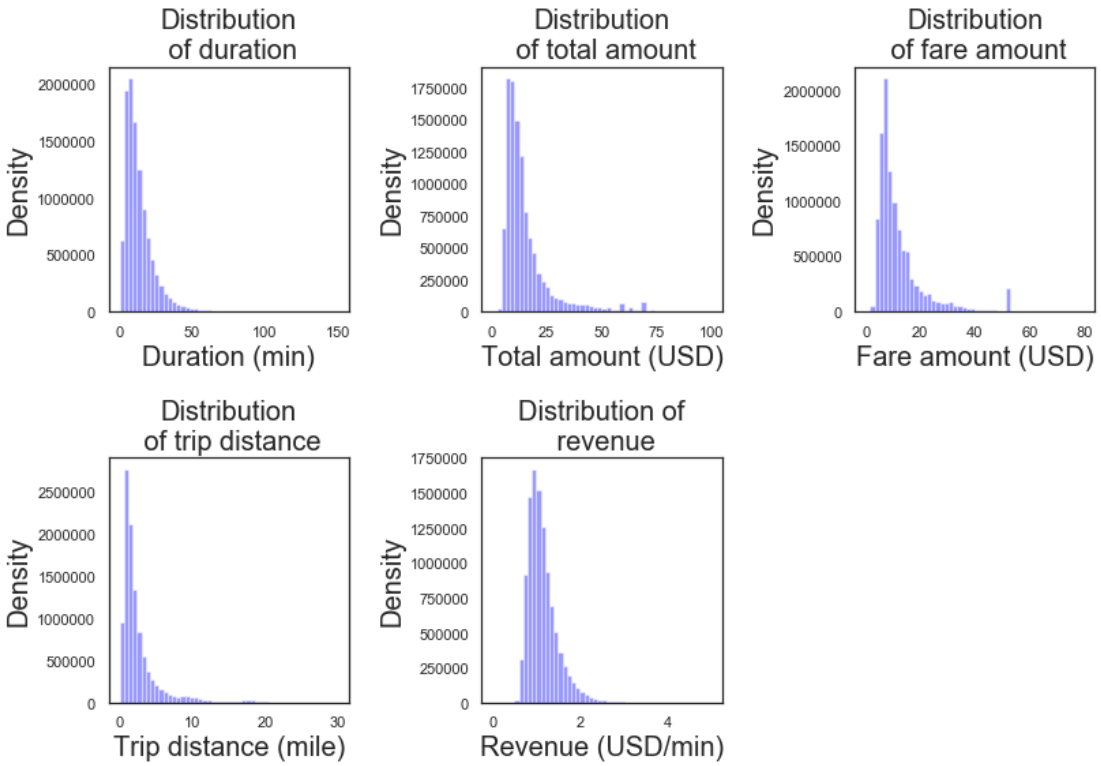


Figure 1 Distribution of some variables

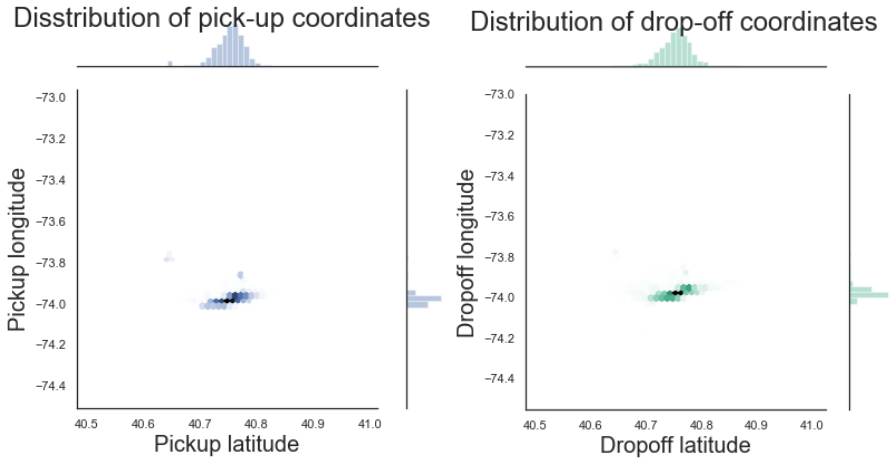


Figure 2 Distribution of coordinates

After data cleaning, we still have about ten million data for data analysis, which should beyond enough.

Data reduction

- In real life, 'VendorID' has no influence in passenger's taxi-taking decisions nor the amount of tip he offers to the driver, as well as the number of passengers. Hence, it can't provide useful

information, so this attribute is deleted.

- Majority of 'store_and_fwd_flag' is N and it seems to have no correlation with the task, so delete this attribute.
- Since we already have pick up time and duration, so drop-off time is not necessary. Hence this variable is deleted.
- Improvement surcharge is a constant tax included in the total amount. Since tax is not of our interest and it has no interaction with other variables, this variable is deleted.

Data processing

Correlation analysis

There are some attributes that are not suitable for linear comparison with reasons listed below:

- Coordinates: Coordinates correspond to the actual geographic location. Longitude or latitude alone cannot provide enough information.
- Pick-up datetime and drop-off datetime: they are reconstructed into pickup hour and duration, so these two attributes are no longer useful.
- Extra: it's a fixed value determined by time.
- The total amount and tolls amount: they are linear combinations of other attributes; we can find their relationship through other attributes.
- Other variables that are not continuous.

After attribute filtering, the Pearson correlation matrix is shown in figure below

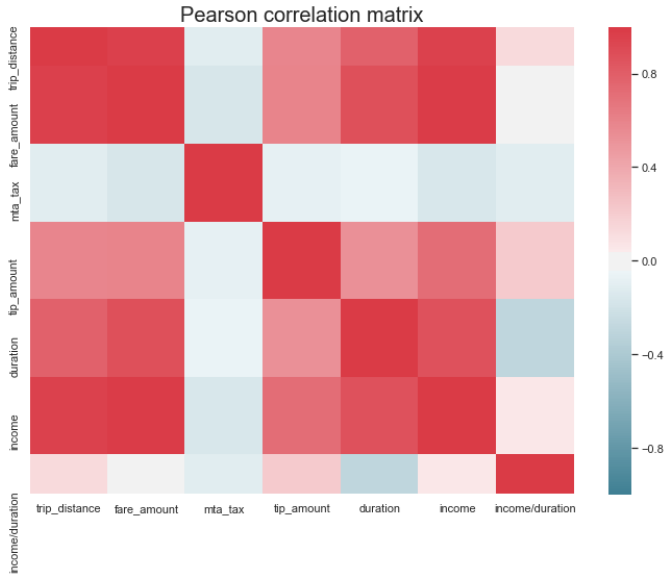


Figure 3 Pearson correlation matrix

Problem Division

As mentioned above, the value we want maximize is

$$y = \frac{x_1 + x_2}{t}$$

where x_1 is the fare amount, x_2 is the tip amount and t is the duration.

The correlation matrix indicates the linear relationship between income and other variables is stronger than the relationship with income over duration. Hence, the contribution of income would be discussed first, which consists of fare and tip. The report would discuss what might affect these two variables respectively first.

Potential factors affecting tip

Assumption:

- Tip might linearly increase as trip distance increase.
- It seems the passenger would not tip the driver using credit card if that is not the payment method they choose for fare. So, payment method should be factor affect tip.
- Tip might increase linearly as trip distance increase.
- Start time may determine the purpose of trip, which might affect the tip.
- Passenger might give more tip on precipitation condition because they are grateful able to take taxi in such condition.

$$y_{ij} = \mu + x\beta + \xi_{22} + \xi_{23} + \varepsilon_{ij}$$

Where y_{ij} is tip amount, $\beta = [\beta_1\ \beta_2\ \tau_1\ \tau_2 \tau_3]^T$, and they are coefficient for trip distance, duration, payment type, start hour and weather respectively. Remove parameter with small coefficient, finally only trip distance and payment type are left (see coefficient and new model in appendix C from block 30 to block 33, start from page 15). But figure 4 shows tip amount doesn't linearly depend on trip distance, figure 5 shows that only certain amount of tip is given regardless of the trip distance. In most cases, it appears 0 tips are given as payments 'other' than 'credit card' would not incur tips (see in figure 5). Hence it seems tip only depends on payment type.

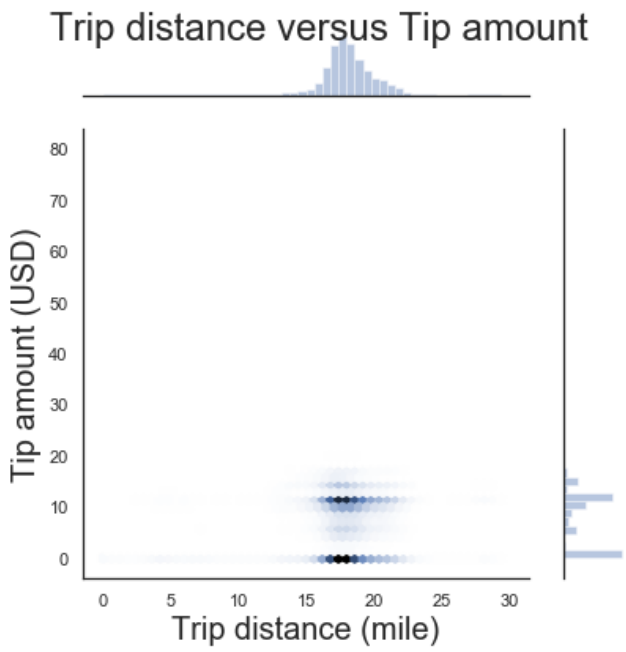


Figure 4 Trip distance versus Tip amount.

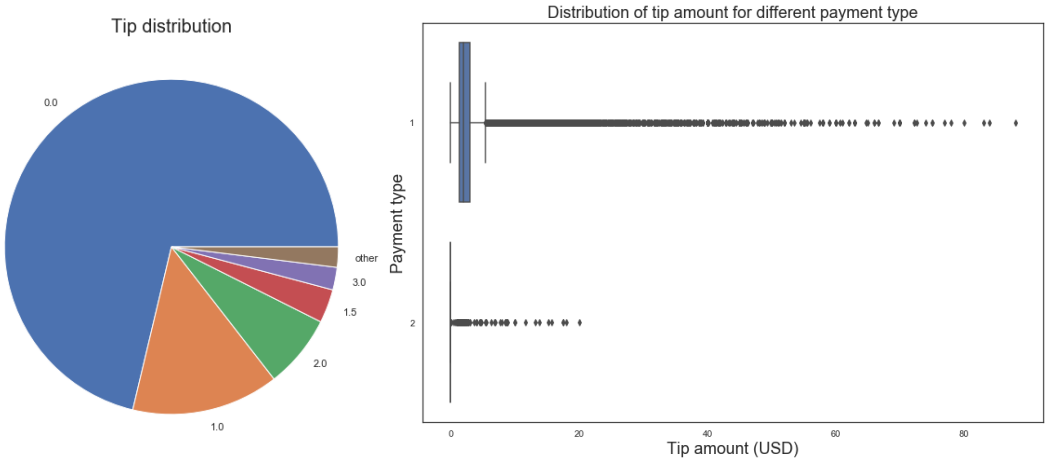


Figure 5 Distribution of tip and boxplot for different payment type where payment type 1 is 'standard rate' and 2 is 'other'.

Potential factors affecting fare amount

In the New York city, the taxi fee included $3.00 basic fee, $1.56 extra per kilometer (about $2.51 USD per mile) and standing/waiting fee which is $0.5 /min (Taxi Fare Calculation in New York City, 2020). Hence, we expect fare have linear relationship with trip distance and duration. In addition, start hour and weather may affect the duration, so they will also be added to the model. The model is:

$$y_{ij} = \mu + x\beta + \xi_{21} + \xi_{22} + \varepsilon_{ij}$$

Where y_i is fare amount, $\beta = [\beta_1\ \beta_2\ \tau_1\ \tau_2]^T$, and they are coefficient for trip distance, duration, start hour and weather respectively.

From the coefficient (see all coefficient for fare amount model in appendix C from block 39 and 40, start from page 28), it's seen that the correlation between duration and weather is weak. So simplify the model to:

$$y_{ij} = \mu + x\beta + \varepsilon_{ij}$$

Where y_i is fare amount, $\beta = [\beta_1\ \beta_2\ \tau_1]^T$, and they are coefficient for trip distance, duration, start hour respectively. All the rest variables appear to be important in the model. Although the variables we used are slightly different from real-life, duration and start hour still capture the effect of waiting time. Hence generally, the coefficient we got is close to what we expected. Furthermore, from the figure below we do observe some salient relationship found in the linear model.

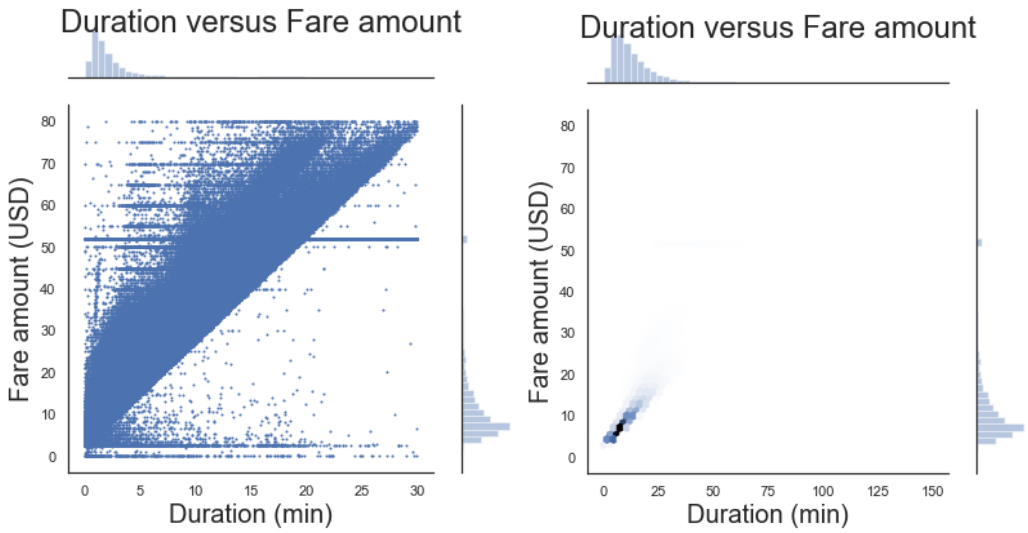


Figure 6 distribution of fare amount for different duration

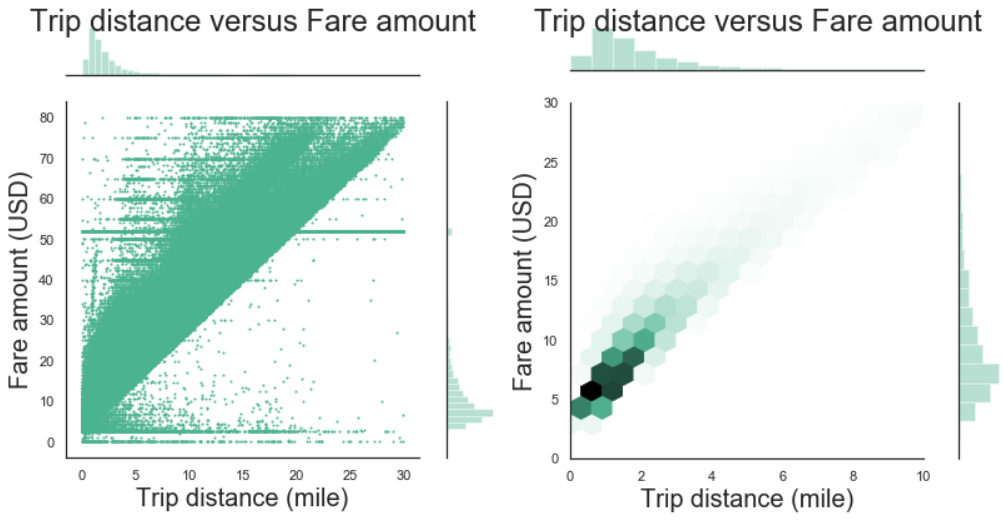


Figure 7 distribution of fare amount for different trip distance

Potential factors affecting trip distance

We still got variable distance in the predator for the fare amount, which is highly related to the duration. In this section, we are trying to find a model could be used to predict distance.

Assumption:
Duration will linearly increase as distance increase.
Start time may determine the purpose of trip, which might affect the trip distance.
Precipitation may disrupt the travel plan, which may affect the trip distance.
RatecodeID may decide the pick-up/drop-off location, which may affect the trip distance.
Payment type may affect the trip distance.

Based on the assumption, since all the relation is linear, linear model is appropriate to use. The linear model is:

$$y_{ij} = \mu + x\beta + \xi_{11} + \xi_{13} + \varepsilon_{ij}$$

Where y_i is trip distance, $\beta = [\beta_1\ \tau_1\ \tau_2 \tau_3\ \tau_4]^T$, and they are coefficient for duration, start hour, payment type, weather and RatecodeID respectively.

From the coefficient we got (see all coefficient for trip distance model in appendix C from block 45 to 53, start from page 37), it can be seen that, the impact of RatecodeID is extraordinarily, figure 8 shows same pattern. Figure 9 shows the revenue when the RatecodeID is 'other' is significantly higher than when RatecodeID is 'standard rate'. But only less than 1% trip didn't choose 'standard rate', which means we can't inform the drivers how to increase their revenue based on RatecodeID. Hence, future study is still necessary in order to observe relationship between revenue and other variables. Since we already know the affect brought by RatecodeId, the rest part of study would focus on other attributes (i.e. delete rows for RatecodeID is 'other').

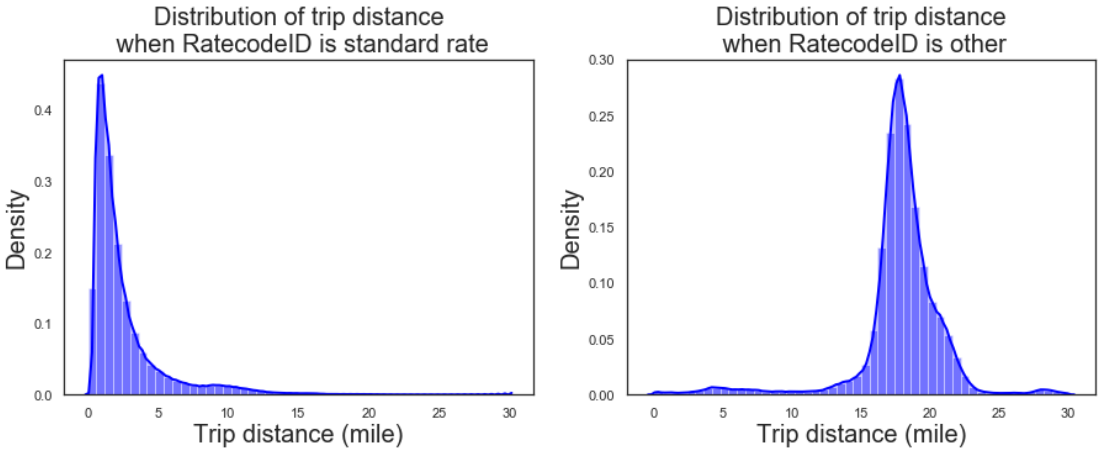


Figure 8 distribution of trip distance for different RatecodeID

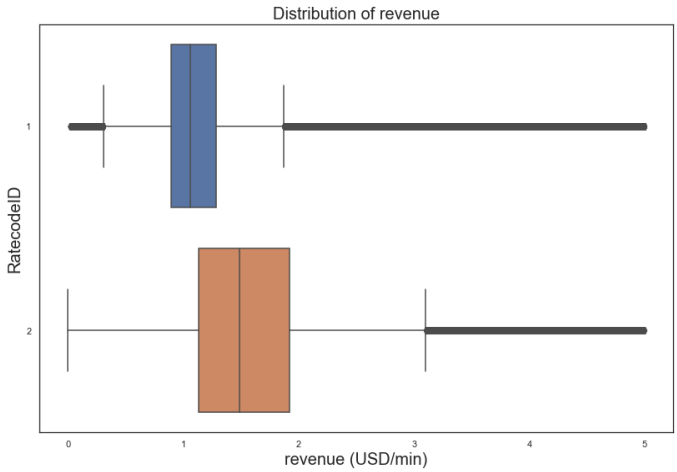


Figure 9 distribution of revenue (RatecodeID 1=standard rate, 2=other).

After delete RatecodeID from all datasets, and delete RatecodeID from the predictor, the model is fitted once again. The result indicate that weather and all the interaction is not significant.

From the figure below we got similar conclusion. So only duration, start hour payment type are kept.

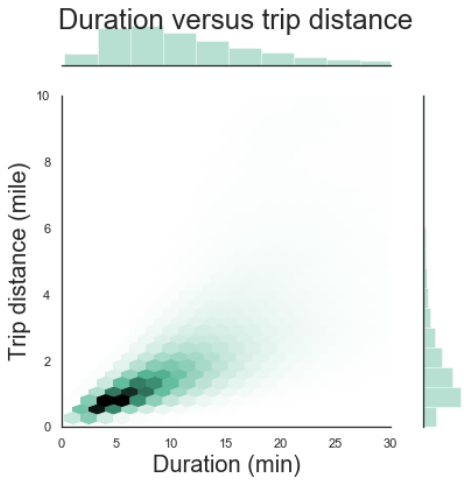


Figure 10 Duration versus trip distance

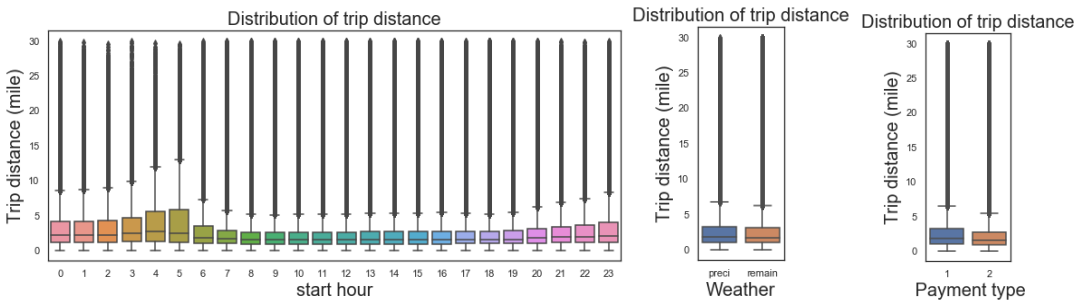


Figure 11 Distribution of tip distance across different attribute

Since we have model predict trip distance, we can use predicted value instead of true value to reduce

number of parameters in the predictor of fare amount.

Potential factors affecting income

Use predicted value of fare amount, tip amount, trip distance instead of actual value, we got:

$$\begin{aligned} y_{income} &= y_{fare} + y_{tip} \\ &= \widehat{y_{fare}} + \widehat{y_{tip}} + \varepsilon_{ij} \\ &= y_{\widehat{trip\ distance}} + x\beta' + \widehat{y_{tip}} + \varepsilon_{ij} \\ &= \mu + x\beta + \xi_{11} + \xi_{12} + \varepsilon_{ij} \end{aligned}$$

Where $\beta = [\beta_1\ \tau_1\ \tau_2 \tau_3]^T$, and they are coefficient for, duration, payment type and start hour respectively.

The interaction isn't significant (see all coefficient for trip distance model in appendix C from block 58 to 59, start from page 53), so the final model is:

$$y_{income} = \mu + x\beta + \varepsilon_{ij}$$

Where $\beta = [\beta_1\ \tau_1\ \tau_2]^T$, and they are coefficient for, duration, payment type and start hour respectively. The figure below shows all these variables significantly affect income, which is same as the conclusion from linear model.

What affecting income/duration

Since we have a model can be used to predicted income, we can get revenue easily.

$$\begin{aligned} y_{\text{revenue}} &= \frac{\mu + x\beta' + \varepsilon_{ij}}{duration} \\ &= 1 + \frac{\mu + x\beta + \varepsilon_{ij}}{duration} \end{aligned}$$

Where Where $\beta = [\tau_1\ \tau_2]^T$, and they are coefficient for payment type and start hour respectively. Since β won't change as duration increase, so we expect revenue decrease as duration increase. And from the figure below, we got exact pattern that we expected.

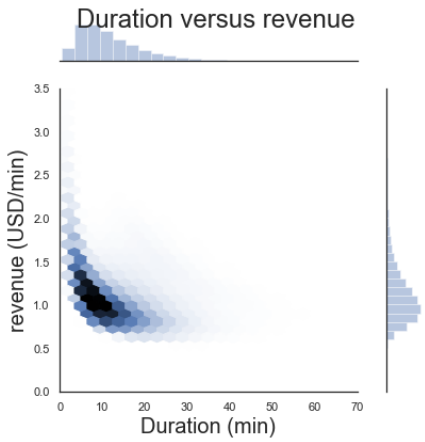


Figure 12 Duration versus revenue

Since the general assumption include the yellow taxi driver work continuously 8 hours per day from the coefficient we get for the revenue, it seems work from 23:00 to 7:00 could maximize the revenue. So the task becomes to find passenger that potentially travels short duration in these period. Since driver can't decide drop-off location, so only pick-up location would be analyzed. Figure 13 shows the distribution across 23:00 to 7:00 for duration equal or shorter than 5 mins and longer than 5 mins, but no significant difference caused by duration is observed (see figure).

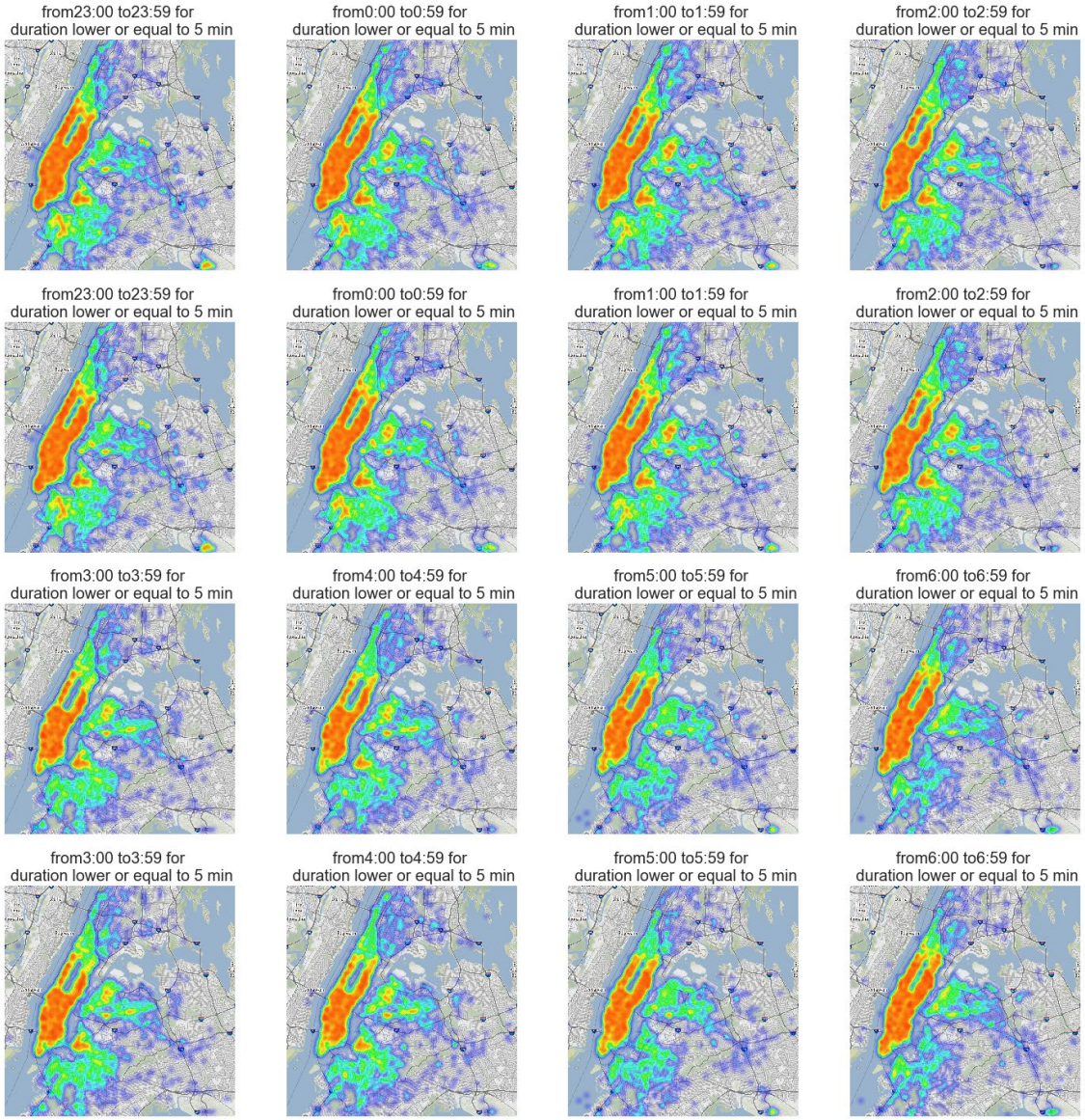


Figure 13 heatmap of pick up location across 23:00 to 7:00

Conclusion

Across all attributes, RatecodeID, pick up time, duration and payment tips are the three main variables affect revenue. Since most RatecodeID is ‘standard rate’, so drivers cannot rely on RatecodeID to increase their revenue even it could increase revenue significantly. As observed from pickup location heatmap, most passengers are picked up from Manhattan from 23:00-7:00 and different duration did not cause obvious difference. The only thing driver could do is choose working hours. So the advice for yellow taxi drivers would be working from 23:00 to 7:00 in Manhattan (although there might be less passengers need to take taxi, the goal of this report is to maximize revenue when the passenger is in the taxi. Hence the constraints were not discussed in the report).

