In [1]:

```
import numpy as np
import pandas as pd
import geopandas as gpd
from shapely.geometry import Point

import matplotlib.pyplot as plt

import json

# Plotly
import chart_studio.plotly as py
import plotly.graph_objs as go
from plotly.offline import download_plotlyjs, init_notebook_mode, plot, iplot
init_notebook_mode(connected=True)
```

Read data

NYC Geometry file

In [2]:

```
nyc_boros = gpd.read_file('./geo/geo_export_b0262261-5940-4b03-b89d-d4eb921ae481.sh
nyc_boros.head(10)
```

Out[2]:

	boro_code	boro_name	county_fip	ntacode	ntaname	shape_area	shape_leng	
0	4.0	Queens	081	QN51	Murray Hill	5.248828e+07	33266.904856	(
1	4.0	Queens	081	QN27	East Elmhurst	1.972685e+07	19816.711894	
2	4.0	Queens	081	QN41	Fresh Meadows-Utopia	2.777485e+07	22106.431272	(
3	1.0	Manhattan	061	MN17	Midtown-Midtown South	3.019153e+07	27032.700375	(
4	2.0	Bronx	005	BX09	Soundview-Castle Hill-Clason Point-Harding Park	5.198380e+07	67340.977626	MULTI ((-7
5	4.0	Queens	081	QN08	St. Albans	7.741275e+07	45401.316898	(
6	3.0	Brooklyn	047	BK69	Clinton Hill	2.052820e+07	23971.466236	(
7	2.0	Bronx	005	BX26	Highbridge	1.645764e+07	18506.310104	(
8	3.0	Brooklyn	047	BK26	Gravesend	3.134195e+07	39922.674490	(
9	3.0	Brooklyn	047	BK46	Ocean Parkway South	1.778210e+07	21975.996042	(

In [3]:

```
nyc_boros.info()
```

```
<class 'geopandas.geodataframe.GeoDataFrame'>
RangeIndex: 195 entries, 0 to 194
Data columns (total 8 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   boro_code   195 non-null    float64
 1   boro_name   195 non-null    object
 2   county_fip  195 non-null    object
 3   ntacode     195 non-null    object
 4   ntaname     195 non-null    object
 5   shape_area  195 non-null    float64
 6   shape_leng  195 non-null    float64
 7   geometry    195 non-null    geometry
dtypes: float64(3), geometry(1), object(4)
memory usage: 12.3+ KB
```

In [4]:

```
nyc_boros.crs
```

Out[4]:

```
<Geographic 2D CRS: GEOGCS["WGS84(DD)",DATUM["WGS84",SPHEROID["WGS84",
...>
Name: WGS84(DD)
Axis Info [ellipsoidal]:
- lon[east]: Longitude (degree)
- lat[north]: Latitude (degree)
Area of Use:
- undefined
Datum: WGS84
- Ellipsoid: WGS84
- Prime Meridian: Greenwich
```

NYC taxis file

In [5]:

```
nyc_taxis = pd.read_csv('clean_nyc_taxis.csv', index_col = 'id')
nyc_taxis.head(10)
```

Out[5]:

	vendor_id	pickup_datetime	dropoff_datetime	passenger_count	pickup_longitude	p
id						
id2875421	2	2016-03-14 17:24:55	2016-03-14 17:32:30	1	-73.982155	
id2377394	1	2016-06-12 00:43:35	2016-06-12 00:54:38	1	-73.980415	
id3858529	2	2016-01-19 11:35:24	2016-01-19 12:10:48	1	-73.979027	
id3504673	2	2016-04-06 19:32:31	2016-04-06 19:39:40	1	-74.010040	
id2181028	2	2016-03-26 13:30:55	2016-03-26 13:38:10	1	-73.973053	
id0801584	2	2016-01-30 22:01:40	2016-01-30 22:09:03	6	-73.982857	
id1813257	1	2016-06-17 22:34:59	2016-06-17 22:40:40	4	-73.969017	
id1324603	2	2016-05-21 07:54:58	2016-05-21 08:20:49	1	-73.969276	
id1301050	1	2016-05-27 23:12:23	2016-05-27 23:16:38	1	-73.999481	
id0012891	2	2016-03-10 21:45:01	2016-03-10 22:05:26	1	-73.981049	

