Mapping Accidents in NYC - Data Bootcamp Project 2018

NYC Open Data

Ian Glennon | May 2018

Contents

1. Overview

What impacts accidents? In particular, what are the major driver-related reasons for accidents, when do they occur, and where do they occur?

Better understanding the factors that lead to accidents can help officials in two main ways. First, this can aid in the allocation of resources. This relates to both city planning - i.e. where to locate hospitals and police stations - as well as city operations - i.e. where and when to place ambulance and police teams on standby. Second, this can help officials and individuals target behaviour. Government can better inform citizens what they should be aware of as well as launch programs to incentivize targeted good behaviours and restrict targeted bad behaviours.

This project looks to leverage NYPD Motor Vehicle Collision data located on NYC Open Data (https://data.cityofnewyork.us/Public-Safety/NYPD-Motor-Vehicle-Collisions/h9gi-nx95/data). It parses collision data for the period of March 21, 2017 to March 21,2018 from the five Boroughs of NYC.

2. Data Source

2.1 Data and Module Import

NYC Open Data is a project aimed at making NYC government data freely available. It is founded thanks to a partnership between The Mayor’s Office of Data Analytics (MODA), the Department of Information Technology and Telecommunications (DoITT), and Socrata. While vast amount of data are available on this site, I chose to look at NYPD Motor Vehicle Collision data.

The data was obtained through a json query via https://data.cityofnewyork.us/Public-Safety/NYPD-Motor-Vehicle-Collisions/h9gi-nx95/data (https://data.cityofnewyork.us/Public-Safety/NYPD-Motor-Vehicle-Collisions/h9gi-nx95/data)

In [1]:
```
#Importing all required libraries
import pandas as pd
import matplotlib.pyplot as plt     # graphics package
import matplotlib as mpl            # graphics package
import numpy as np
import datetime
import matplotlib.patches as mpatches

#Fetching Data from NYC Open Data
url = 'https://data.cityofnewyork.us/resource/qiz3-axqb.json?$limit=500000'
accidents = pd.read_json(url)

accidents.dtypes
```

Out[1]:
```
borough                                   object
contributing_factor_vehicle_1             object
contributing_factor_vehicle_2             object
contributing_factor_vehicle_3             object
contributing_factor_vehicle_4             object
contributing_factor_vehicle_5             object
cross_street_name                         object
date                              datetime64[ns]
latitude                                 float64
location                                  object
longitude                                float64
number_of_cyclist_injured                  int64
number_of_cyclist_killed                   int64
number_of_motorist_injured                 int64
number_of_motorist_killed                  int64
number_of_pedestrians_injured              int64
number_of_pedestrians_killed               int64
number_of_persons_injured                  int64
number_of_persons_killed                   int64
off_street_name                           object
on_street_name                            object
time                                      object
unique_key                                 int64
vehicle_type_code1                        object
vehicle_type_code2                        object
vehicle_type_code_3                       object
vehicle_type_code_4                       object
vehicle_type_code_5                       object
zip_code                                  object
dtype: object
```

In [2]:
```
#Getting borough data for per capita comparisons
url2 = 'https://data.cityofnewyork.us/resource/h2bk-zmw6.json'
boroughpop = pd.read_json(url2)
boroughpop = boroughpop.sort_values(by='borough', ascending=False) #importa
nt for later per capita calculations
boroughpop.dtypes
```

Out[2]:
```
borough            object
sum_population      int64
dtype: object
```

2.2 Data Information

We can see here that there is a lot of information in the accidents file. The important data for the analysis are borough, date and time information, number of people killed and injured, and accident reason. We are also only interested in accidents across one full year, so as not to have overlap in analysis.

In [3]:
```
accidents.dtypes
```

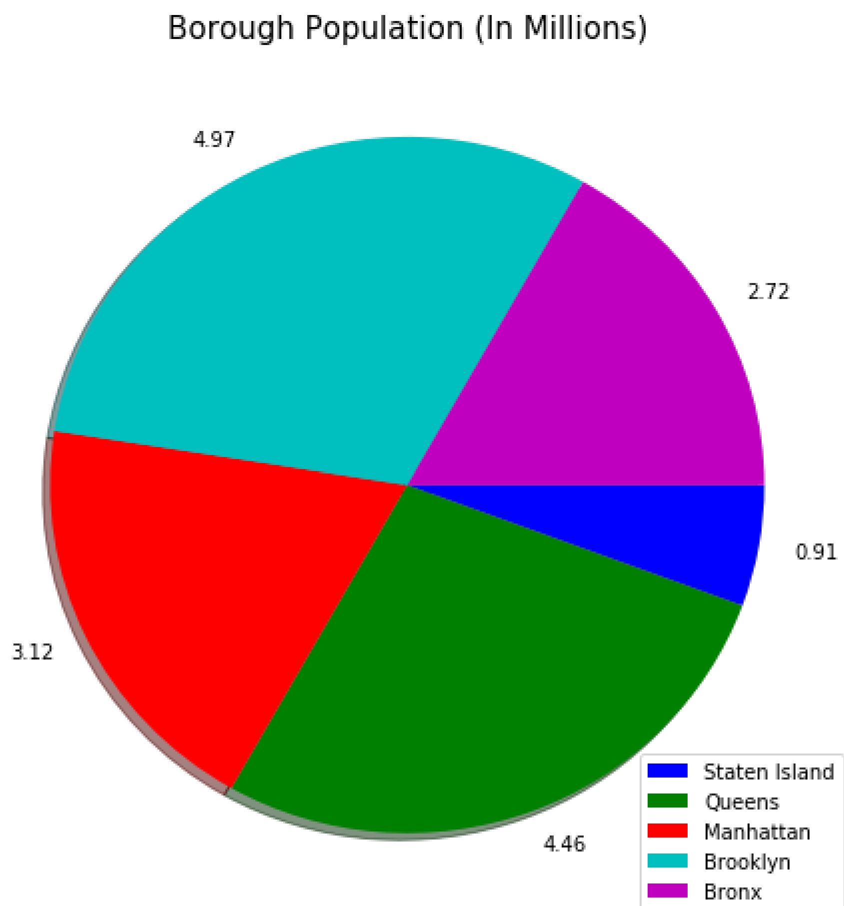
Out[3]:
```
borough                                    object
contributing_factor_vehicle_1              object
contributing_factor_vehicle_2              object
contributing_factor_vehicle_3              object
contributing_factor_vehicle_4              object
contributing_factor_vehicle_5              object
cross_street_name                          object
date                               datetime64[ns]
latitude                                  float64
location                                   object
longitude                                 float64
number_of_cyclist_injured                   int64
number_of_cyclist_killed                    int64
number_of_motorist_injured                  int64
number_of_motorist_killed                   int64
number_of_pedestrians_injured               int64
number_of_pedestrians_killed                int64
number_of_persons_injured                   int64
number_of_persons_killed                    int64
off_street_name                            object
on_street_name                             object
time                                       object
unique_key                                  int64
vehicle_type_code1                         object
vehicle_type_code2                         object
vehicle_type_code_3                        object
vehicle_type_code_4                        object
vehicle_type_code_5                        object
zip_code                                   object
dtype: object
```

In [4]:
```
boroughpop.dtypes
```

Out[4]:
```
borough           object
sum_population     int64
dtype: object
```

It's important to note the relative size of each borough. Staten Island is the smallest borough by far, with Queens and Brooklyn holding the lion's share of New York's population.

In [5]:
```
values = round(boroughpop["sum_population"]/1000000,2)
colors = ['b', 'g', 'r', 'c', 'm', 'y']
labels = boroughpop['borough']

fig, ax = plt.subplots(figsize=(8, 8))
plt.pie(values, colors=colors, labels= values,counterclock=False, shadow
=True)
plt.title('Borough Population (In Millions)',fontsize=15)
plt.legend(labels,loc=4)
plt.show()
```



2.3 Data Cleaning

Below I drop the uneeded columns. I also rename the columns we will be using to make them easy to work with. I also filter out any data that doesn't fall within March 21, 2017 and March 21, 2018.

In [6]:
```
#Understanding data columns. Dropping unneeded columns and renaming
accidents.dtypes
accidents.drop(accidents.columns[[3,4,5,6,8,9,10,17,18,19,20,22,25,26,27,28
]], axis=1, inplace=True)

# rename columns
accidents.columns = ['Borough', 'Contributing_Factor_1', 'Contributing_Fact
or_2','Date','Cyclists Injured','Cyclists Killed','Motorists Injured','Moto
rists Killed','Pedestrians Injured','Pedestrians Killed','Time','Vehicle Ty
pe 1','Vehicle Type 2']

#Get one year's worth of data
accidents[(accidents['Date']>datetime.date(2017,3,21)) & (accidents['Date']
<datetime.date(2018,3,21))]
```

Out[6]:

	Borough	Contributing_Factor_1	Contributing_Factor_2	Date	Cyclists Injured	Cy
385	MANHATTAN	Passing or Lane Usage Improper	Unspecified	2017-03-22	0	0
576	NaN	Unspecified	Unspecified	2017-03-30	0	0
880	NaN	Unspecified	Unspecified	2017-03-31	0	0
1155	NaN	Unsafe Lane Changing	Unspecified	2017-06-24	0	0
1507	NaN	Obstruction/Debris	NaN	2017-03-22	0	0
1686	NaN	Turning Improperly	Unspecified	2017-03-22	0	0
1687	NaN	Unspecified	Unspecified	2017-03-22	0	0
1689	QUEENS	Backing Unsafely	Unspecified	2017-03-22	0	0
1690	BROOKLYN	Driver Inattention/Distraction	Unspecified	2017-03-22	0	0
1691	NaN	Reaction to Other Uninvolved Vehicle	Unspecified	2017-03-22	0	0
1692	QUEENS	Oversized Vehicle	Unspecified	2017-03-22	0	0
1693	BROOKLYN	Unspecified	Unspecified	2017-03-22	0	0

	Borough	Contributing_Factor_1	Contributing_Factor_2	Date	Cyclists Injured	Cy
1694	QUEENS	Backing Unsafely	Passing or Lane Usage Improper	2017-03-22	0	0
1695	MANHATTAN	Unsafe Lane Changing	Unspecified	2017-03-22	0	0
1697	MANHATTAN	Driver Inattention/Distraction	Unspecified	2017-03-22	0	0
1698	MANHATTAN	Following Too Closely	Backing Unsafely	2017-03-22	0	0
1700	NaN	Unspecified	Unspecified	2017-03-22	0	0
1701	MANHATTAN	Driver Inattention/Distraction	Following Too Closely	2017-03-22	0	0
1702	BROOKLYN	Unspecified	Unspecified	2017-03-22	0	0
1703	NaN	Unsafe Lane Changing	Unspecified	2017-03-22	0	0
1704	NaN	Driver Inattention/Distraction	Unspecified	2017-03-22	0	0
1705	NaN	Unsafe Lane Changing	Unspecified	2017-03-22	0	0
1706	NaN	Driver Inexperience	Unspecified	2017-03-22	0	0
1707	BRONX	Turning Improperly	Turning Improperly	2017-03-22	0	0
1709	NaN	Following Too Closely	Unspecified	2017-03-22	0	0
1710	NaN	Driver Inattention/Distraction	Unspecified	2017-03-22	0	0

	Borough	Contributing_Factor_1	Contributing_Factor_2	Date	Cyclists Injured	Cy
1711	NaN	Unspecified	NaN	2017-03-22	0	0
1712	QUEENS	Glare	NaN	2017-03-22	0	0
1713	NaN	Unspecified	NaN	2017-03-22	0	0
1714	MANHATTAN	Unspecified	Unspecified	2017-03-22	0	0
...	...	...	...	...	...	...
307987	NaN	Unspecified	NaN	2018-02-28	0	0
308305	QUEENS	Driver Inattention/Distraction	Unspecified	2018-03-14	0	0
308611	NaN	Driver Inattention/Distraction	Unspecified	2018-03-15	0	0
308648	NaN	Other Vehicular	Other Vehicular	2017-10-24	0	0
308663	MANHATTAN	Driver Inattention/Distraction	NaN	2018-03-06	0	0
308935	BRONX	Aggressive Driving/Road Rage	Unspecified	2018-03-11	0	0
309332	BRONX	Other Vehicular	NaN	2018-03-17	0	0
309338	NaN	Driver Inattention/Distraction	Unspecified	2018-03-14	0	0
309390	BRONX	Unspecified	NaN	2018-03-18	0	0
309638	NaN	Failure to Yield Right-of-Way	NaN	2018-03-18	0	0

	Borough	Contributing_Factor_1	Contributing_Factor_2	Date	Cyclists Injured	Cy
309672	BRONX	View Obstructed/Limited	Unspecified	2018-03-14	0	0
309702	BRONX	Traffic Control Disregarded	Unspecified	2017-04-10	0	0
309933	NaN	Turning Improperly	Unsafe Lane Changing	2017-04-10	0	0
309999	MANHATTAN	Unsafe Lane Changing	Unspecified	2018-03-15	0	0
310014	NaN	Unspecified	Unspecified	2018-03-09	0	0
310449	QUEENS	Unsafe Speed	Unspecified	2018-03-15	0	0
310582	MANHATTAN	Unspecified	NaN	2018-02-15	0	0
310652	BROOKLYN	Unspecified	Unspecified	2018-02-18	0	0
310716	NaN	Unspecified	NaN	2018-03-15	0	0
310926	MANHATTAN	Driver Inattention/Distraction	Driver Inattention/Distraction	2018-03-20	0	0
311251	NaN	Unspecified	Unspecified	2018-02-27	0	0
311266	QUEENS	Failure to Yield Right-of-Way	Unspecified	2018-03-14	0	0

	Borough	Contributing_Factor_1	Contributing_Factor_2	Date	Cyclists Injured	Cy
311498	NaN	Unsafe Speed	Turning Improperly	2018-03-10	0	0
311567	NaN	Backing Unsafely	Unspecified	2018-03-20	0	0
311710	BROOKLYN	Unspecified	NaN	2018-03-05	0	0
311762	MANHATTAN	Turning Improperly	Unspecified	2018-01-10	0	0
311811	NaN	Following Too Closely	Unspecified	2017-04-12	0	0
312000	BROOKLYN	Unsafe Speed	Unspecified	2018-02-28	0	0
443129	NaN	Driver Inattention/Distraction	Unspecified	2018-01-02	0	0
446262	QUEENS	Turning Improperly	Unspecified	2018-01-18	0	0

228400 rows × 13 columns

Here I extract valuable Month and Hour information from the Date and Time columns, respectively. I also opt to add a column for "Total Killed" and "Total Injured". There seems to already be a column for this, but I prefer to do this sumation within Python.