Reference

Weather Underground. (2016). *New York City, NY Weather History* [Daily Observations]. Retrieved from https://www.wunderground.com/history/daily/us/ny/new-york-city/KLGA

Taxi & Limousine Commission. (2016). *TLC Trip Record Data* [2016 January Yellow Taxi Trip Record]. Retrieved from https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page

Data Dictionary - Yellow Taxi Trip Records. (2018). *Yellow Taxi Trip Records*. Retrieved from https://www1.nyc.gov/assets/tlc/downloads/pdf/data_dictionary_trip_records_yellow.pdf

Taxi Fare Calculation in New York City (2020). Retrieved from https://www.taxi-calculator.com/taxi-fare-new-york-city/259

Appendix A

Jupyter Notebook part 1 (the page number is only for this Notebook)

[1]:
```
import pandas as pd
import numpy as np
```

[2]:
```
weather = pd.read_csv('data/2016_01_weather.CSV')
weather
```

[2]:

	time	12:51 AM	1:51 AM	\
0	2016/1/1	Fair	Cloudy	
1	2016/1/2	Fair	Fair	
2	2016/1/3	Fair	Fair	
3	2016/1/4	Partly Cloudy / Windy	Fair	
4	2016/1/5	Fair	Fair / Windy	
5	2016/1/6	Fair	Fair	
6	2016/1/7	Fair	Fair	
7	2016/1/8	Cloudy	Fair	
8	2016/1/9	Cloudy	Cloudy	
9	2016/1/10	Mostly Cloudy	Light Rain	
10	2016/1/11	Fair	Mostly Cloudy	
11	2016/1/12	Partly Cloudy / Windy	Fair	
12	2016/1/13	Cloudy	Fair / Windy	
13	2016/1/14	Cloudy	Mostly Cloudy	
14	2016/1/15	Cloudy	Cloudy	
15	2016/1/16	Mostly Cloudy	Cloudy	
16	2016/1/17	Cloudy	Mostly Cloudy	
17	2016/1/18	Fair	Fair	
18	2016/1/19	Partly Cloudy / Windy	Fair	
19	2016/1/20	Cloudy	Fair / Windy	
20	2016/1/21	Fair / Windy	NaN	
21	2016/1/22	Light Snow	Fair	
22	2016/1/23	Light Snow	Light Snow	
23	2016/1/24	Cloudy	Light Snow / Windy	
24	2016/1/25	Cloudy	Partly Cloudy	
25	2016/1/26	Cloudy	Cloudy	
26	2016/1/27	Cloudy	Cloudy	
27	2016/1/28	Cloudy	Fair	
28	2016/1/29	Fair	Cloudy	
29	2016/1/30	Partly Cloudy	Partly Cloudy	

```
30  2016/1/31                 Cloudy              Cloudy

                 2:51 AM               3:51 AM               4:51 AM  \
0                 Cloudy                Cloudy                Cloudy
1                   Fair          Mostly Cloudy          Mostly Cloudy
2                   Fair                   Fair                   Fair
3           Mostly Cloudy  Mostly Cloudy / Windy          Mostly Cloudy
4                   Fair                   Fair                   Fair
5                   Fair                 Cloudy                   Fair
6                   Fair                   Fair                   Fair
7                   Fair          Partly Cloudy                   Fair
8           Mostly Cloudy          Mostly Cloudy                 Cloudy
9                 Cloudy                Cloudy                   Rain
10  Mostly Cloudy / Windy  Partly Cloudy / Windy  Partly Cloudy / Windy
11          Partly Cloudy          Partly Cloudy          Mostly Cloudy
12  Partly Cloudy / Windy           Fair / Windy          Partly Cloudy
13                Cloudy                Cloudy                 Cloudy
14          Partly Cloudy          Partly Cloudy          Partly Cloudy
15                  Rain             Light Rain             Light Rain
16                Cloudy                 Cloudy          Mostly Cloudy
17                  Fair                   Fair                   Fair
18           Fair / Windy                   Fair                   Fair
19                  Fair                   Fair                   Fair
20                Cloudy                 Cloudy                 Cloudy
21                  Fair                   Fair                   Fair
22     Light Snow / Windy     Light Snow / Windy     Light Snow / Windy
23     Light Snow / Windy         Cloudy / Windy                 Cloudy
24          Partly Cloudy                   Fair                   Fair
25          Mostly Cloudy                 Cloudy                 Cloudy
26          Mostly Cloudy                 Cloudy                 Cloudy
27          Partly Cloudy          Partly Cloudy                   Fair
28                Cloudy                 Cloudy                 Cloudy
29                  Fair          Partly Cloudy          Partly Cloudy
30          Partly Cloudy          Partly Cloudy                 Cloudy

          5:51 AM        6:51 AM        7:51 AM               8:51 AM  …  \
0          Cloudy         Cloudy  Mostly Cloudy                 Cloudy  …
1   Partly Cloudy           Fair  Partly Cloudy          Mostly Cloudy  …
2            Fair           Fair           Fair                   Fair  …
3   Mostly Cloudy  Partly Cloudy  Mostly Cloudy          Mostly Cloudy  …
4            Fair           Fair           Fair                   Fair  …
5          Cloudy         Cloudy         Cloudy                 Cloudy  …
6            Fair  Partly Cloudy  Partly Cloudy          Mostly Cloudy  …
7            Fair           Fair           Fair          Partly Cloudy  …
8          Cloudy         Cloudy         Cloudy          Mostly Cloudy  …
9      Heavy Rain           Rain           Rain             Heavy Rain  …
10           Fair           Fair           Fair                   Fair  …
```

11	Mostly Cloudy	Partly Cloudy	Mostly Cloudy	Mostly Cloudy	…
12	Fair / Windy	Fair / Windy	Fair / Windy	Fair	…
13	Cloudy	Light Snow	Cloudy	Cloudy	…
14	Partly Cloudy	Cloudy	Cloudy	Cloudy	…
15	Light Rain	Cloudy	Light Rain	Light Rain	…
16	Fair	Fair	Mostly Cloudy	Cloudy	…
17	Fair / Windy	Partly Cloudy	Light Snow	Mostly Cloudy / Windy	…
18	Fair	Fair / Windy	Fair	Fair / Windy	…
19	Fair	Partly Cloudy	Fair	Partly Cloudy	…
20	Cloudy	Partly Cloudy	Fair	Fair	…
21	Fair	Fair	Fair	Mostly Cloudy	…
22	Snow / Windy	Snow	Snow / Windy	Heavy Snow / Windy	…
23	Fair / Windy	Fair	Fair	Fair	…
24	Mostly Cloudy	Cloudy	Cloudy	Cloudy	…
25	Cloudy	Cloudy	Cloudy	Cloudy	…
26	Cloudy	Cloudy	Cloudy	Cloudy	…
27	Fair	Fair	Fair	Fair	…
28	Cloudy	Mostly Cloudy	Partly Cloudy	Partly Cloudy	…
29	Fair	Fair	Fair	Fair	…
30	Cloudy	Cloudy	Mostly Cloudy	Mostly Cloudy	…

	2:51 PM	3:51 PM	4:51 PM	\
0	Cloudy	Mostly Cloudy	Mostly Cloudy	
1	Mostly Cloudy	Mostly Cloudy	Mostly Cloudy	
2	Partly Cloudy	Partly Cloudy	Partly Cloudy	
3	Fair / Windy	Fair	Fair	
4	Fair	Fair	Fair	
5	Fair	Fair	Fair	
6	Mostly Cloudy	Mostly Cloudy	Mostly Cloudy	
7	Cloudy	Mostly Cloudy	Cloudy	
8	Cloudy	Cloudy	Cloudy	
9	Fog	Fog	Light Rain	
10	Fair	Fair	Fair / Windy	
11	Cloudy	Cloudy	Cloudy / Windy	
12	Fair / Windy	Fair	Partly Cloudy	
13	Mostly Cloudy	Mostly Cloudy	Cloudy	
14	Cloudy	Cloudy	Cloudy	
15	Mostly Cloudy	Mostly Cloudy	Mostly Cloudy	
16	Cloudy	Light Snow	Light Snow	
17	Fair / Windy	Partly Cloudy / Windy	Mostly Cloudy / Windy	
18	Fair / Windy	Fair / Windy	Partly Cloudy / Windy	
19	Mostly Cloudy	Cloudy	Cloudy	
20	Partly Cloudy	Partly Cloudy	Partly Cloudy / Windy	
21	Cloudy	Cloudy	Cloudy	
22	Heavy Snow / Windy	Heavy Snow / Windy	Heavy Snow	
23	Partly Cloudy	Partly Cloudy	Partly Cloudy	
24	Mostly Cloudy	Cloudy	Cloudy	

25	Cloudy	Cloudy	Fair
26	Cloudy	Cloudy	Fair
27	Fair	Mostly Cloudy	Cloudy
28	Cloudy	Cloudy	Cloudy
29	Mostly Cloudy	Mostly Cloudy	Mostly Cloudy
30	Mostly Cloudy	Cloudy	Mostly Cloudy

	5:51 PM	6:51 PM	7:51 PM
0	Partly Cloudy	Partly Cloudy	Partly Cloudy
1	Mostly Cloudy	Partly Cloudy	Fair
2	Mostly Cloudy	Partly Cloudy	Partly Cloudy
3	Fair / Windy	Fair / Windy	Fair
4	Fair	Fair	Fair
5	Fair	Fair	Fair
6	Mostly Cloudy	Mostly Cloudy	Mostly Cloudy
7	Cloudy	Cloudy	Cloudy
8	Cloudy	Cloudy	Cloudy
9	Mostly Cloudy	Partly Cloudy	Fair
10	Fair	Fair	Fair
11	Light Rain	Cloudy	Mostly Cloudy
12	Partly Cloudy	Mostly Cloudy	Partly Cloudy
13	Cloudy	Cloudy	Cloudy
14	Cloudy	Cloudy	Cloudy
15	Cloudy	Cloudy	Mostly Cloudy
16	Light Snow	Light Snow	Light Snow
17	Partly Cloudy	Partly Cloudy / Windy	Partly Cloudy / Windy
18	Partly Cloudy / Windy	Fair	Fair / Windy
19	Cloudy	Mostly Cloudy	Mostly Cloudy
20	Partly Cloudy	Mostly Cloudy	Partly Cloudy
21	Cloudy	Cloudy	Cloudy
22	Heavy Snow / Windy	Heavy Snow / Windy	Heavy Snow
23	Mostly Cloudy	Mostly Cloudy	Partly Cloudy
24	Cloudy	Cloudy	Mostly Cloudy
25	Cloudy	Cloudy	Cloudy
26	Cloudy	Cloudy	Cloudy
27	Mostly Cloudy	Partly Cloudy	Cloudy
28	Mostly Cloudy / Windy	Mostly Cloudy / Windy	Mostly Cloudy / Windy
29	Mostly Cloudy	Partly Cloudy	Partly Cloudy
30	Cloudy	Mostly Cloudy	Cloudy

	8:51 PM	9:51 PM	10:51 PM
0	Fair	Fair	Fair
1	Fair	Fair	Fair
2	Partly Cloudy	Fair	Fair
3	Partly Cloudy / Windy	Mostly Cloudy / Windy	Mostly Cloudy / Windy
4	Fair	Fair	Fair
5	Fair	Fair	Fair

6	Fair	Fair	Fair
7	Cloudy	Cloudy	Cloudy
8	Cloudy	Light Drizzle	Cloudy
9	Mostly Cloudy	Fair / Windy	Fair / Windy
10	Fair	Fair	Fair
11	Mostly Cloudy	Cloudy / Windy	Mostly Cloudy / Windy
12	Cloudy	Cloudy	Cloudy
13	Cloudy	Partly Cloudy	Mostly Cloudy
14	Cloudy	Cloudy	Cloudy
15	Cloudy	Cloudy	Cloudy
16	Light Snow	Mostly Cloudy	Partly Cloudy
17	Fair / Windy	Fair	Fair / Windy
18	Fair	Fair	Mostly Cloudy / Windy
19	Cloudy	Cloudy	Cloudy
20	Partly Cloudy	Fair	Fair
21	Cloudy	Cloudy	Cloudy
22	Heavy Snow / Windy	Snow / Windy	Light Snow
23	Partly Cloudy	Mostly Cloudy	Mostly Cloudy
24	Fair	Fair	Fair
25	Cloudy	Cloudy	Cloudy
26	Cloudy	Cloudy	Cloudy
27	Cloudy	Cloudy	Cloudy
28	Partly Cloudy / Windy	Fair	Fair
29	Partly Cloudy	Cloudy	Mostly Cloudy
30	Cloudy	Cloudy	Cloudy

	11:51 PM
0	Fair
1	Fair
2	Fair
3	Fair / Windy
4	Fair
5	Fair
6	Fair
7	Cloudy
8	Cloudy
9	Fair
10	Fair
11	Partly Cloudy
12	Cloudy
13	Cloudy
14	Cloudy
15	Mostly Cloudy
16	Partly Cloudy
17	Fair / Windy
18	Mostly Cloudy
19	Cloudy

```
20       Partly Cloudy
21          Light Snow
22  Light Snow / Windy
23              Cloudy
24       Mostly Cloudy
25              Cloudy
26              Cloudy
27              Cloudy
28                Fair
29       Mostly Cloudy
30              Cloudy

[31 rows x 25 columns]
```

[3]:
```
weather.columns = ['time'] + [i for i in range(24)]
weather_list = weather.iloc[:,1:].values.tolist()
weather
```

[3]:

	time	0	1	\
0	2016/1/1	Fair	Cloudy	
1	2016/1/2	Fair	Fair	
2	2016/1/3	Fair	Fair	
3	2016/1/4	Partly Cloudy / Windy	Fair	
4	2016/1/5	Fair	Fair / Windy	
5	2016/1/6	Fair	Fair	
6	2016/1/7	Fair	Fair	
7	2016/1/8	Cloudy	Fair	
8	2016/1/9	Cloudy	Cloudy	
9	2016/1/10	Mostly Cloudy	Light Rain	
10	2016/1/11	Fair	Mostly Cloudy	
11	2016/1/12	Partly Cloudy / Windy	Fair	
12	2016/1/13	Cloudy	Fair / Windy	
13	2016/1/14	Cloudy	Mostly Cloudy	
14	2016/1/15	Cloudy	Cloudy	
15	2016/1/16	Mostly Cloudy	Cloudy	
16	2016/1/17	Cloudy	Mostly Cloudy	
17	2016/1/18	Fair	Fair	
18	2016/1/19	Partly Cloudy / Windy	Fair	
19	2016/1/20	Cloudy	Fair / Windy	
20	2016/1/21	Fair / Windy	NaN	
21	2016/1/22	Light Snow	Fair	
22	2016/1/23	Light Snow	Light Snow	
23	2016/1/24	Cloudy	Light Snow / Windy	
24	2016/1/25	Cloudy	Partly Cloudy	
25	2016/1/26	Cloudy	Cloudy	
26	2016/1/27	Cloudy	Cloudy	
27	2016/1/28	Cloudy	Fair	

```
28  2016/1/29                  Fair              Cloudy
29  2016/1/30          Partly Cloudy       Partly Cloudy
30  2016/1/31                Cloudy              Cloudy

                      2                     3                     4  \
0                Cloudy                Cloudy                Cloudy
1                  Fair          Mostly Cloudy          Mostly Cloudy
2                  Fair                   Fair                   Fair
3          Mostly Cloudy  Mostly Cloudy / Windy          Mostly Cloudy
4                  Fair                   Fair                   Fair
5                  Fair                 Cloudy                   Fair
6                  Fair                   Fair                   Fair
7                  Fair          Partly Cloudy                   Fair
8          Mostly Cloudy          Mostly Cloudy                 Cloudy
9                Cloudy                Cloudy                   Rain
10  Mostly Cloudy / Windy  Partly Cloudy / Windy  Partly Cloudy / Windy
11         Partly Cloudy          Partly Cloudy          Mostly Cloudy
12  Partly Cloudy / Windy           Fair / Windy          Partly Cloudy
13               Cloudy                Cloudy                 Cloudy
14         Partly Cloudy          Partly Cloudy          Partly Cloudy
15                 Rain             Light Rain             Light Rain
16               Cloudy                 Cloudy          Mostly Cloudy
17                 Fair                   Fair                   Fair
18          Fair / Windy                   Fair                   Fair
19                 Fair                   Fair                   Fair
20               Cloudy                 Cloudy                 Cloudy
21                 Fair                   Fair                   Fair
22    Light Snow / Windy     Light Snow / Windy     Light Snow / Windy
23    Light Snow / Windy         Cloudy / Windy                 Cloudy
24         Partly Cloudy                   Fair                   Fair
25         Mostly Cloudy                 Cloudy                 Cloudy
26         Mostly Cloudy                 Cloudy                 Cloudy
27         Partly Cloudy          Partly Cloudy                   Fair
28               Cloudy                 Cloudy                 Cloudy
29                 Fair          Partly Cloudy          Partly Cloudy
30         Partly Cloudy          Partly Cloudy                 Cloudy

                5             6             7                     8  …  \
0          Cloudy        Cloudy  Mostly Cloudy                Cloudy  …
1   Partly Cloudy          Fair  Partly Cloudy          Mostly Cloudy  …
2            Fair          Fair           Fair                   Fair  …
3   Mostly Cloudy  Partly Cloudy  Mostly Cloudy          Mostly Cloudy  …
4            Fair          Fair           Fair                   Fair  …
5          Cloudy        Cloudy         Cloudy                 Cloudy  …
6            Fair  Partly Cloudy  Partly Cloudy          Mostly Cloudy  …
7            Fair          Fair           Fair          Partly Cloudy  …
8          Cloudy        Cloudy         Cloudy          Mostly Cloudy  …
```

9	Heavy Rain	Rain	Rain	Heavy Rain	…
10	Fair	Fair	Fair	Fair	…
11	Mostly Cloudy	Partly Cloudy	Mostly Cloudy	Mostly Cloudy	…
12	Fair / Windy	Fair / Windy	Fair / Windy	Fair	…
13	Cloudy	Light Snow	Cloudy	Cloudy	…
14	Partly Cloudy	Cloudy	Cloudy	Cloudy	…
15	Light Rain	Cloudy	Light Rain	Light Rain	…
16	Fair	Fair	Mostly Cloudy	Cloudy	…
17	Fair / Windy	Partly Cloudy	Light Snow	Mostly Cloudy / Windy	…
18	Fair	Fair / Windy	Fair	Fair / Windy	…
19	Fair	Partly Cloudy	Fair	Partly Cloudy	…
20	Cloudy	Partly Cloudy	Fair	Fair	…
21	Fair	Fair	Fair	Mostly Cloudy	…
22	Snow / Windy	Snow	Snow / Windy	Heavy Snow / Windy	…
23	Fair / Windy	Fair	Fair	Fair	…
24	Mostly Cloudy	Cloudy	Cloudy	Cloudy	…
25	Cloudy	Cloudy	Cloudy	Cloudy	…
26	Cloudy	Cloudy	Cloudy	Cloudy	…
27	Fair	Fair	Fair	Fair	…
28	Cloudy	Mostly Cloudy	Partly Cloudy	Partly Cloudy	…
29	Fair	Fair	Fair	Fair	…
30	Cloudy	Cloudy	Mostly Cloudy	Mostly Cloudy	…

	14	15	16	\
0	Cloudy	Mostly Cloudy	Mostly Cloudy	
1	Mostly Cloudy	Mostly Cloudy	Mostly Cloudy	
2	Partly Cloudy	Partly Cloudy	Partly Cloudy	
3	Fair / Windy	Fair	Fair	
4	Fair	Fair	Fair	
5	Fair	Fair	Fair	
6	Mostly Cloudy	Mostly Cloudy	Mostly Cloudy	
7	Cloudy	Mostly Cloudy	Cloudy	
8	Cloudy	Cloudy	Cloudy	
9	Fog	Fog	Light Rain	
10	Fair	Fair	Fair / Windy	
11	Cloudy	Cloudy	Cloudy / Windy	
12	Fair / Windy	Fair	Partly Cloudy	
13	Mostly Cloudy	Mostly Cloudy	Cloudy	
14	Cloudy	Cloudy	Cloudy	
15	Mostly Cloudy	Mostly Cloudy	Mostly Cloudy	
16	Cloudy	Light Snow	Light Snow	
17	Fair / Windy	Partly Cloudy / Windy	Mostly Cloudy / Windy	
18	Fair / Windy	Fair / Windy	Partly Cloudy / Windy	
19	Mostly Cloudy	Cloudy	Cloudy	
20	Partly Cloudy	Partly Cloudy	Partly Cloudy / Windy	
21	Cloudy	Cloudy	Cloudy	
22	Heavy Snow / Windy	Heavy Snow / Windy	Heavy Snow	

23	Partly Cloudy	Partly Cloudy	Partly Cloudy
24	Mostly Cloudy	Cloudy	Cloudy
25	Cloudy	Cloudy	Fair
26	Cloudy	Cloudy	Fair
27	Fair	Mostly Cloudy	Cloudy
28	Cloudy	Cloudy	Cloudy
29	Mostly Cloudy	Mostly Cloudy	Mostly Cloudy
30	Mostly Cloudy	Cloudy	Mostly Cloudy

	17	18	19	\
0	Partly Cloudy	Partly Cloudy	Partly Cloudy	
1	Mostly Cloudy	Partly Cloudy	Fair	
2	Mostly Cloudy	Partly Cloudy	Partly Cloudy	
3	Fair / Windy	Fair / Windy	Fair	
4	Fair	Fair	Fair	
5	Fair	Fair	Fair	
6	Mostly Cloudy	Mostly Cloudy	Mostly Cloudy	
7	Cloudy	Cloudy	Cloudy	
8	Cloudy	Cloudy	Cloudy	
9	Mostly Cloudy	Partly Cloudy	Fair	
10	Fair	Fair	Fair	
11	Light Rain	Cloudy	Mostly Cloudy	
12	Partly Cloudy	Mostly Cloudy	Partly Cloudy	
13	Cloudy	Cloudy	Cloudy	
14	Cloudy	Cloudy	Cloudy	
15	Cloudy	Cloudy	Mostly Cloudy	
16	Light Snow	Light Snow	Light Snow	
17	Partly Cloudy	Partly Cloudy / Windy	Partly Cloudy / Windy	
18	Partly Cloudy / Windy	Fair	Fair / Windy	
19	Cloudy	Mostly Cloudy	Mostly Cloudy	
20	Partly Cloudy	Mostly Cloudy	Partly Cloudy	
21	Cloudy	Cloudy	Cloudy	
22	Heavy Snow / Windy	Heavy Snow / Windy	Heavy Snow	
23	Mostly Cloudy	Mostly Cloudy	Partly Cloudy	
24	Cloudy	Cloudy	Mostly Cloudy	
25	Cloudy	Cloudy	Cloudy	
26	Cloudy	Cloudy	Cloudy	
27	Mostly Cloudy	Partly Cloudy	Cloudy	
28	Mostly Cloudy / Windy	Mostly Cloudy / Windy	Mostly Cloudy / Windy	
29	Mostly Cloudy	Partly Cloudy	Partly Cloudy	
30	Cloudy	Mostly Cloudy	Cloudy	

	20	21	22	\
0	Fair	Fair	Fair	
1	Fair	Fair	Fair	
2	Partly Cloudy	Fair	Fair	
3	Partly Cloudy / Windy	Mostly Cloudy / Windy	Mostly Cloudy / Windy	

4	Fair	Fair	Fair
5	Fair	Fair	Fair
6	Fair	Fair	Fair
7	Cloudy	Cloudy	Cloudy
8	Cloudy	Light Drizzle	Cloudy
9	Mostly Cloudy	Fair / Windy	Fair / Windy
10	Fair	Fair	Fair
11	Mostly Cloudy	Cloudy / Windy	Mostly Cloudy / Windy
12	Cloudy	Cloudy	Cloudy
13	Cloudy	Partly Cloudy	Mostly Cloudy
14	Cloudy	Cloudy	Cloudy
15	Cloudy	Cloudy	Cloudy
16	Light Snow	Mostly Cloudy	Partly Cloudy
17	Fair / Windy	Fair	Fair / Windy
18	Fair	Fair	Mostly Cloudy / Windy
19	Cloudy	Cloudy	Cloudy
20	Partly Cloudy	Fair	Fair
21	Cloudy	Cloudy	Cloudy
22	Heavy Snow / Windy	Snow / Windy	Light Snow
23	Partly Cloudy	Mostly Cloudy	Mostly Cloudy
24	Fair	Fair	Fair
25	Cloudy	Cloudy	Cloudy
26	Cloudy	Cloudy	Cloudy
27	Cloudy	Cloudy	Cloudy
28	Partly Cloudy / Windy	Fair	Fair
29	Partly Cloudy	Cloudy	Mostly Cloudy
30	Cloudy	Cloudy	Cloudy

	23
0	Fair
1	Fair
2	Fair
3	Fair / Windy
4	Fair
5	Fair
6	Fair
7	Cloudy
8	Cloudy
9	Fair
10	Fair
11	Partly Cloudy
12	Cloudy
13	Cloudy
14	Cloudy
15	Mostly Cloudy
16	Partly Cloudy
17	Fair / Windy

```
18       Mostly Cloudy
19              Cloudy
20       Partly Cloudy
21          Light Snow
22  Light Snow / Windy
23              Cloudy
24       Mostly Cloudy
25              Cloudy
26              Cloudy
27              Cloudy
28                Fair
29       Mostly Cloudy
30              Cloudy

[31 rows x 25 columns]
```

[4]:
```
possible = []
for i in weather_list:
    possible = possible + i

possible = list(set(possible))
possible
len(possible)
```

[4]: 21

[5]:
```
preci = ['Heavy Rain','Light Drizzle', 'Light Snow', 'Rain',
         'Light Rain / Windy', 'Heavy Snow', 'Snow', 'Light Snow / Windy',
         'Snow / Windy', 'Light Rain', 'Heavy Snow / Windy']

remain = ['Partly Cloudy', 'Cloudy / Windy', 'Partly Cloudy / Windy',
          'Fair / Windy', 'Fair', 'Mostly Cloudy',
          'Mostly Cloudy / Windy', 'Fog', 'Cloudy']
```

[ ]:
```

```

[6]:
```
def precipitation_or_not(condition):
    if condition in preci:
        return 'preci'
    elif condition in remain:
        return 'remain'
    else:
        return 'unknown'
```

[7]:
```
for i in range(24):
    weather.loc[:,i] = weather.loc[:,i].apply(precipitation_or_not)
```

[8]:
```
weather.iloc[20,2] = 'remain'
```

[ ]:
```
```

[9]:
```
weather['time'] = pd.to_datetime(weather['time'], format='%Y/%m/%d')
weather
```

[9]:

	time	0	1	2	3	4	5	6	7	\
0	2016-01-01	remain	remain	remain	remain	remain	remain	remain	remain	
1	2016-01-02	remain	remain	remain	remain	remain	remain	remain	remain	
2	2016-01-03	remain	remain	remain	remain	remain	remain	remain	remain	
3	2016-01-04	remain	remain	remain	remain	remain	remain	remain	remain	
4	2016-01-05	remain	remain	remain	remain	remain	remain	remain	remain	
5	2016-01-06	remain	remain	remain	remain	remain	remain	remain	remain	
6	2016-01-07	remain	remain	remain	remain	remain	remain	remain	remain	
7	2016-01-08	remain	remain	remain	remain	remain	remain	remain	remain	
8	2016-01-09	remain	remain	remain	remain	remain	remain	remain	remain	
9	2016-01-10	remain	preci	remain	remain	preci	preci	preci	preci	
10	2016-01-11	remain	remain	remain	remain	remain	remain	remain	remain	
11	2016-01-12	remain	remain	remain	remain	remain	remain	remain	remain	
12	2016-01-13	remain	remain	remain	remain	remain	remain	remain	remain	
13	2016-01-14	remain	remain	remain	remain	remain	remain	preci	remain	
14	2016-01-15	remain	remain	remain	remain	remain	remain	remain	remain	
15	2016-01-16	remain	remain	preci	preci	preci	preci	remain	preci	
16	2016-01-17	remain	remain	remain	remain	remain	remain	remain	remain	
17	2016-01-18	remain	remain	remain	remain	remain	remain	remain	preci	
18	2016-01-19	remain	remain	remain	remain	remain	remain	remain	remain	
19	2016-01-20	remain	remain	remain	remain	remain	remain	remain	remain	
20	2016-01-21	remain	remain	remain	remain	remain	remain	remain	remain	
21	2016-01-22	preci	remain	remain	remain	remain	remain	remain	remain	
22	2016-01-23	preci	preci	preci	preci	preci	preci	preci	preci	
23	2016-01-24	remain	preci	preci	remain	remain	remain	remain	remain	
24	2016-01-25	remain	remain	remain	remain	remain	remain	remain	remain	
25	2016-01-26	remain	remain	remain	remain	remain	remain	remain	remain	
26	2016-01-27	remain	remain	remain	remain	remain	remain	remain	remain	
27	2016-01-28	remain	remain	remain	remain	remain	remain	remain	remain	
28	2016-01-29	remain	remain	remain	remain	remain	remain	remain	remain	
29	2016-01-30	remain	remain	remain	remain	remain	remain	remain	remain	
30	2016-01-31	remain	remain	remain	remain	remain	remain	remain	remain	

	8	…	14	15	16	17	18	19	20	\
0	remain	…	remain	remain	remain	remain	remain	remain	remain	
1	remain	…	remain	remain	remain	remain	remain	remain	remain	
2	remain	…	remain	remain	remain	remain	remain	remain	remain	
3	remain	…	remain	remain	remain	remain	remain	remain	remain	
4	remain	…	remain	remain	remain	remain	remain	remain	remain	
5	remain	…	remain	remain	remain	remain	remain	remain	remain	