In [6]:

```
nyc_taxis.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 1458640 entries, id2875421 to id1209952
Data columns (total 10 columns):
 #   Column              Non-Null Count    Dtype  
---  ------              --------------    -----  
 0   vendor_id           1458640 non-null  int64  
 1   pickup_datetime     1458640 non-null  object 
 2   dropoff_datetime    1458640 non-null  object 
 3   passenger_count     1458640 non-null  int64  
 4   pickup_longitude    1458640 non-null  float64
 5   pickup_latitude     1458640 non-null  float64
 6   dropoff_longitude   1458640 non-null  float64
 7   dropoff_latitude    1458640 non-null  float64
 8   store_and_fwd_flag  1458640 non-null  object 
 9   trip_duration       1458640 non-null  int64  
dtypes: float64(4), int64(3), object(3)
memory usage: 122.4+ MB
```

Create Points

In [7]:

```
gdf_pickup = gpd.GeoDataFrame(geometry=gpd.points_from_xy(nyc_taxis['pickup_longitu
                              index=nyc_taxis.index, crs=nyc_boros.crs)
gdf_pickup.head(10)
```

Out[7]:

id	geometry
id2875421	POINT (-73.98215 40.76794)
id2377394	POINT (-73.98042 40.73856)
id3858529	POINT (-73.97903 40.76394)
id3504673	POINT (-74.01004 40.71997)
id2181028	POINT (-73.97305 40.79321)
id0801584	POINT (-73.98286 40.74220)
id1813257	POINT (-73.96902 40.75784)
id1324603	POINT (-73.96928 40.79778)
id1301050	POINT (-73.99948 40.73840)
id0012891	POINT (-73.98105 40.74434)

In [8]:

```
gdf_dropoff = gpd.GeoDataFrame(geometry=gpd.points_from_xy(nyc_taxis['dropoff_longi
                              index=nyc_taxis.index, crs=nyc_boros.crs)
gdf_dropoff.head(10)
```

Out[8]:

	geometry
id	
id2875421	POINT (-73.96463 40.76560)
id2377394	POINT (-73.99948 40.73115)
id3858529	POINT (-74.00533 40.71009)
id3504673	POINT (-74.01227 40.70672)
id2181028	POINT (-73.97292 40.78252)
id0801584	POINT (-73.99208 40.74918)
id1813257	POINT (-73.95741 40.76590)
id1324603	POINT (-73.92247 40.76056)
id1301050	POINT (-73.98579 40.73281)
id0012891	POINT (-73.97300 40.78999)

1.7 Boroughs

Question 1.7.1

Neighbourhoods for the trip start

In [9]:

```
# Join pickup points with nyc_boros to find neighborhoods/boroughs
trip_start_boros= gpd.sjoin(gdf_pickup, nyc_boros, how='left', predicate = 'within'
```

In [10]:

```
join_trip_start = nyc_taxis.join(trip_start_boros, how='left')
nyc_taxis['trip_start_boro']= join_trip_start['boro_name']
nyc_taxis['trip_start_ntaname'] = join_trip_start['ntaname']
nyc_taxis.head(10)
```

Out[10]:

	vendor_id	pickup_datetime	dropoff_datetime	passenger_count	pickup_longitude	p
id						
id2875421	2	2016-03-14 17:24:55	2016-03-14 17:32:30	1	-73.982155	
id2377394	1	2016-06-12 00:43:35	2016-06-12 00:54:38	1	-73.980415	
id3858529	2	2016-01-19 11:35:24	2016-01-19 12:10:48	1	-73.979027	
id3504673	2	2016-04-06 19:32:31	2016-04-06 19:39:40	1	-74.010040	
id2181028	2	2016-03-26 13:30:55	2016-03-26 13:38:10	1	-73.973053	
id0801584	2	2016-01-30 22:01:40	2016-01-30 22:09:03	6	-73.982857	
id1813257	1	2016-06-17 22:34:59	2016-06-17 22:40:40	4	-73.969017	
id1324603	2	2016-05-21 07:54:58	2016-05-21 08:20:49	1	-73.969276	
id1301050	1	2016-05-27 23:12:23	2016-05-27 23:16:38	1	-73.999481	
id0012891	2	2016-03-10 21:45:01	2016-03-10 22:05:26	1	-73.981049	

