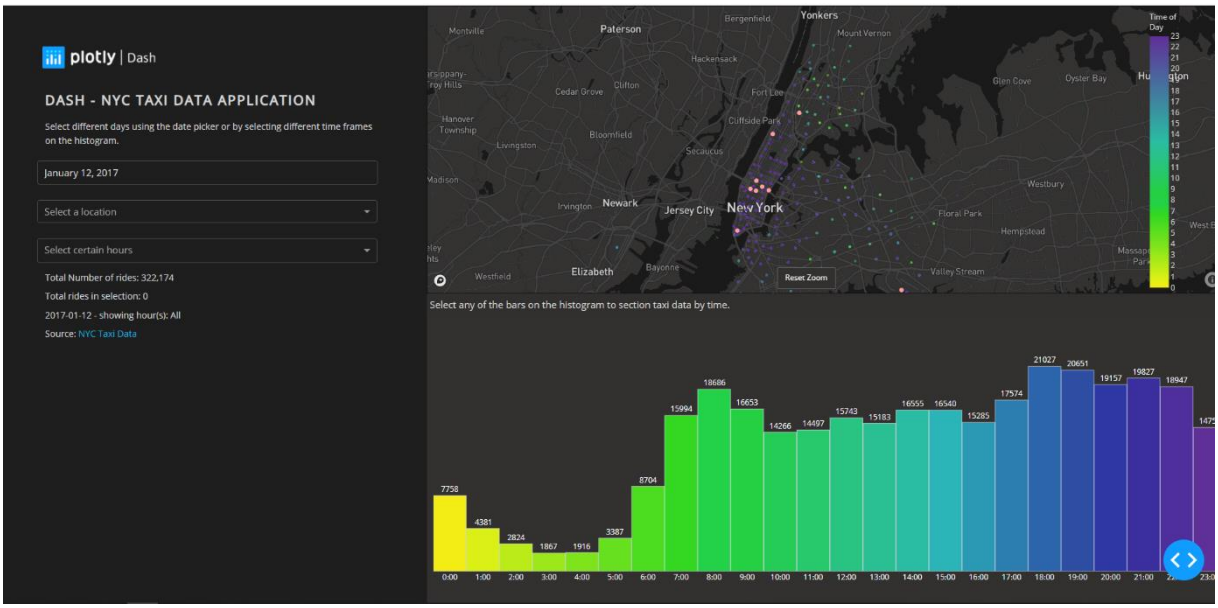
CS 405 Computer Graphics Data Visualization Project with Dash
Sadi Çelik 25542



In this project, I used January 2017 NYC Yellow Taxi Trip Record Data from:
https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page

```
# Download the Trip Record Data (January 2017)
"""urllib.request.urlretrieve("https://s3.amazonaws.com/nyc-tlc/trip+data/"+ \
                              "yellow_tripdata_2017-01.csv",
                              "nyc.2017-01.csv")"""
```

Also used the shape files of NYC zones to get real latitude and longitude values.

```
# Download the Zone Location Data
"""urllib.request.urlretrieve("https://s3.amazonaws.com/nyc-
tlc/misc/taxi_zones.zip", "taxi_zones.zip")
with zipfile.ZipFile("taxi_zones.zip","r") as zip_ref:
    zip_ref.extractall("./shape")"""
```

I converted them into a single csv file with WGS84 output format using:
https://mygeodata.cloud/converter/

To draw a real map, I created an account on MapBox and used my token in my code.

```
# Plotly mapbox public token
mapbox_access_token = "pk.eyJ1Ijoic2FkaWNlbGxpIiwiYSI6ImNra2M5OXhsbDAzbDQyb3A3
bzhud2F5bWoifQ.EkLdQqMa9lWWTE60bALVJA"
```

I recorded the coordinates of important locations in New York City with a dictionary.

```
# Dictionary of important locations in New York
list_of_locations = {
    "Madison Square Garden": {"lat": 40.7505, "lon": -73.9934},
    "Yankee Stadium": {"lat": 40.8296, "lon": -73.9262},
    "Empire State Building": {"lat": 40.7484, "lon": -73.9857},
    "New York Stock Exchange": {"lat": 40.7069, "lon": -74.0113},
    "JFK Airport": {"lat": 40.644987, "lon": -73.785607},
```

```
    "Grand Central Station": {"lat": 40.7527, "lon": -73.9772},
    "Times Square": {"lat": 40.7589, "lon": -73.9851},
    "Columbia University": {"lat": 40.8075, "lon": -73.9626},
    "United Nations HQ": {"lat": 40.7489, "lon": -73.9680},
}
```

Before starting, I checked my data and found that there were some missing values. So, I dropped unnecessary columns.

```
# Initialize NYC Taxi Data
df = pd.read_csv("nyc.2017-01.csv")
datetime_df = df.drop(columns=["VendorID", "passenger_count", "trip_distance",
"RatecodeID","store_and_fwd_flag", "DOLocationID", "payment_type",
"fare_amount","extra", "mta_tax", "tip_amount", "tolls_amount",
"improvement_surcharge", "total_amount"])
datetime_df["tpep_pickup_datetime"] = pd.to_datetime(datetime_df["tpep_pickup_
datetime"], format="%Y-%m-%d %H:%M")
datetime_df["tpep_dropoff_datetime"] = pd.to_datetime(datetime_df["tpep_dropof
f_datetime"], format="%Y-%m-%d %H:%M")
datetime_df.rename(columns={"PULocationID":"LocationID"}, inplace=True)
```

Later converted objects to datetimes to perform calculations more efficiently. Then I merged my datetime_df and zones_df into a single df to plot my histogram and map.

```
# Initialize Location Data
zones_df = pd.read_csv("taxi_zones.csv")
merged_df = pd.merge(datetime_df, zones_df, on="LocationID")
merged_df.index = merged_df["tpep_pickup_datetime"]
merged_df.drop("tpep_pickup_datetime", 1, inplace=True)
merged_df.drop(columns=["OBJECTID", "Shape_Leng", "Shape_Area"], inplace=True)
merged_df.rename(columns={"X":"Lon", "Y":"Lat"}, inplace=True)
```

I created app-callbacks to handle my inputs and outputs. Also updated my map and histogram.

In my histogram I plotted the hourly breakdown of total taxi rides. You can both select on the histogram and from the dropdown on left. Also, you can change the day of the plots with the calendar. (Data from January 1, 2017 to January 31, 2017 is available).

In my map graph I used Map Box's dark mapping. There is a bar on the right which indicates the hours of a day from yellow to purple. With the extracted zone coordinates I showed the most active hour for each region. (example: if a region is purple it means it is most active at 23:00). You can go to some famous places in NYC by using select location dropdown.

Also, there is section under the dropdowns which shows the total rides for selected day and its selected hour breakdown.

Video of my application:
https://drive.google.com/file/d/1gWSlvMauhmPDCypHQmqZyToBdI2bMHyi/view?usp=sharing