6	remain	…	remain	remain	remain	remain	remain	remain	remain
7	remain	…	remain	remain	remain	remain	remain	remain	remain
8	remain	…	remain	remain	remain	remain	remain	remain	remain
9	preci	…	remain	remain	preci	remain	remain	remain	remain
10	remain	…	remain	remain	remain	remain	remain	remain	remain
11	remain	…	remain	remain	remain	preci	remain	remain	remain
12	remain	…	remain	remain	remain	remain	remain	remain	remain
13	remain	…	remain	remain	remain	remain	remain	remain	remain
14	remain	…	remain	remain	remain	remain	remain	remain	remain
15	preci	…	remain	remain	remain	remain	remain	remain	remain
16	remain	…	remain	preci	preci	preci	preci	preci	preci
17	remain	…	remain	remain	remain	remain	remain	remain	remain
18	remain	…	remain	remain	remain	remain	remain	remain	remain
19	remain	…	remain	remain	remain	remain	remain	remain	remain
20	remain	…	remain	remain	remain	remain	remain	remain	remain
21	remain	…	remain	remain	remain	remain	remain	remain	remain
22	preci	…	preci	preci	preci	preci	preci	preci	preci
23	remain	…	remain	remain	remain	remain	remain	remain	remain
24	remain	…	remain	remain	remain	remain	remain	remain	remain
25	remain	…	remain	remain	remain	remain	remain	remain	remain
26	remain	…	remain	remain	remain	remain	remain	remain	remain
27	remain	…	remain	remain	remain	remain	remain	remain	remain
28	remain	…	remain	remain	remain	remain	remain	remain	remain
29	remain	…	remain	remain	remain	remain	remain	remain	remain
30	remain	…	remain	remain	remain	remain	remain	remain	remain

	21	22	23
0	remain	remain	remain
1	remain	remain	remain
2	remain	remain	remain
3	remain	remain	remain
4	remain	remain	remain
5	remain	remain	remain
6	remain	remain	remain
7	remain	remain	remain
8	preci	remain	remain
9	remain	remain	remain
10	remain	remain	remain
11	remain	remain	remain
12	remain	remain	remain
13	remain	remain	remain
14	remain	remain	remain
15	remain	remain	remain
16	remain	remain	remain
17	remain	remain	remain
18	remain	remain	remain
19	remain	remain	remain

```
20  remain  remain  remain
21  remain  remain   preci
22   preci   preci   preci
23  remain  remain  remain
24  remain  remain  remain
25  remain  remain  remain
26  remain  remain  remain
27  remain  remain  remain
28  remain  remain  remain
29  remain  remain  remain
30  remain  remain  remain

[31 rows x 25 columns]
```

[10]:
```
weather.to_csv('data/weather_preprocessed.csv', index=False)
```

[ ]:
```

```

Appendix B

Jupyter Notebook part 2 (the page number is only for this Notebook)

[1]:
```
import pandas as pd
import numpy as np
from numpy import log, sqrt
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
```

[2]:
```
df = pd.read_csv('data/yellow_tripdata_2016-01.csv',error_bad_lines=False)
df
```

[2]:
```
          VendorID tpep_pickup_datetime tpep_dropoff_datetime  \
0                2  2016-01-01 00:00:00   2016-01-01 00:00:00
1                2  2016-01-01 00:00:00   2016-01-01 00:00:00
2                2  2016-01-01 00:00:00   2016-01-01 00:00:00
3                2  2016-01-01 00:00:00   2016-01-01 00:00:00
4                2  2016-01-01 00:00:00   2016-01-01 00:00:00
…              …                    …                    …
10906853         2  2016-01-31 23:30:32   2016-01-31 23:38:18
10906854         1  2016-01-05 00:15:55   2016-01-05 00:16:06
10906855         1  2016-01-05 06:12:46   2016-03-19 20:45:50
10906856         1  2016-01-05 06:21:44   2016-03-28 12:54:26
10906857         1  2016-01-05 06:15:21   2016-01-05 06:15:36

          passenger_count  trip_distance  pickup_longitude  pickup_latitude  \
0                       2           1.10       -73.990372        40.734695
1                       5           4.90       -73.980782        40.729912
2                       1          10.54       -73.984550        40.679565
3                       1           4.75       -73.993469        40.718990
4                       3           1.76       -73.960625        40.781330
…                     …             …                  …                …
10906853                1           2.20       -74.003578        40.751011
10906854                1           0.00       -73.945488        40.751530
10906855                3           1.40       -73.994240        40.766586
10906856                1           2.10       -73.948067        40.776531
10906857                3           0.00       -73.960938        40.758595

          RatecodeID store_and_fwd_flag  dropoff_longitude  dropoff_latitude  \
```

```
0                  1                  N         -73.981842         40.732407
1                  1                  N         -73.944473         40.716679
2                  1                  N         -73.950272         40.788925
3                  1                  N         -73.962242         40.657333
4                  1                  N         -73.977264         40.758514
…                …                  …                  …                …
10906853           1                  N         -73.982651         40.767509
10906854           1                  N         -73.945457         40.751530
10906855           1                  N         -73.984428         40.753922
10906856           1                  N         -73.978188         40.777435
10906857           2                  N         -73.961006         40.758583

          payment_type  fare_amount  extra  mta_tax  tip_amount  tolls_amount  \
0                    2          7.5    0.5      0.5        0.00          0.00
1                    1         18.0    0.5      0.5        0.00          0.00
2                    1         33.0    0.5      0.5        0.00          0.00
3                    2         16.5    0.0      0.5        0.00          0.00
4                    2          8.0    0.0      0.5        0.00          0.00
…                  …           …    …        …          …             …
10906853             2          8.5    0.5      0.5        0.00          0.00
10906854             2          2.5    0.5      0.5        0.00          0.00
10906855             2          7.5    0.5      0.5        0.00          0.00
10906856             1         11.5    0.0      0.5        2.45          0.00
10906857             2         52.0    0.0      0.5        0.00          5.54

          improvement_surcharge  total_amount
0                           0.3          8.80
1                           0.3         19.30
2                           0.3         34.30
3                           0.3         17.30
4                           0.3          8.80
…                         …            …
10906853                    0.3          9.80
10906854                    0.3          3.80
10906855                    0.3          8.80
10906856                    0.3         14.75
10906857                    0.3         58.34

[10906858 rows x 19 columns]
```

[3]:
```
df.columns
```

[3]:
```
Index(['VendorID', 'tpep_pickup_datetime', 'tpep_dropoff_datetime',
       'passenger_count', 'trip_distance', 'pickup_longitude',
       'pickup_latitude', 'RatecodeID', 'store_and_fwd_flag',
       'dropoff_longitude', 'dropoff_latitude', 'payment_type', 'fare_amount',
       'extra', 'mta_tax', 'tip_amount', 'tolls_amount',
```

```
       'improvement_surcharge', 'total_amount'],
      dtype='object')
```

[ ]:

[4]:
```
df.dropna(inplace=True)
df['tpep_pickup_datetime'] = pd.to_datetime(df['tpep_pickup_datetime'],
                                format='%Y/%m/%d %H:%M', errors='coerce')
df['tpep_dropoff_datetime'] = pd.to_datetime(df['tpep_dropoff_datetime'],
                                format='%Y/%m/%d %H:%M', errors='coerce')

df['duration'] = (df['tpep_dropoff_datetime'] -
                  df['tpep_pickup_datetime']).dt.seconds.astype(int) /60
```

[5]:
```
df['start_hour'] =  df['tpep_pickup_datetime'].dt.hour
df['start_date'] = df['tpep_pickup_datetime'].dt.strftime('%Y-%m-%d')
```

[ ]:

[ ]:

[6]:
```
weather = pd.read_csv('data/weather_preprocessed.csv', index_col= 'time')
weather
```

[6]:

time	0	1	2	3	4	5	6	7
2016-01-01	remain	remain	remain	remain	remain	remain	remain	remain
2016-01-02	remain	remain	remain	remain	remain	remain	remain	remain
2016-01-03	remain	remain	remain	remain	remain	remain	remain	remain
2016-01-04	remain	remain	remain	remain	remain	remain	remain	remain
2016-01-05	remain	remain	remain	remain	remain	remain	remain	remain
2016-01-06	remain	remain	remain	remain	remain	remain	remain	remain
2016-01-07	remain	remain	remain	remain	remain	remain	remain	remain
2016-01-08	remain	remain	remain	remain	remain	remain	remain	remain
2016-01-09	remain	remain	remain	remain	remain	remain	remain	remain
2016-01-10	remain	preci	remain	remain	preci	preci	preci	preci
2016-01-11	remain	remain	remain	remain	remain	remain	remain	remain
2016-01-12	remain	remain	remain	remain	remain	remain	remain	remain
2016-01-13	remain	remain	remain	remain	remain	remain	remain	remain
2016-01-14	remain	remain	remain	remain	remain	remain	preci	remain
2016-01-15	remain	remain	remain	remain	remain	remain	remain	remain
2016-01-16	remain	remain	preci	preci	preci	preci	remain	preci
2016-01-17	remain	remain	remain	remain	remain	remain	remain	remain
2016-01-18	remain	remain	remain	remain	remain	remain	remain	preci
2016-01-19	remain	remain	remain	remain	remain	remain	remain	remain
2016-01-20	remain	remain	remain	remain	remain	remain	remain	remain
2016-01-21	remain	remain	remain	remain	remain	remain	remain	remain

2016-01-22	preci	remain	remain	remain	remain	remain	remain	remain
2016-01-23	preci	preci	preci	preci	preci	preci	preci	preci
2016-01-24	remain	preci	preci	remain	remain	remain	remain	remain
2016-01-25	remain	remain	remain	remain	remain	remain	remain	remain
2016-01-26	remain	remain	remain	remain	remain	remain	remain	remain
2016-01-27	remain	remain	remain	remain	remain	remain	remain	remain
2016-01-28	remain	remain	remain	remain	remain	remain	remain	remain
2016-01-29	remain	remain	remain	remain	remain	remain	remain	remain
2016-01-30	remain	remain	remain	remain	remain	remain	remain	remain
2016-01-31	remain	remain	remain	remain	remain	remain	remain	remain

time	8	9	…	14	15	16	17	18	\
2016-01-01	remain	remain	…	remain	remain	remain	remain	remain	
2016-01-02	remain	remain	…	remain	remain	remain	remain	remain	
2016-01-03	remain	remain	…	remain	remain	remain	remain	remain	
2016-01-04	remain	remain	…	remain	remain	remain	remain	remain	
2016-01-05	remain	remain	…	remain	remain	remain	remain	remain	
2016-01-06	remain	remain	…	remain	remain	remain	remain	remain	
2016-01-07	remain	remain	…	remain	remain	remain	remain	remain	
2016-01-08	remain	remain	…	remain	remain	remain	remain	remain	
2016-01-09	remain	remain	…	remain	remain	remain	remain	remain	
2016-01-10	preci	preci	…	remain	remain	preci	remain	remain	
2016-01-11	remain	remain	…	remain	remain	remain	remain	remain	
2016-01-12	remain	remain	…	remain	remain	remain	preci	remain	
2016-01-13	remain	remain	…	remain	remain	remain	remain	remain	
2016-01-14	remain	remain	…	remain	remain	remain	remain	remain	
2016-01-15	remain	remain	…	remain	remain	remain	remain	remain	
2016-01-16	preci	remain	…	remain	remain	remain	remain	remain	
2016-01-17	remain	remain	…	remain	preci	preci	preci	preci	
2016-01-18	remain	remain	…	remain	remain	remain	remain	remain	
2016-01-19	remain	remain	…	remain	remain	remain	remain	remain	
2016-01-20	remain	remain	…	remain	remain	remain	remain	remain	
2016-01-21	remain	remain	…	remain	remain	remain	remain	remain	
2016-01-22	remain	remain	…	remain	remain	remain	remain	remain	
2016-01-23	preci	preci	…	preci	preci	preci	preci	preci	
2016-01-24	remain	remain	…	remain	remain	remain	remain	remain	
2016-01-25	remain	remain	…	remain	remain	remain	remain	remain	
2016-01-26	remain	remain	…	remain	remain	remain	remain	remain	
2016-01-27	remain	remain	…	remain	remain	remain	remain	remain	
2016-01-28	remain	remain	…	remain	remain	remain	remain	remain	
2016-01-29	remain	remain	…	remain	remain	remain	remain	remain	
2016-01-30	remain	remain	…	remain	remain	remain	remain	remain	
2016-01-31	remain	remain	…	remain	remain	remain	remain	remain	

time	19	20	21	22	23

```
2016-01-01  remain  remain  remain  remain  remain
2016-01-02  remain  remain  remain  remain  remain
2016-01-03  remain  remain  remain  remain  remain
2016-01-04  remain  remain  remain  remain  remain
2016-01-05  remain  remain  remain  remain  remain
2016-01-06  remain  remain  remain  remain  remain
2016-01-07  remain  remain  remain  remain  remain
2016-01-08  remain  remain  remain  remain  remain
2016-01-09  remain  remain   preci  remain  remain
2016-01-10  remain  remain  remain  remain  remain
2016-01-11  remain  remain  remain  remain  remain
2016-01-12  remain  remain  remain  remain  remain
2016-01-13  remain  remain  remain  remain  remain
2016-01-14  remain  remain  remain  remain  remain
2016-01-15  remain  remain  remain  remain  remain
2016-01-16  remain  remain  remain  remain  remain
2016-01-17   preci   preci  remain  remain  remain
2016-01-18  remain  remain  remain  remain  remain
2016-01-19  remain  remain  remain  remain  remain
2016-01-20  remain  remain  remain  remain  remain
2016-01-21  remain  remain  remain  remain  remain
2016-01-22  remain  remain  remain  remain   preci
2016-01-23   preci   preci   preci   preci   preci
2016-01-24  remain  remain  remain  remain  remain
2016-01-25  remain  remain  remain  remain  remain
2016-01-26  remain  remain  remain  remain  remain
2016-01-27  remain  remain  remain  remain  remain
2016-01-28  remain  remain  remain  remain  remain
2016-01-29  remain  remain  remain  remain  remain
2016-01-30  remain  remain  remain  remain  remain
2016-01-31  remain  remain  remain  remain  remain

[31 rows x 24 columns]
```

[ ]:

[7]:
```
def fill_weather(weather, date, time):
    return weather.loc[date][time]


df['weather'] = df[['start_date','start_hour']].apply(
    lambda x: fill_weather(weather,x.iloc[0],x.iloc[1]), axis=1)
```

[8]:
```
df.to_feather('data/yellow_tripdata_01_weather.feather')
```

[9]:
```
df
```

[9]:

	VendorID	tpep_pickup_datetime	tpep_dropoff_datetime	\
0	2	2016-01-01 00:00:00	2016-01-01 00:00:00	
1	2	2016-01-01 00:00:00	2016-01-01 00:00:00	
2	2	2016-01-01 00:00:00	2016-01-01 00:00:00	
3	2	2016-01-01 00:00:00	2016-01-01 00:00:00	
4	2	2016-01-01 00:00:00	2016-01-01 00:00:00	
…	…	…	…	
10906853	2	2016-01-31 23:30:32	2016-01-31 23:38:18	
10906854	1	2016-01-05 00:15:55	2016-01-05 00:16:06	
10906855	1	2016-01-05 06:12:46	2016-03-19 20:45:50	
10906856	1	2016-01-05 06:21:44	2016-03-28 12:54:26	
10906857	1	2016-01-05 06:15:21	2016-01-05 06:15:36	

	passenger_count	trip_distance	pickup_longitude	pickup_latitude	\
0	2	1.10	-73.990372	40.734695	
1	5	4.90	-73.980782	40.729912	
2	1	10.54	-73.984550	40.679565	
3	1	4.75	-73.993469	40.718990	
4	3	1.76	-73.960625	40.781330	
…	…	…	…	…	
10906853	1	2.20	-74.003578	40.751011	
10906854	1	0.00	-73.945488	40.751530	
10906855	3	1.40	-73.994240	40.766586	
10906856	1	2.10	-73.948067	40.776531	
10906857	3	0.00	-73.960938	40.758595	

	RatecodeID	store_and_fwd_flag	dropoff_longitude	…	extra	\
0	1	N	-73.981842	…	0.5	
1	1	N	-73.944473	…	0.5	
2	1	N	-73.950272	…	0.5	
3	1	N	-73.962242	…	0.0	
4	1	N	-73.977264	…	0.0	
…	…	…	…	…	…	
10906853	1	N	-73.982651	…	0.5	
10906854	1	N	-73.945457	…	0.5	
10906855	1	N	-73.984428	…	0.5	
10906856	1	N	-73.978188	…	0.0	
10906857	2	N	-73.961006	…	0.0	

	mta_tax	tip_amount	tolls_amount	improvement_surcharge	\
0	0.5	0.00	0.00	0.3	
1	0.5	0.00	0.00	0.3	
2	0.5	0.00	0.00	0.3	
3	0.5	0.00	0.00	0.3	
4	0.5	0.00	0.00	0.3	
…	…	…	…	…	
10906853	0.5	0.00	0.00	0.3	

```
10906854      0.5        0.00          0.00                    0.3
10906855      0.5        0.00          0.00                    0.3
10906856      0.5        2.45          0.00                    0.3
10906857      0.5        0.00          5.54                    0.3

          total_amount    duration  start_hour  start_date  weather
0                 8.80    0.000000           0  2016-01-01   remain
1                19.30    0.000000           0  2016-01-01   remain
2                34.30    0.000000           0  2016-01-01   remain
3                17.30    0.000000           0  2016-01-01   remain
4                 8.80    0.000000           0  2016-01-01   remain
…                 …         …          …          …      …
10906853          9.80    7.766667          23  2016-01-31   remain
10906854          3.80    0.183333           0  2016-01-05   remain
10906855          8.80  873.066667           6  2016-01-05   remain
10906856         14.75  392.700000           6  2016-01-05   remain
10906857         58.34    0.250000           6  2016-01-05   remain

[10906858 rows x 23 columns]
```

Appendix C

Jupyter Notebook part 3(the page number is only for this Notebook)

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import folium
from folium.plugins import FastMarkerCluster
from folium.plugins import HeatMap
from statsmodels.formula.api import *
import matplotlib.image as mpimg
import io
from PIL import Image
```

[2]:
```
df = pd.read_feather('data/yellow_tripdata_01_weather.feather')
df["income"] = df['tip_amount'] + df['fare_amount']
df['income/duration'] = df['income'] / df['duration']
```

1 Data cleaning

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

[3]:

	VendorID	passenger_count	trip_distance	pickup_longitude	\
count	1.090686e+07	1.090686e+07	1.090686e+07	1.090686e+07	
mean	1.535024e+00	1.670847e+00	4.648197e+00	-7.281869e+01	
std	4.987718e-01	1.324891e+00	2.981095e+03	9.168964e+00	
min	1.000000e+00	0.000000e+00	0.000000e+00	-1.219343e+02	
25%	1.000000e+00	1.000000e+00	1.000000e+00	-7.399151e+01	
50%	2.000000e+00	1.000000e+00	1.670000e+00	-7.398138e+01	
75%	2.000000e+00	2.000000e+00	3.080000e+00	-7.396610e+01	
max	2.000000e+00	9.000000e+00	8.000010e+06	0.000000e+00	

	pickup_latitude	RatecodeID	dropoff_longitude	dropoff_latitude	\
count	1.090686e+07	1.090686e+07	1.090686e+07	1.090686e+07	
mean	4.011494e+01	1.039350e+00	-7.288659e+01	4.015315e+01	
std	5.051022e+00	5.186309e-01	8.900841e+00	4.903456e+00	
min	0.000000e+00	1.000000e+00	-1.219335e+02	0.000000e+00	
25%	4.073630e+01	1.000000e+00	-7.399107e+01	4.073481e+01	

```
50%       4.075369e+01  1.000000e+00      -7.397942e+01      4.075413e+01
75%       4.076808e+01  1.000000e+00      -7.396196e+01      4.076962e+01
max       6.090876e+01  9.900000e+01       0.000000e+00      6.090876e+01

       payment_type   fare_amount         extra       mta_tax    tip_amount  \
count  1.090686e+07  1.090686e+07  1.090686e+07  1.090686e+07  1.090686e+07
mean   1.347536e+00  1.248693e+01  3.130757e-01  4.976705e-01  1.750663e+00
std    4.910804e-01  3.556400e+01  4.156792e-01  5.046685e-02  2.623546e+00
min    1.000000e+00 -9.576000e+02 -4.261000e+01 -5.000000e-01 -2.208000e+02
25%    1.000000e+00  6.500000e+00  0.000000e+00  5.000000e-01  0.000000e+00
50%    1.000000e+00  9.000000e+00  0.000000e+00  5.000000e-01  1.260000e+00
75%    2.000000e+00  1.400000e+01  5.000000e-01  5.000000e-01  2.320000e+00
max    5.000000e+00  1.112709e+05  6.488700e+02  8.970000e+01  9.981400e+02

       tolls_amount  improvement_surcharge  total_amount      duration  \
count  1.090686e+07           1.090686e+07  1.090686e+07  1.090686e+07
mean   2.933453e-01           2.997245e-01  1.564140e+01  1.520518e+01
std    1.694572e+00           1.232553e-02  3.641280e+01  5.424797e+01
min   -1.740000e+01          -3.000000e-01 -9.584000e+02  0.000000e+00
25%    0.000000e+00           3.000000e-01  8.300000e+00  6.333333e+00
50%    0.000000e+00           3.000000e-01  1.162000e+01  1.046667e+01
75%    0.000000e+00           3.000000e-01  1.716000e+01  1.688333e+01
max    9.801500e+02           3.000000e-01  1.112716e+05  1.439967e+03

         start_hour        income  income/duration
count  1.090686e+07  1.090686e+07     1.090636e+07
mean   1.354638e+01  1.423759e+01              NaN
std    6.391860e+00  3.609683e+01              NaN
min    0.000000e+00 -9.576000e+02             -inf
25%    9.000000e+00  7.350000e+00     8.976378e-01
50%    1.400000e+01  1.046000e+01     1.065217e+00
75%    1.900000e+01  1.600000e+01     1.303579e+00
max    2.300000e+01  1.112709e+05              inf
```

[4]:
```
rid = df['RatecodeID'].value_counts()
rid
```

[4]:
```
1     10626315
2       225019
5        33688
3        16822
4         4696
99         216
6          102
Name: RatecodeID, dtype: int64
```

[5]:
```
df = df.loc[(df['RatecodeID'] != 99)]
```

[6]:
```
# group payment type
def group_rid(x):
    if x != 1:
        return 2
    else:
        return 1
df['RatecodeID'] = df['RatecodeID'].apply(group_rid)
```

[7]:
```
swf = df['store_and_fwd_flag'].value_counts()
swf
```

[7]:
```
N    10843513
Y       63129
Name: store_and_fwd_flag, dtype: int64
```

[8]:
```
pt = df['payment_type'].value_counts()
pt
```

[8]:
```
1    7181337
2    3673602
3      38292
4      13410
5          1
Name: payment_type, dtype: int64
```

[9]:
```
# group payment type
def group_payment_type(x):
    if x != 1:
        return 2
    else:
        return 1
df['payment_type'] = df['payment_type'].apply(group_payment_type)
```

[10]:
```
passenger = df['passenger_count'].value_counts()
passenger
```

[10]:
```
1    7726830
2    1561966
5     601079
3     436429
6     369155
4     210641
0        471
8         26
9         23
7         22
Name: passenger_count, dtype: int64
```

[11]:
```
df = df.loc[(df["duration"] >= 0.25) &  (df["fare_amount"] > 0) & (df["extra"]␣
↪>= 0) & (df["mta_tax"] >= 0)
            & (df["tip_amount"] >= 0) & (df["tolls_amount"] >=0)  &␣
↪(df["total_amount"]>0) & (df["income"] > 0)
            & (df["improvement_surcharge"] >= 0) & (df["trip_distance"] >= 0.
↪01)]
```

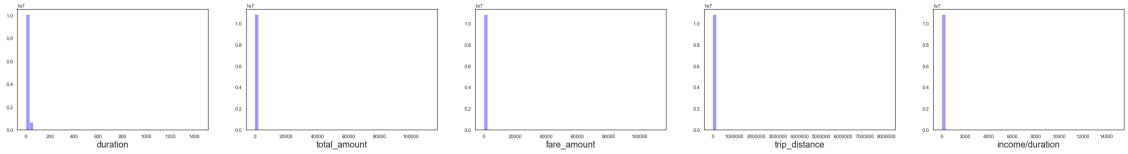
[12]:
```
# lat, long
start_coords = ['pickup_latitude', 'pickup_longitude']
end_coords = ['dropoff_latitude', 'dropoff_longitude']
df[start_coords+ end_coords].describe()
```

[12]:

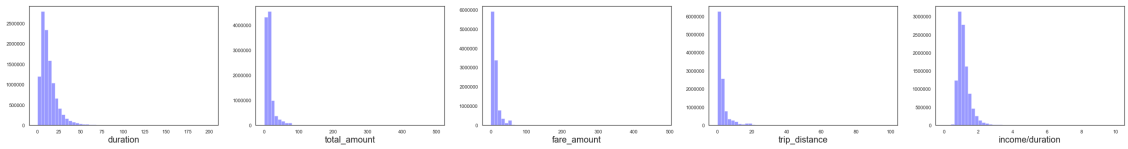
	pickup_latitude	pickup_longitude	dropoff_latitude	dropoff_longitude
count	1.082888e+07	1.082888e+07	1.082888e+07	1.082888e+07
mean	4.016214e+01	-7.290432e+01	4.022927e+01	-7.302472e+01
std	4.863079e+00	8.827634e+00	4.585772e+00	8.323998e+00
min	0.000000e+00	-1.008229e+02	0.000000e+00	-1.008229e+02
25%	4.073648e+01	-7.399152e+01	4.073501e+01	-7.399110e+01
50%	4.075378e+01	-7.398141e+01	4.075424e+01	-7.397948e+01
75%	4.076812e+01	-7.396624e+01	4.076968e+01	-7.396216e+01
max	6.090876e+01	0.000000e+00	6.090876e+01	0.000000e+00

[13]:
```
sns.set(rc={'figure.figsize':(11.7,8.27),"font.size":20,"axes.titlesize":
↪20,"axes.labelsize":20},style="white")
```

[14]:
```
fig, ax = plt.subplots(1, 5)
sns.distplot(df['duration'], kde = False, label = "duration", color ="blue", ax␣
↪= ax[0])
sns.distplot(df['total_amount'], kde = False, label = "total_amount", color␣
↪="blue", ax = ax[1])
sns.distplot(df['fare_amount'], kde = False, label = 'fare_amount', color␣
↪="blue", ax = ax[2])
sns.distplot(df['trip_distance'], kde = False, label = "duration", color␣
↪="blue", ax = ax[3])
sns.distplot(df['income/duration'], kde = False, label = "duration", color␣
↪="blue", ax = ax[4])


fig.set_figheight(5)
fig.set_figwidth(15)
fig.set_figwidth(25)
fig.set_figwidth(35)
fig.set_figwidth(45)
```



[15]:
```
df = df.loc[(df["duration"] <= 200) & (df["total_amount"]<= 500)
            & (df['fare_amount'] <= 500) & (df['trip_distance'] <= 100) &␣
 ↪(df["income/duration"] <= 10)]

fig, ax = plt.subplots(1, 5)
sns.distplot(df['duration'], kde = False, label = "duration", color ="blue", ax␣
 ↪= ax[0])
sns.distplot(df['total_amount'], kde = False, label = "total_amount", color␣
 ↪="blue", ax = ax[1])
sns.distplot(df['fare_amount'], kde = False, label = 'fare_amount', color␣
 ↪="blue", ax = ax[2])
sns.distplot(df['trip_distance'], kde = False, label = "duration", color␣
 ↪="blue", ax = ax[3])
sns.distplot(df['income/duration'], kde = False, label = "duration", color␣
 ↪="blue", ax = ax[4])

fig.set_figheight(5)
fig.set_figwidth(15)
fig.set_figwidth(25)
fig.set_figwidth(35)
fig.set_figwidth(45)
```

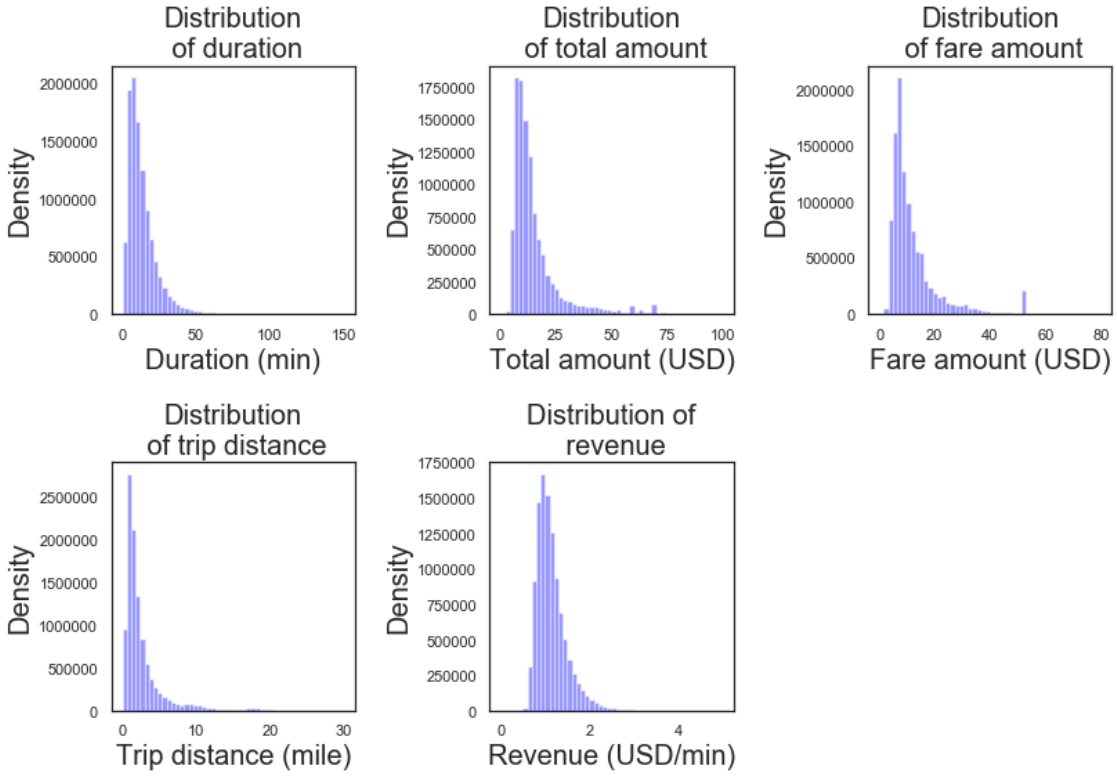


[16]:
```
df = df.loc[(df["duration"] <= 150) & (df["total_amount"]<=100) &␣
 ↪(df['fare_amount'] <=80) & (df['trip_distance'] <= 30) & (df["income/
 ↪duration"] <= 5)]