Neighbourhoods for the trip end

In [11]:

```
# Join dropoff points with nyc_boros to find neighborhoods/boroughs
trip_end_boros = gpd.sjoin(gdf_dropoff, nyc_boros, how='left', predicate = 'within'
```

In [12]:

```
join_trip_end = nyc_taxis.join(trip_end_boros, how='left')
nyc_taxis['trip_end_boro'] = join_trip_end['boro_name']
nyc_taxis['trip_end_ntaname'] = join_trip_end['ntaname']
nyc_taxis.head(10)
```

Out[12]:

	vendor_id	pickup_datetime	dropoff_datetime	passenger_count	pickup_longitude	p
id						
id2875421	2	2016-03-14 17:24:55	2016-03-14 17:32:30	1	-73.982155	
id2377394	1	2016-06-12 00:43:35	2016-06-12 00:54:38	1	-73.980415	
id3858529	2	2016-01-19 11:35:24	2016-01-19 12:10:48	1	-73.979027	
id3504673	2	2016-04-06 19:32:31	2016-04-06 19:39:40	1	-74.010040	
id2181028	2	2016-03-26 13:30:55	2016-03-26 13:38:10	1	-73.973053	
id0801584	2	2016-01-30 22:01:40	2016-01-30 22:09:03	6	-73.982857	
id1813257	1	2016-06-17 22:34:59	2016-06-17 22:40:40	4	-73.969017	
id1324603	2	2016-05-21 07:54:58	2016-05-21 08:20:49	1	-73.969276	
id1301050	1	2016-05-27 23:12:23	2016-05-27 23:16:38	1	-73.999481	
id0012891	2	2016-03-10 21:45:01	2016-03-10 22:05:26	1	-73.981049	

Question 1.7.2

In [13]:

```
with open('./geo/2010 Neighborhood Tabulation Areas (NTAs).geojson') as file:
    boroughs = json.load(file)
```

Chloropleth of pickups

In [14]:

```
# Add value counts for the start/pickup for each neighbourhood
temp_df = pd.DataFrame({'ntaname':nyc_taxis['trip_start_ntaname'].value_counts(sort
                       'start_ntaname_counts':nyc_taxis['trip_start_ntaname'].value

nyc_boros = pd.merge(nyc_boros,temp_df, on='ntaname', how='left' )
nyc_boros['start_ntaname_counts'].fillna(0.0)
nyc_boros.head()
```

Out[14]:

	boro_code	boro_name	county_fip	ntacode	ntaname	shape_area	shape_leng	
0	4.0	Queens	081	QN51	Murray Hill	5.248828e+07	33266.904856	(
1	4.0	Queens	081	QN27	East Elmhurst	1.972685e+07	19816.711894	
2	4.0	Queens	081	QN41	Fresh Meadows-Utopia	2.777485e+07	22106.431272	(
3	1.0	Manhattan	061	MN17	Midtown-Midtown South	3.019153e+07	27032.700375	(
4	2.0	Bronx	005	BX09	Soundview-Castle Hill-Clason Point-Harding Park	5.198380e+07	67340.977626	MULTI ((-7

In [15]:

```
data = dict(
    type ='choropleth', locations=nyc_boros['ntaname'],featureidkey="properties.nta
    locationmode='geojson-id', geojson=boroughs,
    z = nyc_boros['start_ntaname_counts'],
    colorbar={'title':'Number of pickups'},
)

layout = dict(
    title='Number of all pickups in NYC',
    geo = dict(
        showframe=False,
        projection = {'type':'mercator'},
        visible=False,
        fitbounds = 'locations'
    )
)
```

In [16]:

```
chloromap = go.Figure(data=[data], layout=layout)
iplot(chloromap, validate=False)
```

Here we can see that there is a high distribution of pickups in Midtown-Midtown South which is located in Manhattan.