In [7]:
```
#obtaining total injured, total killed, and month + hour of accident

accidents['Total Injured'] = accidents['Cyclists Injured'] + accidents[
'Motorists Injured'] + accidents['Pedestrians Injured']
accidents['Total Killed'] = accidents['Cyclists Killed'] + accidents['Mo
torists Killed'] + accidents['Pedestrians Killed']
accidents['MonthNumber'] = pd.DatetimeIndex(accidents['Date']).month
accidents["Time"] = pd.to_datetime(accidents["Time"])
accidents["Hour"] = accidents["Time"].dt.hour

#Converting Month to text
accidents['Month'] = accidents['MonthNumber'].map({1:'January',2:'Februa
ry',3:'March',4:'April',5:'May',6:'June',7:'July',8:'August',9:'Septembe
r',10:'October',11:'November',12:'December'})


print(accidents)
```

```
                Borough           Contributing_Factor_1  \
0              BROOKLYN                     Unspecified
1                   NaN               Oversized Vehicle
2                   NaN                     Fell Asleep
3                   NaN                    Unsafe Speed
4              BROOKLYN                     Unspecified
5                   NaN                Other Vehicular
6                QUEENS  Driver Inattention/Distraction
7                   NaN                     Unspecified
8                   NaN  Driver Inattention/Distraction
9             MANHATTAN              Turning Improperly
10               QUEENS             Alcohol Involvement
11                  NaN  Driver Inattention/Distraction
12                BRONX  Driver Inattention/Distraction
13                  NaN  Passing or Lane Usage Improper
14                  NaN           Following Too Closely
15               QUEENS                           Glare
16                  NaN           Following Too Closely
17                  NaN               Pavement Slippery
18               QUEENS                     Unspecified
19            MANHATTAN  Driver Inattention/Distraction
20            MANHATTAN             Unsafe Lane Changing
21             BROOKLYN                     Unspecified
22            MANHATTAN                     Unspecified
23                  NaN                     Unspecified
24               QUEENS                    Unsafe Speed
25                  NaN           Following Too Closely
26               QUEENS                     Unspecified
27               QUEENS                     Unspecified
28                  NaN              Driver Inexperience
29               QUEENS                     Unspecified
...                 ...                             ...
499970    STATEN ISLAND                     Unspecified
499971    STATEN ISLAND                     Unspecified
499972    STATEN ISLAND                     Unspecified
499973    STATEN ISLAND                     Unspecified
499974    STATEN ISLAND                     Unspecified
499975        MANHATTAN                     Unspecified
499976    STATEN ISLAND                     Unspecified
499977    STATEN ISLAND                     Unspecified
499978            BRONX  Driver Inattention/Distraction
499979    STATEN ISLAND                     Unspecified
499980    STATEN ISLAND                 Fatigued/Drowsy
499981    STATEN ISLAND                     Unspecified
499982    STATEN ISLAND                     Unspecified
499983            BRONX                     Unspecified
499984    STATEN ISLAND                     Unspecified
499985    STATEN ISLAND             Driver Inexperience
499986    STATEN ISLAND             Driver Inexperience
499987    STATEN ISLAND   Failure to Yield Right-of-Way
499988    STATEN ISLAND                     Unspecified
499989        MANHATTAN                Other Vehicular
499990        MANHATTAN               Oversized Vehicle
499991        MANHATTAN          Prescription Medication
499992        MANHATTAN                Other Vehicular
499993        MANHATTAN                Other Vehicular
499994            BRONX                     Unspecified
```

```
499995      MANHATTAN               Fatigued/Drowsy
499996      MANHATTAN          Passenger Distraction
499997      MANHATTAN                    Unspecified
499998      MANHATTAN  Driver Inattention/Distraction
499999      MANHATTAN  Driver Inattention/Distraction

                   Contributing_Factor_2       Date  Cyclists Injured  \
0                                    NaN 2017-03-20                 0
1                            Unspecified 2017-03-20                 0
2                                    NaN 2017-03-20                 0
3                            Unspecified 2017-02-05                 0
4                                    NaN 2017-03-20                 0
5                            Unspecified 2017-03-20                 0
6                                    NaN 2017-03-20                 0
7                            Unspecified 2017-02-05                 0
8                    Unsafe Lane Changing 2017-03-20                 0
9                            Unspecified 2017-03-20                 0
10                           Unspecified 2017-03-20                 0
11                           Unspecified 2017-03-20                 0
12        Driver Inattention/Distraction 2017-03-20                 0
13                           Unspecified 2017-03-20                 0
14                  Following Too Closely 2017-03-20                 0
15                           Unspecified 2017-03-20                 0
16                  Following Too Closely 2017-03-20                 0
17                           Unspecified 2017-03-20                 0
18                                   NaN 2017-03-20                 0
19                           Unspecified 2017-03-20                 0
20                   Unsafe Lane Changing 2017-03-20                 0
21                                   NaN 2017-03-20                 0
22                                   NaN 2017-03-20                 0
23                           Unspecified 2017-03-20                 0
24                           Unspecified 2017-03-20                 0
25                           Unspecified 2017-03-20                 0
26                                   NaN 2017-03-20                 0
27                           Unspecified 2017-03-20                 0
28                           Unspecified 2017-03-20                 0
29                                   NaN 2017-03-20                 0
...                                  ...        ...               ...
499970                       Unspecified 2014-08-24                 0
499971                       Unspecified 2014-08-24                 0
499972                       Unspecified 2014-08-24                 0
499973                               NaN 2014-08-24                 0
499974                       Unspecified 2014-08-22                 0
499975                       Unspecified 2014-08-25                 0
499976                       Unspecified 2014-08-22                 0
499977                       Unspecified 2014-08-23                 0
499978  Driver Inattention/Distraction 2013-05-27                 0
499979                       Unspecified 2014-08-23                 0
499980     Traffic Control Disregarded 2014-08-20                 0
499981                       Unspecified 2014-08-25                 0
499982                       Unspecified 2014-08-22                 0
499983                       Unspecified 2013-05-27                 0
499984                       Unspecified 2014-08-24                 1
499985                       Unspecified 2014-08-25                 0
499986                       Unspecified 2014-08-25                 0
499987                       Unspecified 2014-08-25                 0
499988                       Unspecified 2014-08-21                 0
```

```
499989           Passenger Distraction 2014-08-25                  1
499990                             NaN 2014-08-26                  0
499991                     Unspecified 2014-08-26                  0
499992                Other Vehicular 2014-08-20                  0
499993                Backing Unsafely 2014-08-26                  0
499994                             NaN 2013-05-27                  0
499995                Other Vehicular 2014-08-26                  0
499996                     Unspecified 2014-08-26                  1
499997                     Unspecified 2014-08-25                  0
499998  Driver Inattention/Distraction 2014-08-26                  0
499999                     Unspecified 2014-08-25                  0

        Cyclists Killed  Motorists Injured  Motorists Killed  \
0                     0                  0                 0
1                     0                  0                 0
2                     0                  1                 0
3                     0                  0                 0
4                     0                  0                 0
5                     0                  0                 0
6                     0                  0                 0
7                     0                  0                 0
8                     0                  0                 0
9                     0                  2                 0
10                    0                  0                 0
11                    0                  0                 0
12                    0                  0                 0
13                    0                  0                 0
14                    0                  2                 0
15                    0                  0                 0
16                    0                  0                 0
17                    0                  0                 0
18                    0                  0                 0
19                    0                  0                 0
20                    0                  0                 0
21                    0                  0                 0
22                    0                  0                 0
23                    0                  0                 0
24                    0                  0                 0
25                    0                  1                 0
26                    0                  0                 0
27                    0                  1                 0
28                    0                  1                 0
29                    0                  0                 0
...                 ...                ...               ...
499970                0                  1                 0
499971                0                  0                 0
499972                0                  0                 0
499973                0                  0                 0
499974                0                  0                 0
499975                0                  0                 0
499976                0                  0                 0
499977                0                  0                 0
499978                0                  0                 0
499979                0                  0                 0
499980                0                  0                 0
499981                0                  0                 0
499982                0                  0                 0
```

```
499983             0                0               0
499984             0                0               0
499985             0                0               0
499986             0                0               0
499987             0                0               0
499988             0                0               0
499989             0                0               0
499990             0                0               0
499991             0                0               0
499992             0                0               0
499993             0                0               0
499994             0                0               0
499995             0                0               0
499996             0                0               0
499997             0                0               0
499998             0                0               0
499999             0                0               0

        Pedestrians Injured  Pedestrians Killed                Time  \
0                         0                   0 2018-04-16 00:30:00
1                         0                   0 2018-04-16 01:50:00
2                         0                   0 2018-04-16 05:30:00
3                         0                   0 2018-04-16 23:20:00
4                         0                   0 2018-04-16 01:00:00
5                         0                   0 2018-04-16 04:30:00
6                         1                   0 2018-04-16 06:45:00
7                         0                   0 2018-04-16 17:00:00
8                         0                   0 2018-04-16 04:33:00
9                         0                   0 2018-04-16 02:15:00
10                        0                   0 2018-04-16 03:40:00
11                        0                   0 2018-04-16 05:10:00
12                        0                   0 2018-04-16 00:00:00
13                        0                   0 2018-04-16 05:55:00
14                        0                   0 2018-04-16 08:50:00
15                        0                   0 2018-04-16 08:05:00
16                        0                   0 2018-04-16 08:00:00
17                        0                   0 2018-04-16 08:40:00
18                        0                   0 2018-04-16 08:00:00
19                        0                   0 2018-04-16 06:36:00
20                        0                   0 2018-04-16 07:30:00
21                        0                   0 2018-04-16 07:00:00
22                        0                   0 2018-04-16 07:40:00
23                        0                   0 2018-04-16 09:04:00
24                        0                   0 2018-04-16 06:10:00
25                        0                   0 2018-04-16 09:00:00
26                        0                   0 2018-04-16 08:53:00
27                        0                   0 2018-04-16 08:20:00
28                        0                   0 2018-04-16 07:50:00
29                        0                   0 2018-04-16 09:05:00
...                     ...                 ...                 ...
499970                    0                   0 2018-04-16 14:15:00
499971                    0                   0 2018-04-16 15:30:00
499972                    0                   0 2018-04-16 18:08:00
499973                    0                   0 2018-04-16 14:10:00
499974                    0                   0 2018-04-16 11:25:00
499975                    0                   0 2018-04-16 20:44:00
499976                    0                   0 2018-04-16 18:51:00
```

```
499977              0                   0 2018-04-16 11:53:00
499978              0                   0 2018-04-16 13:47:00
499979              0                   0 2018-04-16 12:55:00
499980              0                   0 2018-04-16 18:12:00
499981              0                   0 2018-04-16 15:50:00
499982              0                   0 2018-04-16 16:08:00
499983              0                   0 2018-04-16 13:30:00
499984              0                   0 2018-04-16 13:30:00
499985              0                   0 2018-04-16 14:37:00
499986              0                   0 2018-04-16 13:30:00
499987              0                   0 2018-04-16 11:00:00
499988              0                   0 2018-04-16 14:00:00
499989              0                   0 2018-04-16 21:50:00
499990              0                   0 2018-04-16 10:26:00
499991              0                   0 2018-04-16 10:00:00
499992              0                   0 2018-04-16 09:48:00
499993              0                   0 2018-04-16 14:15:00
499994              1                   0 2018-04-16 15:20:00
499995              0                   0 2018-04-16 16:45:00
499996              0                   0 2018-04-16 00:55:00
499997              0                   0 2018-04-16 23:30:00
499998              0                   0 2018-04-16 01:40:00
499999              0                   0 2018-04-16 15:48:00

                        Vehicle Type 1                  Vehicle Type 2
\
0                         PICK-UP TRUCK                            NaN

1                     PASSENGER VEHICLE              PASSENGER VEHICLE

2                     PASSENGER VEHICLE                            NaN

3                     PASSENGER VEHICLE  SPORT UTILITY / STATION WAGON

4                     PASSENGER VEHICLE                            NaN

5                     PASSENGER VEHICLE              PASSENGER VEHICLE

6         SPORT UTILITY / STATION WAGON                            NaN

7         SPORT UTILITY / STATION WAGON                            NaN

8                     PASSENGER VEHICLE  SPORT UTILITY / STATION WAGON

9                     PASSENGER VEHICLE  SPORT UTILITY / STATION WAGON

10                    PASSENGER VEHICLE              PASSENGER VEHICLE

11        SPORT UTILITY / STATION WAGON                            NaN

12        SPORT UTILITY / STATION WAGON                            VAN

13                    PASSENGER VEHICLE                            NaN

14        SPORT UTILITY / STATION WAGON  SPORT UTILITY / STATION WAGON

15                    PASSENGER VEHICLE                            NaN
```