fig, ax = plt.subplots(1, 5)
sns.distplot(df['duration'], kde = False,  color ="blue", ax = ax[0])
sns.distplot(df['total_amount'], kde = False,  color ="blue", ax = ax[1])
sns.distplot(df['fare_amount'], kde = False, color ="blue", ax = ax[2])
sns.distplot(df['trip_distance'], kde = False,  color ="blue", ax = ax[3])
sns.distplot(df['income/duration'], kde = False,  color ="blue", ax = ax[4])
```

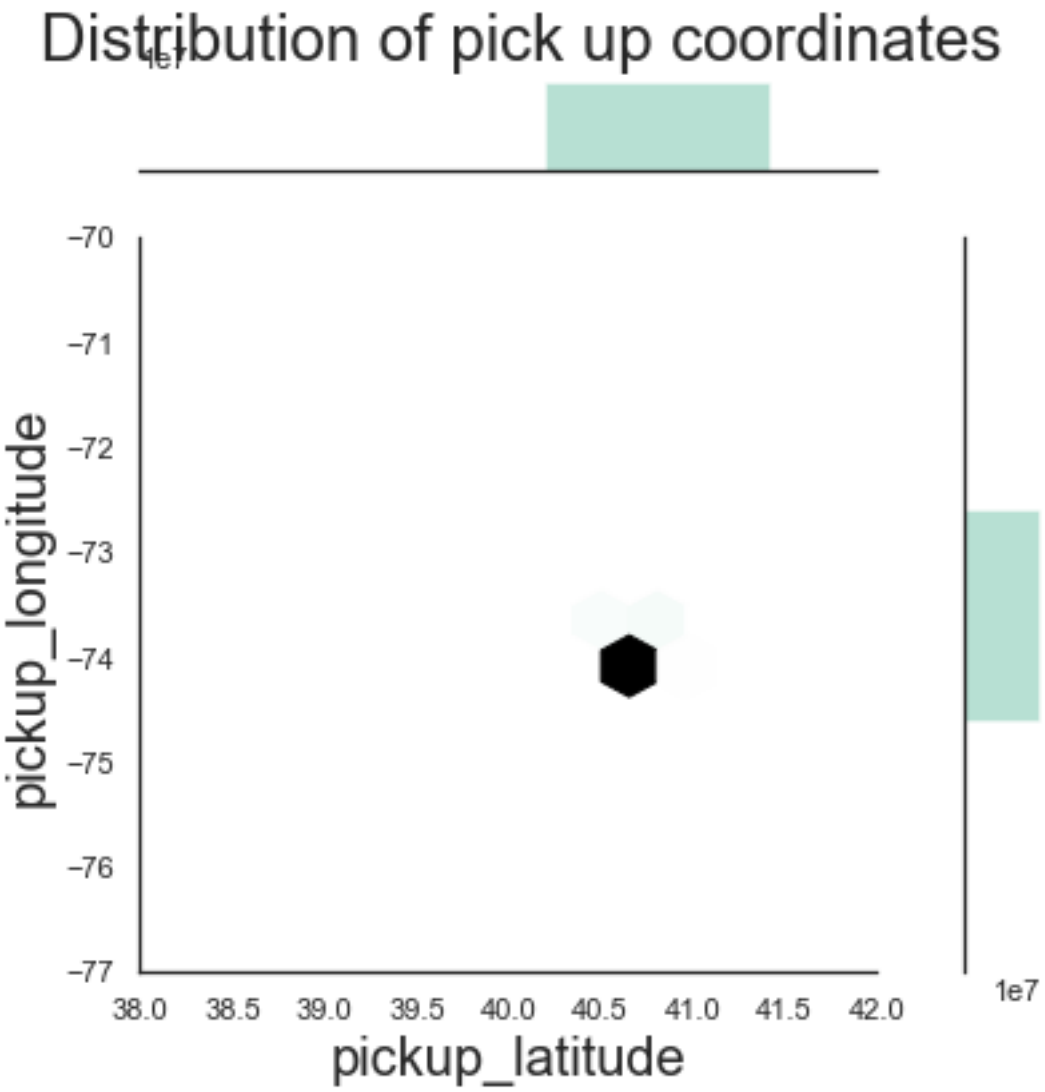
```
ax[0].set_title("Distribution of duration ")
ax[1].set_title("Distribution of total amount ")
ax[2].set_title("Distribution of fare amount ")
ax[3].set_title("Distribution of trip distance ")
ax[4].set_title("Distribution of income per minute")

ax[0].set_xlabel("Duration (min)")
ax[1].set_xlabel("Total amount (USD)")
ax[2].set_xlabel("Fare amount (USD)")
ax[3].set_xlabel("Trip distance (mile)")
ax[4].set_xlabel("Income per minute (USD/min)")

ax[0].set_ylabel("Density")
ax[1].set_ylabel("Density")
ax[2].set_ylabel("Density")
ax[3].set_ylabel("Density")
ax[4].set_ylabel("Density")

fig.set_figheight(5)
fig.set_figwidth(15)
fig.set_figwidth(25)
fig.set_figwidth(35)
fig.set_figwidth(45)


fig.savefig('plots/distibution_narrowed_value.png')
```

[17]:
```
plt.subplot(231)
sns.distplot(df['duration'], kde = False, label = "duration", color ="blue")
plt.title("Distribution\n of duration")
plt.xlabel('Duration (min)')
plt.ylabel("Density")

plt.subplot(232)
sns.distplot(df['total_amount'], kde = False, label = "total_amount", color␣
 ↪="blue")
plt.title("Distribution\n of total amount")
plt.xlabel('Total amount (USD)')
plt.ylabel("Density")
```

```
plt.subplot(233)
sns.distplot(df['fare_amount'], kde = False, label = 'fare_amount', color␣
 ↪="blue")
plt.title("Distribution\n of fare amount")
plt.xlabel('Fare amount (USD)')
plt.ylabel("Density")

plt.subplot(234)
sns.distplot(df['trip_distance'], kde = False, label = "duration", color␣
 ↪="blue")
plt.title("Distribution\n of trip distance")
plt.xlabel('Trip distance (mile)')
plt.ylabel("Density")

plt.subplot(235)
sns.distplot(df['income/duration'], kde = False, label = "duration", color␣
 ↪="blue")
plt.title("Distribution of\n revenue")
plt.xlabel('Revenue (USD/min)')
plt.ylabel("Density")

plt.tight_layout()
plt.show()
```



[18]:
```
p = sns.jointplot(x='pickup_latitude',y='pickup_longitude' , data=␣
↪df,kind="hex",
                  color="#4CB391", xlim=(38,42), ylim=(-77, -70), gridsize=200)

p.fig.suptitle("Distribution of pick up coordinates")
p.fig.tight_layout()
```



[19]:
```
df = df.loc[(df["pickup_longitude"] < -73) & (df["pickup_longitude"] > -74.5) &
            (df["pickup_latitude"] > 40.5) & (df["pickup_latitude"] < 41) &
            (df["dropoff_longitude"] < -73) & (df["dropoff_longitude"] > -74.5)␣
↪&
```

```
            (df["dropoff_latitude"] > 40.5) & (df["dropoff_latitude"] < 41)]
```

[20]:
```
df.reset_index(inplace=True, drop=True)
df
```

[20]:

```
          VendorID tpep_pickup_datetime tpep_dropoff_datetime  \
0                2  2016-01-01 00:00:00   2016-01-01 00:18:30
1                2  2016-01-01 00:00:00   2016-01-01 00:26:45
2                1  2016-01-01 00:00:01   2016-01-01 00:11:55
3                1  2016-01-01 00:00:02   2016-01-01 00:11:14
4                2  2016-01-01 00:00:02   2016-01-01 00:11:08
…              …                   …                   …
10624517         2  2016-01-31 21:28:59   2016-01-31 22:01:58
10624518         2  2016-01-31 22:36:41   2016-01-31 22:45:04
10624519         2  2016-01-31 22:53:00   2016-01-31 22:59:37
10624520         2  2016-01-31 23:00:11   2016-01-31 23:12:08
10624521         2  2016-01-31 23:30:32   2016-01-31 23:38:18

          passenger_count  trip_distance  pickup_longitude  pickup_latitude  \
0                       2           5.52        -73.980118        40.743050
1                       2           7.45        -73.994057        40.719990
2                       1           1.20        -73.979424        40.744614
3                       1           6.00        -73.947151        40.791046
4                       1           3.21        -73.998344        40.723896
…                     …             …                 …                …
10624517                1           7.83        -74.002953        40.750481
10624518                1           2.50        -74.009277        40.717049
10624519                1           1.68        -74.003578        40.750751
10624520                1           2.65        -74.002159        40.734852
10624521                1           2.20        -74.003578        40.751011

          RatecodeID store_and_fwd_flag  dropoff_longitude  …  tip_amount  \
0                  1                  N        -73.913490  …        0.00
1                  1                  N        -73.966362  …        0.00
2                  1                  N        -73.992035  …        0.00
3                  1                  N        -73.920769  …        0.00
4                  1                  N        -73.995850  …        0.00
…                …                  …                  …  …          …
10624517           1                  N        -73.958153  …        5.00
10624518           1                  N        -73.994637  …        2.16
10624519           1                  N        -74.002159  …        1.00
10624520           1                  N        -73.999680  …        1.00
10624521           1                  N        -73.982651  …        0.00

          tolls_amount  improvement_surcharge  total_amount   duration  \
0                  0.0                    0.3         20.30  18.500000
1                  0.0                    0.3         27.30  26.750000
```

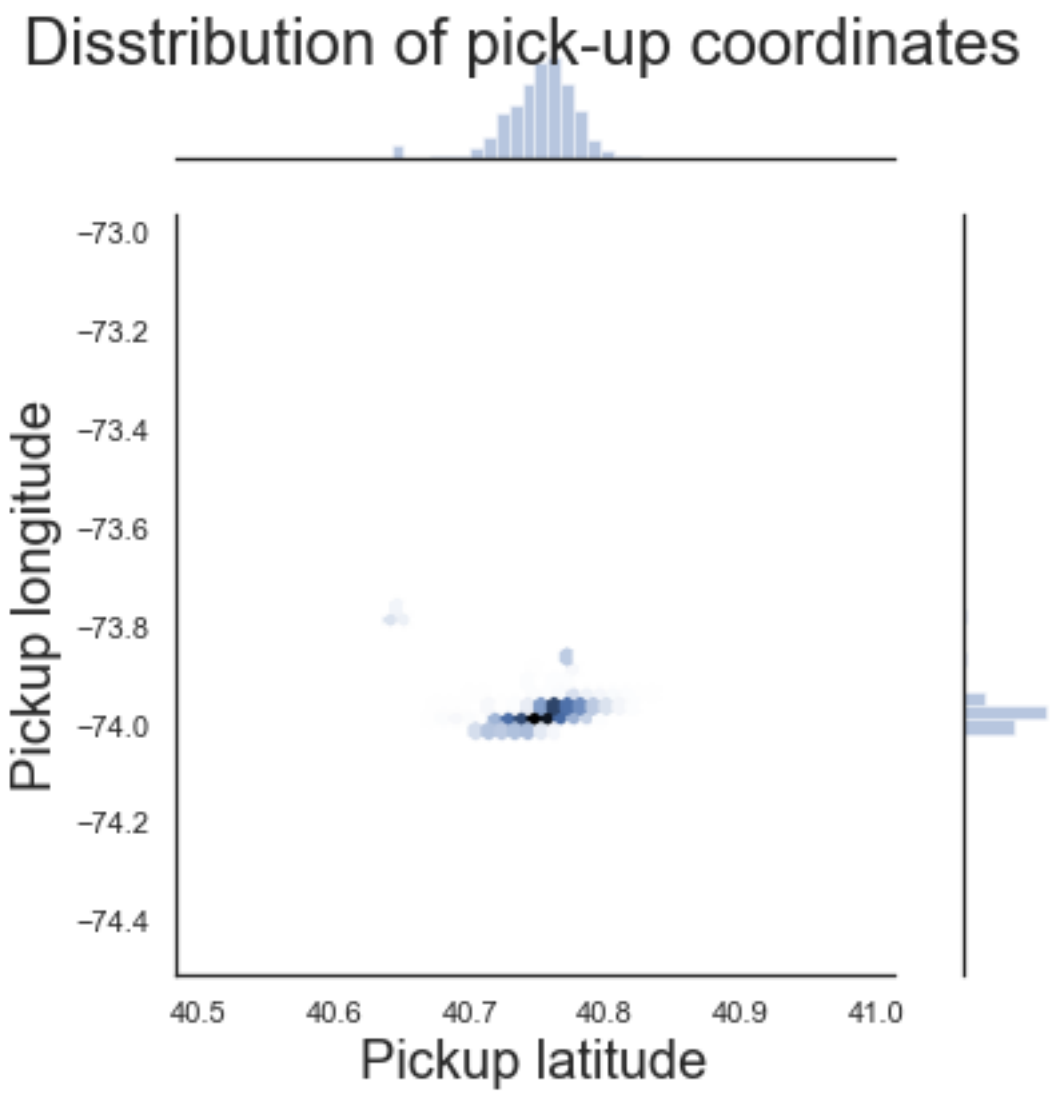
```
2                  0.0                    0.3         10.30  11.900000
3                  0.0                    0.3         19.30  11.200000
4                  0.0                    0.3         12.80  11.100000
…                  …                      …           …          …
10624517           0.0                    0.3         35.30  32.983333
10624518           0.0                    0.3         12.96   8.383333
10624519           0.0                    0.3          9.30   6.616667
10624520           0.0                    0.3         13.30  11.950000
10624521           0.0                    0.3          9.80   7.766667

          start_hour  start_date  weather  income  income/duration
0                  0  2016-01-01   remain   19.00         1.027027
1                  0  2016-01-01   remain   26.00         0.971963
2                  0  2016-01-01   remain    9.00         0.756303
3                  0  2016-01-01   remain   18.00         1.607143
4                  0  2016-01-01   remain   11.50         1.036036
…                  …           …       …       …              …
10624517          21  2016-01-31   remain   34.00         1.030824
10624518          22  2016-01-31   remain   11.66         1.390855
10624519          22  2016-01-31   remain    8.00         1.209068
10624520          23  2016-01-31   remain   12.00         1.004184
10624521          23  2016-01-31   remain    8.50         1.094421

[10624522 rows x 25 columns]
```

[21]:
```
df.describe()
```

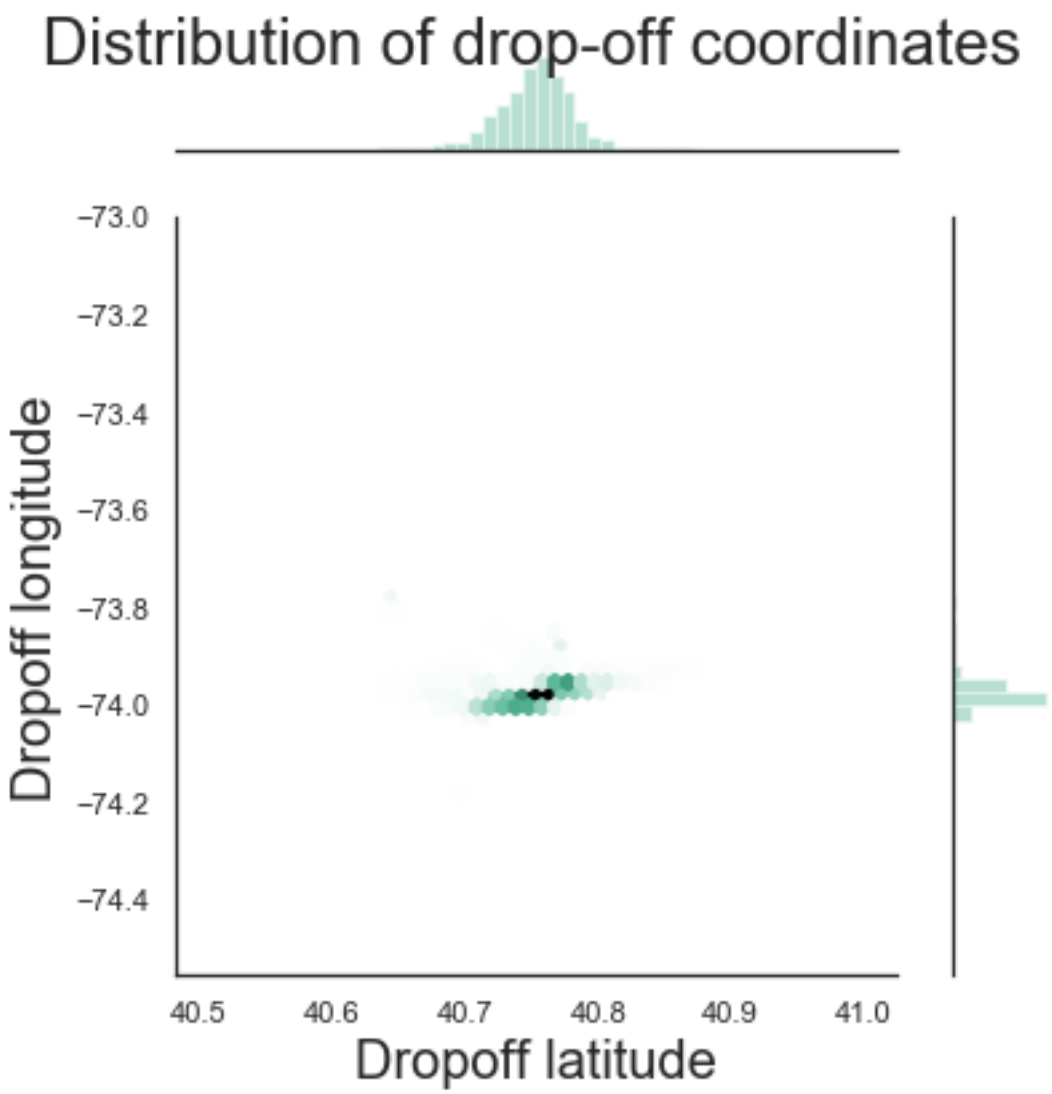
[21]:
```
           VendorID  passenger_count  trip_distance  pickup_longitude  \
count  1.062452e+07     1.062452e+07   1.062452e+07      1.062452e+07
mean   1.538963e+00     1.675594e+00   2.888937e+00     -7.397336e+01
std    4.984796e-01     1.329766e+00   3.503629e+00      3.796886e-02
min    1.000000e+00     0.000000e+00   1.000000e-02     -7.443886e+01
25%    1.000000e+00     1.000000e+00   1.000000e+00     -7.399164e+01
50%    2.000000e+00     1.000000e+00   1.690000e+00     -7.398169e+01
75%    2.000000e+00     2.000000e+00   3.100000e+00     -7.396725e+01
max    2.000000e+00     9.000000e+00   3.000000e+01     -7.303445e+01

       pickup_latitude    RatecodeID  dropoff_longitude  dropoff_latitude  \
count     1.062452e+07  1.062452e+07       1.062452e+07      1.062452e+07
mean      4.075104e+01  1.021407e+00      -7.397358e+01      4.075202e+01
std       2.779560e-02  1.447374e-01       3.365993e-02      3.144207e-02
min       4.050597e+01  1.000000e+00      -7.448333e+01      4.050733e+01
25%       4.073759e+01  1.000000e+00      -7.399119e+01      4.073634e+01
50%       4.075442e+01  1.000000e+00      -7.397976e+01      4.075478e+01
75%       4.076839e+01  1.000000e+00      -7.396313e+01      4.076998e+01
max       4.098892e+01  2.000000e+00      -7.306847e+01      4.099876e+01
```

```
       payment_type   fare_amount         extra       mta_tax    tip_amount  \
count  1.062452e+07  1.062452e+07  1.062452e+07  1.062452e+07  1.062452e+07
mean   1.339203e+00  1.229613e+01  3.143274e-01  4.991960e-01  1.723470e+00
std    4.734389e-01  9.823765e+00  3.656012e-01  2.003404e-02  2.243046e+00
min    1.000000e+00  1.000000e-02  0.000000e+00  0.000000e+00  0.000000e+00
25%    1.000000e+00  6.500000e+00  0.000000e+00  5.000000e-01  0.000000e+00
50%    1.000000e+00  9.000000e+00  0.000000e+00  5.000000e-01  1.260000e+00
75%    2.000000e+00  1.400000e+01  5.000000e-01  5.000000e-01  2.320000e+00
max    2.000000e+00  8.000000e+01  8.500000e+00  8.900000e-01  8.800000e+01

       tolls_amount  improvement_surcharge  total_amount      duration  \
count  1.062452e+07           1.062452e+07  1.062452e+07  1.062452e+07
mean   2.763625e-01           2.999957e-01  1.540948e+01  1.319822e+01
std    1.280491e+00           1.138437e-03  1.213060e+01  1.011057e+01
min    0.000000e+00           0.000000e+00  3.100000e-01  2.666667e-01
25%    0.000000e+00           3.000000e-01  8.300000e+00  6.416667e+00
50%    0.000000e+00           3.000000e-01  1.162000e+01  1.051667e+01
75%    0.000000e+00           3.000000e-01  1.716000e+01  1.688333e+01
max    9.782000e+01           3.000000e-01  1.000000e+02  1.497833e+02

         start_hour        income  income/duration
count  1.062452e+07  1.062452e+07     1.062452e+07
mean   1.355524e+01  1.401960e+01     1.148759e+00
std    6.386805e+00  1.131511e+01     3.862012e-01
min    0.000000e+00  1.000000e-02     8.424600e-05
25%    9.000000e+00  7.360000e+00     8.974359e-01
50%    1.400000e+01  1.046000e+01     1.063235e+00
75%    1.900000e+01  1.595000e+01     1.295681e+00
max    2.300000e+01  9.950000e+01     5.000000e+00
```

[22]:
```
p = sns.jointplot(x='pickup_latitude',y='pickup_longitude' , data=␣
 ↪df,kind="hex")
p.ax_joint.set_xlabel('Pickup latitude')
p.ax_joint.set_ylabel('Pickup longitude')
p.fig.suptitle("Disstribution of pick-up coordinates ")
p.fig.tight_layout()


p.fig.savefig('plots/disstribution of pick up coordinates.png')
```



[23]:
```
p = sns.jointplot(x='dropoff_latitude',y='dropoff_longitude' , data=␣
↪df,kind="hex",color="#4CB391")

p.ax_joint.set_xlabel('Dropoff latitude')
p.ax_joint.set_ylabel('Dropoff longitude')
p.fig.suptitle("Distribution of drop-off coordinates")
p.fig.tight_layout()
p.fig.savefig('plots/disstribution of drop off coordinates.png')
```



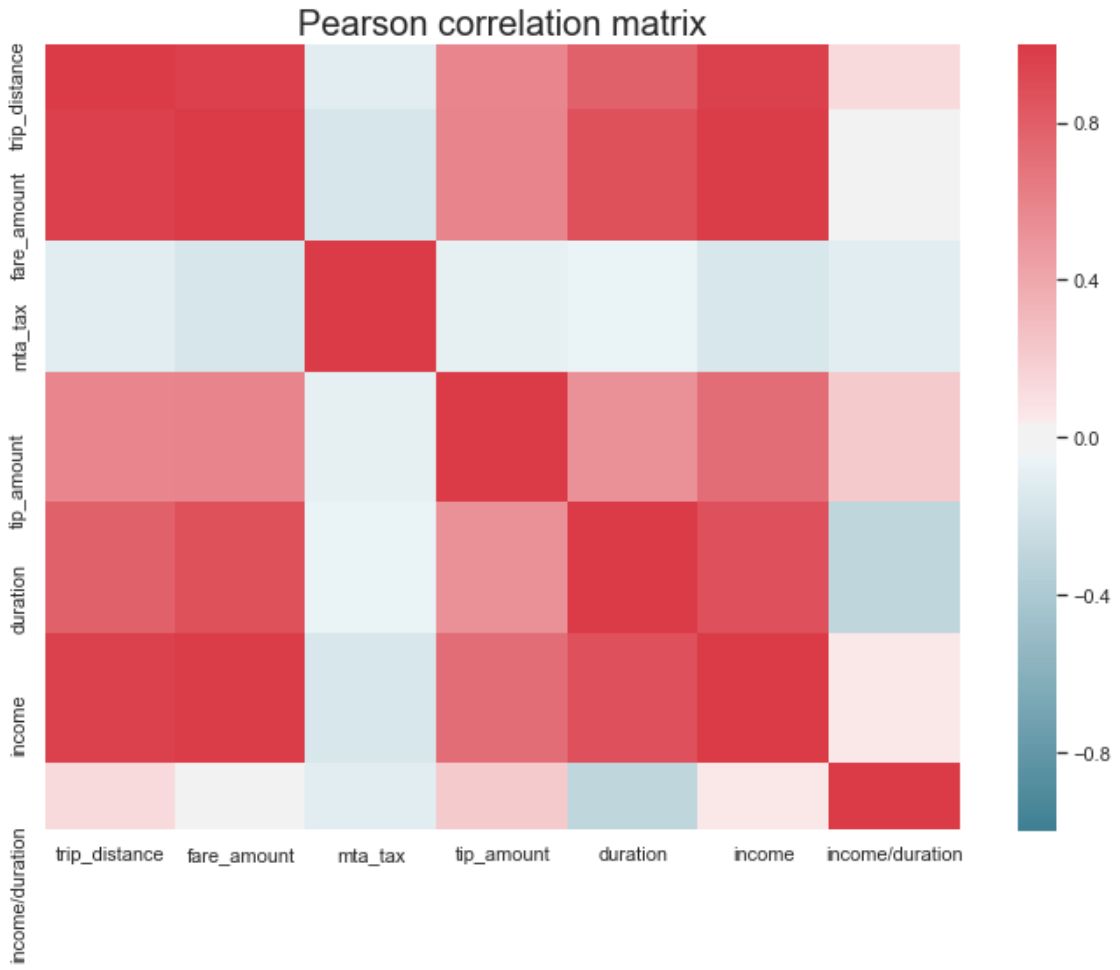
[24]:
```
df.drop(['VendorID','tpep_pickup_datetime',␣
 ↪'tpep_dropoff_datetime','store_and_fwd_flag', 'improvement_surcharge',
        'start_date'], axis=1, inplace = True)
```

[ ]:
```

```

2 Change data type

[25]:
```
df['start_hour'] = df['start_hour'].astype('category')
df['payment_type'] = df['payment_type'].astype('category')
df['RatecodeID'] = df['RatecodeID'].astype('category')
df['weather'] = df['weather'].astype('category')
```

[26]:
```
corr_attr =['trip_distance', 'payment_type', 'fare_amount', 'mta_tax',
 'tip_amount', 'duration', 'start_hour', 'weather','income','income/duration']
corr = df[corr_attr].corr()
sns.heatmap(corr,cmap = sns.diverging_palette(220, 10, as_cmap=True),␣
 ↪square=True, center=0, vmin=-1, vmax=1)
plt.title('Pearson correlation matrix')
plt.savefig('plots/correlation.png')
plt.show()
```



[ ]:

3 Sampling

[27]:
```
sub_df1 = df.sample(n=1000000, random_state=100)
sub_df2 = df.sample(n=1000000, random_state=50)
sub_df3 = df.sample(n=1000000, random_state=30)
```

[28]:
```
card_df = df.loc[(df["payment_type"] == 1)]
cash_df = df.loc[(df["payment_type"] == 2)]
preci_df =  df.loc[(df["weather"] != 'remain')]
remain_df = df.loc[(df["weather"] == 'remain')]
stand_df = df.loc[(df["RatecodeID"] == 1)]
other_df = df.loc[(df["RatecodeID"] == 2)]
```

[29]:
```
df.columns
```

[29]:
```
Index(['passenger_count', 'trip_distance', 'pickup_longitude',
       'pickup_latitude', 'RatecodeID', 'dropoff_longitude',
       'dropoff_latitude', 'payment_type', 'fare_amount', 'extra', 'mta_tax',
       'tip_amount', 'tolls_amount', 'total_amount', 'duration', 'start_hour',
       'weather', 'income', 'income/duration'],
      dtype='object')
```

[ ]:
```
```

4 What is related to tip amount

[30]:
```
fit = ols(formula="tip_amount ~ trip_distance + duration  +payment_type +␣
 ↪start_hour  + weather + duration * start_hour + duration * weather",
        data=sub_df1).fit()
print(fit.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:             tip_amount   R-squared:                       0.623
Model:                            OLS   Adj. R-squared:                  0.623
Method:                 Least Squares   F-statistic:                 3.235e+04
Date:                Fri, 04 Sep 2020   Prob (F-statistic):               0.00
Time:                        20:39:32   Log-Likelihood:            -1.7400e+06
No. Observations:             1000000   AIC:                         3.480e+06
Df Residuals:                  999948   BIC:                         3.481e+06
Df Model:                          51
Covariance Type:            nonrobust
==============================================================================
==============
                                 coef    std err          t      P>|t|
[0.025      0.975]
```