Chloropleth of dropoffs

In [17]:

```
# Add value counts for the end/dropoff for each neighbourhood
temp_df = pd.DataFrame({'ntaname':nyc_taxis['trip_end_ntaname'].value_counts(sort=F
                       'end_ntaname_counts':nyc_taxis['trip_end_ntaname'].value_cou

nyc_boros = pd.merge(nyc_boros,temp_df, on='ntaname', how='left' )
nyc_boros['end_ntaname_counts'].fillna(0.0)
nyc_boros.head()
```

Out[17]:

	boro_code	boro_name	county_fip	ntacode	ntaname	shape_area	shape_leng	
0	4.0	Queens	081	QN51	Murray Hill	5.248828e+07	33266.904856	(
1	4.0	Queens	081	QN27	East Elmhurst	1.972685e+07	19816.711894	
2	4.0	Queens	081	QN41	Fresh Meadows-Utopia	2.777485e+07	22106.431272	(
3	1.0	Manhattan	061	MN17	Midtown-Midtown South	3.019153e+07	27032.700375	(
4	2.0	Bronx	005	BX09	Soundview-Castle Hill-Clason Point-Harding Park	5.198380e+07	67340.977626	MULTI ((-7

In [18]:

```
data_2 = dict(
    type ='choropleth', locations=nyc_boros['ntaname'],featureidkey="properties.nta
    locationmode='geojson-id', geojson=boroughs,
    z = nyc_boros['end_ntaname_counts'],
    colorbar={'title':'Number of pickups'},
)

layout_2 = dict(
    title='Number of all dropoffs in NYC',
    geo = dict(
        showframe=False,
        projection = {'type':'mercator'},
        visible=False,
        fitbounds = 'locations'
    )
)
```

In [19]:

```
chloromap2 = go.Figure(data=[data_2], layout=layout_2)
iplot(chloromap2, validate=False)
```

It is the same for the dropoffs there is a high number of dropoffs in Midtown-Midtown South which is in Manhattan. There is also less dropoffs 30k at the airport than pickups 68k.

Question 1.7.3

In [20]:

```
nyc_boros.groupby(['boro_name'])['start_ntaname_counts'].sum()
```

Out[20]:

```
boro_name
Bronx               1257.0
Brooklyn           26394.0
Manhattan        1343791.0
Queens             85918.0
Staten Island         57.0
Name: start_ntaname_counts, dtype: float64
```

We can see that Manhattan has the most outgoing trips

In [21]:

```
nyc_boros.groupby(['boro_name'])['end_ntaname_counts'].sum()
```

Out[21]:

```
boro_name
Bronx               9389.0
Brooklyn           77516.0
Manhattan        1288937.0
Queens             76089.0
Staten Island        376.0
Name: end_ntaname_counts, dtype: float64
```

We can see that Manhattan also has the most outgoing trips

Question 1.7.4 & 17.5

In [32]:

```
nyc_taxis['pickup_datetime'] = pd.to_datetime(nyc_taxis['pickup_datetime'])
nyc_taxis['dropoff_datetime'] = pd.to_datetime(nyc_taxis['dropoff_datetime'])
```

In [23]:

```
# Create timesamps
midnight = pd.Timestamp(2018, 1, 5, 0).time()
five_am = pd.Timestamp(2018, 1, 5, 5).time()
```

In [24]:

```
bool_pickup = (nyc_taxis['pickup_datetime'].dt.time >= midnight) & (nyc_taxis['pick
bool_dropoff = (nyc_taxis['dropoff_datetime'].dt.time >= midnight) & (nyc_taxis['dr
```

In [30]:

```
nyc_taxis[bool_dropoff]['trip_end_boro'].value_counts()
```

Out[30]:

```
Manhattan        125556
Brooklyn          23756
Queens            15340
Bronx              2817
Staten Island       100
Name: trip_end_boro, dtype: int64
```

In [31]:

```
nyc_taxis[bool_pickup]['trip_start_boro'].value_counts()
```

Out[31]:

```
Manhattan        140581
Brooklyn           7756
Queens             7589
Bronx               279
Staten Island         7
Name: trip_start_boro, dtype: int64
```

From above we can see that Manhattan is the busiest at night and Staten Island is the quietest at night