16	SPORT UTILITY / STATION WAGON	SPORT UTILITY / STATION WAGON
17	PASSENGER VEHICLE	SPORT UTILITY / STATION WAGON
18	PASSENGER VEHICLE	NaN
19	SPORT UTILITY / STATION WAGON	PASSENGER VEHICLE
20	PASSENGER VEHICLE	NaN
21	SPORT UTILITY / STATION WAGON	NaN
22	SPORT UTILITY / STATION WAGON	NaN
23	PASSENGER VEHICLE	SPORT UTILITY / STATION WAGON
24	PASSENGER VEHICLE	SPORT UTILITY / STATION WAGON
25	SPORT UTILITY / STATION WAGON	PICK-UP TRUCK
26	NaN	NaN
27	SPORT UTILITY / STATION WAGON	SPORT UTILITY / STATION WAGON
28	PASSENGER VEHICLE	SPORT UTILITY / STATION WAGON
29	PICK-UP TRUCK	NaN
...	...	...
499970	SPORT UTILITY / STATION WAGON	SPORT UTILITY / STATION WAGON
499971	SPORT UTILITY / STATION WAGON	SPORT UTILITY / STATION WAGON
499972	SPORT UTILITY / STATION WAGON	SPORT UTILITY / STATION WAGON
499973	SPORT UTILITY / STATION WAGON	NaN
499974	SPORT UTILITY / STATION WAGON	SPORT UTILITY / STATION WAGON
499975	SPORT UTILITY / STATION WAGON	UNKNOWN
499976	SPORT UTILITY / STATION WAGON	SPORT UTILITY / STATION WAGON
499977	SPORT UTILITY / STATION WAGON	SPORT UTILITY / STATION WAGON
499978	PASSENGER VEHICLE	PASSENGER VEHICLE
499979	SPORT UTILITY / STATION WAGON	SPORT UTILITY / STATION WAGON
499980	SPORT UTILITY / STATION WAGON	SPORT UTILITY / STATION WAGON
499981	SPORT UTILITY / STATION WAGON	PASSENGER VEHICLE
499982	PASSENGER VEHICLE	PASSENGER VEHICLE

499983	PASSENGER VEHICLE	PASSENGER VEHICLE
499984	PASSENGER VEHICLE	BICYCLE
499985	PASSENGER VEHICLE	PASSENGER VEHICLE
499986	MOTORCYCLE	VAN
499987	PASSENGER VEHICLE	PASSENGER VEHICLE
499988	SPORT UTILITY / STATION WAGON	PASSENGER VEHICLE
499989	SPORT UTILITY / STATION WAGON	BICYCLE
499990	LARGE COM VEH(6 OR MORE TIRES)	NaN
499991	BUS	LARGE COM VEH(6 OR MORE TIRES)
499992	BUS	OTHER
499993	PASSENGER VEHICLE	OTHER
499994	PASSENGER VEHICLE	NaN
499995	SMALL COM VEH(4 TIRES)	PASSENGER VEHICLE
499996	LIVERY VEHICLE	BICYCLE
499997	SPORT UTILITY / STATION WAGON	UNKNOWN
499998	PASSENGER VEHICLE	TAXI
499999	LARGE COM VEH(6 OR MORE TIRES)	PASSENGER VEHICLE

	Total Injured	Total Killed	MonthNumber	Hour	Month
0	0	0	3	0	March
1	0	0	3	1	March
2	1	0	3	5	March
3	0	0	2	23	February
4	0	0	3	1	March
5	0	0	3	4	March
6	1	0	3	6	March
7	0	0	2	17	February
8	0	0	3	4	March
9	2	0	3	2	March
10	0	0	3	3	March
11	0	0	3	5	March
12	0	0	3	0	March
13	0	0	3	5	March
14	2	0	3	8	March
15	0	0	3	8	March
16	0	0	3	8	March
17	0	0	3	8	March
18	0	0	3	8	March
19	0	0	3	6	March
20	0	0	3	7	March

```
21              0          0         3     7    March
22              0          0         3     7    March
23              0          0         3     9    March
24              0          0         3     6    March
25              1          0         3     9    March
26              0          0         3     8    March
27              1          0         3     8    March
28              1          0         3     7    March
29              0          0         3     9    March
...           ...        ...       ...   ...      ...
499970          1          0         8    14   August
499971          0          0         8    15   August
499972          0          0         8    18   August
499973          0          0         8    14   August
499974          0          0         8    11   August
499975          0          0         8    20   August
499976          0          0         8    18   August
499977          0          0         8    11   August
499978          0          0         5    13      May
499979          0          0         8    12   August
499980          0          0         8    18   August
499981          0          0         8    15   August
499982          0          0         8    16   August
499983          0          0         5    13      May
499984          1          0         8    13   August
499985          0          0         8    14   August
499986          0          0         8    13   August
499987          0          0         8    11   August
499988          0          0         8    14   August
499989          1          0         8    21   August
499990          0          0         8    10   August
499991          0          0         8    10   August
499992          0          0         8     9   August
499993          0          0         8    14   August
499994          1          0         5    15      May
499995          0          0         8    16   August
499996          1          0         8     0   August
499997          0          0         8    23   August
499998          0          0         8     1   August
499999          0          0         8    15   August

[500000 rows x 18 columns]
```

3. Visualizing the Data

3.1 Where Accidents Occur

Accidents broken down by the five boroughs of New York City.

3.1.1 By Borough and Traveler

Injuries and deaths by borough and by type of traveler.

In [8]:
```
accidentsgrouped = accidents.groupby(['Borough']).sum()

#sorting accidents grouped by borough so that it matches up with our boroug
h population dataframe. This will be used later
accidentsgrouped['boroughsort']=accidentsgrouped.index
accidentsgrouped = accidentsgrouped.sort_values(by='boroughsort', ascending
=False)

accidentsgrouped['Pedestrians Injured Per Cap'] = np.divide(accidentsgroupe
d['Pedestrians Injured'],boroughpop['sum_population'])
accidentsgrouped['Motorists Injured Per Cap'] = np.divide(accidentsgrouped[
'Motorists Injured'],boroughpop['sum_population'])
accidentsgrouped['Cyclists Injured Per Cap'] = np.divide(accidentsgrouped[
'Cyclists Injured'],boroughpop['sum_population'])
accidentsgrouped['Pedestrians Killed Per Cap'] = np.divide(accidentsgrouped
['Pedestrians Killed'],boroughpop['sum_population'])
accidentsgrouped['Motorists Killed Per Cap'] = np.divide(accidentsgrouped[
'Motorists Killed'],boroughpop['sum_population'])
accidentsgrouped['Cyclists Killed Per Cap'] = np.divide(accidentsgrouped['C
yclists Killed'],boroughpop['sum_population'])

#Setting of dataframes around types of injuries and deaths
pedestinjuriesbyborough = accidentsgrouped.sort_values(by='Pedestrians Inju
red', ascending=False)
pedestdeathsbyborough = accidentsgrouped.sort_values(by='Pedestrians Kille
d', ascending=False)
motorinjuriesbyborough = accidentsgrouped.sort_values(by='Motorists Injure
d', ascending=False)
motordeathsbyborough = accidentsgrouped.sort_values(by='Motorists Killed',
ascending=False)
cyclistinjuriesbyborough = accidentsgrouped.sort_values(by='Cyclists Injure
d', ascending=False)
cyclistdeathsbyborough = accidentsgrouped.sort_values(by='Cyclists Killed',
 ascending=False)

print(accidentsgrouped)
```

```
               Cyclists Injured  Cyclists Killed  Motorists Injured  \
Borough
STATEN ISLAND                71                0               1867
QUEENS                     1119                6              12055
MANHATTAN                  1655                9               4454
BROOKLYN                   2120               12              13008
BRONX                       560                2               6728

               Motorists Killed  Pedestrians Injured  Pedestrians Killed  \
Borough
STATEN ISLAND                 8                  368                   7
QUEENS                       19                 3208                  47
MANHATTAN                     4                 2913                  35
BROOKLYN                     20                 4396                  36
BRONX                         8                 2237                  18

               Total Injured  Total Killed  MonthNumber    Hour  \
Borough
STATEN ISLAND           2306            15        59284  127464
QUEENS                 16382            72       382941  805525
MANHATTAN               9022            48       315336  650439
BROOKLYN               19524            68       431961  919128
BRONX                   9525            28       208192  439047

                 boroughsort  Pedestrians Injured Per Cap  \
Borough
STATEN ISLAND  STATEN ISLAND                     0.000403
QUEENS                QUEENS                     0.000719
MANHATTAN          MANHATTAN                     0.000933
BROOKLYN            BROOKLYN                     0.000885
BRONX                  BRONX                     0.000823

               Motorists Injured Per Cap  Cyclists Injured Per Cap  \
Borough
STATEN ISLAND                   0.002046                  0.000078
QUEENS                          0.002703                  0.000251
MANHATTAN                       0.001426                  0.000530
BROOKLYN                        0.002617                  0.000427
BRONX                           0.002476                  0.000206

               Pedestrians Killed Per Cap  Motorists Killed Per Cap  \
Borough
STATEN ISLAND                    0.000008                  0.000009
QUEENS                           0.000011                  0.000004
MANHATTAN                        0.000011                  0.000001
BROOKLYN                         0.000007                  0.000004
BRONX                            0.000007                  0.000003

               Cyclists Killed Per Cap
Borough
STATEN ISLAND             0.000000e+00
QUEENS                    1.345261e-06
MANHATTAN                 2.881782e-06
BROOKLYN                  2.414474e-06
BRONX                     7.359007e-07
```

Here we see a comparison between Pedestrian, Motorists, and Cyclist injuries and deaths across Buroughs. Interestingly, the ranking between boroughs changes across types of commuters and across injuries and deaths.

Observations

1. Why are almost as many motorists killed on Staten Island as in the Bronx (3rd), even though Staten Island is consistently ranked lowest in all other areas?
2. Why does Queens take the lead for pedestrians killed, even though it is ranked 2nd and 3rd in all other types of accidents?
3. Why does Manhattan seem to have the highest variability amongst buroughs, being last in Motorists injured/killed, 2nd in cyclists injured/killed, and middle of the pack for pedestrian injuries?
4. Why does Staten Island have such low rates of Cyclist injuries and deaths?

Potential Additions

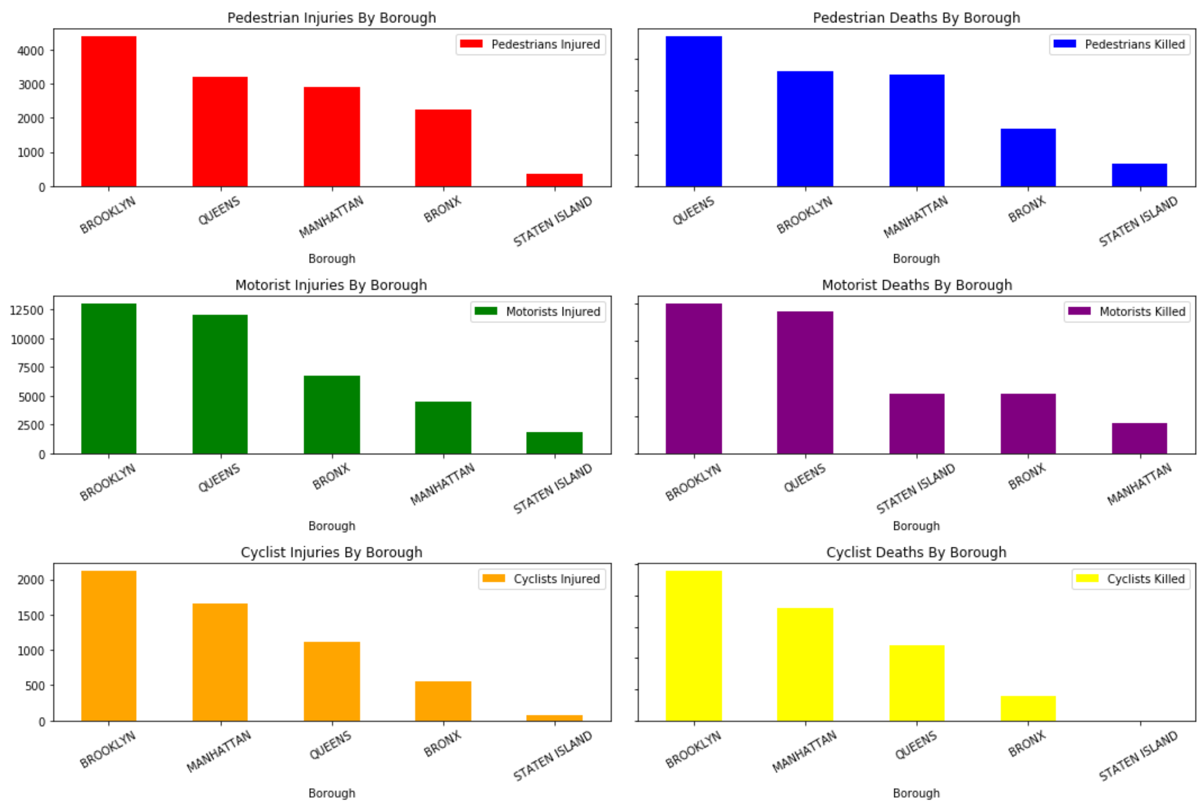
Of course, we would have to understand these numbers better in the context of per capita rates. This could be done by merging with a population dataset and dividing these rates by number of inhabitants. However, this would still not account for the fact that some areas have very high commuter rates.

Perhaps better would be to look at geographic size, or even total traffic flow. Geographic size may paint a better picture, but this still lacks information on density. In regards to total traffic flow, well, let me know if that dataset exists!

Takeaways

1. Do many vehicles pass through Staten Island, lending to its high motorist deaths? Or, does Staten Island have inadequate speed limits, traffic lights, stop signs, etc.? Higher speed tolerances may explain why Staten Island jumps more in deaths than in injuries. Perhaps Staten Island's motor policy needs review.
2. Does Queens have a higher proportion of people who walk? Or, are sidewalks inadequate, crosswalks ignored, and residents lacking general safety habits like looking both ways? In this area, perhaps Queen's outreach and development programs need review.
3. Does Manhattan see such high variability because it is the most diverse burough of them all? Or, does it clearly have a problem providing adequate laneways for cyclists? I know I personally see cyclists navigating erractically, crossing intersections when lights are red and even failing to look both ways when doing so. The data indicates a clear problem, supported by anecdotal evidence. Perhaps Manhattan needs to look in providing better bicycly lanes as well as fining cyclists for ignoring lights in the same way they would motorists.
4. Does Staten Island have few cyclists? Or does it adequately provide for them and enforce cyclist law? Perhaps Manhattan and Brooklyn could learn from them.

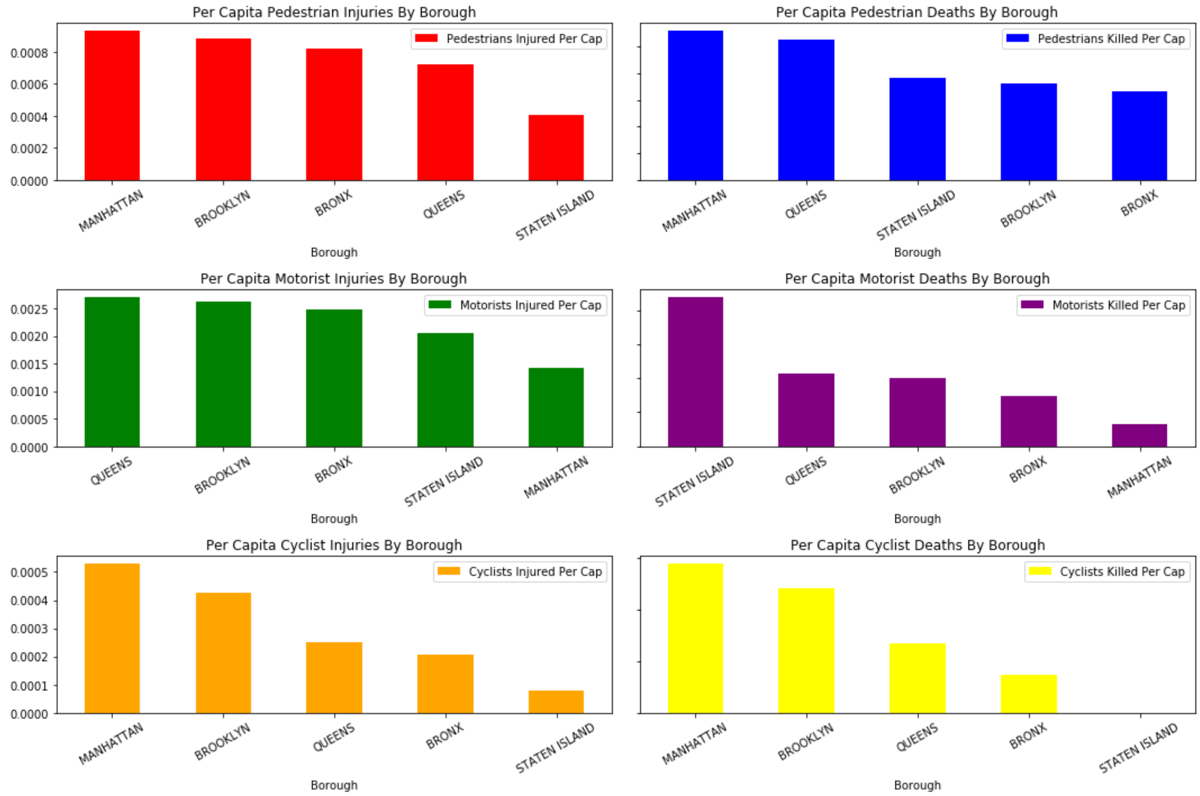
In [9]:
```
fig, ax = plt.subplots(nrows=3, ncols=2)
pedestinjuriesbyborough[['Pedestrians Injured']].plot(kind='bar', ax=ax[0,0
], color=['red'], width=0.5, title ="Pedestrian Injuries By Borough", figsi
ze=(15, 10), legend=True,fontsize=10, rot = 30)
pedestdeathsbyborough[['Pedestrians Killed']].plot(kind='bar', ax=ax[0,1],
color=['blue'],width=0.5, title ="Pedestrian Deaths By Borough", figsize=(1
5, 10), legend=True, fontsize=10,sharey=ax[0,0],rot = 30 )
motorinjuriesbyborough[['Motorists Injured']].plot(kind='bar', ax=ax[1,0],
color=['green'], width=0.5, title ="Motorist Injuries By Borough", figsize=
(15, 10), legend=True,fontsize=10, rot = 30)
motordeathsbyborough[['Motorists Killed']].plot(kind='bar', ax=ax[1,1], col
or=['purple'],width=0.5, title ="Motorist Deaths By Borough", figsize=(15,
10), legend=True, fontsize=10,sharey=ax[1,0],rot = 30 )
cyclistinjuriesbyborough[['Cyclists Injured']].plot(kind='bar', ax=ax[2,0],
 color=['orange'],width=0.5, title ="Cyclist Injuries By Borough", figsize=
(15, 10), legend=True, fontsize=10,rot = 30 )
cyclistdeathsbyborough[['Cyclists Killed']].plot(kind='bar', ax=ax[2,1], co
lor=['yellow'],width=0.5, title ="Cyclist Deaths By Borough", figsize=(15,
10), legend=True, fontsize=10,sharey=ax[2,0],rot = 30)

fig.tight_layout()
fig.subplots_adjust(wspace=0.05, hspace=0.7)
plt.show()
```



Taking into account borough populations, some observations change or become clearer.

1. Staten Island is far ahead of the other buroughs in terms of motorists killed per capita - something is clearly wrong.
2. Pedestrian deaths for Queens seem even more shocking now given its per capita ranking against other buroughs.
3. Manhattan's variability now appears largely explained. It has by far the lowest motorist deaths and injuries per capita of any borough and yet it has the highest per capita rate of injuries and deaths of cyclists and pedestrians per capita than any other burough. This is likely explained by the foot traffic density and slow traffic speed in the city. It may also indicate inadequate protection for non-motorists
4. The low level of cyclist injuries and deaths in Staten Island is even more impressive when looked at per capita.
5. Finally, a general observation is that far more drivers are killed and injured by accidents than pedestrians and cyclists (see axes). This is perhaps not surprising, though it is certainly reassuring that those who are "at fault" are those who are most affected.

In [10]:
```
#setting up sorting for graphs
pedestinjuriesbyborough = accidentsgrouped.sort_values(by='Pedestrians Inju
red Per Cap', ascending=False)
pedestdeathsbyborough = accidentsgrouped.sort_values(by='Pedestrians Killed
 Per Cap', ascending=False)
motorinjuriesbyborough = accidentsgrouped.sort_values(by='Motorists Injured
 Per Cap', ascending=False)
motordeathsbyborough = accidentsgrouped.sort_values(by='Motorists Killed Pe
r Cap', ascending=False)
cyclistinjuriesbyborough = accidentsgrouped.sort_values(by='Cyclists Injure
d Per Cap', ascending=False)
cyclistdeathsbyborough = accidentsgrouped.sort_values(by='Cyclists Killed P
er Cap', ascending=False)

fig, ax = plt.subplots(nrows=3, ncols=2)
pedestinjuriesbyborough[['Pedestrians Injured Per Cap']].plot(kind='bar', a
x=ax[0,0], color=['red'], width=0.5, title ="Per Capita Pedestrian Injuries
 By Borough", figsize=(15, 10), legend=True,fontsize=10, rot = 30,sort_colu
mns=True)
pedestdeathsbyborough[['Pedestrians Killed Per Cap']].plot(kind='bar', ax=a
x[0,1], color=['blue'],width=0.5, title ="Per Capita Pedestrian Deaths By B
orough", figsize=(15, 10), legend=True, fontsize=10,sharey=ax[0,0],rot = 30
,sort_columns=True )
motorinjuriesbyborough[['Motorists Injured Per Cap']].plot(kind='bar', ax=a
x[1,0], color=['green'], width=0.5, title ="Per Capita Motorist Injuries By
 Borough", figsize=(15, 10), legend=True,fontsize=10, rot = 30,sort_columns
=True)
motordeathsbyborough[['Motorists Killed Per Cap']].plot(kind='bar', ax=ax[1
,1], color=['purple'],width=0.5, title ="Per Capita Motorist Deaths By Boro
ugh", figsize=(15, 10), legend=True, fontsize=10,sharey=ax[1,0],rot = 30,so
rt_columns=True )
cyclistinjuriesbyborough[['Cyclists Injured Per Cap']].plot(kind='bar', ax=
ax[2,0], color=['orange'],width=0.5, title ="Per Capita Cyclist Injuries By
 Borough", figsize=(15, 10), legend=True, fontsize=10,rot = 30,sort_columns
=True )
cyclistdeathsbyborough[['Cyclists Killed Per Cap']].plot(kind='bar', ax=ax[
2,1], color=['yellow'],width=0.5, title ="Per Capita Cyclist Deaths By Boro
ugh", figsize=(15, 10), legend=True, fontsize=10,sharey=ax[2,0],rot = 30,so
rt_columns=True)

fig.tight_layout()
fig.subplots_adjust(wspace=0.05, hspace=0.7)
plt.show()
```