```
------------------------------------------------------------------------------
--------------
Intercept                      1.3750      0.017     82.495      0.000
1.342       1.408
payment_type[T.2]             -2.4419      0.003   -833.912      0.000
-2.448      -2.436
start_hour[T.1]                0.0979      0.020      4.944      0.000
0.059       0.137
start_hour[T.2]                0.1737      0.022      8.024      0.000
0.131       0.216
start_hour[T.3]                0.2309      0.024      9.812      0.000
0.185       0.277
start_hour[T.4]                0.3003      0.026     11.567      0.000
0.249       0.351
start_hour[T.5]               -0.0813      0.025     -3.211      0.001
-0.131      -0.032
start_hour[T.6]               -0.1434      0.019     -7.705      0.000
-0.180      -0.107
start_hour[T.7]               -0.0834      0.017     -4.960      0.000
-0.116      -0.050
start_hour[T.8]               -0.1438      0.017     -8.617      0.000
-0.177      -0.111
start_hour[T.9]               -0.1555      0.017     -9.215      0.000
-0.189      -0.122
start_hour[T.10]              -0.0996      0.017     -5.913      0.000
-0.133      -0.067
start_hour[T.11]              -0.0894      0.017     -5.317      0.000
-0.122      -0.056
start_hour[T.12]              -0.0435      0.017     -2.619      0.009
-0.076      -0.011
start_hour[T.13]               0.0180      0.017      1.084      0.278
-0.015       0.051
start_hour[T.14]               0.0095      0.016      0.583      0.560
-0.022       0.041
start_hour[T.15]               0.0570      0.016      3.569      0.000
0.026       0.088
start_hour[T.16]               0.0489      0.016      3.024      0.002
0.017       0.081
start_hour[T.17]               0.0359      0.016      2.263      0.024
0.005       0.067
start_hour[T.18]               0.0183      0.016      1.155      0.248
-0.013       0.049
start_hour[T.19]              -0.0120      0.016     -0.745      0.456
-0.044       0.020
start_hour[T.20]              -0.0189      0.016     -1.151      0.250
-0.051       0.013
start_hour[T.21]              -0.0351      0.017     -2.115      0.034
-0.068      -0.003
```

```
start_hour[T.22]              -0.0265      0.017     -1.571      0.116
-0.059       0.007
start_hour[T.23]               0.0038      0.018      0.215      0.830
-0.031       0.038
weather[T.remain]             -0.0365      0.011     -3.179      0.001
-0.059      -0.014
trip_distance                  0.2935      0.001    439.276      0.000
0.292       0.295
duration                       0.0196      0.001     17.933      0.000
0.017       0.022
duration:start_hour[T.1]      -0.0144      0.001    -10.871      0.000
-0.017      -0.012
duration:start_hour[T.2]      -0.0246      0.001    -16.615      0.000
-0.027      -0.022
duration:start_hour[T.3]      -0.0322      0.002    -20.246      0.000
-0.035      -0.029
duration:start_hour[T.4]      -0.0321      0.002    -18.811      0.000
-0.035      -0.029
duration:start_hour[T.5]       0.0112      0.002      6.833      0.000
0.008       0.014
duration:start_hour[T.6]       0.0089      0.001      7.399      0.000
0.007       0.011
duration:start_hour[T.7]       0.0026      0.001      2.455      0.014
0.001       0.005
duration:start_hour[T.8]       0.0113      0.001     10.770      0.000
0.009       0.013
duration:start_hour[T.9]       0.0151      0.001     14.149      0.000
0.013       0.017
duration:start_hour[T.10]      0.0133      0.001     12.498      0.000
0.011       0.015
duration:start_hour[T.11]      0.0141      0.001     13.256      0.000
0.012       0.016
duration:start_hour[T.12]      0.0102      0.001      9.654      0.000
0.008       0.012
duration:start_hour[T.13]      0.0046      0.001      4.395      0.000
0.003       0.007
duration:start_hour[T.14]      0.0052      0.001      5.181      0.000
0.003       0.007
duration:start_hour[T.15]      0.0013      0.001      1.301      0.193
-0.001       0.003
duration:start_hour[T.16]      0.0055      0.001      5.575      0.000
0.004       0.007
duration:start_hour[T.17]      0.0048      0.001      4.870      0.000
0.003       0.007
duration:start_hour[T.18]      0.0050      0.001      4.993      0.000
0.003       0.007
duration:start_hour[T.19]      0.0063      0.001      5.967      0.000
0.004       0.008
```

```
duration:start_hour[T.20]      0.0043      0.001      3.992      0.000
0.002       0.006
duration:start_hour[T.21]      0.0076      0.001      6.892      0.000
0.005       0.010
duration:start_hour[T.22]      0.0065      0.001      5.858      0.000
0.004       0.009
duration:start_hour[T.23]      0.0018      0.001      1.553      0.120
-0.000       0.004
duration:weather[T.remain]     0.0045      0.001      6.221      0.000
0.003       0.006
==============================================================================
Omnibus:                   622805.953   Durbin-Watson:                   2.000
Prob(Omnibus):                  0.000   Jarque-Bera (JB):        145312046.952
Skew:                           1.901   Prob(JB):                         0.00
Kurtosis:                      61.932   Cond. No.                     1.12e+03
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 1.12e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

[31]:
```
fit = ols(formula="tip_amount ~ trip_distance + duration  +payment_type +␣
 ↪start_hour  + weather + duration * start_hour + duration * weather",
          data=sub_df2).fit()
print(fit.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:             tip_amount   R-squared:                       0.626
Model:                            OLS   Adj. R-squared:                  0.626
Method:                 Least Squares   F-statistic:                 3.278e+04
Date:                Fri, 04 Sep 2020   Prob (F-statistic):               0.00
Time:                        20:39:39   Log-Likelihood:            -1.7348e+06
No. Observations:             1000000   AIC:                         3.470e+06
Df Residuals:                  999948   BIC:                         3.470e+06
Df Model:                          51
Covariance Type:            nonrobust
==============================================================================
==============
                                 coef    std err          t      P>|t|
[0.025      0.975]
------------------------------------------------------------------------------
--------------
Intercept                      1.3755      0.017     83.172      0.000
1.343       1.408
payment_type[T.2]             -2.4370      0.003   -836.442      0.000
```

```
-2.443      -2.431
start_hour[T.1]              0.0572      0.020      2.905      0.004
0.019       0.096
start_hour[T.2]              0.1291      0.021      6.033      0.000
0.087       0.171
start_hour[T.3]              0.1337      0.024      5.682      0.000
0.088       0.180
start_hour[T.4]              0.2179      0.026      8.423      0.000
0.167       0.269
start_hour[T.5]             -0.0485      0.025     -1.924      0.054
-0.098       0.001
start_hour[T.6]             -0.1698      0.019     -9.135      0.000
-0.206      -0.133
start_hour[T.7]             -0.1338      0.017     -8.040      0.000
-0.166      -0.101
start_hour[T.8]             -0.1768      0.017    -10.695      0.000
-0.209      -0.144
start_hour[T.9]             -0.1775      0.017    -10.589      0.000
-0.210      -0.145
start_hour[T.10]            -0.1067      0.017     -6.393      0.000
-0.139      -0.074
start_hour[T.11]            -0.0648      0.017     -3.872      0.000
-0.098      -0.032
start_hour[T.12]            -0.0270      0.017     -1.636      0.102
-0.059       0.005
start_hour[T.13]             0.0170      0.017      1.031      0.303
-0.015       0.049
start_hour[T.14]             0.0168      0.016      1.040      0.298
-0.015       0.049
start_hour[T.15]            -0.0011      0.016     -0.067      0.946
-0.032       0.030
start_hour[T.16]             0.0489      0.016      3.044      0.002
0.017       0.080
start_hour[T.17]             0.0255      0.016      1.612      0.107
-0.005       0.056
start_hour[T.18]             0.0106      0.016      0.676      0.499
-0.020       0.042
start_hour[T.19]            -0.0188      0.016     -1.165      0.244
-0.050       0.013
start_hour[T.20]            -0.0759      0.016     -4.630      0.000
-0.108      -0.044
start_hour[T.21]            -0.0774      0.016     -4.694      0.000
-0.110      -0.045
start_hour[T.22]            -0.0421      0.017     -2.512      0.012
-0.075      -0.009
start_hour[T.23]            -0.0274      0.017     -1.566      0.117
-0.062       0.007
weather[T.remain]           -0.0200      0.011     -1.757      0.079
```

```
-0.042       0.002
trip_distance                  0.2956      0.001    445.463      0.000
0.294       0.297
duration                       0.0197      0.001     18.115      0.000
0.018       0.022
duration:start_hour[T.1]      -0.0102      0.001     -7.689      0.000
-0.013      -0.008
duration:start_hour[T.2]      -0.0217      0.001    -14.757      0.000
-0.025      -0.019
duration:start_hour[T.3]      -0.0255      0.002    -15.906      0.000
-0.029      -0.022
duration:start_hour[T.4]      -0.0274      0.002    -16.282      0.000
-0.031      -0.024
duration:start_hour[T.5]       0.0064      0.002      3.872      0.000
0.003       0.010
duration:start_hour[T.6]       0.0119      0.001      9.932      0.000
0.010       0.014
duration:start_hour[T.7]       0.0070      0.001      6.612      0.000
0.005       0.009
duration:start_hour[T.8]       0.0144      0.001     13.846      0.000
0.012       0.016
duration:start_hour[T.9]       0.0177      0.001     16.667      0.000
0.016       0.020
duration:start_hour[T.10]      0.0150      0.001     14.281      0.000
0.013       0.017
duration:start_hour[T.11]      0.0114      0.001     10.755      0.000
0.009       0.013
duration:start_hour[T.12]      0.0087      0.001      8.266      0.000
0.007       0.011
duration:start_hour[T.13]      0.0040      0.001      3.822      0.000
0.002       0.006
duration:start_hour[T.14]      0.0045      0.001      4.487      0.000
0.003       0.006
duration:start_hour[T.15]      0.0065      0.001      6.677      0.000
0.005       0.008
duration:start_hour[T.16]      0.0058      0.001      5.844      0.000
0.004       0.008
duration:start_hour[T.17]      0.0063      0.001      6.468      0.000
0.004       0.008
duration:start_hour[T.18]      0.0058      0.001      5.729      0.000
0.004       0.008
duration:start_hour[T.19]      0.0073      0.001      6.866      0.000
0.005       0.009
duration:start_hour[T.20]      0.0097      0.001      8.989      0.000
0.008       0.012
duration:start_hour[T.21]      0.0116      0.001     10.576      0.000
0.009       0.014
duration:start_hour[T.22]      0.0073      0.001      6.651      0.000
```

```
0.005       0.009
duration:start_hour[T.23]      0.0044      0.001      3.892      0.000
0.002       0.007
duration:weather[T.remain]     0.0024      0.001      3.249      0.001
0.001       0.004
==============================================================================
Omnibus:                   530326.393   Durbin-Watson:                   1.997
Prob(Omnibus):                  0.000   Jarque-Bera (JB):         74084741.181
Skew:                           1.544   Prob(JB):                         0.00
Kurtosis:                      45.054   Cond. No.                     1.12e+03
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 1.12e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

[32]:
```
fit = ols(formula="tip_amount ~ trip_distance + duration  +payment_type +␣
 ↪start_hour  + weather + duration * start_hour + duration * weather",
        data=sub_df3).fit()
print(fit.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:             tip_amount   R-squared:                       0.622
Model:                            OLS   Adj. R-squared:                  0.622
Method:                 Least Squares   F-statistic:                 3.225e+04
Date:                Fri, 04 Sep 2020   Prob (F-statistic):               0.00
Time:                        20:39:46   Log-Likelihood:            -1.7395e+06
No. Observations:             1000000   AIC:                         3.479e+06
Df Residuals:                  999948   BIC:                         3.480e+06
Df Model:                          51
Covariance Type:            nonrobust
==============================================================================
==============
                                 coef    std err          t      P>|t|
[0.025      0.975]
------------------------------------------------------------------------------
--------------
Intercept                      1.3999      0.017     84.202      0.000
1.367       1.432
payment_type[T.2]             -2.4413      0.003   -833.862      0.000
-2.447      -2.436
start_hour[T.1]                0.0751      0.020      3.775      0.000
0.036       0.114
start_hour[T.2]                0.1229      0.022      5.697      0.000
0.081       0.165
```

```
start_hour[T.3]              0.1560      0.024      6.576      0.000
0.110       0.202
start_hour[T.4]              0.2289      0.026      8.871      0.000
0.178       0.279
start_hour[T.5]             -0.0020      0.025     -0.077      0.938
-0.051       0.048
start_hour[T.6]             -0.1406      0.019     -7.557      0.000
-0.177      -0.104
start_hour[T.7]             -0.1495      0.017     -8.895      0.000
-0.182      -0.117
start_hour[T.8]             -0.1527      0.017     -9.146      0.000
-0.185      -0.120
start_hour[T.9]             -0.1564      0.017     -9.292      0.000
-0.189      -0.123
start_hour[T.10]            -0.0727      0.017     -4.324      0.000
-0.106      -0.040
start_hour[T.11]            -0.0774      0.017     -4.611      0.000
-0.110      -0.044
start_hour[T.12]            -0.0186      0.017     -1.116      0.265
-0.051       0.014
start_hour[T.13]             0.0204      0.017      1.227      0.220
-0.012       0.053
start_hour[T.14]             0.0118      0.016      0.727      0.468
-0.020       0.044
start_hour[T.15]             0.0310      0.016      1.935      0.053
-0.000       0.062
start_hour[T.16]             0.0343      0.016      2.120      0.034
0.003       0.066
start_hour[T.17]             0.0325      0.016      2.036      0.042
0.001       0.064
start_hour[T.18]             0.0305      0.016      1.924      0.054
-0.001       0.061
start_hour[T.19]            -0.0596      0.016     -3.690      0.000
-0.091      -0.028
start_hour[T.20]            -0.0377      0.016     -2.292      0.022
-0.070      -0.005
start_hour[T.21]            -0.0947      0.017     -5.680      0.000
-0.127      -0.062
start_hour[T.22]            -0.0405      0.017     -2.405      0.016
-0.074      -0.008
start_hour[T.23]             0.0069      0.017      0.394      0.694
-0.027       0.041
weather[T.remain]           -0.0515      0.011     -4.507      0.000
-0.074      -0.029
trip_distance                0.2929      0.001    438.074      0.000
0.292       0.294
duration                     0.0158      0.001     14.499      0.000
0.014       0.018
```

```
duration:start_hour[T.1]      -0.0106      0.001     -7.872      0.000
-0.013      -0.008
duration:start_hour[T.2]      -0.0178      0.001    -12.055      0.000
-0.021      -0.015
duration:start_hour[T.3]      -0.0256      0.002    -15.903      0.000
-0.029      -0.022
duration:start_hour[T.4]      -0.0255      0.002    -15.149      0.000
-0.029      -0.022
duration:start_hour[T.5]       0.0039      0.002      2.387      0.017
0.001       0.007
duration:start_hour[T.6]       0.0104      0.001      8.741      0.000
0.008       0.013
duration:start_hour[T.7]       0.0105      0.001      9.789      0.000
0.008       0.013
duration:start_hour[T.8]       0.0135      0.001     12.823      0.000
0.011       0.016
duration:start_hour[T.9]       0.0171      0.001     15.960      0.000
0.015       0.019
duration:start_hour[T.10]      0.0125      0.001     11.780      0.000
0.010       0.015
duration:start_hour[T.11]      0.0132      0.001     12.381      0.000
0.011       0.015
duration:start_hour[T.12]      0.0093      0.001      8.817      0.000
0.007       0.011
duration:start_hour[T.13]      0.0055      0.001      5.224      0.000
0.003       0.008
duration:start_hour[T.14]      0.0060      0.001      5.991      0.000
0.004       0.008
duration:start_hour[T.15]      0.0043      0.001      4.330      0.000
0.002       0.006
duration:start_hour[T.16]      0.0082      0.001      8.283      0.000
0.006       0.010
duration:start_hour[T.17]      0.0064      0.001      6.535      0.000
0.005       0.008
duration:start_hour[T.18]      0.0060      0.001      5.934      0.000
0.004       0.008
duration:start_hour[T.19]      0.0121      0.001     11.424      0.000
0.010       0.014
duration:start_hour[T.20]      0.0082      0.001      7.533      0.000
0.006       0.010
duration:start_hour[T.21]      0.0146      0.001     13.182      0.000
0.012       0.017
duration:start_hour[T.22]      0.0089      0.001      8.074      0.000
0.007       0.011
duration:start_hour[T.23]      0.0022      0.001      1.909      0.056
-5.88e-05       0.004
duration:weather[T.remain]     0.0062      0.001      8.519      0.000
0.005       0.008
```

```
==============================================================================
Omnibus:                   639753.491   Durbin-Watson:                   2.000
Prob(Omnibus):                  0.000   Jarque-Bera (JB):        156474629.192
Skew:                           1.982   Prob(JB):                         0.00
Kurtosis:                      64.153   Cond. No.                     1.12e+03
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 1.12e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

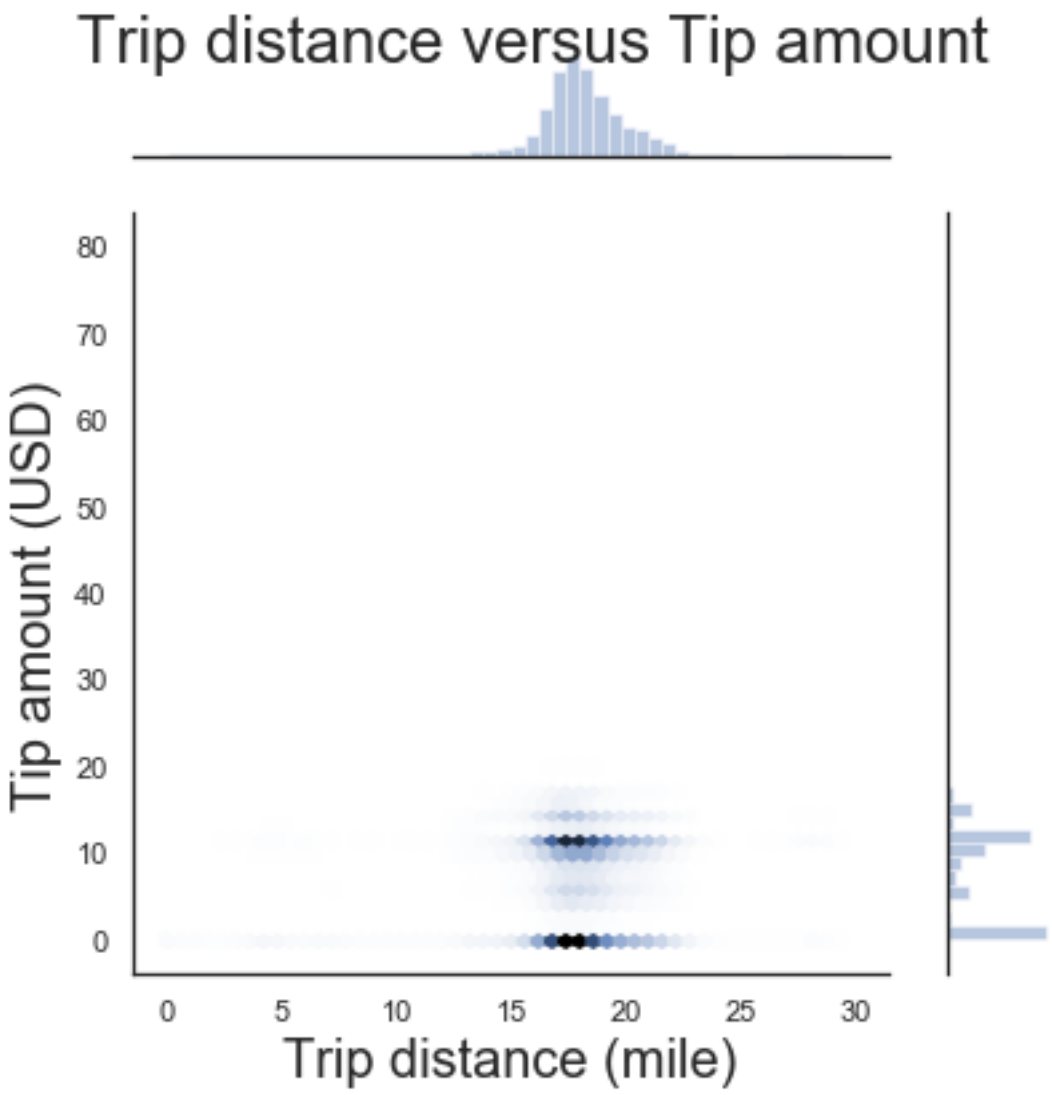
[33]:
```
fit = ols(formula="tip_amount ~ trip_distance +payment_type",
          data=df).fit()
print(fit.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:             tip_amount   R-squared:                       0.613
Model:                            OLS   Adj. R-squared:                  0.613
Method:                 Least Squares   F-statistic:                 8.406e+06
Date:                Fri, 04 Sep 2020   Prob (F-statistic):               0.00
Time:                        20:39:51   Log-Likelihood:            -1.8619e+07
No. Observations:            10624522   AIC:                         3.724e+07
Df Residuals:                10624519   BIC:                         3.724e+07
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
=====
                       coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
-----
Intercept              1.5257      0.001   2367.297      0.000       1.524
1.527
payment_type[T.2]     -2.4566      0.001  -2711.455      0.000      -2.458
-2.455
trip_distance          0.3569      0.000   2915.138      0.000       0.357
0.357
==============================================================================
Omnibus:                   6729583.760   Durbin-Watson:                   1.979
Prob(Omnibus):                  0.000   Jarque-Bera (JB):       1537681233.635
Skew:                           1.965   Prob(JB):                         0.00
Kurtosis:                      61.805   Cond. No.                         10.6
==============================================================================

Warnings:
```

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

[ ]:

[34]:
```
p = sns.jointplot(x='trip_distance',y= 'tip_amount',data= other_df, kind="hex",␣
↪color="b")
p.ax_joint.set_xlabel('Trip distance (mile)')
p.ax_joint.set_ylabel('Tip amount (USD)')
p.fig.suptitle("Trip distance versus Tip amount")
p.fig.tight_layout()
```



[ ]:

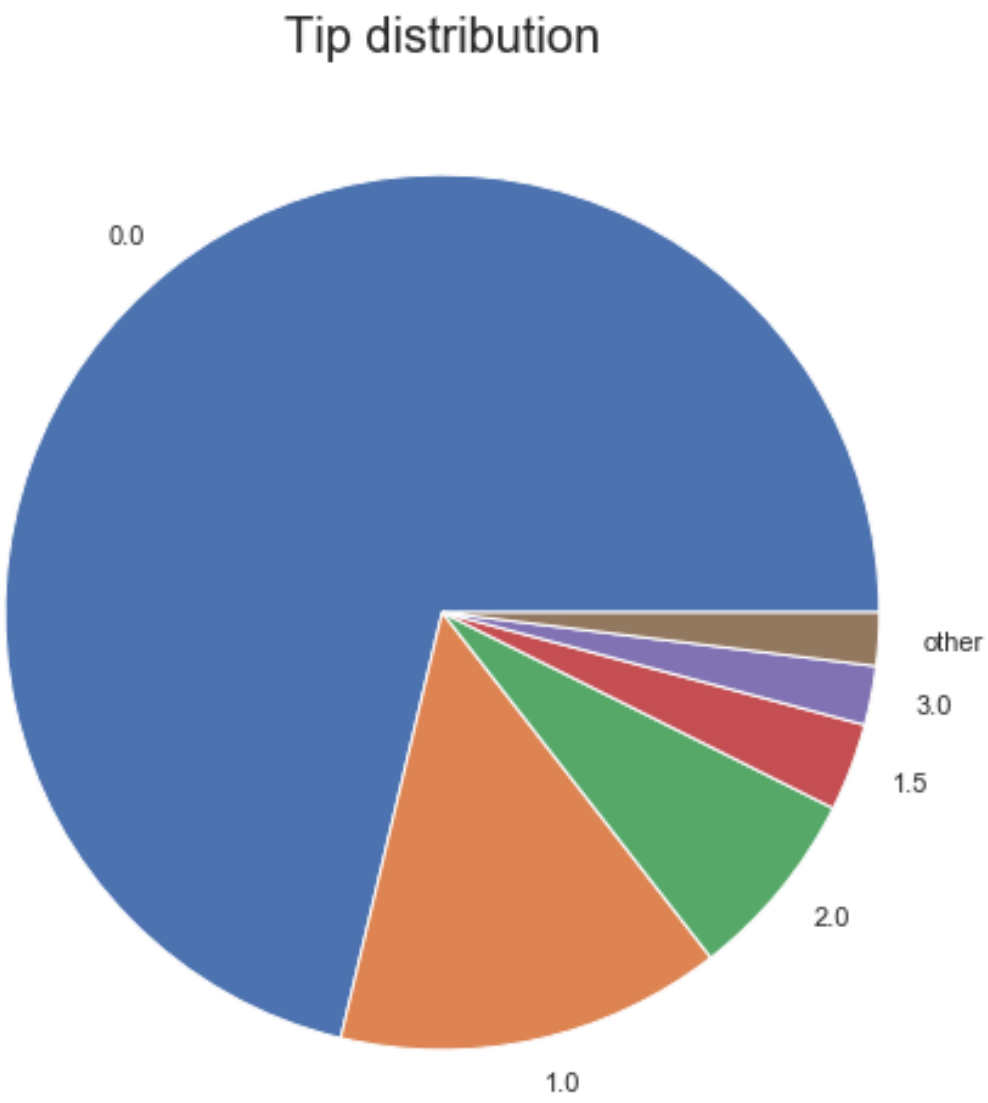
[35]:
```
tip_count = df['tip_amount'].value_counts()
tip_count
```

[35]:
```
0.00     3840212
1.00      768220
2.00      381510
1.50      174963
3.00      117458
          …
78.00          1
77.00          1
11.12          1
11.13          1
64.69          1
Name: tip_amount, Length: 2074, dtype: int64
```

[36]:
```
tip_count[5] = tip_count.iloc[5:].sum()
tip_count = tip_count.iloc[:6]
tip_count.index = tip_count.index.tolist()[:5] + ['other']
tip_count
```

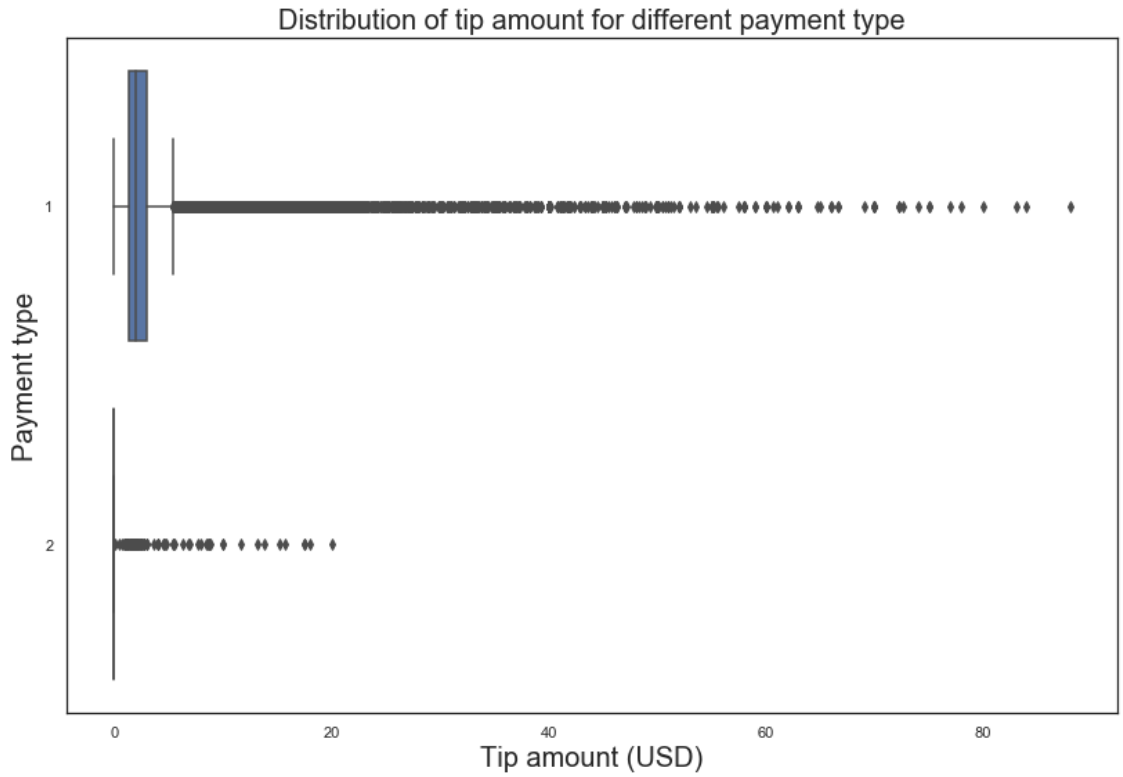
[36]:
```
0.0      3840212
1.0       768220
2.0       381510
1.5       174963
3.0       117458
other     105529
Name: tip_amount, dtype: int64
```

[37]:
```
plt.pie(tip_count.values,labels=tip_count.index)
plt.title("Tip distribution")
plt.show()
```



[ ]:

[38]:
```
sns.boxplot(x="tip_amount", y="payment_type", data=df)
plt.title("Distribution of tip amount for different payment type")
plt.xlabel('Tip amount (USD)')
plt.ylabel("Payment type")
plt.tight_layout()
```



[ ]:

5 What is related to fare amount

[39]:
```
fit = ols(formula="fare_amount ~ trip_distance + duration + start_hour +␣
 ↪weather + duration*start_hour + duration*weather",
          data=sub_df1).fit()
print(fit.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:            fare_amount   R-squared:                       0.974
Model:                            OLS   Adj. R-squared:                  0.974
Method:                 Least Squares   F-statistic:                 7.451e+05
Date:                Fri, 04 Sep 2020   Prob (F-statistic):               0.00
Time:                        20:40:04   Log-Likelihood:            -1.8819e+06
No. Observations:             1000000   AIC:                         3.764e+06
Df Residuals:                  999949   BIC:                         3.765e+06
Df Model:                          50
Covariance Type:            nonrobust
==============================================================================
==============
```

```
                                coef    std err          t      P>|t|
[0.025      0.975]
------------------------------------------------------------------------------
--------------
Intercept                      1.9486      0.019    101.664      0.000
1.911       1.986
start_hour[T.1]                0.0509      0.023      2.232      0.026
0.006       0.096
start_hour[T.2]                0.0606      0.025      2.430      0.015
0.012       0.110
start_hour[T.3]                0.0087      0.027      0.320      0.749
-0.044       0.062
start_hour[T.4]               -0.0441      0.030     -1.473      0.141
-0.103       0.015
start_hour[T.5]                0.0339      0.029      1.161      0.246
-0.023       0.091
start_hour[T.6]                0.5401      0.021     25.182      0.000
0.498       0.582
start_hour[T.7]                0.6817      0.019     35.166      0.000
0.644       0.720
start_hour[T.8]                0.5188      0.019     26.974      0.000
0.481       0.557
start_hour[T.9]                0.2486      0.019     12.784      0.000
0.210       0.287
start_hour[T.10]               0.2999      0.019     15.453      0.000
0.262       0.338
start_hour[T.11]               0.1468      0.019      7.581      0.000
0.109       0.185
start_hour[T.12]               0.2428      0.019     12.679      0.000
0.205       0.280
start_hour[T.13]               0.2887      0.019     15.077      0.000
0.251       0.326
start_hour[T.14]               0.6010      0.019     32.093      0.000
0.564       0.638
start_hour[T.15]               0.7723      0.018     41.934      0.000
0.736       0.808
start_hour[T.16]               0.7072      0.019     37.968      0.000
0.671       0.744
start_hour[T.17]               0.6678      0.018     36.521      0.000
0.632       0.704
start_hour[T.18]               0.4819      0.018     26.384      0.000
0.446       0.518
start_hour[T.19]               0.2590      0.019     13.957      0.000
0.223       0.295
start_hour[T.20]               0.0958      0.019      5.053      0.000
0.059       0.133
start_hour[T.21]               0.1093      0.019      5.711      0.000
0.072       0.147
```

```
start_hour[T.22]                0.0626      0.019      3.228      0.001
0.025       0.101
start_hour[T.23]               -0.0171      0.020     -0.847      0.397
-0.057       0.023
weather[T.remain]               0.0996      0.013      7.528      0.000
0.074       0.126
trip_distance                   2.0124      0.001   2613.856      0.000
2.011       2.014
duration                        0.3261      0.001    258.939      0.000
0.324       0.329
duration:start_hour[T.1]       -0.0073      0.002     -4.754      0.000
-0.010      -0.004
duration:start_hour[T.2]       -0.0099      0.002     -5.814      0.000
-0.013      -0.007
duration:start_hour[T.3]       -0.0026      0.002     -1.426      0.154
-0.006       0.001
duration:start_hour[T.4]        0.0147      0.002      7.472      0.000
0.011       0.019
duration:start_hour[T.5]        0.0196      0.002     10.324      0.000
0.016       0.023
duration:start_hour[T.6]       -0.0548      0.001    -39.698      0.000
-0.057      -0.052
duration:start_hour[T.7]       -0.0573      0.001    -46.421      0.000
-0.060      -0.055
duration:start_hour[T.8]       -0.0233      0.001    -19.245      0.000
-0.026      -0.021
duration:start_hour[T.9]        0.0023      0.001      1.868      0.062
-0.000       0.005
duration:start_hour[T.10]      -0.0034      0.001     -2.807      0.005
-0.006      -0.001
duration:start_hour[T.11]       0.0090      0.001      7.313      0.000
0.007       0.011
duration:start_hour[T.12]       0.0020      0.001      1.622      0.105
-0.000       0.004
duration:start_hour[T.13]      -0.0040      0.001     -3.299      0.001
-0.006      -0.002
duration:start_hour[T.14]      -0.0322      0.001    -27.775      0.000
-0.034      -0.030
duration:start_hour[T.15]      -0.0512      0.001    -45.212      0.000
-0.053      -0.049
duration:start_hour[T.16]      -0.0510      0.001    -44.767      0.000
-0.053      -0.049
duration:start_hour[T.17]      -0.0488      0.001    -43.103      0.000
-0.051      -0.047
duration:start_hour[T.18]      -0.0349      0.001    -30.004      0.000
-0.037      -0.033
duration:start_hour[T.19]      -0.0194      0.001    -15.978      0.000
-0.022      -0.017
```

```
duration:start_hour[T.20]     -0.0086      0.001     -6.836      0.000
-0.011      -0.006
duration:start_hour[T.21]     -0.0091      0.001     -7.179      0.000
-0.012      -0.007
duration:start_hour[T.22]     -0.0028      0.001     -2.205      0.027
-0.005      -0.000
duration:start_hour[T.23]      0.0033      0.001      2.515      0.012
0.001       0.006
duration:weather[T.remain]     0.0045      0.001      5.302      0.000
0.003       0.006
==============================================================================
Omnibus:                  1016425.708   Durbin-Watson:                   2.000
Prob(Omnibus):                  0.000   Jarque-Bera (JB):       2567932846.754
Skew:                           3.830   Prob(JB):                         0.00
Kurtosis:                     251.136   Cond. No.                     1.12e+03
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 1.12e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

[40]:
```
fit = ols(formula="fare_amount ~ trip_distance + duration + start_hour",
          data=df).fit()
print(fit.summary())
```

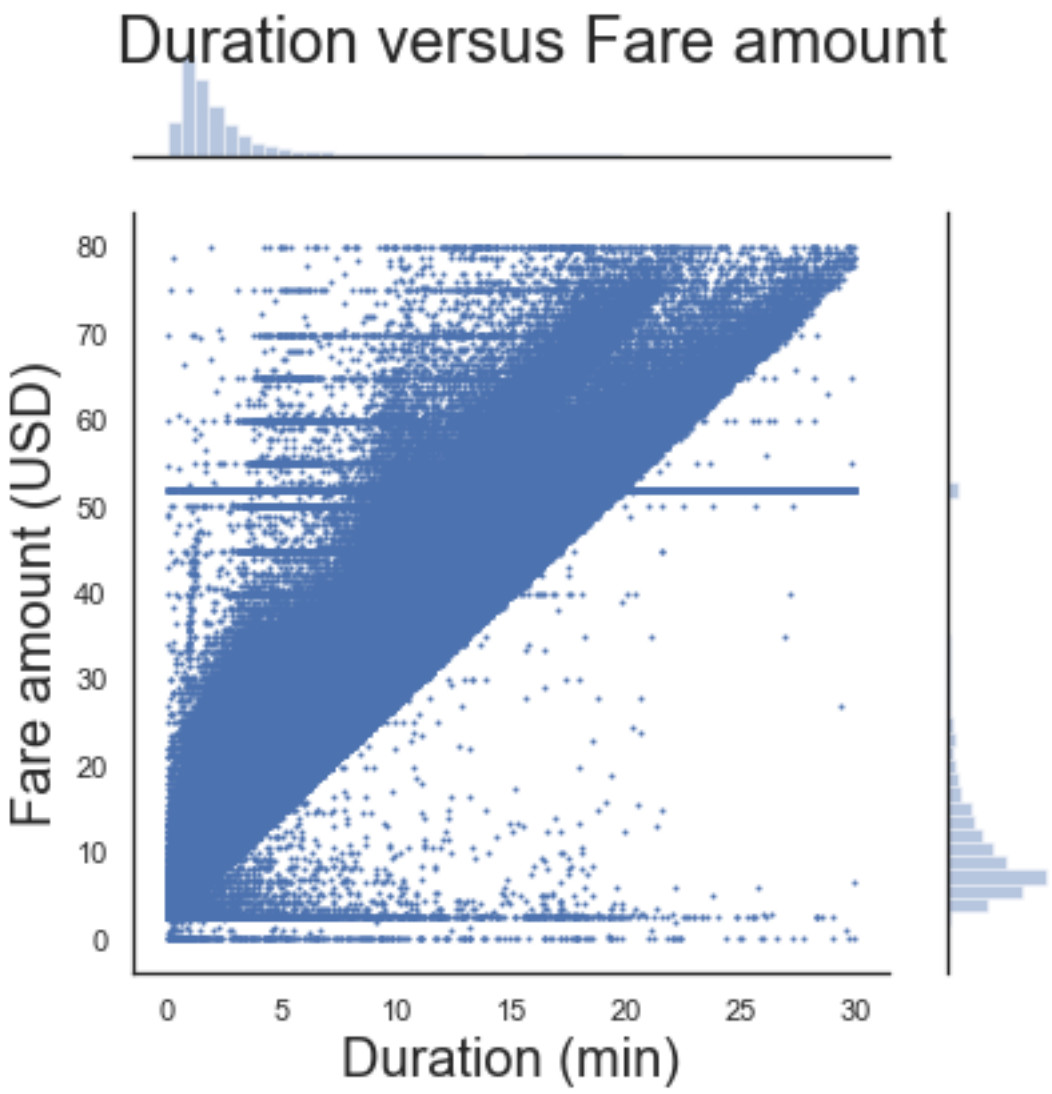
```
                            OLS Regression Results
==============================================================================
Dep. Variable:            fare_amount   R-squared:                       0.973
Model:                            OLS   Adj. R-squared:                  0.973
Method:                 Least Squares   F-statistic:                 1.532e+07
Date:                Fri, 04 Sep 2020   Prob (F-statistic):               0.00
Time:                        20:41:02   Log-Likelihood:            -2.0162e+07
No. Observations:            10624522   AIC:                         4.032e+07
Df Residuals:                10624496   BIC:                         4.032e+07
Df Model:                           25
Covariance Type:            nonrobust
==============================================================================
====
                       coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
----
Intercept            2.3201      0.003    867.692      0.000       2.315
2.325
start_hour[T.1]     -0.0492      0.004    -12.404      0.000      -0.057
-0.041
```

```
start_hour[T.2]     -0.0706      0.004    -16.410      0.000      -0.079
-0.062
start_hour[T.3]     -0.0396      0.005     -8.312      0.000      -0.049
-0.030
start_hour[T.4]      0.1059      0.005     19.868      0.000       0.095
0.116
start_hour[T.5]      0.2606      0.006     46.680      0.000       0.250
0.272
start_hour[T.6]     -0.0910      0.004    -21.361      0.000      -0.099
-0.083
start_hour[T.7]     -0.0289      0.004     -7.903      0.000      -0.036
-0.022
start_hour[T.8]      0.2317      0.004     66.160      0.000       0.225
0.239
start_hour[T.9]      0.3171      0.004     90.572      0.000       0.310
0.324
start_hour[T.10]     0.2916      0.004     82.908      0.000       0.285
0.299
start_hour[T.11]     0.3046      0.003     87.433      0.000       0.298
0.311
start_hour[T.12]     0.2999      0.003     87.276      0.000       0.293
0.307
start_hour[T.13]     0.2674      0.003     77.717      0.000       0.261
0.274
start_hour[T.14]     0.2011      0.003     59.117      0.000       0.194
0.208
start_hour[T.15]     0.1137      0.003     33.389      0.000       0.107
0.120
start_hour[T.16]     0.0058      0.003      1.662      0.097      -0.001
0.013
start_hour[T.17]     0.0148      0.003      4.392      0.000       0.008
0.021
start_hour[T.18]     0.0423      0.003     12.939      0.000       0.036
0.049
start_hour[T.19]     0.0288      0.003      8.803      0.000       0.022
0.035
start_hour[T.20]     0.0071      0.003      2.140      0.032       0.001
0.014
start_hour[T.21]     0.0093      0.003      2.769      0.006       0.003
0.016
start_hour[T.22]     0.0405      0.003     12.021      0.000       0.034
0.047
start_hour[T.23]     0.0334      0.004      9.507      0.000       0.026
0.040
trip_distance        2.0156      0.000   8559.985      0.000       2.015
2.016
duration             0.3068   8.14e-05   3767.531      0.000       0.307
0.307
```

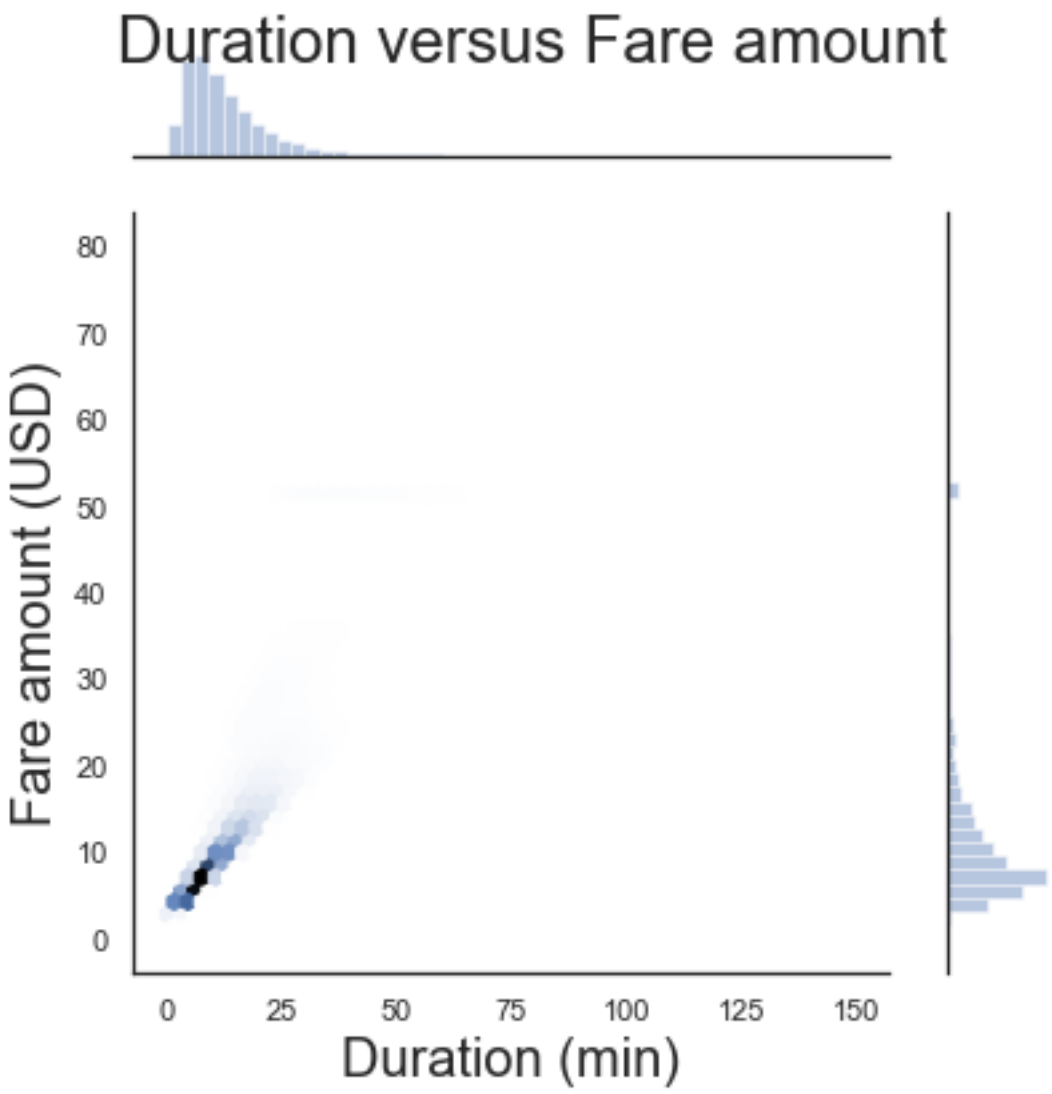
```
==============================================================================
Omnibus:                 10420296.736   Durbin-Watson:                   1.978
Prob(Omnibus):                 0.000   Jarque-Bera (JB):      24913210541.062
Skew:                          3.578   Prob(JB):                         0.00
Kurtosis:                    240.120   Cond. No.                         451.
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
```

[ ]:

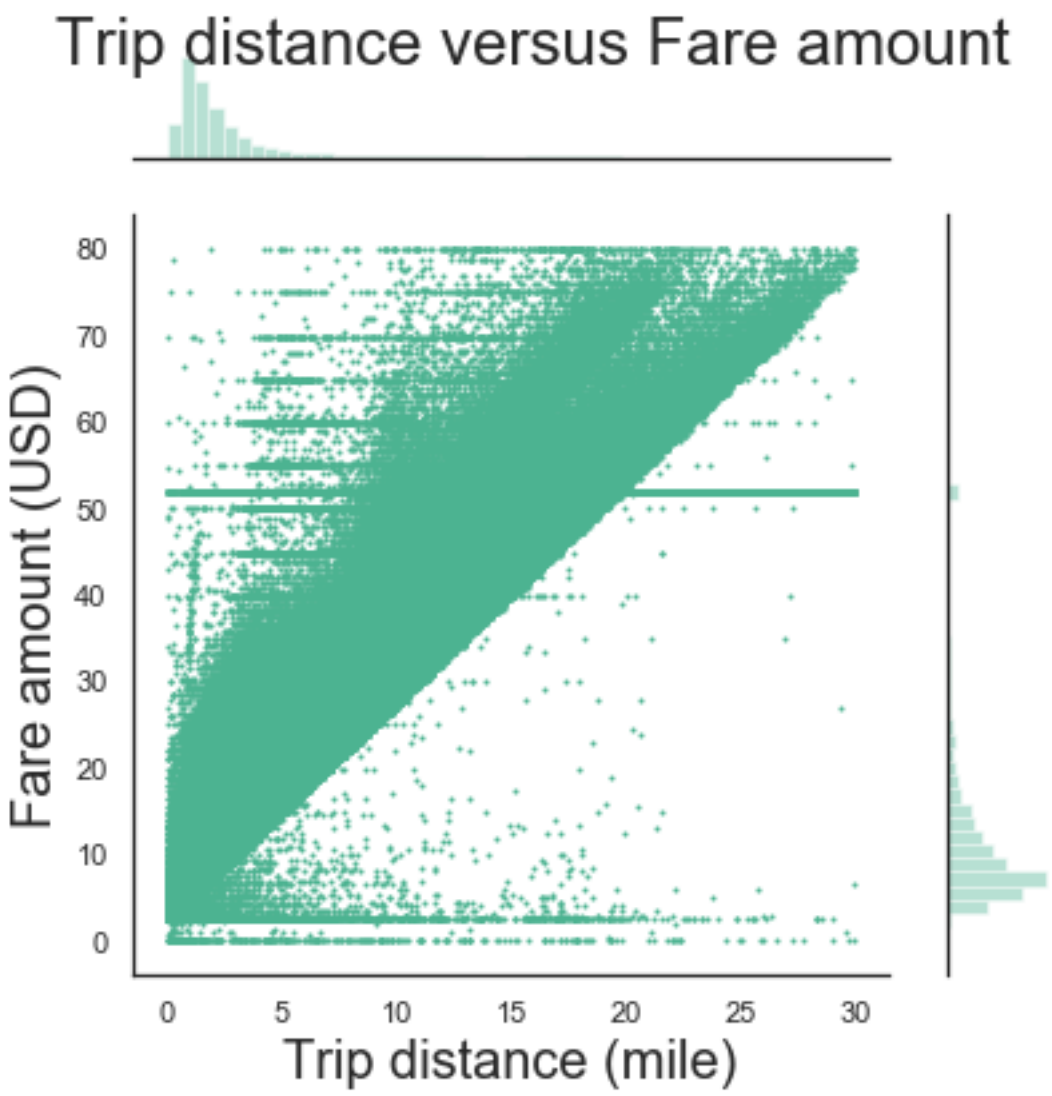
[41]:
```
p = sns.jointplot(x='trip_distance',y= 'fare_amount', data= df, s=1)
p.ax_joint.set_xlabel('Duration (min)')
p.ax_joint.set_ylabel('Fare amount (USD)')
p.fig.suptitle("Duration versus Fare amount")
p.fig.tight_layout()
```



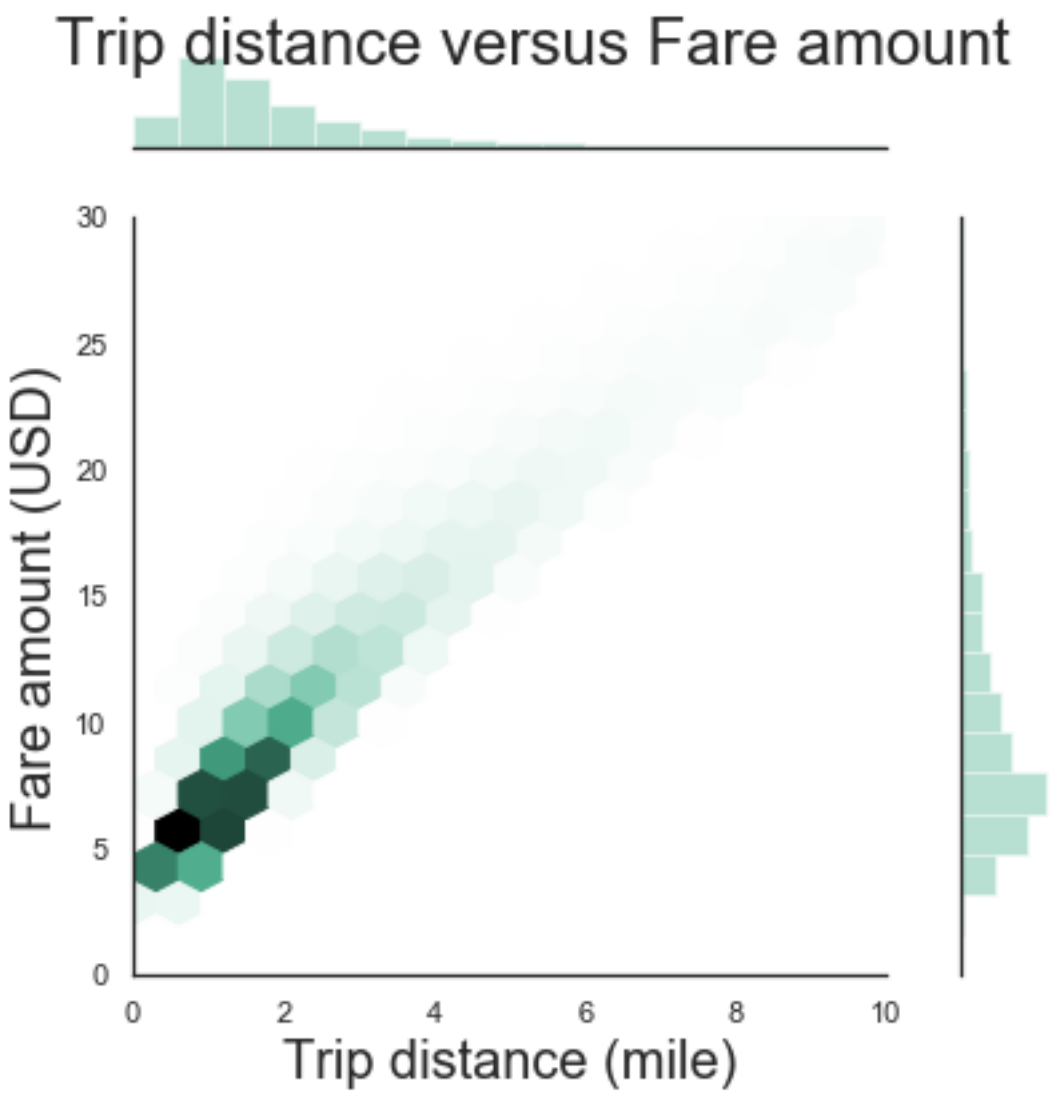
[42]:
```
p = sns.jointplot(x='duration',y= 'fare_amount',data= df, kind="hex")
p.ax_joint.set_xlabel('Duration (min)')
p.ax_joint.set_ylabel('Fare amount (USD)')
p.fig.suptitle("Duration versus Fare amount")
p.fig.tight_layout()
```



[43]:
```
p = sns.jointplot(x='trip_distance',y= 'fare_amount', data= df, s=1,␣
↪color="#4CB391")
p.ax_joint.set_xlabel('Trip distance (mile)')
p.ax_joint.set_ylabel('Fare amount (USD)')
p.fig.suptitle("Trip distance versus Fare amount")
p.fig.tight_layout()
```



[44]:
```
p = sns.jointplot(x='trip_distance',y= 'fare_amount',data= df, kind="hex",␣
↪color="#4CB391", xlim=(0,10), ylim=(0, 30))
p.ax_joint.set_xlabel('Trip distance (mile)')
p.ax_joint.set_ylabel('Fare amount (USD)')
p.fig.suptitle("Trip distance versus Fare amount")
p.fig.tight_layout()
```



[ ]:

6 What is related to trip distance

[45]:
```
fit = ols(formula="trip_distance ~ tip_amount + duration + start_hour + payment_type + weather + RatecodeID +\
                   duration* start_hour + duration* weather",
          data=sub_df1).fit()
print(fit.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:          trip_distance   R-squared:                       0.780
```

```
Model:                            OLS   Adj. R-squared:                  0.780
Method:                 Least Squares   F-statistic:                 6.828e+04
Date:                Fri, 04 Sep 2020   Prob (F-statistic):               0.00
Time:                        20:44:11   Log-Likelihood:            -1.9156e+06
No. Observations:             1000000   AIC:                         3.831e+06
Df Residuals:                  999947   BIC:                         3.832e+06
Df Model:                          52
Covariance Type:            nonrobust
================================================================================
==============
                                 coef    std err          t      P>|t|
[0.025      0.975]
--------------------------------------------------------------------------------
--------------
Intercept                     -0.9023      0.020    -45.309      0.000
-0.941      -0.863
start_hour[T.1]               -0.0319      0.024     -1.353      0.176
-0.078       0.014
start_hour[T.2]               -0.0777      0.026     -3.010      0.003
-0.128      -0.027
start_hour[T.3]               -0.0639      0.028     -2.278      0.023
-0.119      -0.009
start_hour[T.4]            -6.041e-05      0.031     -0.002      0.998
-0.061       0.061
start_hour[T.5]                0.3590      0.030     11.883      0.000
0.300       0.418
start_hour[T.6]                0.6620      0.022     29.821      0.000
0.618       0.705
start_hour[T.7]                0.4677      0.020     23.324      0.000
0.428       0.507
start_hour[T.8]                0.2624      0.020     13.189      0.000
0.223       0.301
start_hour[T.9]                0.2310      0.020     11.486      0.000
0.192       0.270
start_hour[T.10]               0.3012      0.020     15.002      0.000
0.262       0.341
start_hour[T.11]               0.2251      0.020     11.234      0.000
0.186       0.264
start_hour[T.12]               0.1378      0.020      6.955      0.000
0.099       0.177
start_hour[T.13]              -0.0157      0.020     -0.792      0.428
-0.055       0.023
start_hour[T.14]               0.0112      0.019      0.580      0.562
-0.027       0.049
start_hour[T.15]               0.0917      0.019      4.811      0.000
0.054       0.129
start_hour[T.16]               0.1529      0.019      7.934      0.000
0.115       0.191
```

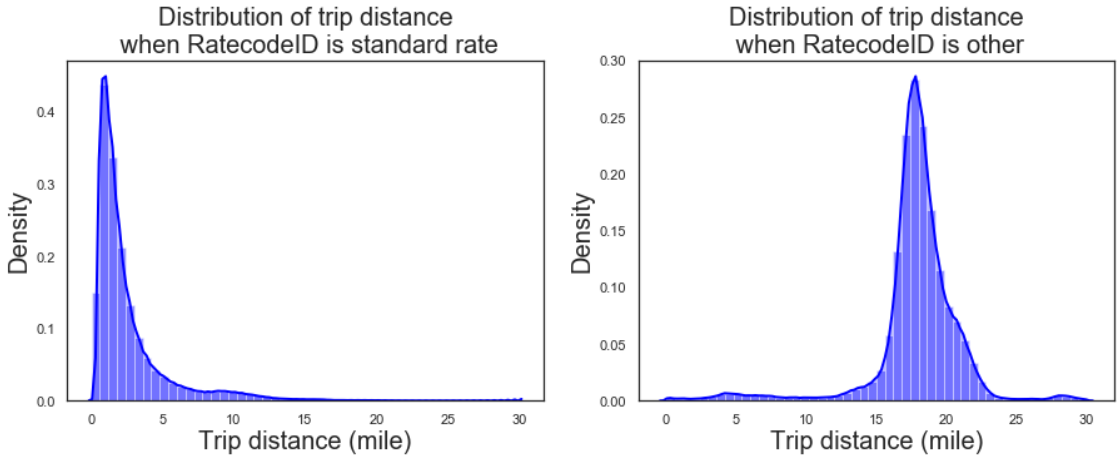
```
start_hour[T.17]               0.1084      0.019      5.730      0.000
0.071       0.146
start_hour[T.18]              -0.0506      0.019     -2.677      0.007
-0.088      -0.014
start_hour[T.19]              -0.1338      0.019     -6.969      0.000
-0.171      -0.096
start_hour[T.20]              -0.2008      0.020    -10.237      0.000
-0.239      -0.162
start_hour[T.21]              -0.1633      0.020     -8.249      0.000
-0.202      -0.125
start_hour[T.22]              -0.1494      0.020     -7.441      0.000
-0.189      -0.110
start_hour[T.23]              -0.1592      0.021     -7.610      0.000
-0.200      -0.118
payment_type[T.2]              0.9278      0.004    208.424      0.000
0.919       0.937
weather[T.remain]              0.0412      0.014      3.008      0.003
0.014       0.068
RatecodeID[T.2]                7.4829      0.013    567.486      0.000
7.457       7.509
tip_amount                     0.3994      0.001    355.463      0.000
0.397       0.402
duration                       0.2453      0.001    191.044      0.000
0.243       0.248
duration:start_hour[T.1]       0.0119      0.002      7.538      0.000
0.009       0.015
duration:start_hour[T.2]       0.0240      0.002     13.594      0.000
0.021       0.027
duration:start_hour[T.3]       0.0373      0.002     19.655      0.000
0.034       0.041
duration:start_hour[T.4]       0.0544      0.002     26.758      0.000
0.050       0.058
duration:start_hour[T.5]       0.0377      0.002     19.221      0.000
0.034       0.042
duration:start_hour[T.6]      -0.0391      0.001    -27.326      0.000
-0.042      -0.036
duration:start_hour[T.7]      -0.0723      0.001    -56.581      0.000
-0.075      -0.070
duration:start_hour[T.8]      -0.0911      0.001    -72.812      0.000
-0.094      -0.089
duration:start_hour[T.9]      -0.0877      0.001    -69.046      0.000
-0.090      -0.085
duration:start_hour[T.10]     -0.0897      0.001    -70.883      0.000
-0.092      -0.087
duration:start_hour[T.11]     -0.0881      0.001    -69.437      0.000
-0.091      -0.086
duration:start_hour[T.12]     -0.0807      0.001    -64.150      0.000
-0.083      -0.078
```

