Question 4

Group 19

Abhisek Mohanty
Abhishek Nigam

Loading all datasets

In [2]:
```
import matplotlib.pyplot as plt
import pandas as pd
import mplleaflet

%matplotlib inline
```

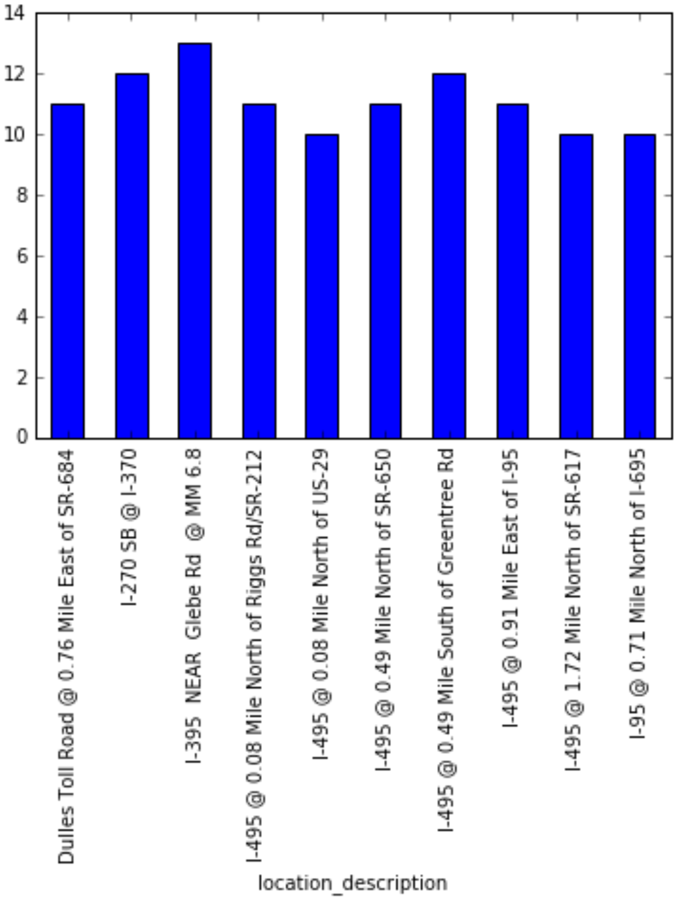
In [3]:
```
q1_data = pd.read_csv('datasets/detector_lane_inventory.tsv', delimiter='\t')
q2_data = pd.read_csv('datasets/events_train_holdout.tsv', delimiter='\t', error_bad_lines=False)
q3_data = pd.read_csv('datasets/cleaning_test_06_09.tsv', delimiter='\t')
```

```
Skipping line 45149: expected 13 fields, saw 15
```

Q1 data

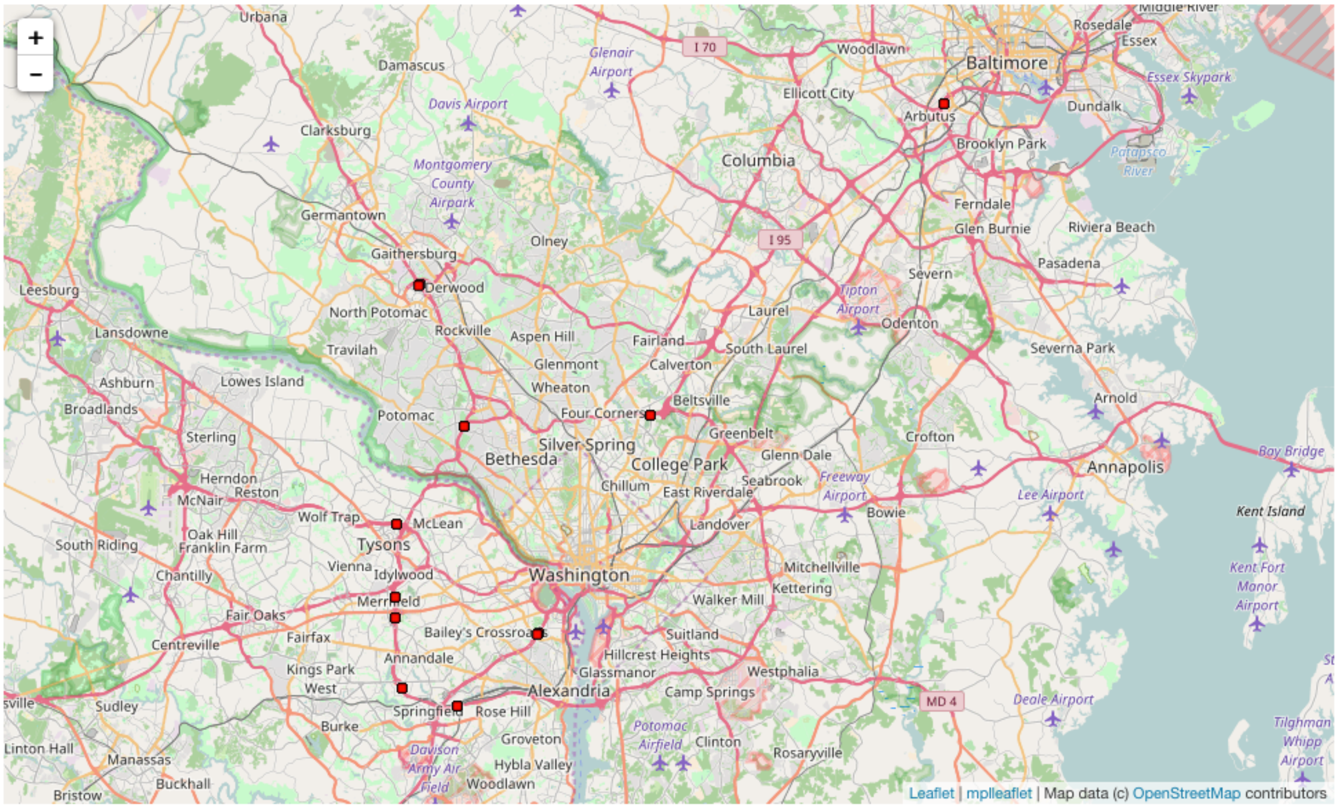
This data shows the most frequently used locations in the city. The graph builds on 10 such highly used locations/roads and points out the number of occourances of these roads in the dataset.

In [9]:
```
max_used = q1_data.groupby('location_description').size().sort_values().tail(10)
topusedroads = df_lane[df_lane['location_description'].isin(max_used.keys())]
topusedroads.groupby('location_description').size().plot.bar()
```



We are the mapping this data on the city map to find these spots which would be benificial in identifying most commonly used roads within the city.

In [10]:
```
plt.figure(figsize=(10,10))
plt.hold(True)
plt.plot(topusedroads['longitude'], topusedroads['latitude'], 'rs') # D
raw red squares
mplleaflet.display()
```



Q2 data

In the first graph we are plotting the Incidents reported aong the hours of the day. This shows us that the graphs peaks at 9am-10am n the morning and 4-5 pm in the evening. This is in correlation with the hoffice hours during which people commute i.e when more people are travelling for office the number of incidents are more. These incidents include traffic congestion, vechile faliure, obstruction etc.

```
In [15]: clean_TS = q2_data[q2_data["created_tstamp"].isnull() == False]
         clean_TS = clean_TS[clean_TS["created_tstamp"] != "NaN"]
         clean_TS = clean_TS[clean_TS["created_tstamp"] != "0"]
         clean_TS = clean_TS[clean_TS["created_tstamp"] != "1"]
         clean_TS = clean_TS[clean_TS["created_tstamp"] != "2"]
         clean_TS = clean_TS[clean_TS["created_tstamp"] != "3"]
         clean_TS = clean_TS[clean_TS["created_tstamp"] != "4"]
         clean_TS = clean_TS[clean_TS["created_tstamp"] != "5"]
```

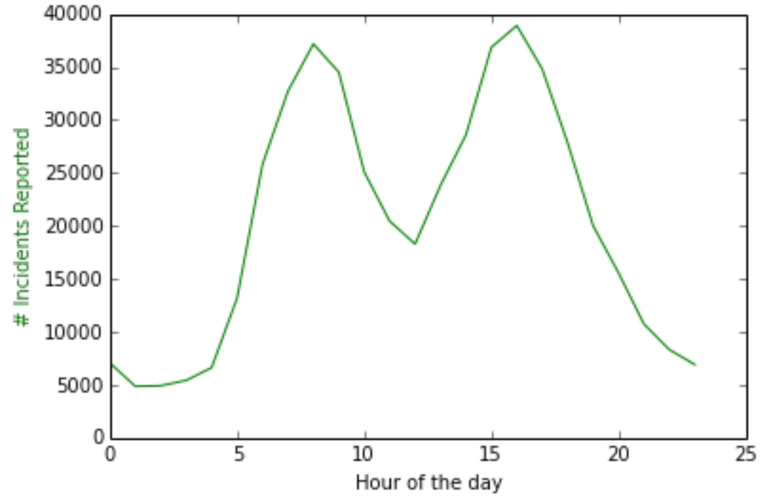
```
In [16]: clean_TS['hours'] = clean_TS.apply(lambda row: float(row['created_tstam
         p'].split("T")[1].split(":")[0]), axis=1)
         hour_grouped = clean_TS.groupby('hours')
```

In [17]:

```
fig, ax1 = plt.subplots()
x = hour_grouped.size().index
ax1.plot(x, hour_grouped.size(), 'g-')
ax1.set_xlabel('Hour of the day')
ax1.set_ylabel('# Incidents Reported', color='g')
```

Out[17]:

```
<matplotlib.text.Text at 0x7f3259da4810>
```



In the second graph, we drill down the incident level further. We try to plot the vechile accidents which have happened along the city. This doesn't include any other type of incident. The graph again peaks at around 8am-10am in the morning and 5pm-7pm in the evening. This is consistent with the office hours of the people when the rush is more.

In [18]:

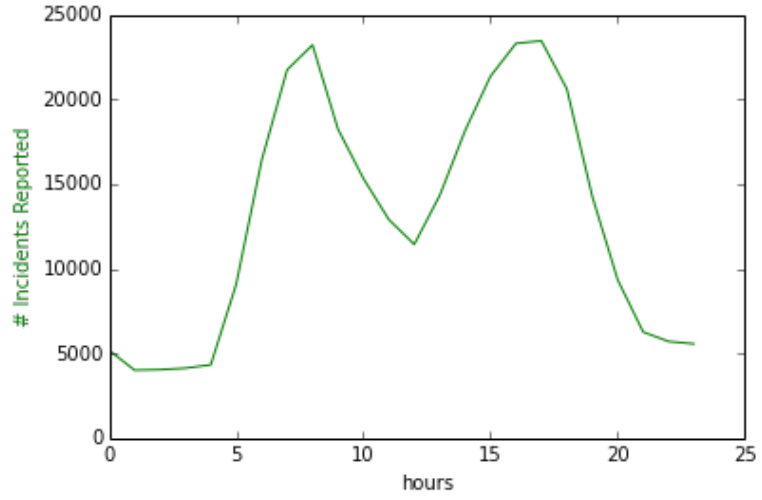
```
clean_TS_AI = clean_TS[clean_TS['event_type'] == 'accidentsAndIncidents
']
AI_grouped = clean_TS_AI.groupby('hours')
```

In [19]:

```
fig, ax1 = plt.subplots()
x = AI_grouped.size().index
ax1.plot(x, AI_grouped.size(), 'g-')
ax1.set_xlabel('hours')
ax1.set_ylabel('# Incidents Reported', color='g')
```

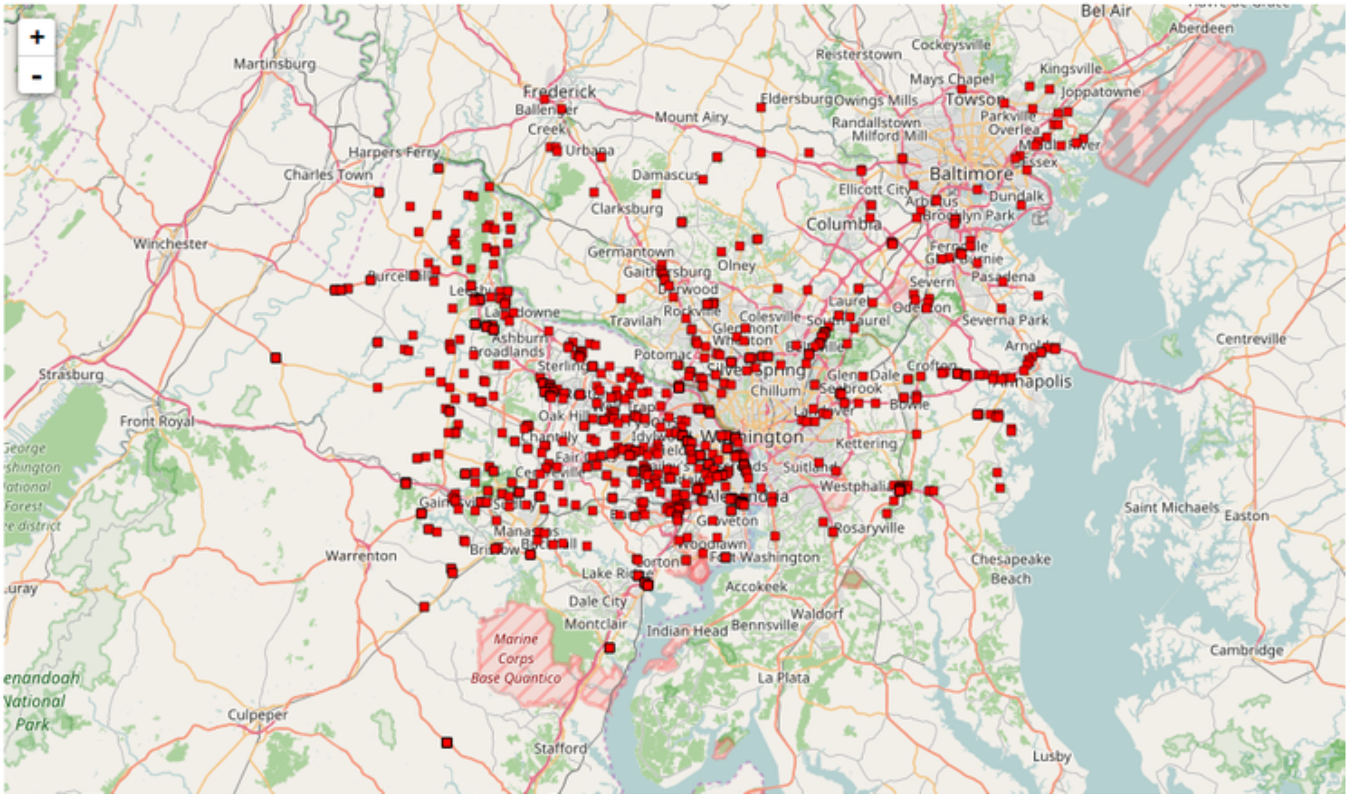
Out[19]:

```
<matplotlib.text.Text at 0x7f3259d265d0>
```



This is a very interesting grph which shows where does these accidents occour. We can analyze the density of this data to find out the areas which have a lot of accidents. Moreover, by providing this data to the city administration, we can take steps to reduce these number of accidents.

In [ ]:
```
plt.figure(figsize=(10,10))
plt.hold(True)
plt.plot(clean_TS_AI['longitude'], clean_TS_AI['latitude'], 'rs') # Dra
w red squares
mplleaflet.display()
```



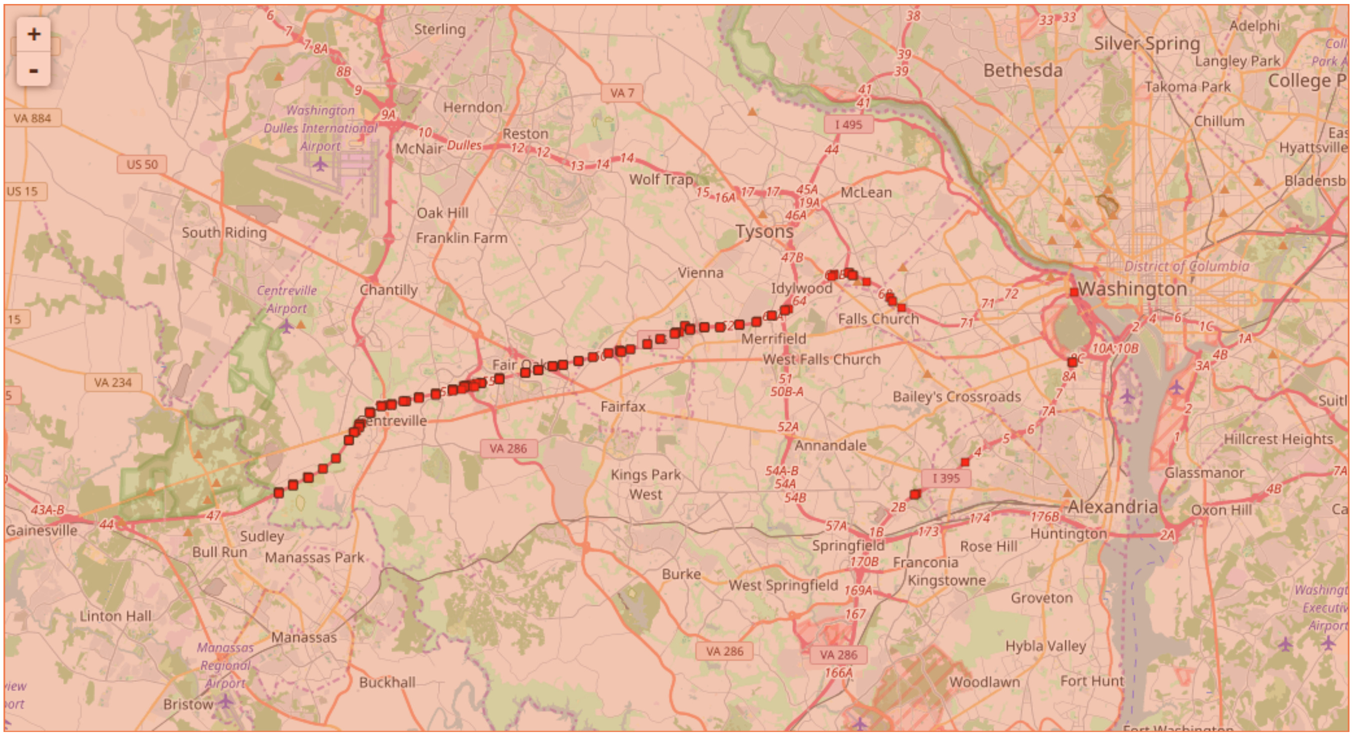
Q3 data

We first remove the negative speeds.
Do a groupby 'lane_id' column.
Calculate Average speeds for each lane.
Join with the map dataset to get the latitude and longitude of the lane_id
Plot the points with the lowest speeds in the first map and then with the highest flow in the second map

In [5]:
```
q3_data = q3_data[q3_data['speed'] > 0]
group_q3_data = q3_data.groupby(['lane_id'])
```

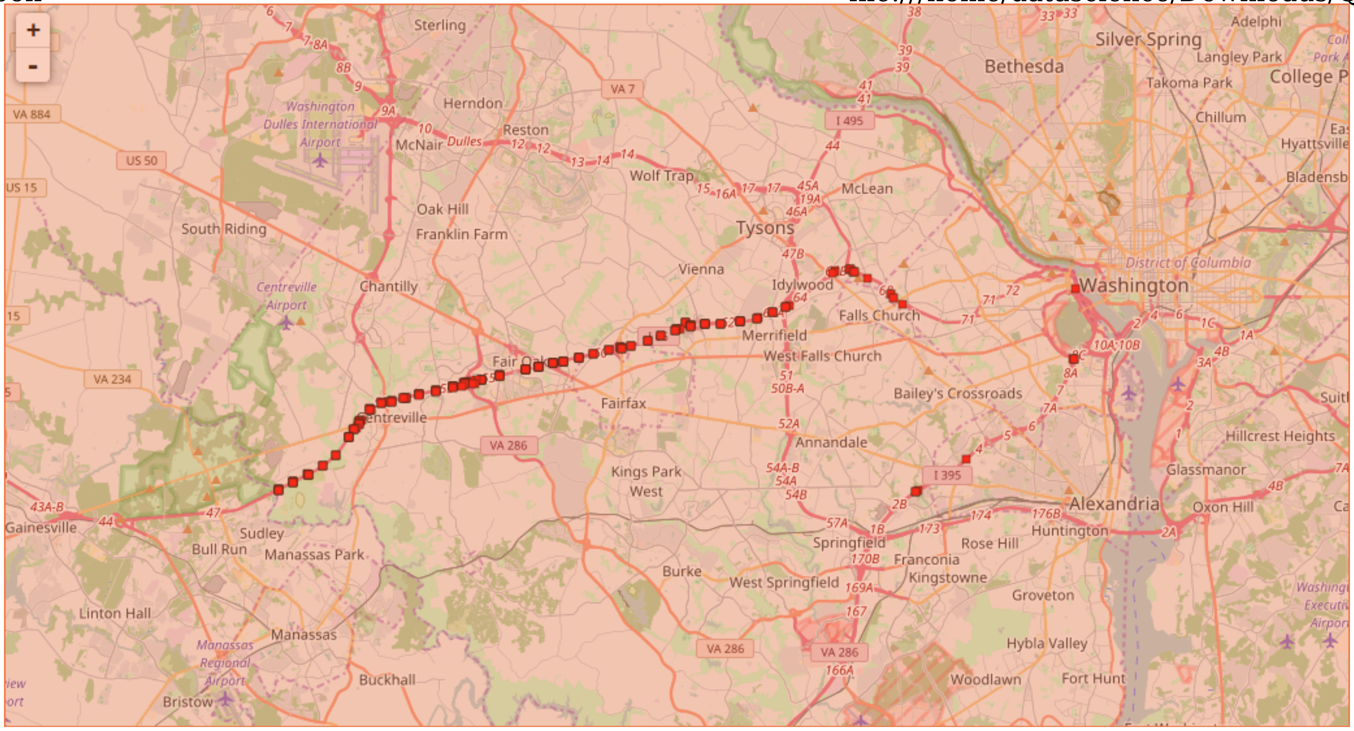
In [11]:

```
mean_speeds_by_lane = pd.DataFrame(group_q3_data.speed.mean()).reset_in
dex()
joined_maps_speed = pd.merge(mean_speeds_by_lane, q1_data, how='inner')
joined_maps_speed = joined_maps_speed.sort(['speed'])

plt.figure(figsize=(10,10))
plt.hold(True)
plt.plot(joined_maps_speed['longitude'], joined_maps_speed['latitude'],
 'rs') # Draw red squares
mplleaflet.display()
```



In [2]:

```
total_flow_by_lane = pd.DataFrame(group_q3_data.flow.count().reset_inde
x())
joined_maps_flow = pd.merge(total_flow_by_lane, q1_data, how='inner')
joined_maps_flow = joined_maps_flow.sort(['flow'])

plt.figure(figsize=(10,10))
plt.hold(True)
plt.plot(joined_maps_flow['longitude'], joined_maps_flow['latitude'], '
rs') # Draw red squares
mplleaflet.display()
```



As we can see above, the locations with the min speeds have the maximum flow

```
In [ ]:
```