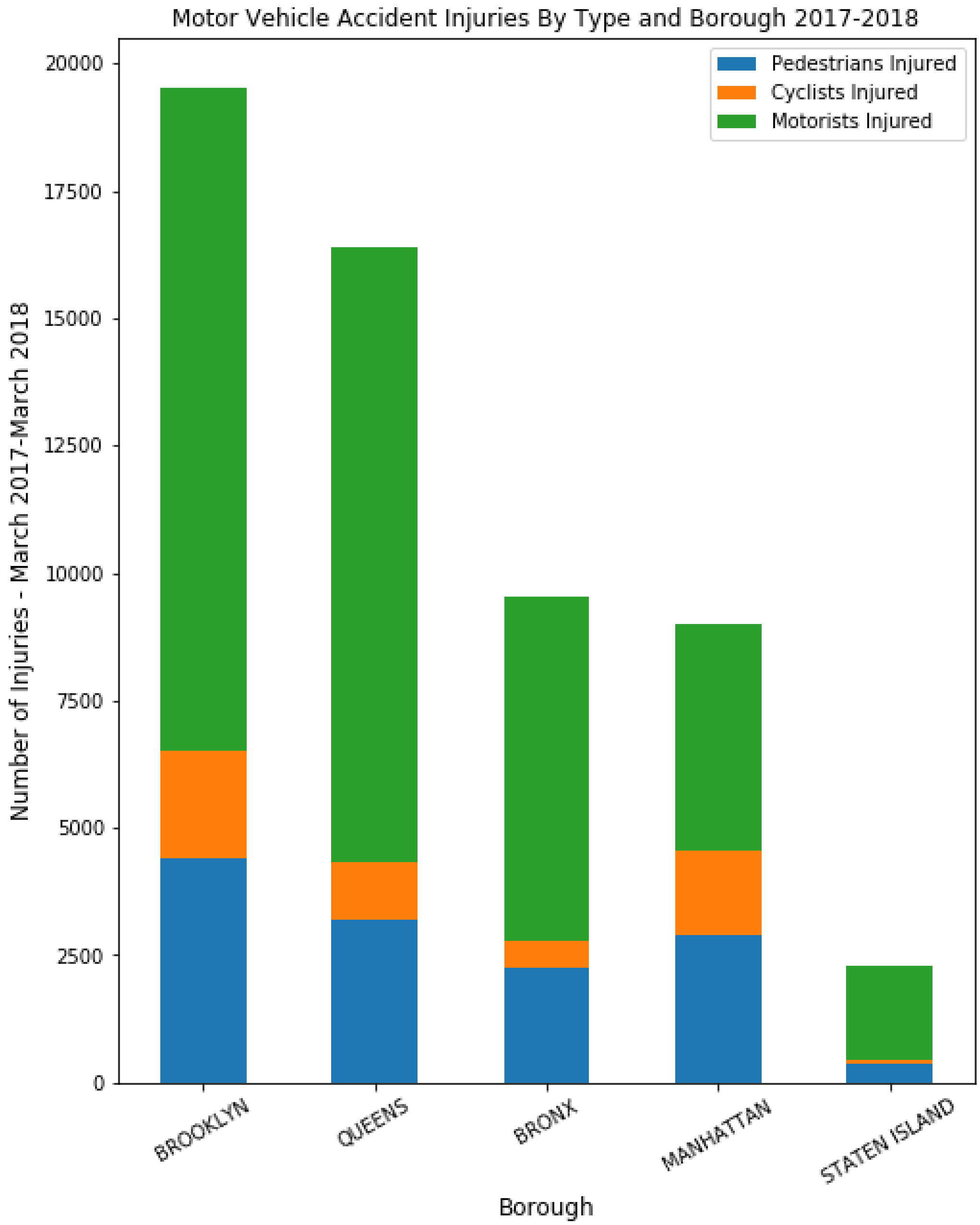
3.1.2 Borough and Traveler Summarized

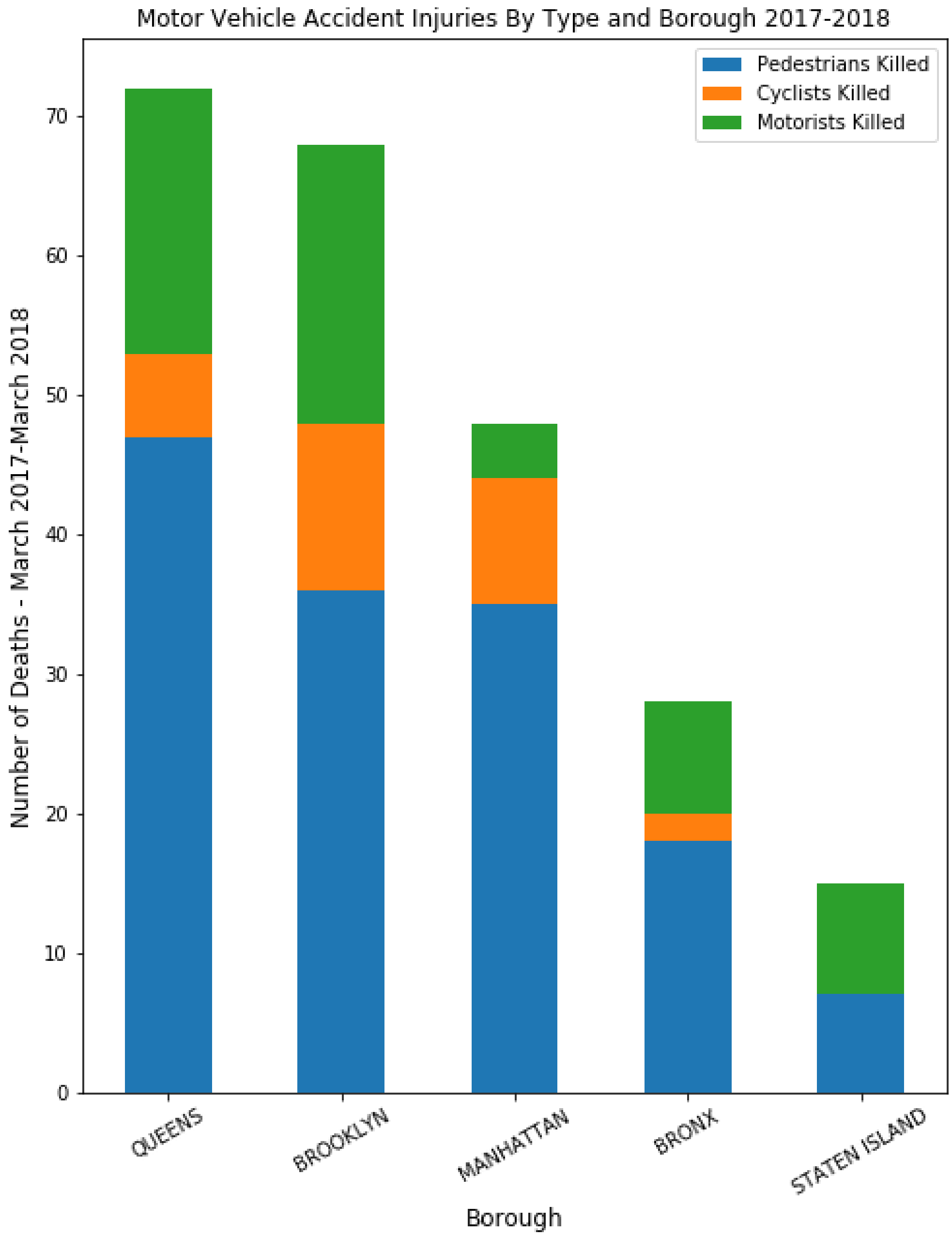
Observations and Takeaways

Looking at the numbers from another angle, we see that Brooklyn has the most injuries, but Queens has the most deaths. Furthermore, we see that this difference is largely due to pedestrian deaths, and this is despite the fact that Queens has fewer pedestrian injuries than Brooklyn. Does Queens have a poor hospital system that is either not getting injured pedestrians to treatment quickly enough or is losing them in the hospital themselves. Ultimately, are there more deaths that could have been just injuries?

We also see that while Manhattan and The Bronx have about the same amount of injuries, Manhattan has almost twice as many deaths as The Bronx. Again, this difference seems to be due to pedestrian deaths. This might inform us that Amblance ride windows are too long or hospital ER treatment not adequate for pedestrian type injuries.

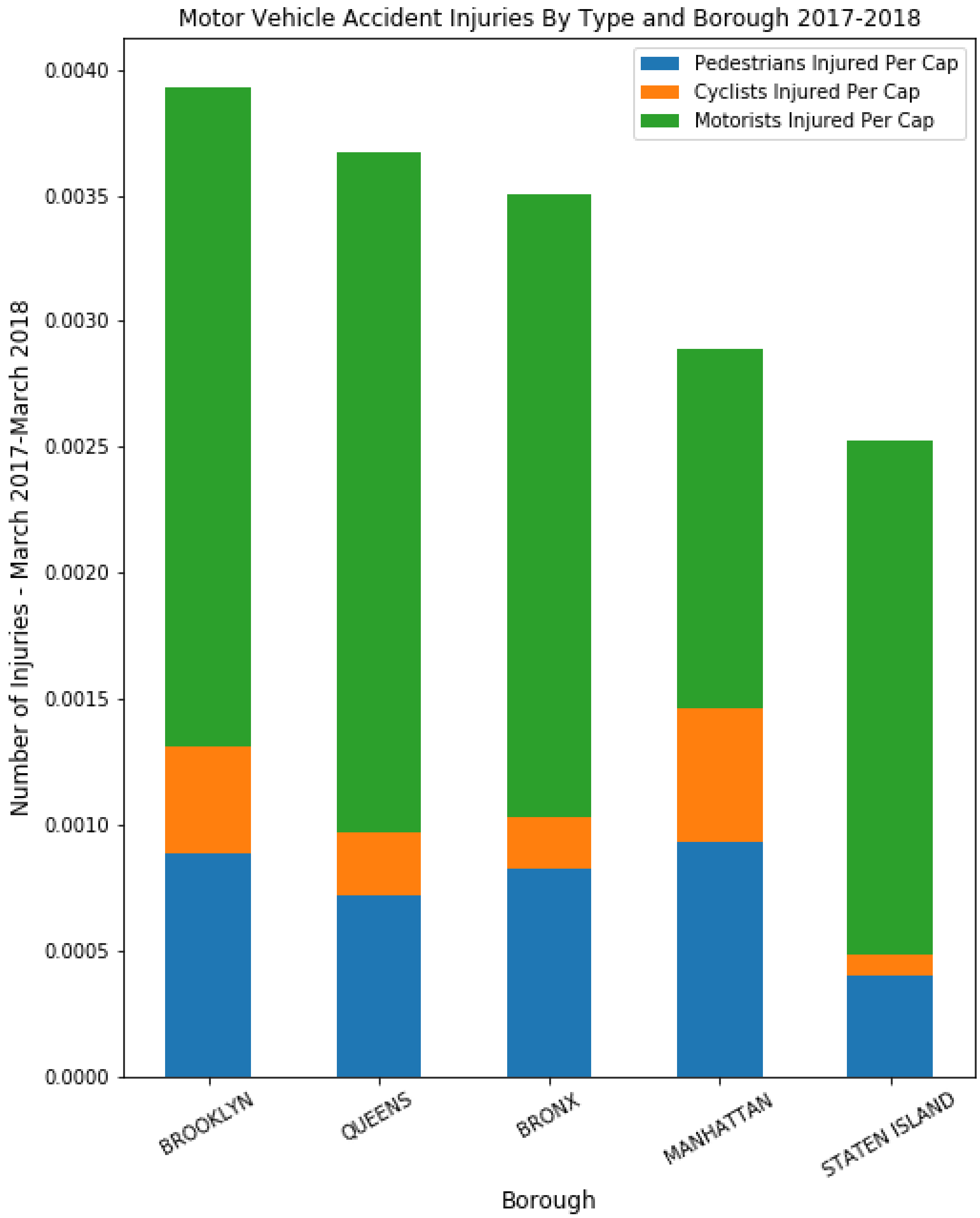
In [18]:
```
injuriesbyborough = accidentsgrouped.sort_values(by='Total Injured', ascend
ing=False)
deathsbyborough = accidentsgrouped.sort_values(by='Total Killed', ascending
=False)

ax = injuriesbyborough[['Pedestrians Injured','Cyclists Injured', 'Motorist
s Injured']].plot(kind='bar', stacked = True,title ="Motor Vehicle Accident
 Injuries By Type and Borough 2017-2018", figsize=(8, 10), legend=True, fon
tsize=10,rot=30)
ax.set_xlabel("Borough", fontsize=12)
ax.set_ylabel("Number of Injuries - March 2017-March 2018", fontsize=12)

ax = deathsbyborough[['Pedestrians Killed','Cyclists Killed', 'Motorists Ki
lled']].plot(kind='bar', stacked = True,title ="Motor Vehicle Accident Inju
ries By Type and Borough 2017-2018", figsize=(8, 10), legend=True, fontsize
=10,rot=30)
ax.set_xlabel("Borough", fontsize=12)
ax.set_ylabel("Number of Deaths - March 2017-March 2018", fontsize=12)

plt.show()
```


Motor Vehicle Accident Injuries By Type and Borough 2017-2018
Pedestrians Injured
Cyclists Injured
Motorists Injured
Number of Injuries - March 2017-March 2018
20000
17500
15000
12500
10000
7500
5000
2500
0
BROOKLYN
QUEENS
BRONX
MANHATTAN
STATEN ISLAND
Borough

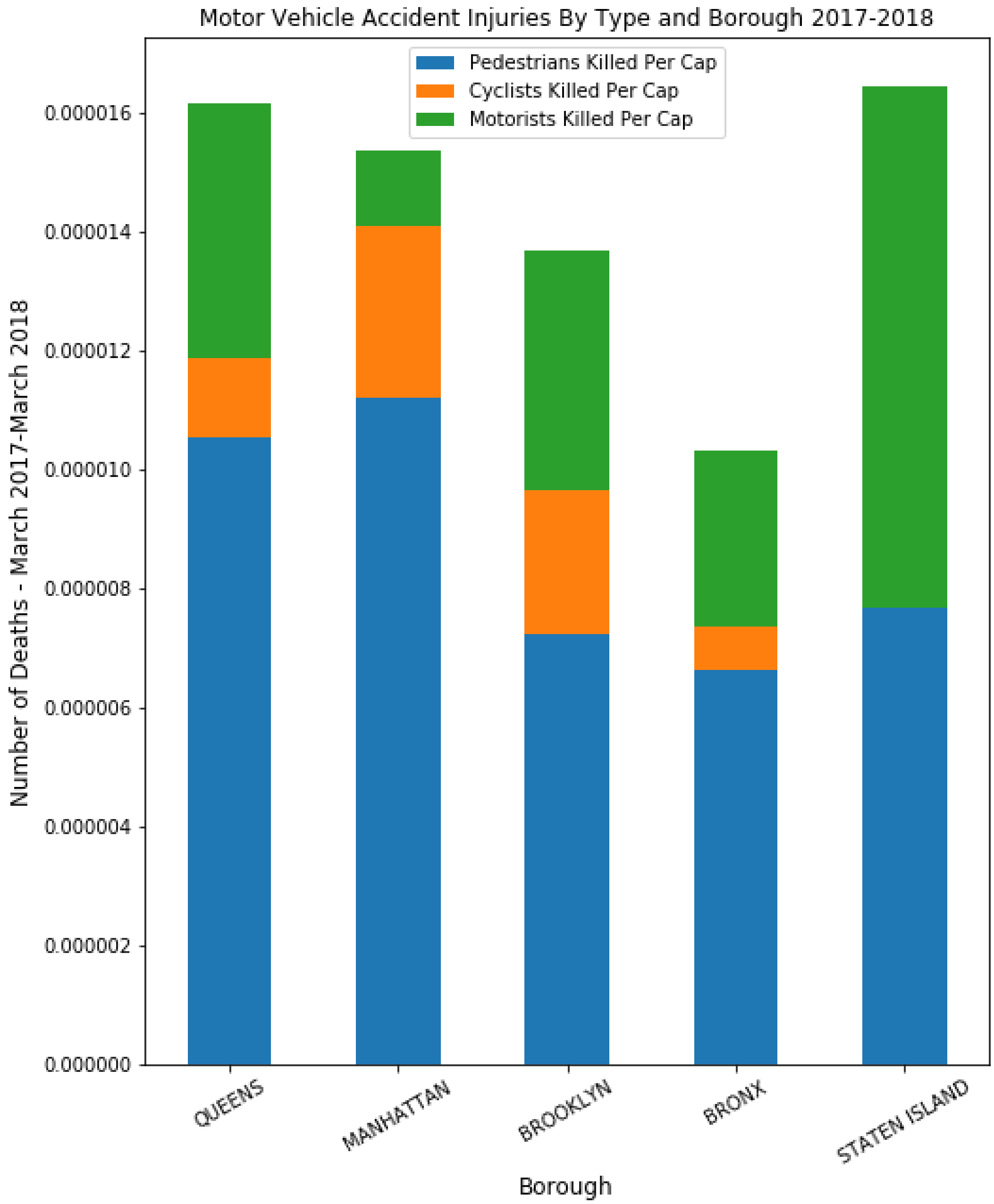

Motor Vehicle Accident Injuries By Type and Borough 2017-2018
Pedestrians Killed
Cyclists Killed
Motorists Killed
Number of Deaths - March 2017-March 2018
0
10
20
30
40
50
60
70
QUEENS
BROOKLYN
MANHATTAN
BRONX
STATEN ISLAND
Borough

In [25]:

```
injuriesbyborough['Total Injured Per Cap'] = np.divide(injuriesbyborough
['Total Injured'],boroughpop['sum_population'])
deathsbyborough['Total Killed Per Cap'] = np.divide(deathsbyborough['Tot
al Killed'],boroughpop['sum_population'])
injuriesbyborough = injuriesbyborough.sort_values(by='Total Injured Per
 Cap', ascending=False)
deathsbyborough = deathsbyborough.sort_values(by='Total Killed Per Cap',
 ascending=False)

ax = injuriesbyborough[['Pedestrians Injured Per Cap','Cyclists Injured
 Per Cap', 'Motorists Injured Per Cap']].plot(kind='bar', stacked = True
,title ="Motor Vehicle Accident Injuries By Type and Borough 2017-2018",
 figsize=(8, 10), legend=True, fontsize=10,rot=30)
ax.set_xlabel("Borough", fontsize=12)
ax.set_ylabel("Number of Injuries - March 2017-March 2018", fontsize=12)

ax = deathsbyborough[['Pedestrians Killed Per Cap','Cyclists Killed Per
 Cap', 'Motorists Killed Per Cap']].plot(kind='bar', stacked = True,titl
e ="Motor Vehicle Accident Injuries By Type and Borough 2017-2018", figs
ize=(8, 10), legend=True, fontsize=10,rot=30)
ax.set_xlabel("Borough", fontsize=12)
ax.set_ylabel("Number of Deaths - March 2017-March 2018", fontsize=12)

plt.show()
```



Motor Vehicle Accident Injuries By Type and Borough 2017-2018
Pedestrians Injured Per Cap
Cyclists Injured Per Cap
Motorists Injured Per Cap
Number of Injuries - March 2017-March 2018
0.0040
0.0035
0.0030
0.0025
0.0020
0.0015
0.0010
0.0005
0.0000
BROOKLYN
QUEENS
BRONX
MANHATTAN
STATEN ISLAND
Borough



Checking the largest accidents we see that there are no major outliers contributing to this discrepency.

In [12]:
```
accidents = accidents.sort_values(by='Total Killed', ascending=False)
accidentstop10 =accidents.head(10)
print(accidentstop10)
```

```
          Borough Contributing_Factor_1 Contributing_Factor_2       Date  \
193648  MANHATTAN       Other Vehicular           Unspecified 2017-10-31   

158552     QUEENS          Unsafe Speed           Unspecified 2017-09-18   

384584        NaN           Unspecified           Unspecified 2016-06-19   

230057        NaN           Unspecified           Unspecified 2016-08-31   

86937         NaN          Unsafe Speed           Unspecified 2017-07-16   

97738         NaN           Unspecified                   NaN 2014-10-17   

66576      QUEENS          Unsafe Speed                   NaN 2017-06-17   

394334   BROOKLYN           Unspecified           Unspecified 2016-08-01   

383460        NaN           Unspecified                   NaN 2016-06-13   

20443         NaN           Unspecified           Unspecified 2017-04-17   


        Cyclists Injured  Cyclists Killed  Motorists Injured  \
193648                 1                2                  4   
158552                 0                0                 13   
384584                 0                0                  9   
230057                 0                0                  3   
86937                  0                0                  8   
97738                  0                0                  0   
66576                  0                0                  1   
394334                 0                0                  0   
383460                 0                0                  0   
20443                  0                0                  0   

        Motorists Killed  Pedestrians Injured  Pedestrians Killed  \
193648                 0                    8                   8   
158552                 1                    3                   2   
384584                 3                    0                   0   
230057                 3                    0                   0   
86937                  2                    0                   0   
97738                  2                    0                   0   
66576                  2                    0                   0   
394334                 0                    0                   2   
383460                 2                    0                   0   
20443                  2                    0                   0   

                      Time                 Vehicle Type 1  \
193648 2018-04-16 15:08:00                             FB   
158552 2018-04-16 06:17:00  SPORT UTILITY / STATION WAGON   
384584 2018-04-16 13:45:00              PASSENGER VEHICLE   
230057 2018-04-16 04:17:00              PASSENGER VEHICLE   
86937  2018-04-16 06:26:00              PASSENGER VEHICLE   
97738  2018-04-16 02:04:00                     MOTORCYCLE   
66576  2018-04-16 00:55:00              PASSENGER VEHICLE   
394334 2018-04-16 00:08:00              PASSENGER VEHICLE   
383460 2018-04-16 02:17:00              PASSENGER VEHICLE   
```

```
20443  2018-04-16 18:21:00            PASSENGER VEHICLE

                      Vehicle Type 2  Total Injured  Total Killed  \
193648                            BU             13            10
158552  SPORT UTILITY / STATION WAGON             16             3
384584              PASSENGER VEHICLE              9             3
230057              PASSENGER VEHICLE              3             3
86937   SPORT UTILITY / STATION WAGON              8             2
97738                             NaN              0             2
66576                             NaN              1             2
394334              PASSENGER VEHICLE              0             2
383460                            NaN              0             2
20443   SPORT UTILITY / STATION WAGON              0             2

        MonthNumber  Hour      Month
193648           10    15    October
158552            9     6  September
384584            6    13       June
230057            8     4     August
86937             7     6       July
97738            10     2    October
66576             6     0       June
394334            8     0     August
383460            6     2       June
20443             4    18      April
```

3. Visualizing the Data

3.2 When Accidents Occur

Not where, but when accidents occur. Based on time of day as well as month.

3.2.1 Time of Day

Observations and Takeaways

Injuries

We see below that injuries for accidents are low in the wee hours of the night. They start rising at around 6 am, with a mini-peak at around 8 am. They fall until 10, at which point they steadily rise again before peaking at around 5pm.

It seems that accidents are low from midnights to 6 am due to few drivers on the road. Accidents spike between 6-8 am as more drivers rush to work, perhaps falling afterwards as roads get too congested for real injuries to take place.

As rushhour ends at 10pm, the rise up to 5pm can be explained by crowded, but not completely congested streets.

Interestingly, accidents occur far more at the end of the day than at the beginning. One may posit that this is due to factors such as road-rage after a stressful day at work, fatigure after a long day, or eagerness to arrive home.

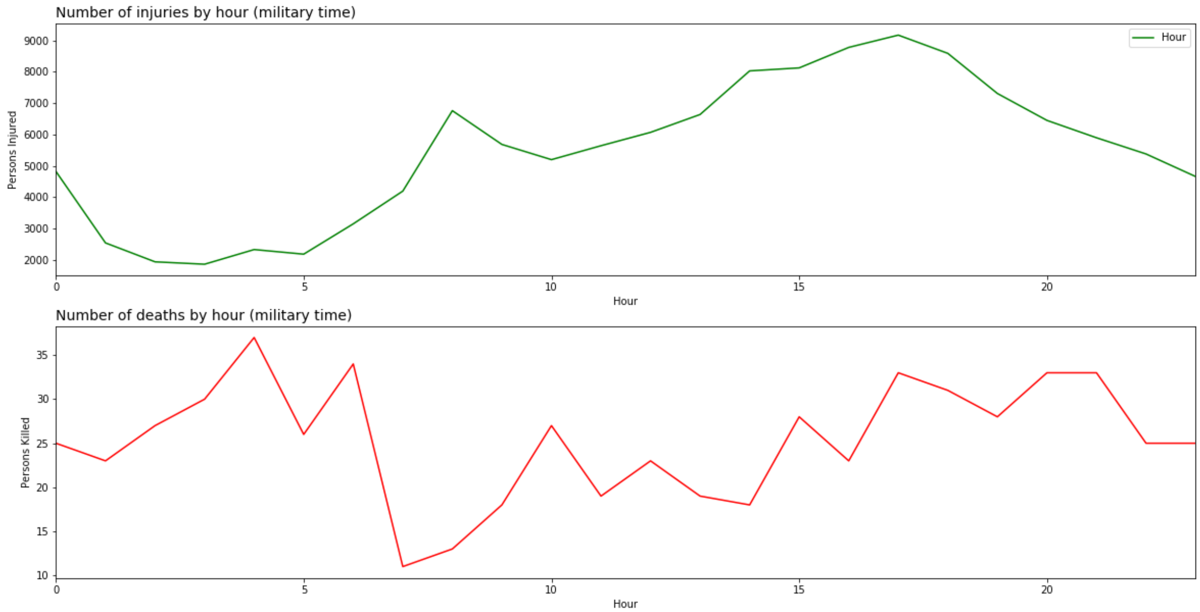
In reaction, officials may determine that targeting messaging to drivers may be a good way to alert them to the risks when driving home in the evening. They may also opt to post more traffic cops and general officers throughout the city during these times.

Deaths

In direct contrast with injuries, deaths occur the most in the wee hours of the night. This could be due to a number of factors including driver fatigue, drunk driving, and higher car speeds thanks to fewer drivers on the road.

Deaths plummet during the morning rush hour and interestingly rise steadily throughout the day until rush hour is over. This might be because as drivers reach the tail end of their commute home, they are more tired and more eager to get home, and, importantly, no longer in slow-moving traffic.

In [13]:
```
accidentsbyhour = accidents.groupby(['Hour']).sum()

fig, ax = plt.subplots(nrows=2, ncols=1, figsize=(20, 10))
accidentsbyhour['Total Injured'].plot(ax=ax[0], color='green')
ax[0].set_title('Number of injuries by hour (military time)', fontsize=14,
loc='left')
ax[0].set_ylabel('Persons Injured')
ax[0].legend(['Hour', 'Persons Injured'])
accidentsbyhour['Total Killed'].plot(ax=ax[1], color='red')
ax[1].set_title('Number of deaths by hour (military time)', fontsize=14, lo
c='left')
ax[1].set_ylabel('Persons Killed')
plt.show()
```



3.2.2 Month

Observations and Takeaways

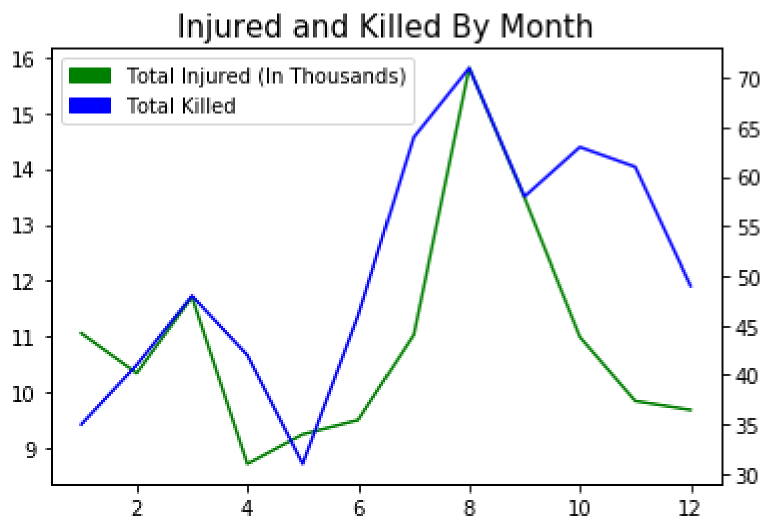
Injuries and deaths seems to show a similar seasonal trend. That being said, there is a consistently low level of deaths from January through to May.

Spring is a visibly a period of low injuries and deaths, which continues into most of summer. From August to September accidents are at their highest.

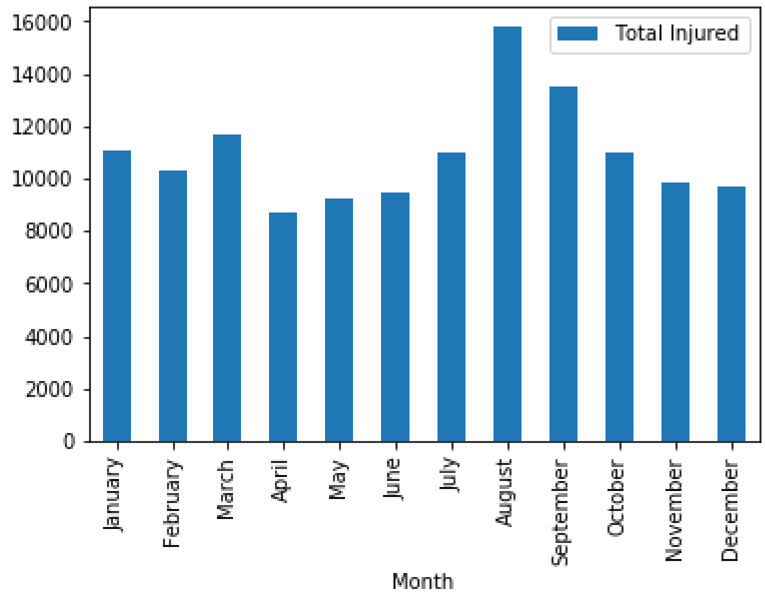
This may be weather related. More people (and tourists) might be out from August to September. It could also have to do with commuter levels - springtime is a period when many choose to take vacations. Finally, fewer deaths may occur during the cold months as weather prevents motorists from driving too quickly.

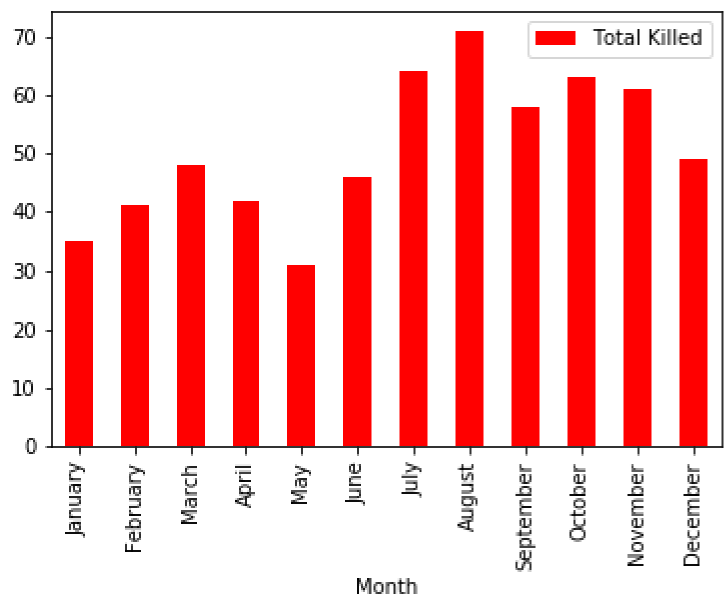
City officials could be put on "high alert," or government services increased during this fall and winter period. To balance the budget, this could be met with reduced government services during the low spring and summer months.

In [14]:
```
injuriesbymonth = accidents.groupby(['MonthNumber']).sum()
injuriesbymonth['Month'] = ['January','February','March','April','May',
'June','July','August','September','October','November','December']

fig, ax1 = plt.subplots()

x = injuriesbymonth.index
y1 = injuriesbymonth['Total Injured']/1000
y2 = injuriesbymonth['Total Killed']
ax1.set_title("Injured and Killed By Month",fontsize=15)

ax2 = ax1.twinx()

ax1.plot(x, y1, 'g-', label = "Total Injured")
ax2.plot(x, y2, 'b-', label = "Total Killed")

injured = mpatches.Patch(color='Green', label='Total Injured (In Thousan
ds)')
killed = mpatches.Patch(color='Blue', label='Total Killed')
plt.legend(handles=[injured,killed])

plt.show()
```



In [31]:
```
injuriesbymonth.plot(y='Total Injured',kind='bar')


deathsbymonth = accidents.groupby(['MonthNumber']).sum()
deathsbymonth['Month'] = ['January','February','March','April','May','June'
,'July','August','September','October','November','December']
deathsbymonth = deathsbymonth.set_index('Month')
deathsbymonth.plot(y='Total Killed',kind='bar', color='red')
plt.show()
```



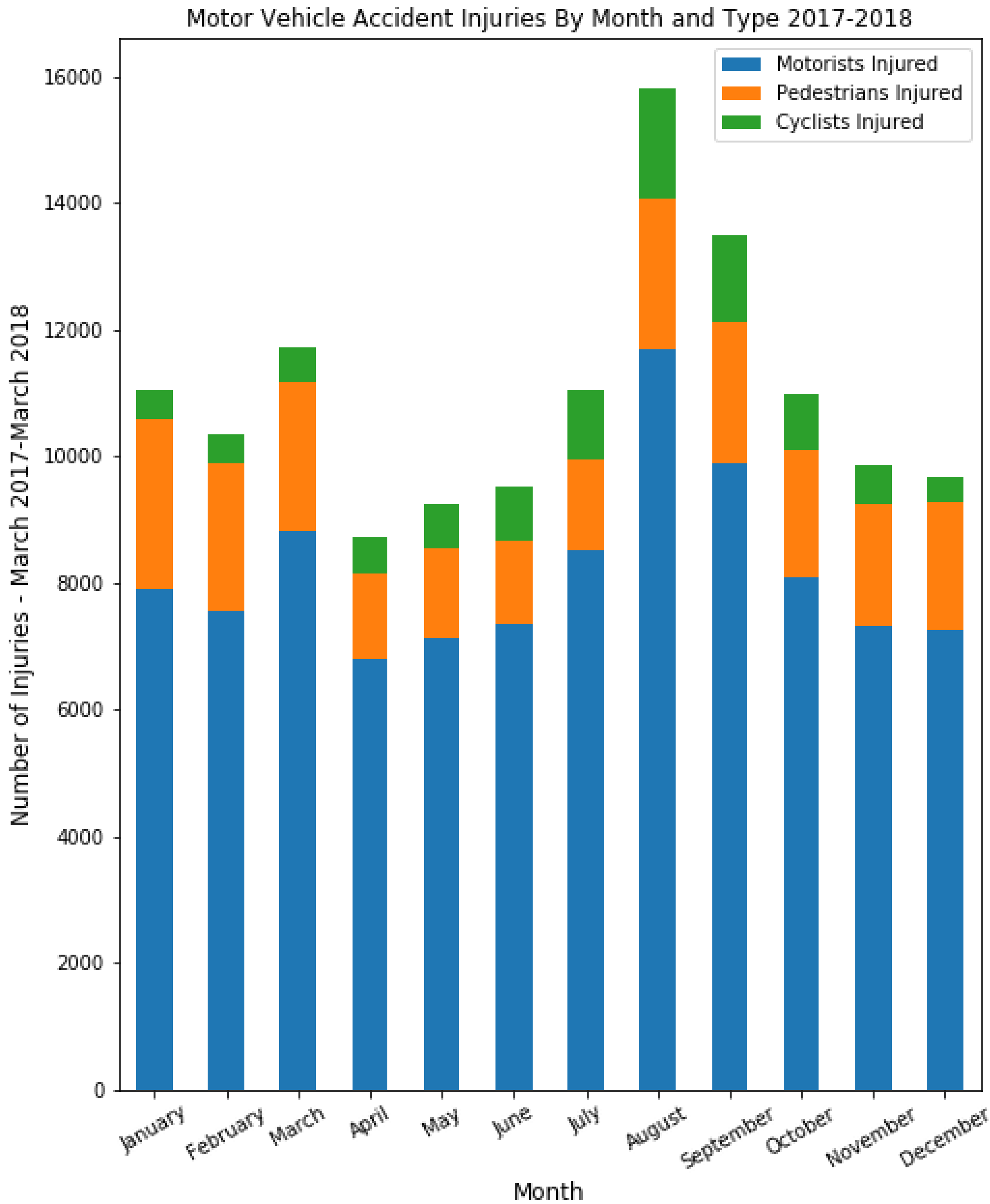


3.2.3 Month and Traveler Type

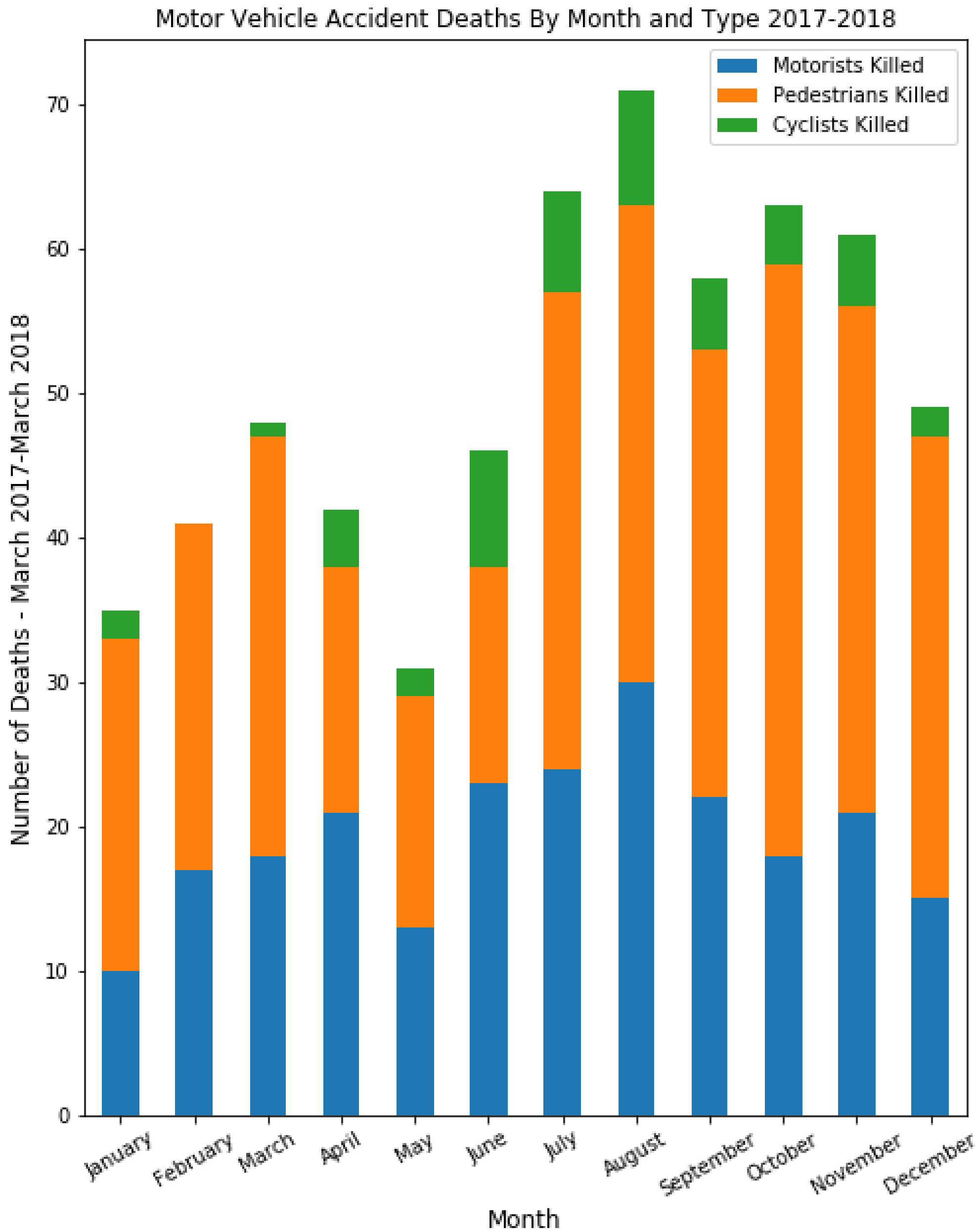
Falls and rises in injuries are driven primarily by changes in motorist injuries. This does not come as a huge surprise, given that they account for the vast majority of injuries.

However, the delta in deaths between months is caused by motorist, pedestrian, and cyclist deaths alike. Frightenenly, about half of deaths are not to the motorists, but to pedestrians and cyclists. This demonstrates the importance of protecting those on foot and bike with sidewalks, crosswalks, walking/biking lanes, etc; Motorists are protected de-facto by their cars, while those outside are extremely vulnerable.

In [32]:
```
ax = injuriesbymonth[['Motorists Injured','Pedestrians Injured','Cyclists I
njured',]].plot(kind='bar', stacked = True,title ="Motor Vehicle Accident I
njuries By Month and Type 2017-2018", figsize=(8, 10), legend=True, fontsiz
e=10,rot=30)
ax.set_xlabel("Month", fontsize=12)
ax.set_ylabel("Number of Injuries - March 2017-March 2018", fontsize=12)

ax = deathsbymonth[['Motorists Killed','Pedestrians Killed','Cyclists Kille
d']].plot(kind='bar', stacked = True,title ="Motor Vehicle Accident Deaths
 By Month and Type 2017-2018", figsize=(8, 10), legend=True, fontsize=10,ro
t=30)
ax.set_xlabel("Month", fontsize=12)
ax.set_ylabel("Number of Deaths - March 2017-March 2018", fontsize=12)

plt.show()
```



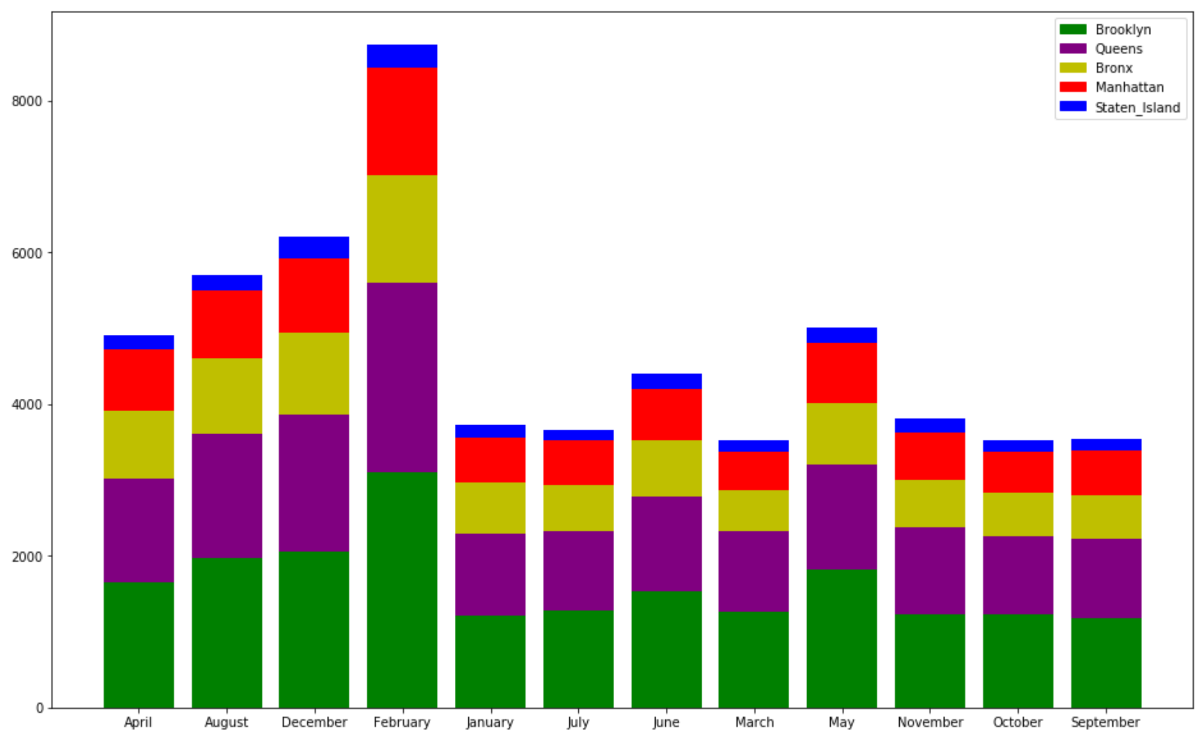
Motor Vehicle Accident Injuries By Month and Type 2017-2018
Motorists Injured
Pedestrians Injured
Cyclists Injured
Number of Injuries - March 2017-March 2018
0
2000
4000
6000
8000
10000
12000
14000
16000
January
February
March
April
May
June
July
August
September
October
November
December
Month



3.2.4 Month and Borough

Data seems to suggest that changes in injury rates by month do not favor particular boroughs. While we observed earlier that boroughs are responsible for differering absolute amounts of accidents, their relative importance does not shift noticeably from month to month. In other words, injuries rise and fall in unison across boroughs throughout the year.

In [33]:
```
accidentsbymonthborough = accidents.groupby(['Month', 'Borough'], as_index=
False).sum()

fig, ax = plt.subplots(nrows=1, ncols=1, figsize=(16, 10))
A1 = accidentsbymonthborough.query("Borough=='BROOKLYN'")['Total Injured']
B2 = np.add(A1,accidentsbymonthborough.query("Borough=='QUEENS'")['Total In
jured'])
C3= np.add(B2,accidentsbymonthborough.query("Borough=='BRONX'")['Total Inju
red'])
D4 = np.add(C3,accidentsbymonthborough.query("Borough=='MANHATTAN'")['Total
 Injured'])
E5 = np.add(D4,accidentsbymonthborough.query("Borough=='STATEN ISLAND'")['T
otal Injured'])


X = injuriesbymonth.index.tolist()

plt.bar(X, E5, color = 'b')
plt.bar(X, D4, color = 'r')
plt.bar(X, C3, color = 'y')
plt.bar(X, B2, color = 'purple')
plt.bar(X, A1, color = 'g')

Brooklyn = mpatches.Patch(color='g', label='Brooklyn')
Queens = mpatches.Patch(color='Purple', label='Queens')
Bronx = mpatches.Patch(color='y', label='Bronx')
Manhattan = mpatches.Patch(color='r', label='Manhattan')
Staten_Island= mpatches.Patch(color='b', label='Staten_Island')
plt.legend(handles=[Brooklyn,Queens,Bronx,Manhattan,Staten_Island])


plt.show()
```



3. Visualizing the Data

3.3 Why Accidents Occur

Not where or when, but why accidents occur. The causes of accidents based on the primary car/driver at fault.

Observations and Takeaways

In the charts below I look at the top eight contributing factors to injuries and fatalities. The percentages relate to the percent that each item accounts for in the group of eight.

The biggest culprit by far for injuries is driver inattention. This is followed by a failure to yield right-of-way and following too closely.

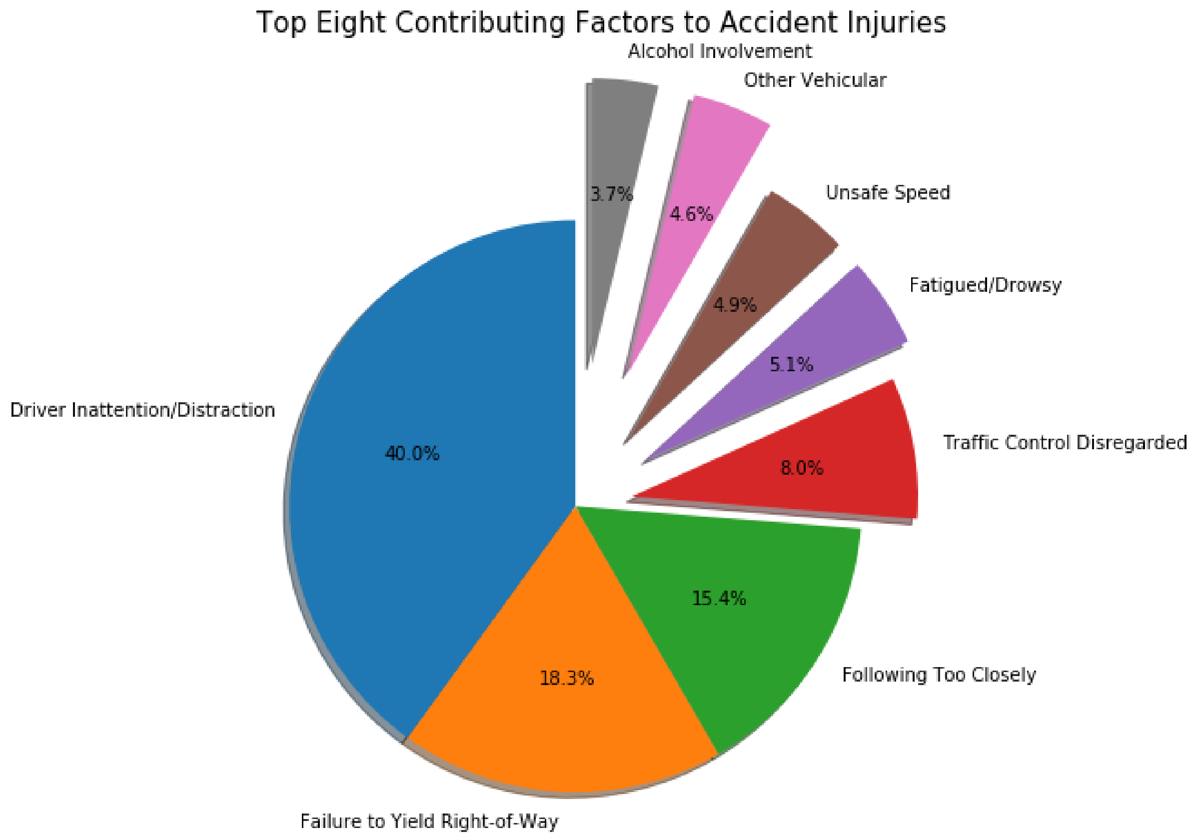
Perhaps unsurprisingly, the main causes of accident deaths are different. While the leading factor is still driver inattention, 2nd and 3rd major contributing factors are failure to yield right of way and unsafe speeds.

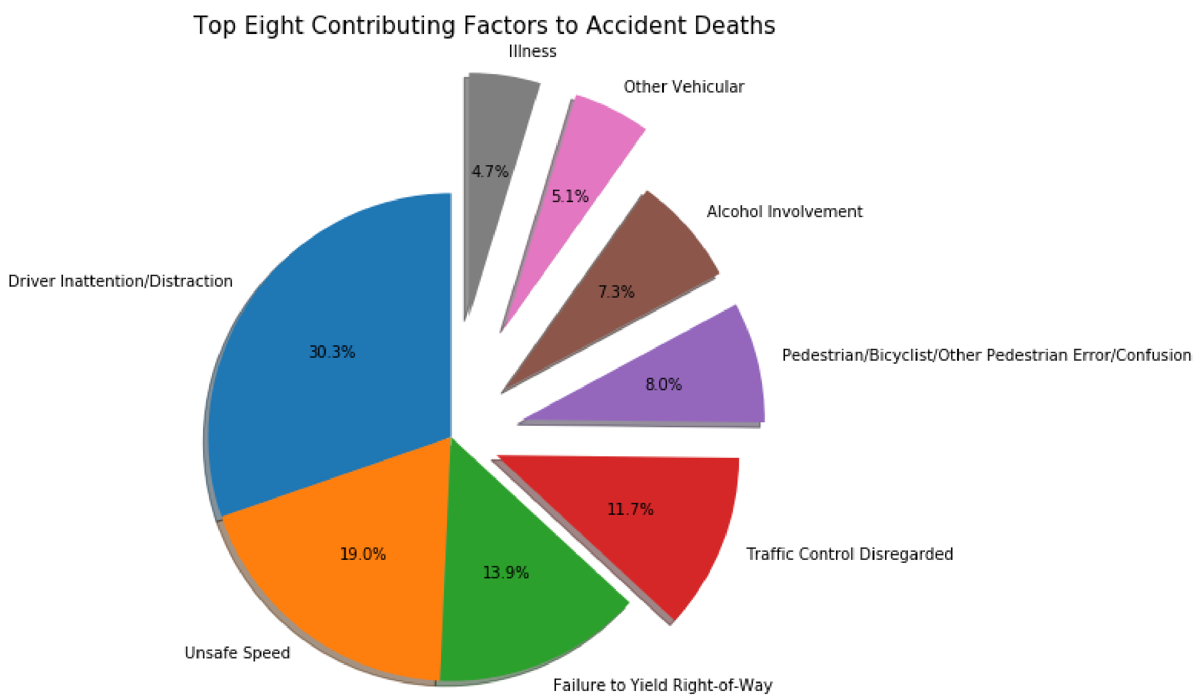
Surprisingly, alcohol consumption accounts for a relatively low percentage of injuries and deaths.

Officials looking to address injuries and/or fatalities can look to these results for direction in which practices to pursue and which behaviours to change.

In [34]:
```
#remove unspecified as an option
accidents = accidents[accidents.Contributing_Factor_1 != 'Unspecified']
```

In [35]:
```
contributingfactor = accidents.groupby(['Contributing_Factor_1']).sum() #gr
oup dataset by contributing factor

contributingfactorinjured = contributingfactor.sort_values(by='Total Injure
d', ascending=False) #get top 8 injuries
contributingfactorinjured =contributingfactorinjured.head(8)

contributingfactorkilled = contributingfactor.sort_values(by='Total Killed'
, ascending=False) #get top 8 deaths
contributingfactorkilled =contributingfactorkilled.head(8)

#Plot injured pie chart
labels = contributingfactorinjured.index
sizes =contributingfactorinjured["Total Injured"]
explode = (0,0,0,.2,.3,.3,.5,.5)
fig, ax = plt.subplots(figsize=(8, 8))
ax.pie(sizes, explode=explode, labels=labels, autopct='%1.1f%%',
        shadow=True, startangle=90)
ax.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circl
e.
ax.set_title("Top Eight Contributing Factors to Accident Injuries",fontsize
=15)

#Plot deaths pie chart
labels = contributingfactorkilled.index
sizes =contributingfactorkilled["Total Killed"]
explode = (0,0,0,.2,.3,.3,.5,.5)
fig, ax = plt.subplots(figsize=(8, 8))
ax.pie(sizes, explode=explode, labels=labels, autopct='%1.1f%%',
        shadow=True, startangle=90)
ax.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circl
e.
ax.set_title("Top Eight Contributing Factors to Accident Deaths",fontsize=1
5)

plt.show()
```


Top Eight Contributing Factors to Accident Injuries
Alcohol Involvement
3.7%
Other Vehicular
4.6%
Unsafe Speed
4.9%
Fatigued/Drowsy
5.1%
Traffic Control Disregarded
8.0%
Driver Inattention/Distraction
40.0%
Following Too Closely
15.4%
Failure to Yield Right-of-Way
18.3%


Top Eight Contributing Factors to Accident Deaths
Illness
4.7%
Other Vehicular
5.1%
Alcohol Involvement
7.3%
Pedestrian/Bicyclist/Other Pedestrian Error/Confusion
8.0%
Driver Inattention/Distraction
30.3%
Traffic Control Disregarded
11.7%
Unsafe Speed
19.0%
Failure to Yield Right-of-Way
13.9%

4. Conclusion

If NYC government officials are not already leveraging accident data to target efforts, there are many areas in which this data can guide their efforts. Citizens of and travelers to NYC can also stand to gain from greater knowledge of when and where to be most aware in NYC. Some observations in this dataset may also be applicable across other cities and countries, as it is reasonable to suggest that many trends around time and cause of accident are ubiquitous.

Key Observations

Key accident observations for the city of New York include:

Where

1. Staten Island has a significantly higher per capita death rate of motorists than other boroughs.
2. Manhattan is the most dangerous place for cyclists and pedestrians, but the safest place for motorists.
3. Brooklyn has the most injuries due to accidents per capita while Staten Island has the least.
4. However, Staten Island has the most deaths due to accidents per capita while the Bronx has by far the least.

When

1. Accidents occur the most during both morning and evening rush hour.
2. Even so, the most deaths occur between 4 and 7 am, plummeting in the morning rush hour but rising again during the evening rush hour.
3. Spring is the period with the fewest amount of injuries and deaths due to accidents.
4. The late summer and fall periods see the most accidents resulting in injuries and deaths.

How/Why

1. While the vast majority of injuries are to the motorists themselves, half of accidents deaths are born by pedestrians and cyclists.
2. The top cause of both injuries and deaths in accidents is Driver Inattention/Distraction
3. The top three causes of death in accidents are distracted drivers, unsafe speeds, and a failure to yield the right-of-way. These causes account for almost two-thirds of all deaths in accidents.