```
duration:start_hour[T.13]     -0.0647      0.001    -51.737      0.000
-0.067      -0.062
duration:start_hour[T.14]     -0.0704      0.001    -58.626      0.000
-0.073      -0.068
duration:start_hour[T.15]     -0.0810      0.001    -69.002      0.000
-0.083      -0.079
duration:start_hour[T.16]     -0.0846      0.001    -71.752      0.000
-0.087      -0.082
duration:start_hour[T.17]     -0.0840      0.001    -71.667      0.000
-0.086      -0.082
duration:start_hour[T.18]     -0.0680      0.001    -56.440      0.000
-0.070      -0.066
duration:start_hour[T.19]     -0.0413      0.001    -32.877      0.000
-0.044      -0.039
duration:start_hour[T.20]     -0.0141      0.001    -10.898      0.000
-0.017      -0.012
duration:start_hour[T.21]     -0.0066      0.001     -5.063      0.000
-0.009      -0.004
duration:start_hour[T.22]     -0.0046      0.001     -3.451      0.001
-0.007      -0.002
duration:start_hour[T.23]      0.0104      0.001      7.594      0.000
0.008       0.013
duration:weather[T.remain]    -0.0029      0.001     -3.306      0.001
-0.005      -0.001
==============================================================================
Omnibus:                   448238.632   Durbin-Watson:                   2.004
Prob(Omnibus):                  0.000   Jarque-Bera (JB):         14404965.415
Skew:                           1.536   Prob(JB):                         0.00
Kurtosis:                      21.338   Cond. No.                     1.11e+03
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 1.11e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

[ ]:

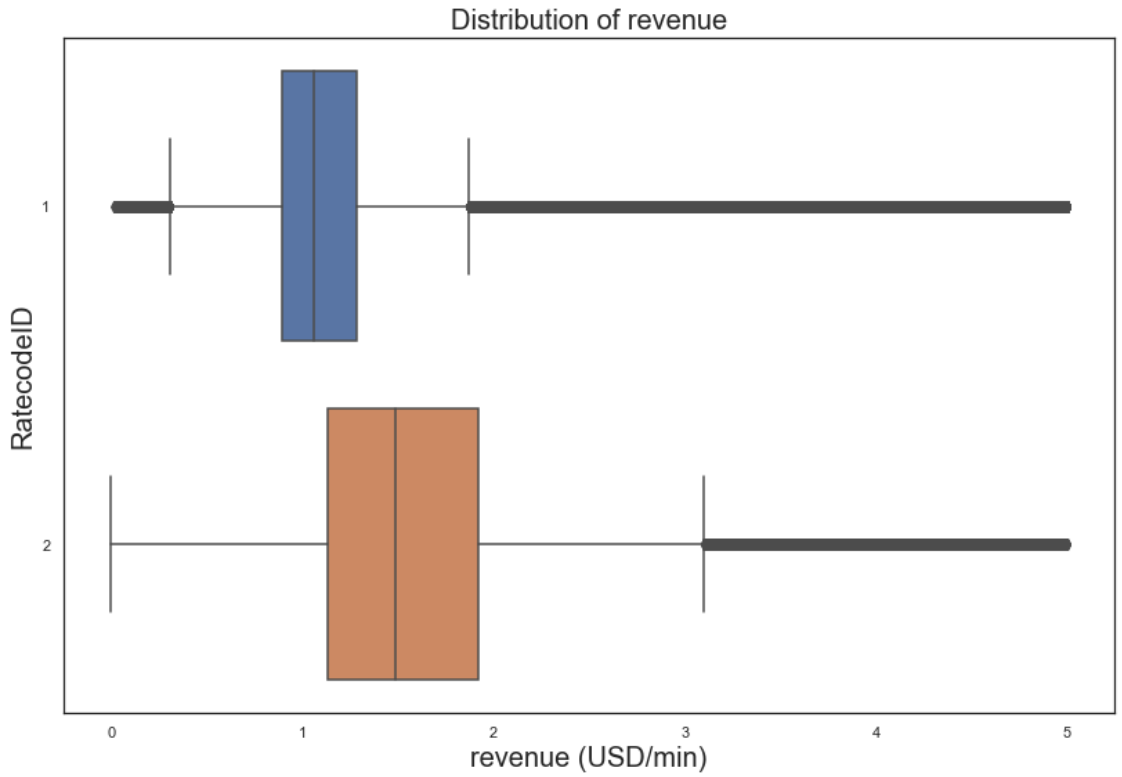
[46]:
```
plt.figure(figsize=(15, 5))
plt.subplot(121)
sns.distplot(stand_df ['trip_distance'], kde = True,
             kde_kws = {'shade': True, 'linewidth': 2}, color ="blue")
plt.title("Distribution of trip distance\n when RatecodeID is standard rate")
plt.xlabel('Trip distance (mile)')
plt.ylabel("Density")
```

```
plt.subplot(122)
sns.distplot(other_df['trip_distance'], kde = True,
             kde_kws = {'shade': True, 'linewidth': 2}, color ="blue")
plt.title("Distribution of trip distance\n when RatecodeID is other")
plt.xlabel('Trip distance (mile)')
plt.ylabel("Density")

fig.set_figheight(5)
fig.set_figwidth(15)
```



[ ]:

[47]:
```
sns.boxplot(x="income/duration", y="RatecodeID", data=df)
plt.title("Distribution of revenue")
plt.xlabel('revenue (USD/min)')
plt.tight_layout()
```



[ ]:

7 Delete ratecodeID = 2 from all dataset and resampling

[48]:
```
df = df.loc[(df["RatecodeID"] == 1)]
```

[49]:
```
df.shape
```

[49]: (10397081, 19)

[50]:
```
sub_df1 = sub_df1.loc[(sub_df1["RatecodeID"] == 1)]
sub_df2 = sub_df2.loc[(sub_df2["RatecodeID"] == 1)]
sub_df3 = sub_df3.loc[(sub_df3["RatecodeID"] == 1)]
```

[51]:
```
fit = ols(formula="trip_distance ~  duration + start_hour + payment_type +␣
 ↪weather +\
                    duration* start_hour + duration* weather",
          data=sub_df1).fit()
print(fit.summary())
```

```
                            OLS Regression Results
==============================================================================
```

```
Dep. Variable:          trip_distance   R-squared:                       0.616
Model:                            OLS   Adj. R-squared:                  0.616
Method:                 Least Squares   F-statistic:                 3.141e+04
Date:                Fri, 04 Sep 2020   Prob (F-statistic):               0.00
Time:                        20:44:26   Log-Likelihood:            -1.8964e+06
No. Observations:              978619   AIC:                         3.793e+06
Df Residuals:                  978568   BIC:                         3.794e+06
Df Model:                          50
Covariance Type:            nonrobust
==============================================================================
==============
                                 coef    std err          t      P>|t|
[0.025      0.975]
------------------------------------------------------------------------------
--------------
Intercept                     -0.2641      0.021    -12.674      0.000
-0.305      -0.223
start_hour[T.1]               -0.0232      0.024     -0.953      0.341
-0.071       0.025
start_hour[T.2]               -0.0621      0.027     -2.337      0.019
-0.114      -0.010
start_hour[T.3]               -0.0251      0.029     -0.868      0.385
-0.082       0.032
start_hour[T.4]                0.0572      0.032      1.780      0.075
-0.006       0.120
start_hour[T.5]               -0.0487      0.033     -1.495      0.135
-0.113       0.015
start_hour[T.6]                0.0312      0.024      1.285      0.199
-0.016       0.079
start_hour[T.7]                0.0735      0.022      3.417      0.001
0.031       0.116
start_hour[T.8]                0.1065      0.021      5.081      0.000
0.065       0.148
start_hour[T.9]                0.1482      0.021      7.058      0.000
0.107       0.189
start_hour[T.10]               0.2211      0.021     10.511      0.000
0.180       0.262
start_hour[T.11]               0.1996      0.021      9.527      0.000
0.159       0.241
start_hour[T.12]               0.1146      0.021      5.513      0.000
0.074       0.155
start_hour[T.13]              -0.0582      0.021     -2.785      0.005
-0.099      -0.017
start_hour[T.14]              -0.1435      0.021     -6.975      0.000
-0.184      -0.103
start_hour[T.15]              -0.1224      0.020     -6.038      0.000
-0.162      -0.083
start_hour[T.16]              -0.0595      0.020     -2.909      0.004
```

```
-0.100      -0.019
start_hour[T.17]              -0.0660      0.020     -3.293      0.001
-0.105      -0.027
start_hour[T.18]              -0.1711      0.020     -8.590      0.000
-0.210      -0.132
start_hour[T.19]              -0.1985      0.020     -9.862      0.000
-0.238      -0.159
start_hour[T.20]              -0.2477      0.021    -12.068      0.000
-0.288      -0.207
start_hour[T.21]              -0.2165      0.021    -10.494      0.000
-0.257      -0.176
start_hour[T.22]              -0.1700      0.021     -8.145      0.000
-0.211      -0.129
start_hour[T.23]              -0.1840      0.022     -8.476      0.000
-0.227      -0.141
payment_type[T.2]             -0.0411      0.004    -11.409      0.000
-0.048      -0.034
weather[T.remain]             -0.0828      0.015     -5.616      0.000
-0.112      -0.054
duration                       0.2732      0.001    195.946      0.000
0.271       0.276
duration:start_hour[T.1]       0.0105      0.002      6.316      0.000
0.007       0.014
duration:start_hour[T.2]       0.0224      0.002     12.172      0.000
0.019       0.026
duration:start_hour[T.3]       0.0343      0.002     17.326      0.000
0.030       0.038
duration:start_hour[T.4]       0.0557      0.002     25.492      0.000
0.051       0.060
duration:start_hour[T.5]       0.0969      0.002     40.489      0.000
0.092       0.102
duration:start_hour[T.6]       0.0321      0.002     17.880      0.000
0.029       0.036
duration:start_hour[T.7]      -0.0421      0.001    -28.743      0.000
-0.045      -0.039
duration:start_hour[T.8]      -0.0881      0.001    -64.460      0.000
-0.091      -0.085
duration:start_hour[T.9]      -0.0900      0.001    -65.911      0.000
-0.093      -0.087
duration:start_hour[T.10]     -0.0901      0.001    -65.809      0.000
-0.093      -0.087
duration:start_hour[T.11]     -0.0933      0.001    -68.174      0.000
-0.096      -0.091
duration:start_hour[T.12]     -0.0862      0.001    -62.963      0.000
-0.089      -0.084
duration:start_hour[T.13]     -0.0677      0.001    -49.210      0.000
-0.070      -0.065
duration:start_hour[T.14]     -0.0642      0.001    -48.065      0.000
```

```
-0.067      -0.062
duration:start_hour[T.15]     -0.0696      0.001    -53.217      0.000
-0.072      -0.067
duration:start_hour[T.16]     -0.0708      0.001    -53.906      0.000
-0.073      -0.068
duration:start_hour[T.17]     -0.0746      0.001    -57.583      0.000
-0.077      -0.072
duration:start_hour[T.18]     -0.0634      0.001    -48.105      0.000
-0.066      -0.061
duration:start_hour[T.19]     -0.0398      0.001    -29.163      0.000
-0.042      -0.037
duration:start_hour[T.20]     -0.0128      0.001     -9.145      0.000
-0.016      -0.010
duration:start_hour[T.21]     -0.0024      0.001     -1.728      0.084
-0.005       0.000
duration:start_hour[T.22]     -0.0029      0.001     -2.047      0.041
-0.006      -0.000
duration:start_hour[T.23]      0.0134      0.001      9.185      0.000
0.011       0.016
duration:weather[T.remain]     0.0091      0.001      9.010      0.000
0.007       0.011
==============================================================================
Omnibus:                   479239.040   Durbin-Watson:                   2.003
Prob(Omnibus):                  0.000   Jarque-Bera (JB):         12934893.544
Skew:                           1.801   Prob(JB):                         0.00
Kurtosis:                      20.443   Cond. No.                     1.02e+03
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 1.02e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

[ ]:

[52]:
```
fit = ols(formula="trip_distance ~  duration + start_hour + payment_type +␣
 ↪weather",
          data=sub_df1).fit()
print(fit.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:          trip_distance   R-squared:                       0.598
Model:                            OLS   Adj. R-squared:                  0.598
Method:                 Least Squares   F-statistic:                 5.602e+04
Date:                Fri, 04 Sep 2020   Prob (F-statistic):               0.00
Time:                        20:44:28   Log-Likelihood:           -1.9188e+06
```

```
No. Observations:              978619   AIC:                          3.838e+06
Df Residuals:                  978592   BIC:                          3.838e+06
Df Model:                          26
Covariance Type:            nonrobust
==============================================================================
=====
                         coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
-----
Intercept              0.2814      0.012     22.661      0.000       0.257
0.306
start_hour[T.1]        0.0862      0.014      6.204      0.000       0.059
0.113
start_hour[T.2]        0.1784      0.015     11.865      0.000       0.149
0.208
start_hour[T.3]        0.3641      0.017     21.893      0.000       0.332
0.397
start_hour[T.4]        0.7026      0.019     37.372      0.000       0.666
0.739
start_hour[T.5]        0.8762      0.020     43.714      0.000       0.837
0.915
start_hour[T.6]        0.2237      0.015     14.859      0.000       0.194
0.253
start_hour[T.7]       -0.4487      0.013    -35.005      0.000      -0.474
-0.424
start_hour[T.8]       -1.0021      0.012    -81.852      0.000      -1.026
-0.978
start_hour[T.9]       -0.9907      0.012    -80.976      0.000      -1.015
-0.967
start_hour[T.10]      -0.9136      0.012    -74.194      0.000      -0.938
-0.889
start_hour[T.11]      -0.9772      0.012    -80.174      0.000      -1.001
-0.953
start_hour[T.12]      -0.9621      0.012    -79.917      0.000      -0.986
-0.939
start_hour[T.13]      -0.9005      0.012    -74.763      0.000      -0.924
-0.877
start_hour[T.14]      -0.9453      0.012    -79.053      0.000      -0.969
-0.922
start_hour[T.15]      -0.9996      0.012    -83.880      0.000      -1.023
-0.976
start_hour[T.16]      -0.9531      0.012    -78.304      0.000      -0.977
-0.929
start_hour[T.17]      -1.0099      0.012    -85.745      0.000      -1.033
-0.987
start_hour[T.18]      -0.9570      0.011    -83.659      0.000      -0.979
-0.935
```

```
start_hour[T.19]     -0.6886      0.011    -60.096      0.000      -0.711
-0.666
start_hour[T.20]     -0.4191      0.012    -35.972      0.000      -0.442
-0.396
start_hour[T.21]     -0.2589      0.012    -22.118      0.000      -0.282
-0.236
start_hour[T.22]     -0.2031      0.012    -17.163      0.000      -0.226
-0.180
start_hour[T.23]     -0.0070      0.012     -0.569      0.569      -0.031
0.017
payment_type[T.2]    -0.0385      0.004    -10.447      0.000      -0.046
-0.031
weather[T.remain]    -0.0264      0.009     -3.033      0.002      -0.044
-0.009
duration              0.2331      0.000   1185.661      0.000       0.233
0.233
==============================================================================
Omnibus:                   468487.939   Durbin-Watson:                   2.003
Prob(Omnibus):                  0.000   Jarque-Bera (JB):          9516502.815
Skew:                           1.832   Prob(JB):                         0.00
Kurtosis:                      17.831   Cond. No.                         406.
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
```

[53]:
```
fit = ols(formula="trip_distance ~ duration + start_hour + payment_type",
          data=df).fit()
print(fit.summary())
```

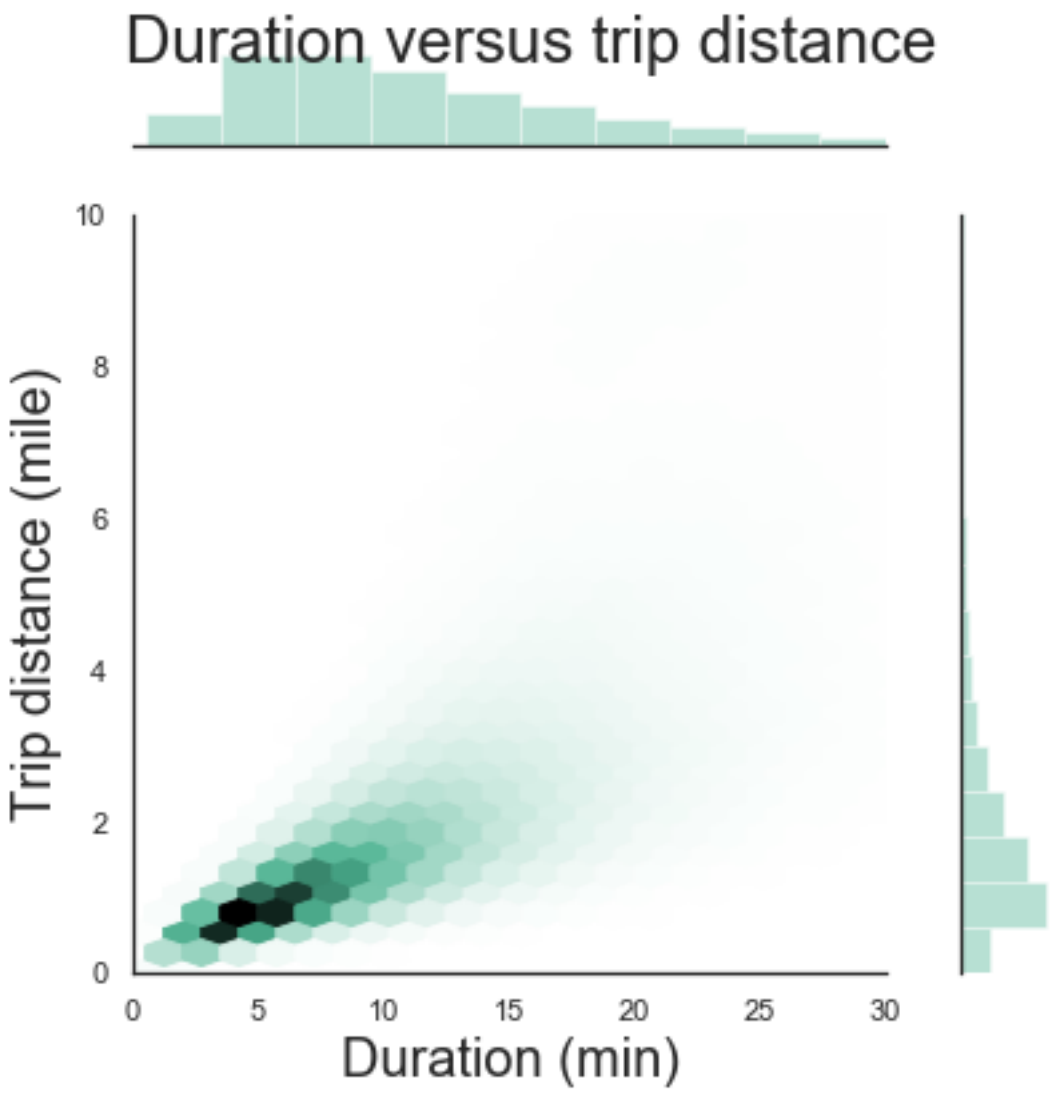
```
                            OLS Regression Results
==============================================================================
Dep. Variable:          trip_distance   R-squared:                       0.598
Model:                            OLS   Adj. R-squared:                  0.598
Method:                 Least Squares   F-statistic:                 6.189e+05
Date:                Fri, 04 Sep 2020   Prob (F-statistic):               0.00
Time:                        20:45:03   Log-Likelihood:            -2.0368e+07
No. Observations:            10397081   AIC:                         4.074e+07
Df Residuals:                10397055   BIC:                         4.074e+07
Df Model:                          25
Covariance Type:            nonrobust
==============================================================================
=====
                         coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
-----
```

```
Intercept             0.2651      0.003     90.971      0.000        0.259
0.271
start_hour[T.1]       0.0749      0.004     17.670      0.000        0.067
0.083
start_hour[T.2]       0.1835      0.005     39.971      0.000        0.175
0.193
start_hour[T.3]       0.3630      0.005     71.304      0.000        0.353
0.373
start_hour[T.4]       0.6985      0.006    121.783      0.000        0.687
0.710
start_hour[T.5]       0.8469      0.006    138.262      0.000        0.835
0.859
start_hour[T.6]       0.2392      0.005     51.817      0.000        0.230
0.248
start_hour[T.7]      -0.4673      0.004   -118.987      0.000       -0.475
-0.460
start_hour[T.8]      -1.0053      0.004   -268.657      0.000       -1.013
-0.998
start_hour[T.9]      -1.0082      0.004   -269.610      0.000       -1.016
-1.001
start_hour[T.10]     -0.9097      0.004   -241.856      0.000       -0.917
-0.902
start_hour[T.11]     -0.9729      0.004   -261.153      0.000       -0.980
-0.966
start_hour[T.12]     -0.9663      0.004   -262.822      0.000       -0.974
-0.959
start_hour[T.13]     -0.9143      0.004   -247.929      0.000       -0.921
-0.907
start_hour[T.14]     -0.9579      0.004   -262.449      0.000       -0.965
-0.951
start_hour[T.15]     -1.0181      0.004   -279.000      0.000       -1.025
-1.011
start_hour[T.16]     -0.9476      0.004   -254.267      0.000       -0.955
-0.940
start_hour[T.17]     -1.0236      0.004   -283.774      0.000       -1.031
-1.016
start_hour[T.18]     -0.9577      0.003   -274.151      0.000       -0.965
-0.951
start_hour[T.19]     -0.7014      0.004   -200.385      0.000       -0.708
-0.695
start_hour[T.20]     -0.4229      0.004   -118.682      0.000       -0.430
-0.416
start_hour[T.21]     -0.2713      0.004    -75.786      0.000       -0.278
-0.264
start_hour[T.22]     -0.2045      0.004    -56.614      0.000       -0.212
-0.197
start_hour[T.23]     -0.0228      0.004     -6.060      0.000       -0.030
-0.015
```

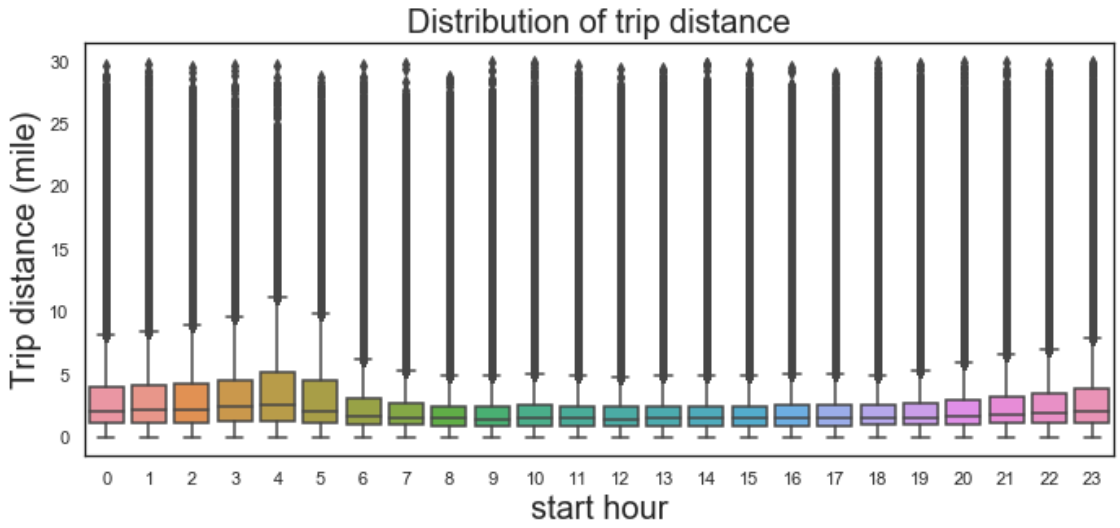
```
payment_type[T.2]     -0.0320      0.001    -28.313      0.000      -0.034
-0.030
duration               0.2326   6.02e-05   3865.297      0.000       0.232
0.233
==============================================================================
Omnibus:                  4920433.098   Durbin-Watson:                   1.839
Prob(Omnibus):                  0.000   Jarque-Bera (JB):         97517120.372
Skew:                           1.811   Prob(JB):                         0.00
Kurtosis:                      17.560   Cond. No.                         404.
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
```

[ ]:

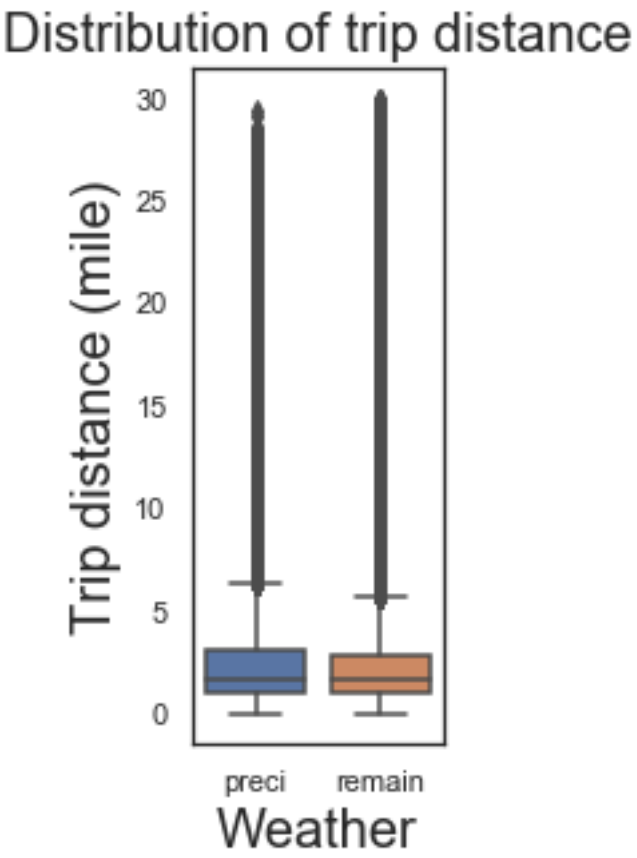
[54]:
```
p = sns.jointplot(x='duration',y= 'trip_distance',data= df, kind="hex",␣
 ↪color="#4CB391", xlim=(0,30), ylim=(0,10), gridsize=100)
p.ax_joint.set_xlabel('Duration (min)')
p.ax_joint.set_ylabel('Trip distance (mile)')
p.fig.suptitle("Duration versus trip distance")
p.fig.tight_layout()
```



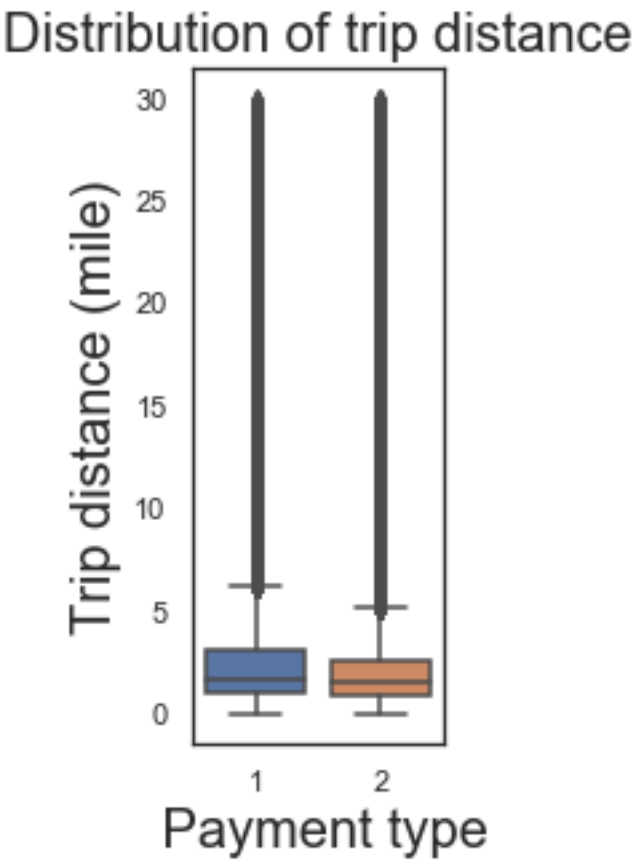
[55]:
```
plt.figure(figsize=(10, 5))
sns.boxplot(x="start_hour", y="trip_distance", data=df)
plt.title("Distribution of trip distance")
plt.ylabel('Trip distance (mile)')
plt.xlabel('start hour')
plt.tight_layout()
```



[56]:
```
plt.figure(figsize=(3, 5))
sns.boxplot(x="weather", y="trip_distance", data=df)
plt.title("Distribution of trip distance")
plt.ylabel('Trip distance (mile)')
plt.xlabel('Weather')
plt.tight_layout()
```



[57]:
```
plt.figure(figsize=(3, 5))
sns.boxplot(x="payment_type", y="trip_distance", data=df)
plt.title("Distribution of trip distance")
plt.ylabel('Trip distance (mile)')
plt.xlabel(' Payment type')
plt.tight_layout()
```



[ ]:

[ ]:

8 what related to income

[58]:
```
fit = ols(formula="income ~ duration + payment_type  +start_hour + duration * ␣
 ↪payment_type + duration * start_hour",data=sub_df1).fit()
print(fit.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 income   R-squared:                       0.812
Model:                            OLS   Adj. R-squared:                  0.812
Method:                 Least Squares   F-statistic:                 8.606e+04
Date:                Fri, 04 Sep 2020   Prob (F-statistic):               0.00
Time:                        20:45:42   Log-Likelihood:            -2.7340e+06
No. Observations:              978619   AIC:                         5.468e+06
Df Residuals:                  978569   BIC:                         5.469e+06
```

```
Df Model:                           49
Covariance Type:            nonrobust
==============================================================================
==============
                                 coef    std err          t      P>|t|
[0.025      0.975]
------------------------------------------------------------------------------
--------------
Intercept                      1.8370      0.038     48.529      0.000
1.763       1.911
payment_type[T.2]             -0.2886      0.014    -19.991      0.000
-0.317      -0.260
start_hour[T.1]                0.1167      0.057      2.034      0.042
0.004       0.229
start_hour[T.2]                0.0243      0.063      0.389      0.697
-0.098       0.147
start_hour[T.3]                0.0637      0.068      0.937      0.349
-0.070       0.197
start_hour[T.4]                0.2699      0.076      3.570      0.000
0.122       0.418
start_hour[T.5]               -0.0248      0.077     -0.323      0.747
-0.175       0.126
start_hour[T.6]                0.0018      0.057      0.031      0.975
-0.110       0.114
start_hour[T.7]                0.0725      0.051      1.434      0.152
-0.027       0.172
start_hour[T.8]                0.2776      0.049      5.632      0.000
0.181       0.374
start_hour[T.9]                0.2367      0.049      4.793      0.000
0.140       0.333
start_hour[T.10]               0.3729      0.049      7.548      0.000
0.276       0.470
start_hour[T.11]               0.2180      0.049      4.426      0.000
0.121       0.315
start_hour[T.12]               0.0616      0.049      1.260      0.208
-0.034       0.157
start_hour[T.13]              -0.3020      0.049     -6.148      0.000
-0.398      -0.206
start_hour[T.14]              -0.3351      0.048     -6.935      0.000
-0.430      -0.240
start_hour[T.15]              -0.1915      0.048     -4.019      0.000
-0.285      -0.098
start_hour[T.16]              -0.0129      0.048     -0.268      0.789
-0.107       0.081
start_hour[T.17]              -0.0552      0.047     -1.171      0.242
-0.148       0.037
start_hour[T.18]              -0.2565      0.047     -5.480      0.000
-0.348      -0.165
```

```
start_hour[T.19]              -0.3286      0.047     -6.944      0.000
-0.421      -0.236
start_hour[T.20]              -0.5343      0.048    -11.075      0.000
-0.629      -0.440
start_hour[T.21]              -0.4027      0.049     -8.301      0.000
-0.498      -0.308
start_hour[T.22]              -0.3164      0.049     -6.455      0.000
-0.412      -0.220
start_hour[T.23]              -0.4277      0.051     -8.377      0.000
-0.528      -0.328
duration                       1.0509      0.003    414.220      0.000
1.046       1.056
duration:payment_type[T.2]    -0.1738      0.001   -179.995      0.000
-0.176      -0.172
duration:start_hour[T.1]       0.0015      0.004      0.385      0.701
-0.006       0.009
duration:start_hour[T.2]       0.0262      0.004      6.043      0.000
0.018       0.035
duration:start_hour[T.3]       0.0552      0.005     11.853      0.000
0.046       0.064
duration:start_hour[T.4]       0.1040      0.005     20.237      0.000
0.094       0.114
duration:start_hour[T.5]       0.2084      0.006     36.990      0.000
0.197       0.219
duration:start_hour[T.6]       0.0673      0.004     15.914      0.000
0.059       0.076
duration:start_hour[T.7]      -0.0849      0.003    -24.682      0.000
-0.092      -0.078
duration:start_hour[T.8]      -0.1818      0.003    -56.633      0.000
-0.188      -0.176
duration:start_hour[T.9]      -0.1695      0.003    -52.874      0.000
-0.176      -0.163
duration:start_hour[T.10]     -0.1655      0.003    -51.481      0.000
-0.172      -0.159
duration:start_hour[T.11]     -0.1631      0.003    -50.726      0.000
-0.169      -0.157
duration:start_hour[T.12]     -0.1487      0.003    -46.301      0.000
-0.155      -0.142
duration:start_hour[T.13]     -0.1105      0.003    -34.218      0.000
-0.117      -0.104
duration:start_hour[T.14]     -0.1194      0.003    -38.116      0.000
-0.126      -0.113
duration:start_hour[T.15]     -0.1443      0.003    -46.944      0.000
-0.150      -0.138
duration:start_hour[T.16]     -0.1509      0.003    -48.848      0.000
-0.157      -0.145
duration:start_hour[T.17]     -0.1592      0.003    -52.251      0.000
-0.165      -0.153
```

```
duration:start_hour[T.18]     -0.1377      0.003    -44.496      0.000
-0.144      -0.132
duration:start_hour[T.19]     -0.0875      0.003    -27.299      0.000
-0.094      -0.081
duration:start_hour[T.20]     -0.0243      0.003     -7.357      0.000
-0.031      -0.018
duration:start_hour[T.21]     -0.0050      0.003     -1.509      0.131
-0.011       0.001
duration:start_hour[T.22]     -0.0039      0.003     -1.166      0.243
-0.010       0.003
duration:start_hour[T.23]      0.0360      0.003     10.525      0.000
0.029       0.043
==============================================================================
Omnibus:                   472872.262   Durbin-Watson:                   2.001
Prob(Omnibus):                  0.000   Jarque-Bera (JB):         55605092.808
Skew:                           1.350   Prob(JB):                         0.00
Kurtosis:                      39.829   Cond. No.                         780.
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
```

[59]:
```
fit = ols(formula="income ~ duration + payment_type  +start_hour",data=df).fit()
print(fit.summary())
```

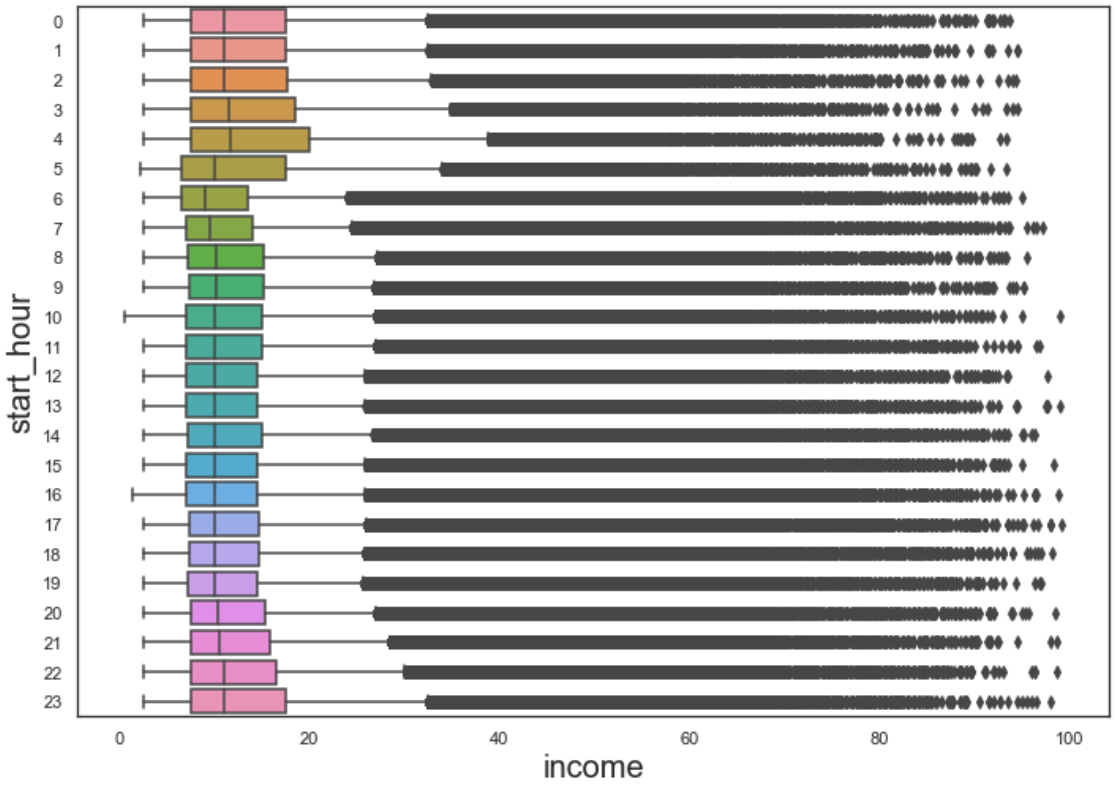
```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 income   R-squared:                       0.800
Model:                            OLS   Adj. R-squared:                  0.800
Method:                 Least Squares   F-statistic:                 1.664e+06
Date:                Fri, 04 Sep 2020   Prob (F-statistic):               0.00
Time:                        20:46:16   Log-Likelihood:            -2.9344e+07
No. Observations:            10397081   AIC:                         5.869e+07
Df Residuals:                10397055   BIC:                         5.869e+07
Df Model:                          25
Covariance Type:            nonrobust
==============================================================================
=====
                         coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
-----
Intercept              3.7478      0.007    542.448      0.000       3.734
3.761
payment_type[T.2]     -2.3755      0.003   -887.310      0.000      -2.381
-2.370
start_hour[T.1]        0.0956      0.010      9.517      0.000       0.076
```

```
0.115
start_hour[T.2]       0.2958      0.011     27.172      0.000       0.275
0.317
start_hour[T.3]       0.6549      0.012     54.249      0.000       0.631
0.679
start_hour[T.4]       1.4539      0.014    106.900      0.000       1.427
1.481
start_hour[T.5]       1.9355      0.015    133.275      0.000       1.907
1.964
start_hour[T.6]       0.4873      0.011     44.514      0.000       0.466
0.509
start_hour[T.7]      -0.9951      0.009   -106.861      0.000      -1.013
-0.977
start_hour[T.8]      -1.9803      0.009   -223.193      0.000      -1.998
-1.963
start_hour[T.9]      -1.8977      0.009   -214.025      0.000      -1.915
-1.880
start_hour[T.10]     -1.6679      0.009   -187.026      0.000      -1.685
-1.650
start_hour[T.11]     -1.7957      0.009   -203.303      0.000      -1.813
-1.778
start_hour[T.12]     -1.7816      0.009   -204.361      0.000      -1.799
-1.764
start_hour[T.13]     -1.6951      0.009   -193.869      0.000      -1.712
-1.678
start_hour[T.14]     -1.8381      0.009   -212.389      0.000      -1.855
-1.821
start_hour[T.15]     -2.0287      0.009   -234.465      0.000      -2.046
-2.012
start_hour[T.16]     -1.8938      0.009   -214.319      0.000      -1.911
-1.876
start_hour[T.17]     -2.0793      0.009   -243.129      0.000      -2.096
-2.063
start_hour[T.18]     -1.9590      0.008   -236.508      0.000      -1.975
-1.943
start_hour[T.19]     -1.4165      0.008   -170.673      0.000      -1.433
-1.400
start_hour[T.20]     -0.8467      0.008   -100.230      0.000      -0.863
-0.830
start_hour[T.21]     -0.4974      0.008    -58.603      0.000      -0.514
-0.481
start_hour[T.22]     -0.3422      0.009    -39.944      0.000      -0.359
-0.325
start_hour[T.23]      0.0212      0.009      2.382      0.017       0.004
0.039
duration              0.8976      0.000   6291.336      0.000       0.897
0.898
==============================================================================
```

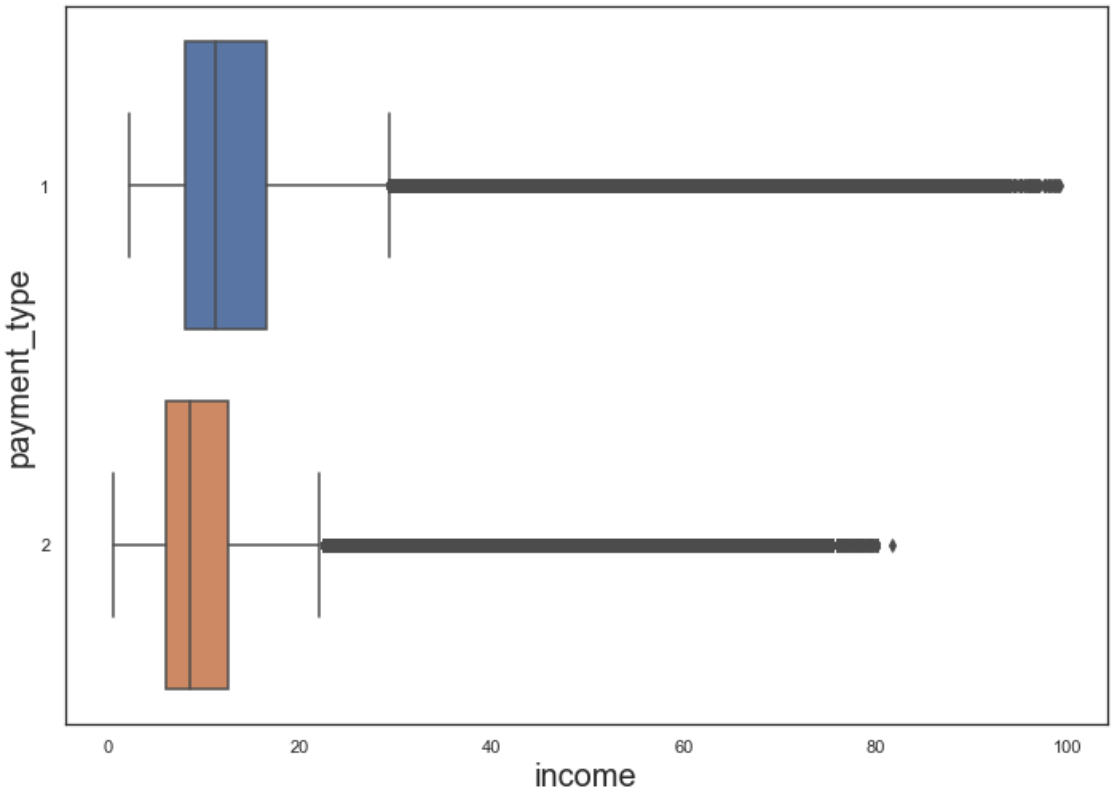
```
Omnibus:                  5137951.303   Durbin-Watson:                   1.883
Prob(Omnibus):                  0.000   Jarque-Bera (JB):        414165231.594
Skew:                           1.507   Prob(JB):                         0.00
Kurtosis:                      33.773   Cond. No.                         404.
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
```

[ ]:

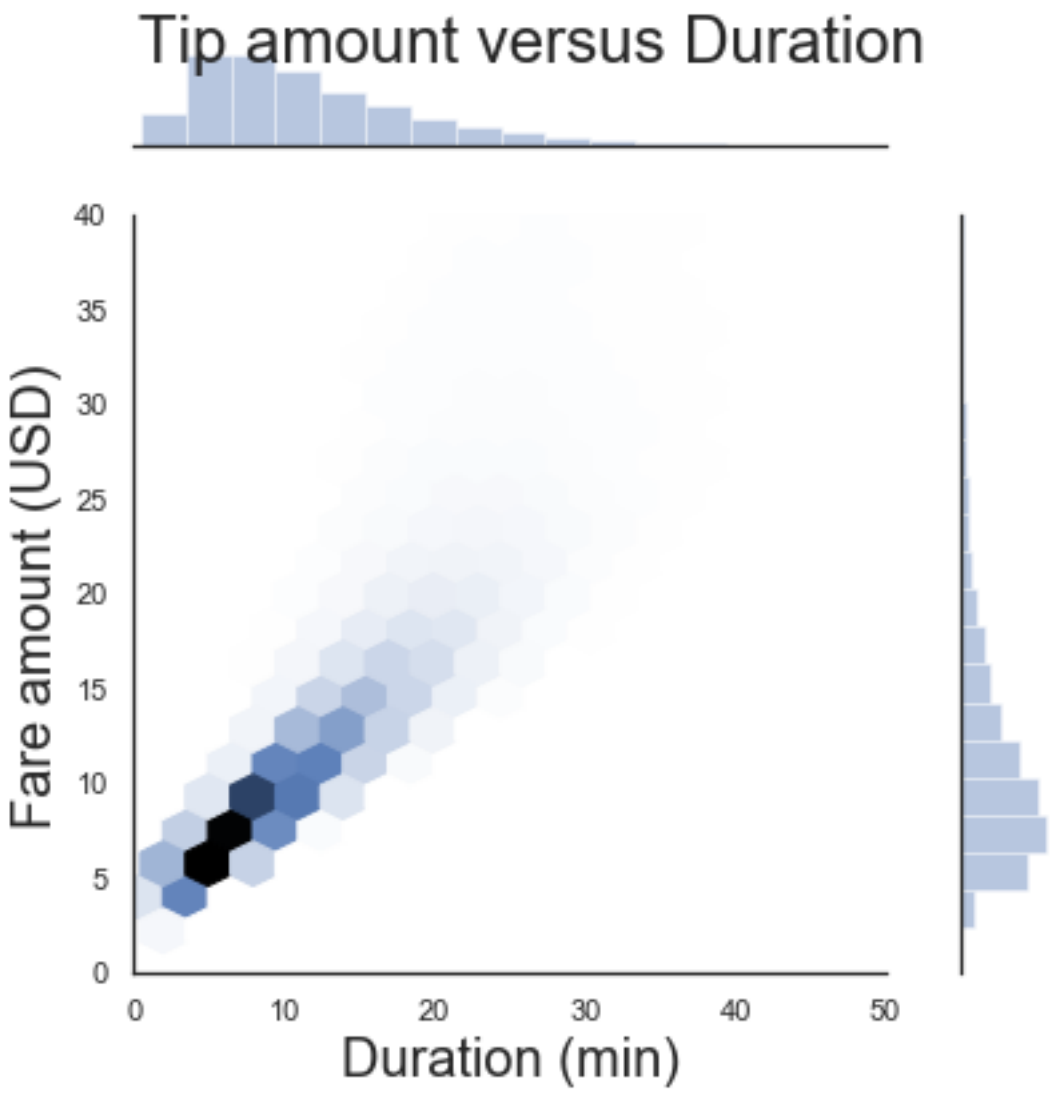
[60]:
```
ax = sns.boxplot(x="income", y="start_hour", data=df)
```



[61]:
```
ax = sns.boxplot(x="income", y="payment_type", data=df)
```

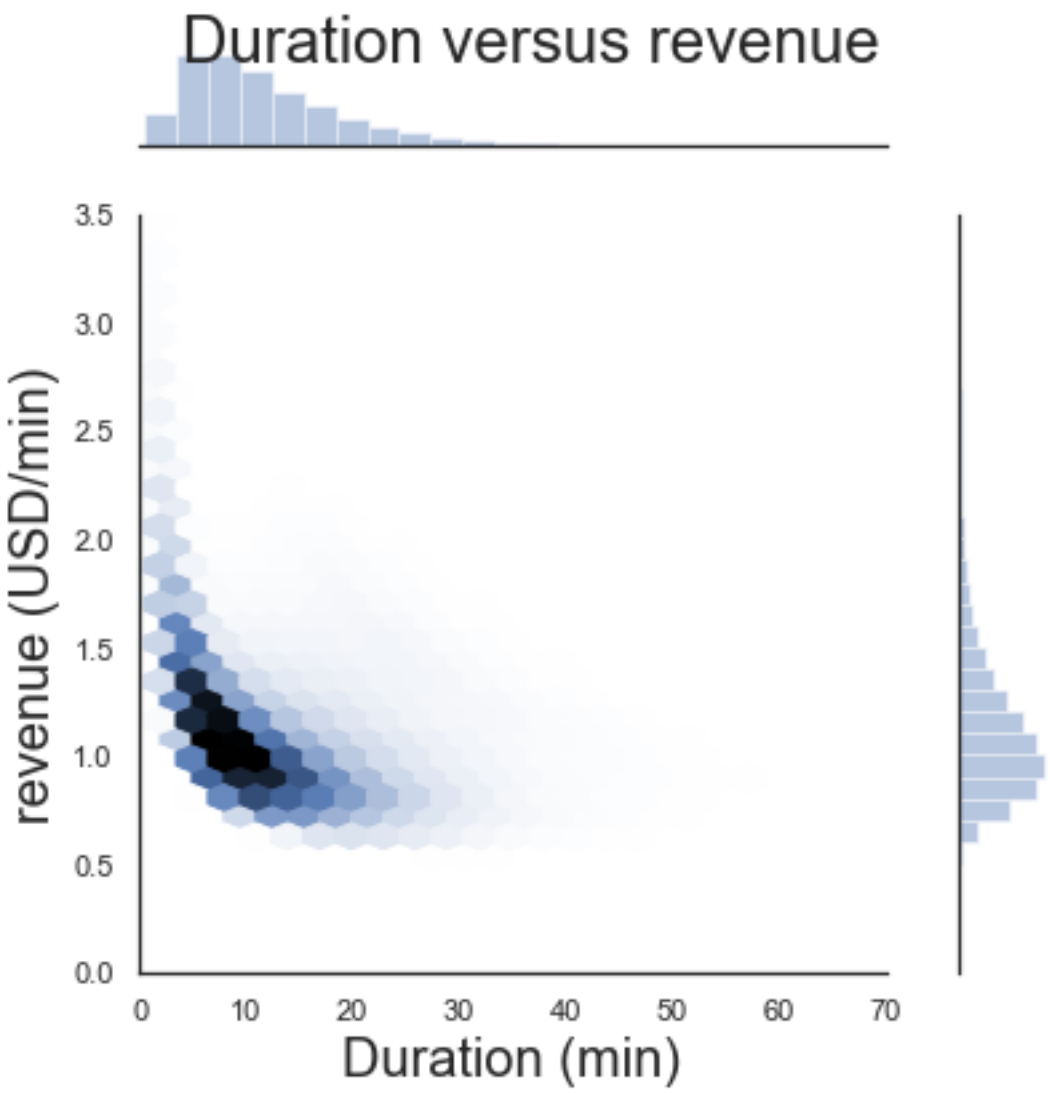


[62]:
```
p = sns.jointplot(x='duration',y= 'income',data= stand_df, kind="hex",␣
↪color="b", xlim=(0,50), ylim=(0,40))
p.ax_joint.set_xlabel('Duration (min)')
p.ax_joint.set_ylabel('Fare amount (USD)')
p.fig.suptitle("Tip amount versus Duration")
p.fig.tight_layout()
```



[ ]:

[63]:
```
p = sns.jointplot(x='duration',y= 'income/duration',data= stand_df, kind="hex",␣
↪color="b", xlim=(0,70), ylim=(0,3.5))
p.ax_joint.set_xlabel('Duration (min)')
p.ax_joint.set_ylabel('revenue (USD/min)')
p.fig.suptitle("Duration versus revenue")
p.fig.tight_layout()
```



[ ]:

[64]:
```
for i in range(24):
    curr_data = stand_df[start_coords].loc[(df['start_hour'] == i) &␣
↪(df['duration'] <= 10)]
    curr = folium.Map(location=[40.75, -73.9], tiles="Stamen Terrain",␣
↪zoom_start=12)
    curr.add_child(HeatMap(curr_data[start_coords].values, radius=10))
    curr.save('plots/start_Heatmap_high_revenue_in' + str(i) + '.html')
```

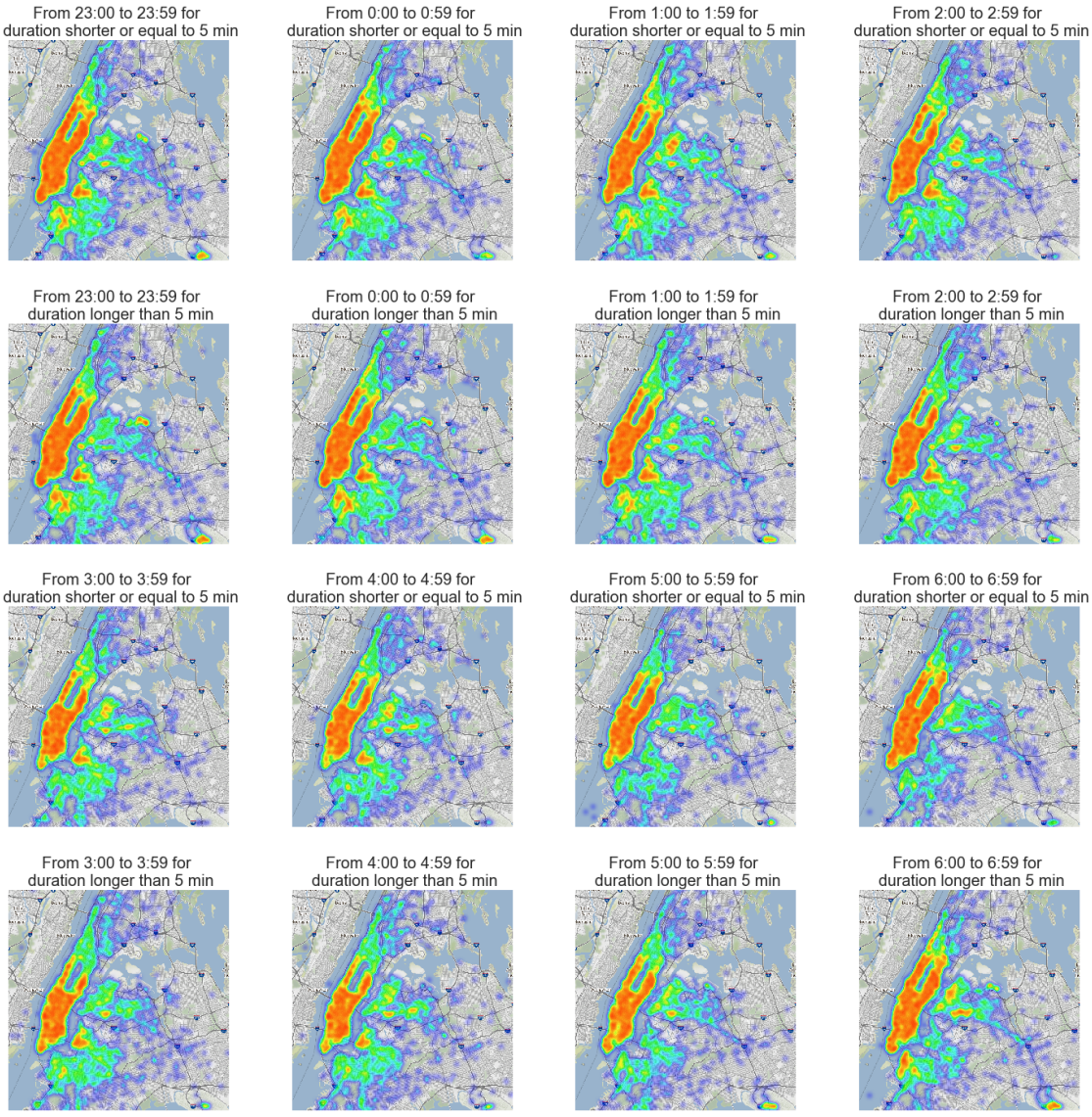
[65]:
```
for i in range(24):
```

```
    curr_data = stand_df[start_coords].loc[(df['start_hour'] == i) &␣
↪(df['duration'] > 10)]
    curr = folium.Map(location=[40.75, -73.9], tiles="Stamen Terrain",␣
↪zoom_start=12)
    curr.add_child(HeatMap(curr_data[start_coords].values, radius=10))
    curr.save('plots/start_Heatmap_low_revenue_in' + str(i) + '.html')
```

[ ]:

[66]:
```
time = [0,1,2,3,4,5,6,23]
time = [str(i) for i in time]
posi = [i for i in range(1,5)] + [i for i in range(9,13)]
```

[67]:
```
plt.figure(figsize=(20, 20))
for i in range(8):
    plt.subplot(4,4,posi[i])
    plt.title("From " + time[i-1] + ":00 to " + time[i-1] + ":59"\
              + " for \n duration shorter or equal to 5 min" )
    img = mpimg.imread("plots/heatmap/high " + time[i-1] +'.png')
    plt.imshow(img)
    plt.axis('off')
    plt.subplot(4,4, posi[i]+4)
    plt.title("From " + time[i-1] + ":00 to " + time[i-1] + ":59"\
              + " for \n duration longer than 5 min" )
    img = mpimg.imread("plots/heatmap/low " + time[i-1] +'.png')
    plt.imshow(img)
    plt.axis('off')
plt.tight_layout()
plt.show()
```


From 23:00 to 23:59 for duration shorter or equal to 5 min
From 0:00 to 0:59 for duration shorter or equal to 5 min
From 1:00 to 1:59 for duration shorter or equal to 5 min
From 2:00 to 2:59 for duration shorter or equal to 5 min
From 23:00 to 23:59 for duration longer than 5 min
From 0:00 to 0:59 for duration longer than 5 min
From 1:00 to 1:59 for duration longer than 5 min
From 2:00 to 2:59 for duration longer than 5 min
From 3:00 to 3:59 for duration shorter or equal to 5 min
From 4:00 to 4:59 for duration shorter or equal to 5 min
From 5:00 to 5:59 for duration shorter or equal to 5 min
From 6:00 to 6:59 for duration shorter or equal to 5 min
From 3:00 to 3:59 for duration longer than 5 min
From 4:00 to 4:59 for duration longer than 5 min
From 5:00 to 5:59 for duration longer than 5 min
From 6:00 to 6:59 for duration longer than 5 min

[ ]:

[ ]: