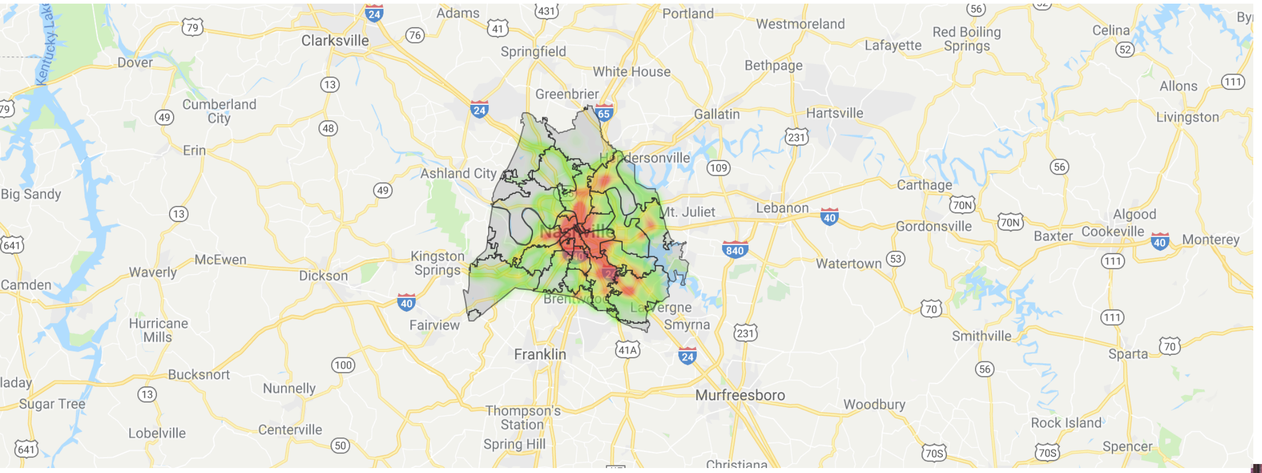
METRO NASHVILLE & DAVIDSON COUNTY
TRAFFIC CRASHES 2018

~~TEAM 1~~

…- Zachary Nguyen

OUTLINE



1. Crashes vs. Location

- Plot all reported accidents in Davidson County
- Heat map of all reported accidents in Davidson County, grouped by Zip-code Areas

2. Crashes vs. Time

- Number of all accidents vs Hours
- Number of all accidents vs Weekdays

Hypothesis: whether or not our observed frequencies of traffic accidents occur equally frequently for the different weekdays

DATA SOURCES & TOOLS

Data.Nashville.Gov

- 34.5K rows dataset (.csv) provides details about traffic crashes that occur in Davidson County
- Shape-file (.geojson) indicating the boundaries of Zip Code areas of Davidson County

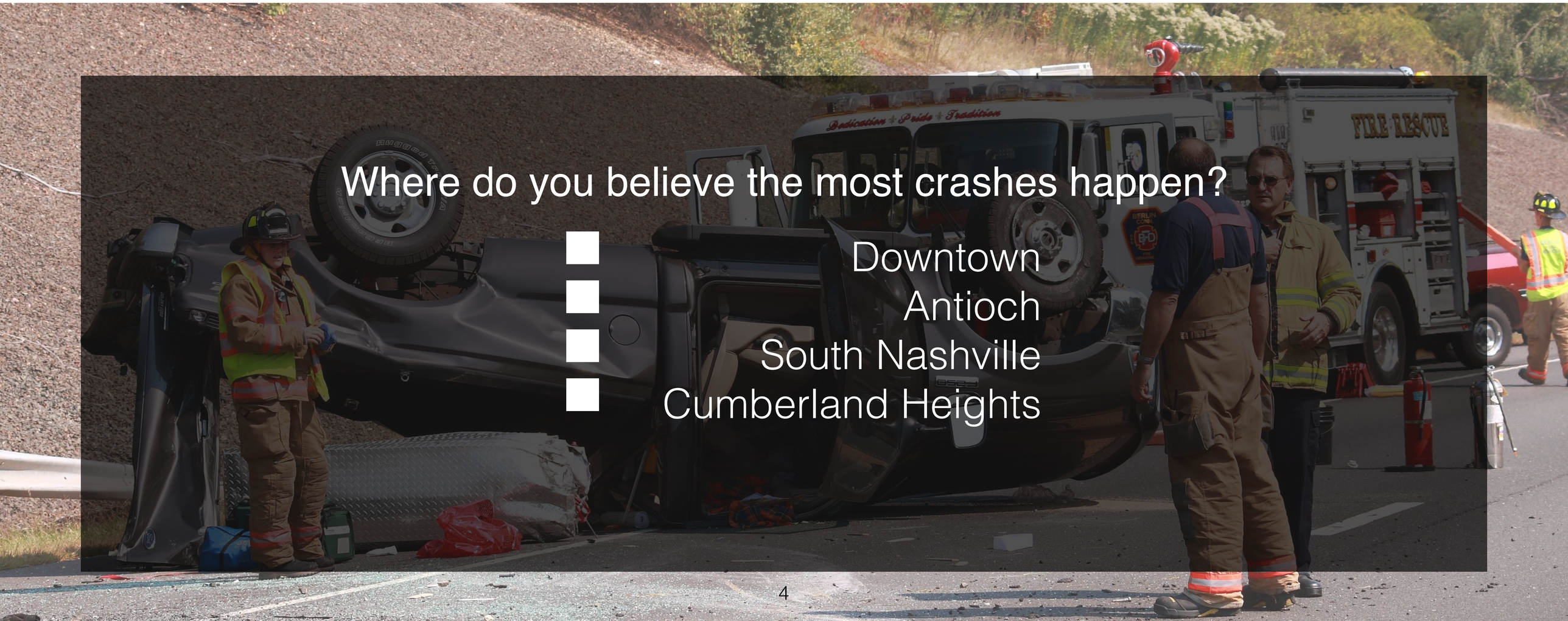
GOOGLE API

- Gmaps library to visualize and interact with geographical data.

Jupyter Notebook & Python

- Pandas
- NumPy
- Matplotlib
- Json
- Gmaps
- Scipy.stats

PART 1
NUMBER CRASHES VS. LOCATION



Where do you believe the most crashes happen?

- Downtown
- Antioch
- South Nashville
- Cumberland Heights

PART 1: CRASHES VS. LOCATION

Analysis 1- Plot All Reported Accidents in Davidson County With Gmaps

Import CSV file into Dataframe

```
df=pd.read_csv("../Metro_Nashville_Davidson_County_Traffic_Crashes_2018_.csv"
```

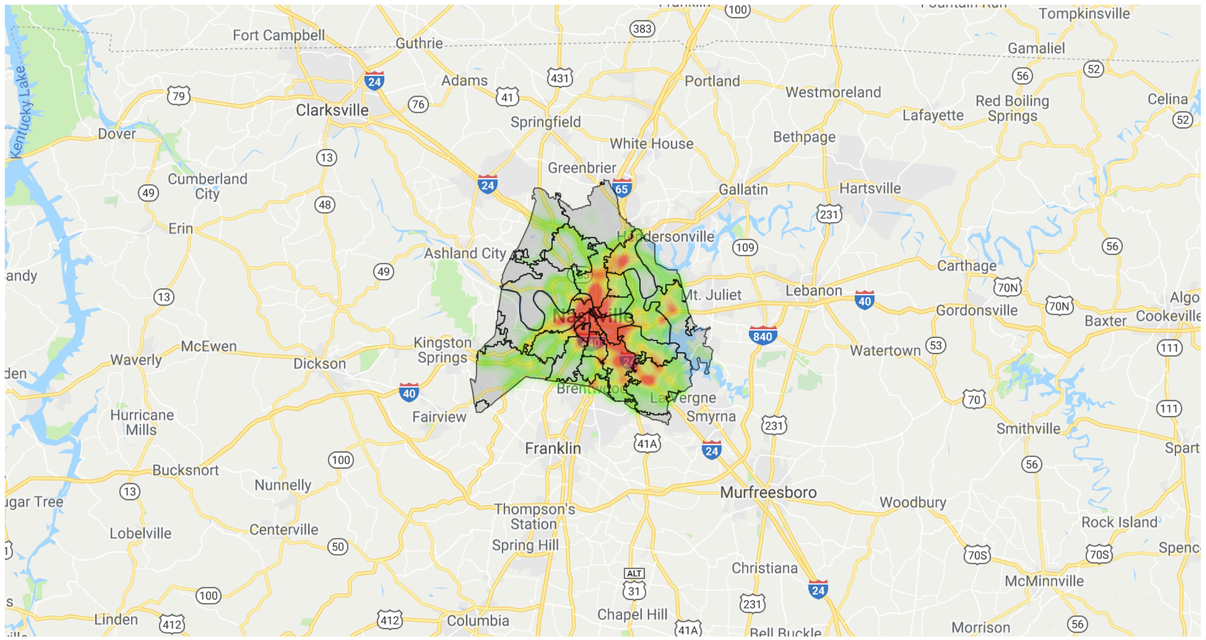
	Accident Number	Date and Time	Number of Motor Vehicles	Number of Injuries	Number of Fatalities	Property Damage	Hit and Run	Reporting Officer	Collision Type Code	Collision Type Description	...	Harmful Code Description	Street Address	City
0	20181075326	12/31/2018 11:10:00 PM	1.0	0	0	NaN	N	240964.0	0.0	NOT COLLISION W/MOTOR VEHICLE-TRANSPORT	...	CURB;SHRUBBERY	ROCKWOOD DR & SADDLESTONE DR	HERMITAG
1	20181075323	12/31/2018 11:09:00 PM	3.0	1	0	NaN	N	110062.0	4.0	ANGLE	...	MOTOR VEHICLE IN TRANSPORT;PARKED MOTOR VEHICLE	SPRING ST & N 1ST ST	NASHVILL

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 34479 entries, 0 to 34478
Data columns (total 25 columns):
Accident Number                 34479 non-null int64
Date and Time                   34479 non-null object
Number of Motor Vehicles        34478 non-null float64
Number of Injuries              34479 non-null int64
Number of Fatalities            34479 non-null int64
Property Damage                 2386 non-null object
Hit and Run                     34479 non-null object
Reporting Officer               34478 non-null float64
Collision Type Code             34478 non-null float64
Collision Type Description      34478 non-null object
Weather Code                    34407 non-null float64
Weather Description             34407 non-null object
Illumination Code               34448 non-null float64
Illumination Description        34448 non-null object
Harmful Code                    34479 non-null object
Harmful Code Description        34152 non-null object
Street Address                  34478 non-null object
City                            34479 non-null object
State                           34479 non-null object
Zip                             34423 non-null float64
RPA                             34458 non-null float64
Precinct                        34453 non-null object
Latitude                        34446 non-null float64
Longitude                       34446 non-null float64
Mapped Location                 34446 non-null object
dtypes: float64(9), int64(3), object(13)
memory usage: 6.6+ MB
```

Removing any rows with NULL values in Latitude OR Longitude columns

```
filtered_df=df[((df.Latitude.notnull())|(df.Longitude.notnull()))]
```

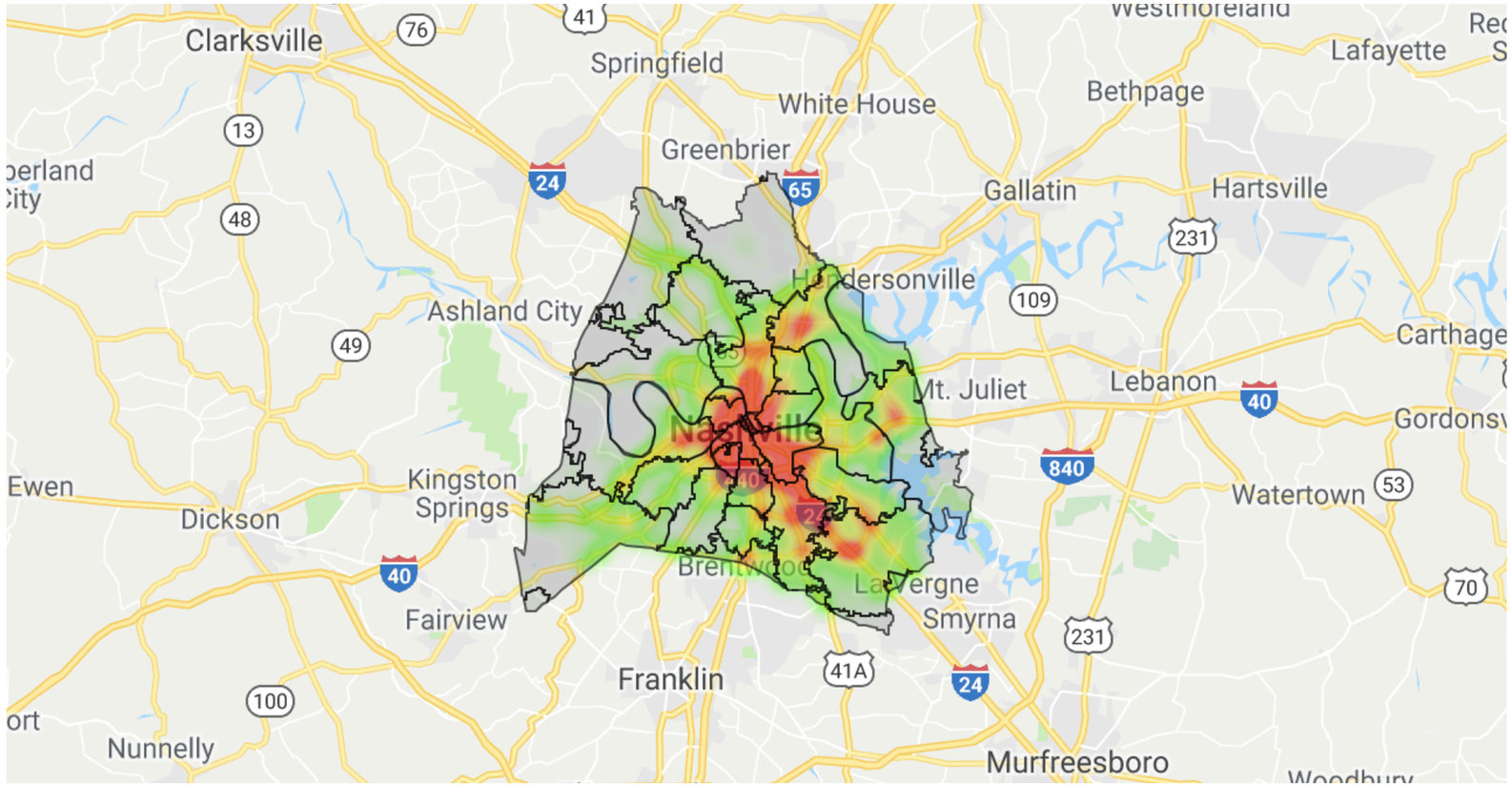
{DataFrame:34446,25}




```
import gmaps
gmaps.configure(api_key='AIz…')
…
import json
with open('../Data_Sources/Davidson.geojson') as f:
    zip_geo_json = json.load(f)

…
fig.add_layer(gmaps.heatmap_layer(filtered_df[['Latitude','Longitude']]))
```

PART 1: CRASHES VS. LOCATION

Analysis 1 - all Reported Accidents in Davidson County With Gmaps



- Most of car accidents (red dots) was happened on Highways: I24, I65, I40, I440

PART 1: CRASHES VS. LOCATION (CONT.)

Analysis 2-Heat Map of all Reported Accidents in Davidson County, Grouped by Zip-Code Areas

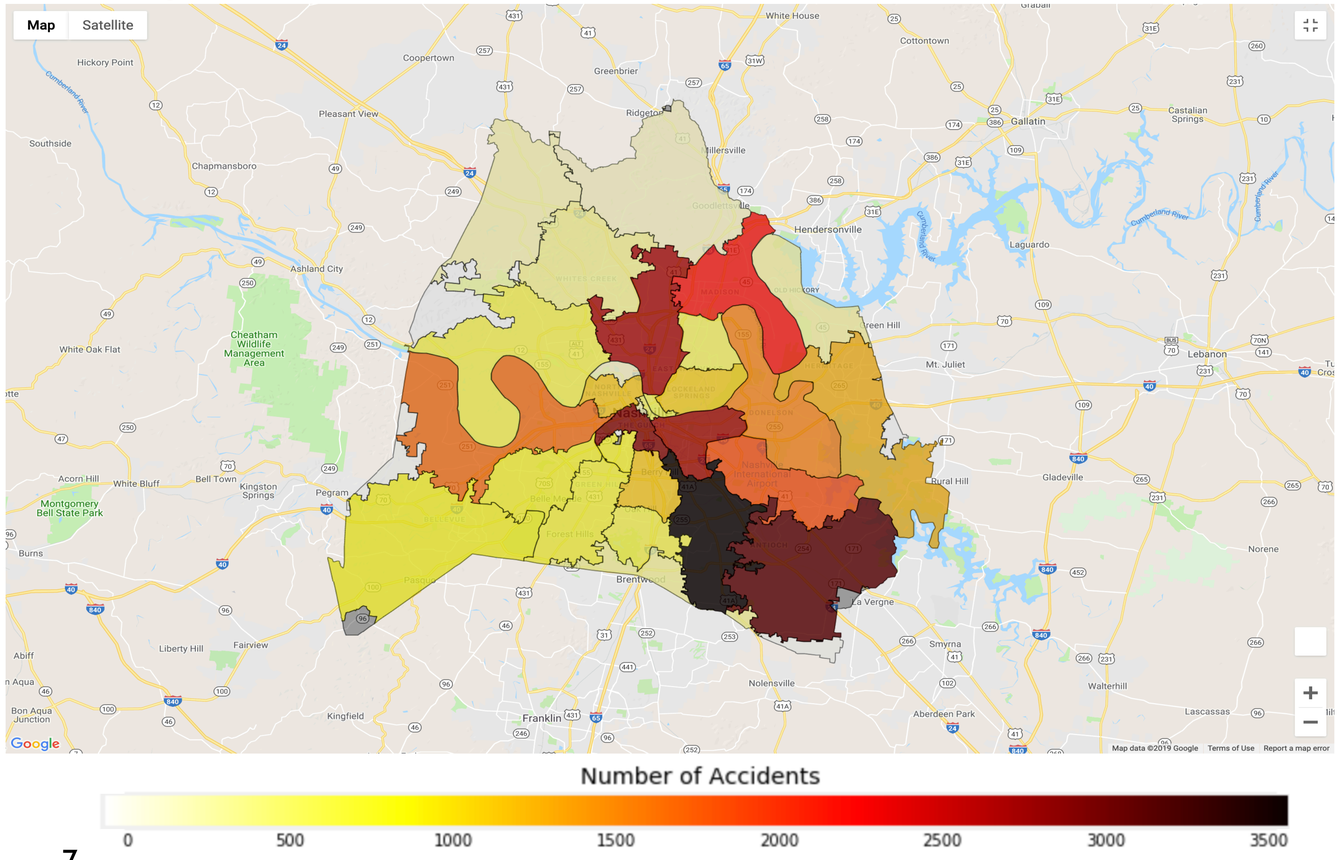
Number of car accidents in each Zip-code area

```
zip_crashes=filtered_df.groupby(['Zip']).agg({"Accident Number":"count"})
```

Zip	Accident Number
37211	3516
37013	3076
37203	2835
37210	2691
37207	2655

Create function: "calculate_color" to generate the color code, based on the corresponding Accident number

```
# We will need to scale the Accident values to lie between 0 and 1
min_crash = min(zip_crashes["Accident Number"])
max_crash = max(zip_crashes["Accident Number"])
crash_range = max_crash - min_crash

# find the color code, based on the Accident number
def calculate_color(crash):
    """
    Convert the Accident Number to a color
    """
    # make the number between 0 and 1
    normalized_crash = (crash - min_crash) / crash_range

    # invert the number  so that high inequality gives dark color
    inverse_crash = 1.0 - normalized_crash
    #inverse_crash = normalized_crash

    # transform the number to a matplotlib color
    mpl_color = hot(inverse_crash)

    # transform from a matplotlib color to a valid CSS color
    gmaps_color = to_hex(mpl_color, keep_alpha=False)

    return gmaps_color
```



PART 1: CRASHES VS. LOCATION (CONT.)

Analysis 2-Heat Map of all Reported Accidents in Davidson County, Grouped by Zip-Code Areas

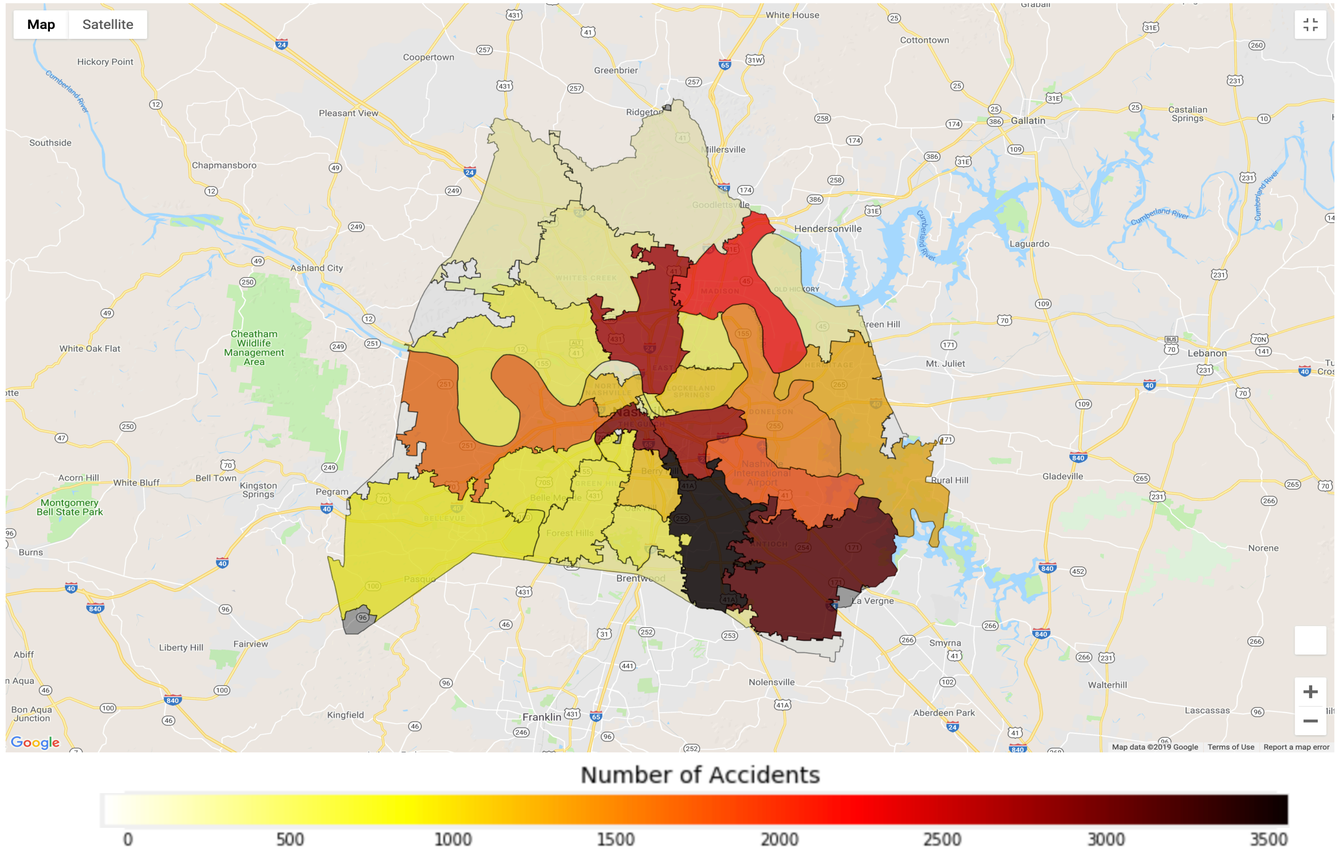
Genreate a list of colors for all Zipcode areas in Davidson County based on the Accident numbers

```
colors = []
accident_nums = []
for feature in zip_geo_json['features']:
    zipcode = int(feature['properties']['zip'])
    try:
     # get nummber of accidents and the corresponding color for each
     Zipcode found from the report
        accident_num = zip_crashes.loc[zipcode]['Accident Number']
        color = calculate_color(accident_num)

    # set the color for those Zipcode areas that are not in the report
    except KeyError:
        # no nummber of accidents for that Zipcode return default
color
        color = (0, 0, 0, 0.3)
        accident_num = 0

    colors.append(color)
    accident_nums.append(accident_num)
```

Zip	Accident Number
37211	3516
37013	3076
37203	2835
37210	2691
37207	2655

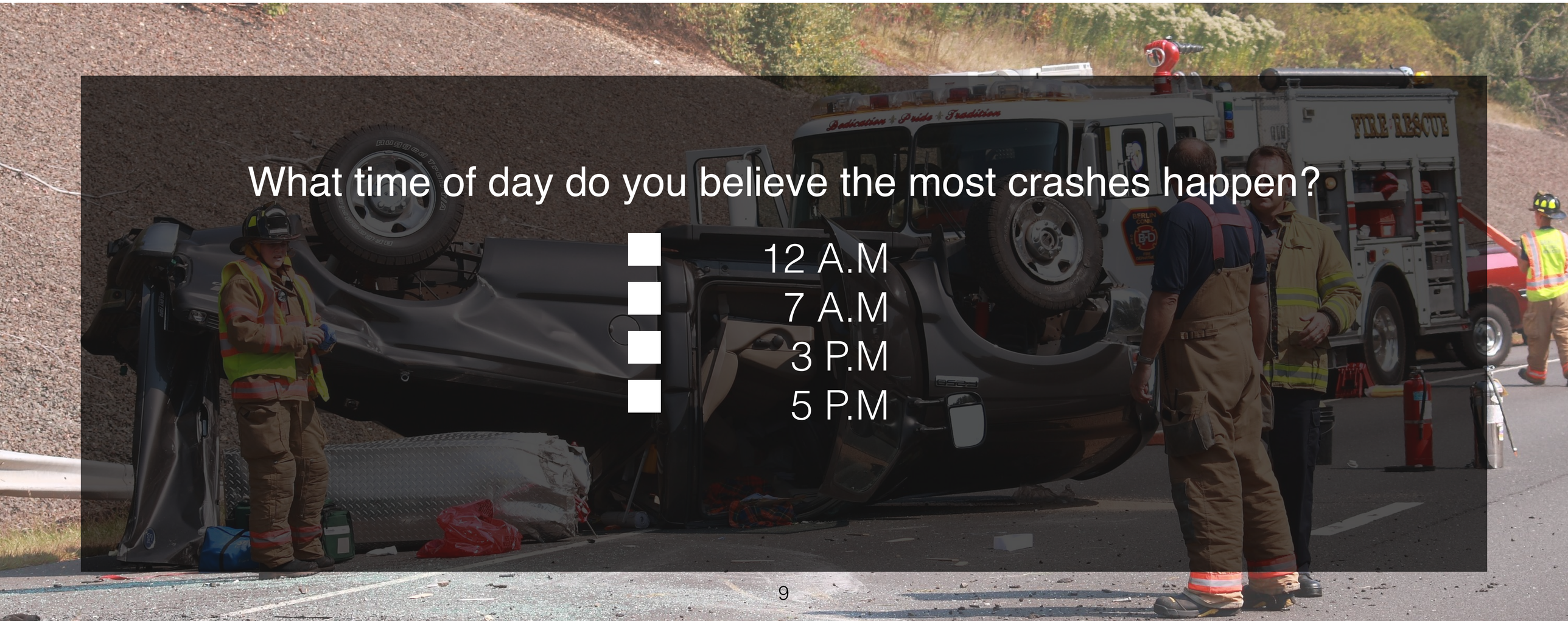
Plotting the heat map

```
fig = gmaps.figure()
crash_layer = gmaps.geojson_layer(
    zip_geo_json,
    fill_color=colors,
    stroke_color=colors,
    fill_opacity=0.8)
fig.add_layer(crash_layer)
# border
fig.add_layer(gmaps.geojson_layer(zip_geo_json,fill_opacity=0.05))
```



PART 2: CRASHES VS. TIME

Analysis 1 - Crashes vs. Hours



What time of day do you believe the most crashes happen?

- 12 A.M
- 7 A.M
- 3 P.M
- 5 P.M

PART 2: CRASHES VS. TIME

Analysis 1 - Number of all Reported Crashes by 24 Hours

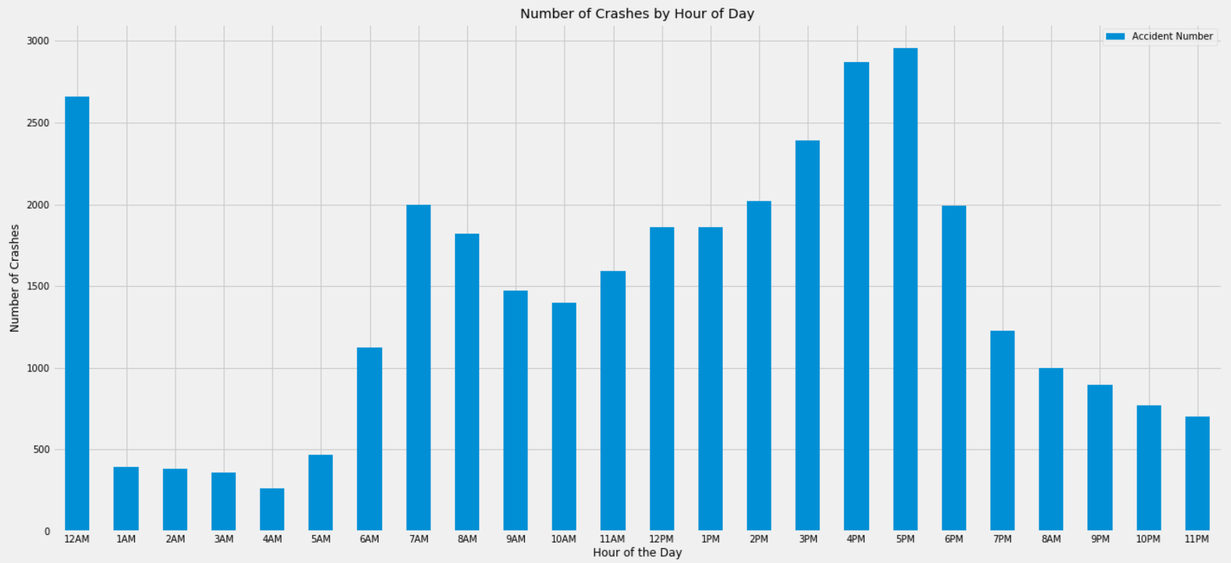
Data manipulation: Add columns of detailed Time info (from "Date and Time" column)

```
df['Date']=pd.to_datetime(df['Date and Time'])
df['WeekDay']=df.Date.dt.weekday_name
df['Hour']=df.Date.dt.hour
df['Minute']=df.Date.dt.minute
df['Second']=df.Date.dt.second
```

```
df[['Accident Number', 'Date and Time','Date',
       'WeekDay', 'Hour', 'Minute', 'Second']].head()
```

Out[29]:

	Accident Number	Date and Time	Date	WeekDay	Hour	Minute	Second
0	20181075326	12/31/2018 11:10:00 PM	2018-12-31 23:10:00	Monday	23	10	0
1	20181075323	12/31/2018 11:09:00 PM	2018-12-31 23:09:00	Monday	23	9	0
2	20181075301	12/31/2018 10:55:00 PM	2018-12-31 22:55:00	Monday	22	55	0
3	20181075265	12/31/2018 10:44:00 PM	2018-12-31 22:44:00	Monday	22	44	0
4	20181075263	12/31/2018 10:42:00 PM	2018-12-31 22:42:00	Monday	22	42	0



Number of car accidents grouped by Hours

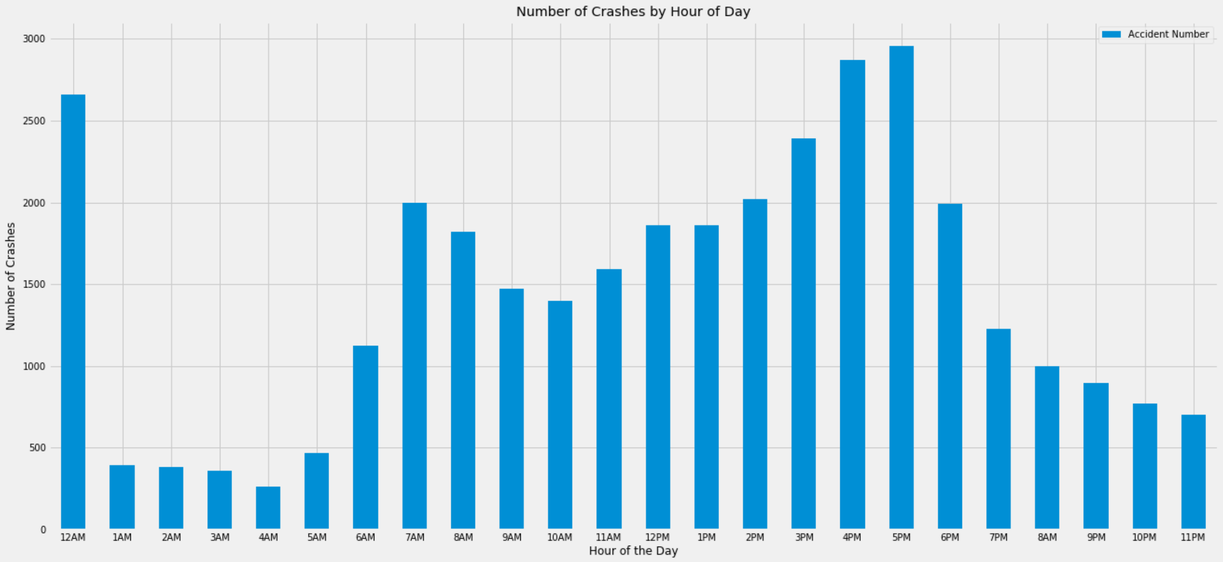
```
hour_crashes=df.groupby(['Hour']).agg({"Accident Number":"count"})
```

Bar Graph

```
# Plotting
hour_crashes.plot(kind='bar',figsize=(20,10))
plt.title("Number of Crashes by Hour of Day")
plt.xlabel("Hour of the Day")
plt.ylabel("Number of Crashes")
plt.xticks(np.arange(0,24,step=1),
('12AM','1AM',‘2AM','3AM','4AM','5AM','6AM','7AM','8AM','9AM','10AM'
,'11AM','12PM',‘1PM','2PM','3PM','4PM','5PM','6PM','7PM','8AM','9PM'
,'10PM','11PM'),rotation=0)
plt.savefig("Number of Crashes by Hour of Day - All")
plt.show()
```

PART 2: CRASHES VS. TIME

Analysis 1 - Number of all Reported Crashes by 24 Hours



Note:

- The number of all reported crashes by hour of day in Davidson County is likely related to the traffic hour.
  - More accidents occurred at rush-hour ranges: 7:00-9:00 A.M. and 4:00-6:00P.M.
- There was an abnormal data in this plot at the range of 12:00-1:00 A.M where the number of crashes was relatively high compared with its neighbor ranges (i.e., 11:00 P.M. -12:00:00 A.M and 1:00 - 2:00 A.M.)
  - Car accidents with unknown time were automatically assigned as exact the midnight (i.e., 12:00:00 A.M.)
- We need to identify and handle those unknown-time data properly.

PART 2: CRASHES VS. TIME

Analysis 1 - Number of all Reported Crashes by 24 Hours

Fig1: Including unknown-timed accidents

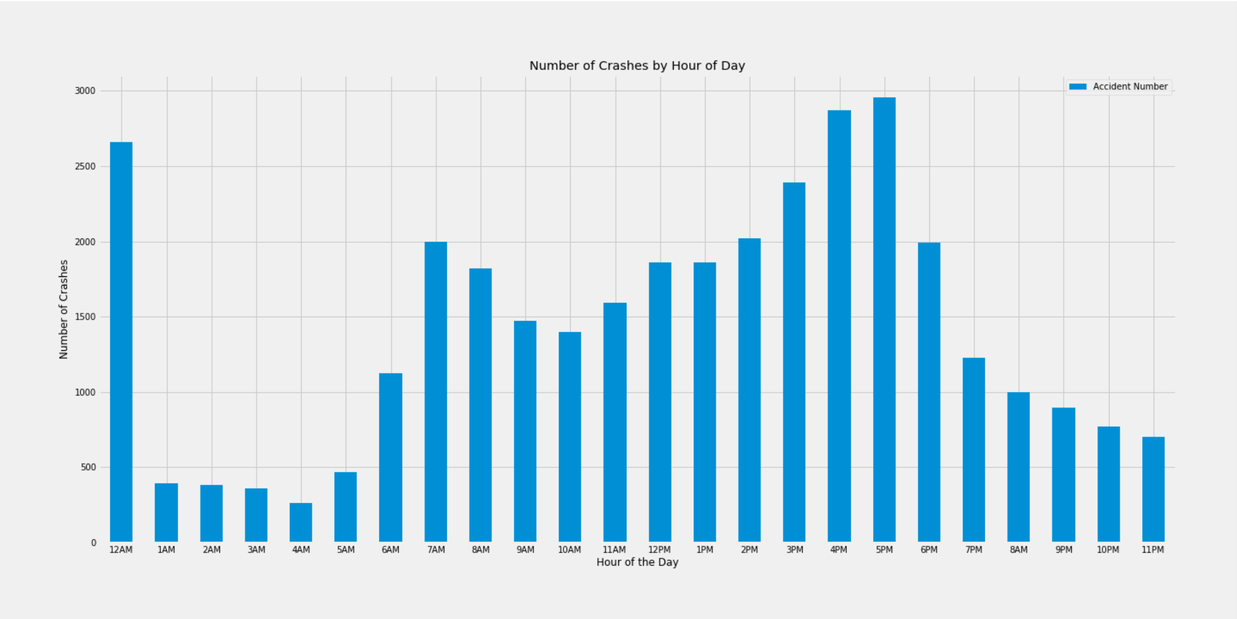
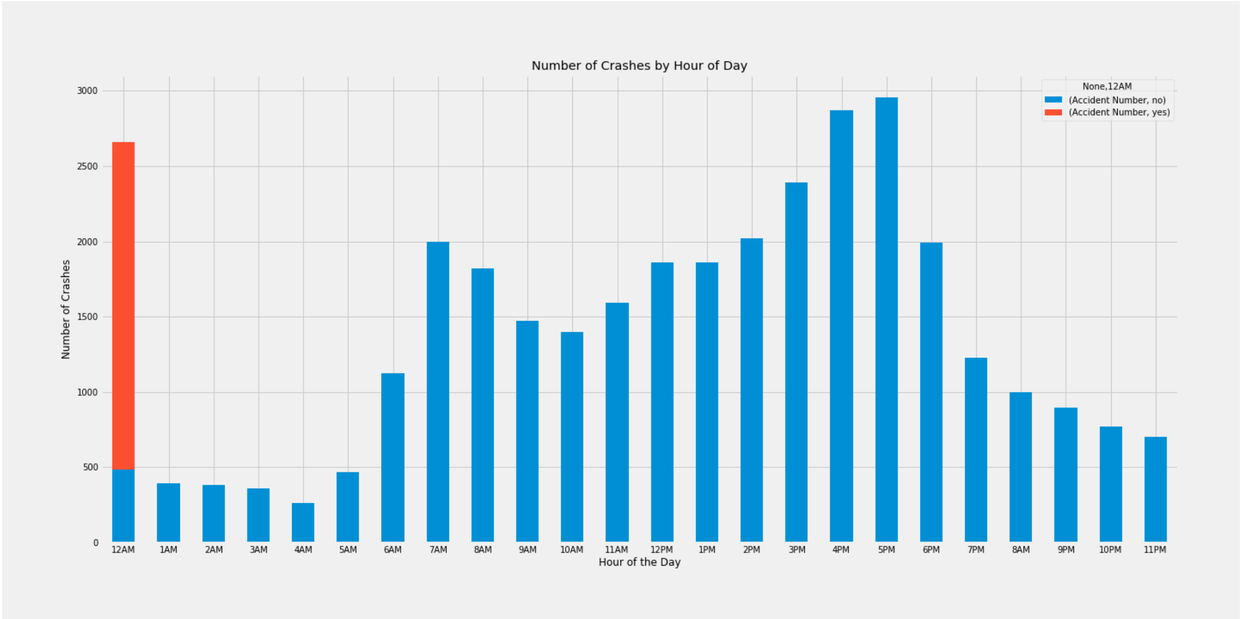


Fig2: Separation of unknown-timed accidents

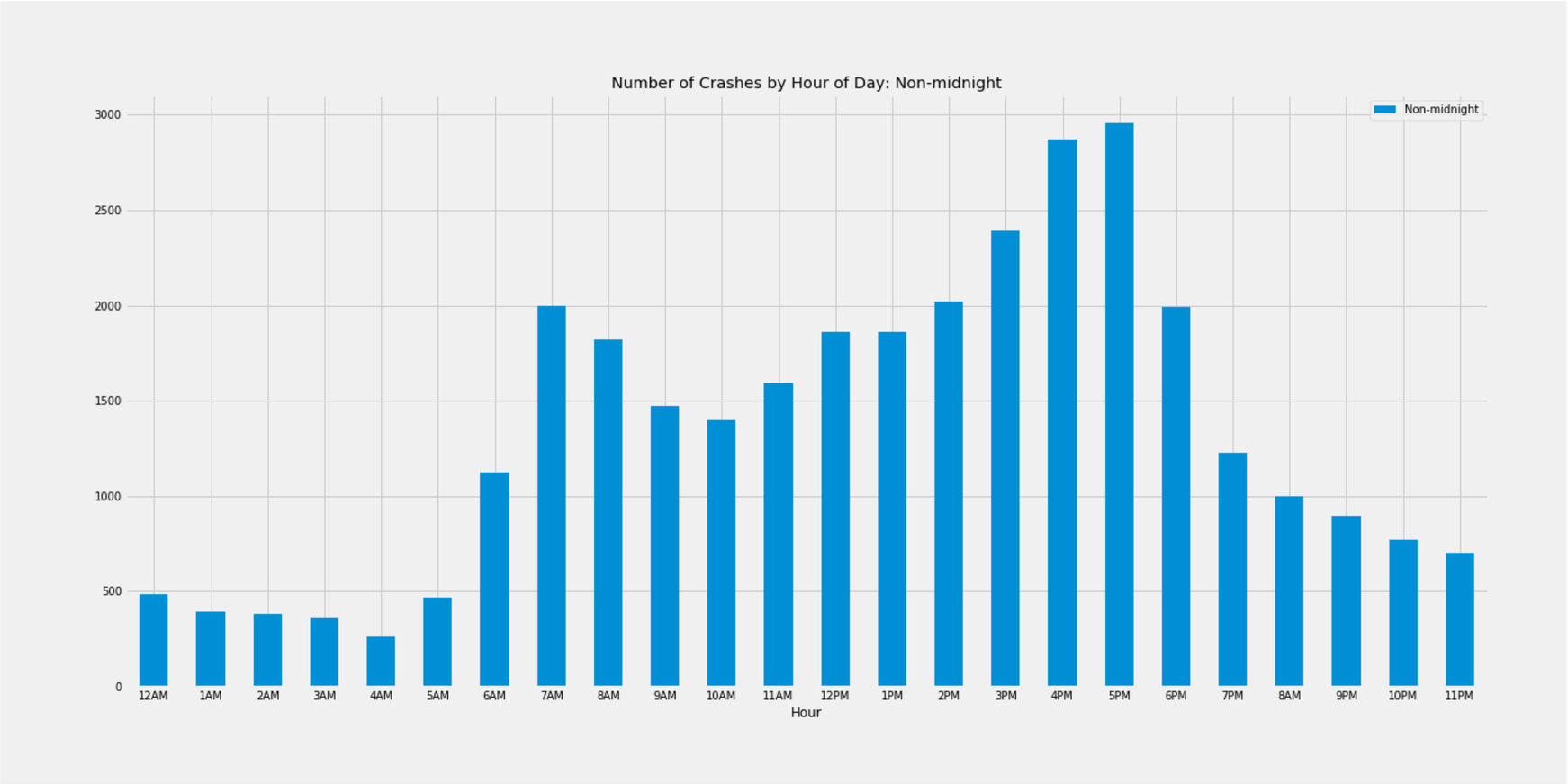



```
midnight_df=df[(df.Hour == 0)&(df.Minute==0)&(df.Second==0)]
notmidnight_df=df[(df.Hour != 0)|(df.Minute!=0)|(df.Second!=0)]
```

- **2,169 rows** from the data frame with a time of 12:00 AM, as police officers use this time if the exact time of the accident is unknown

PART 2: CRASHES VS. TIME

Analysis 1 - Number of all Reported Crashes by 24 Hours

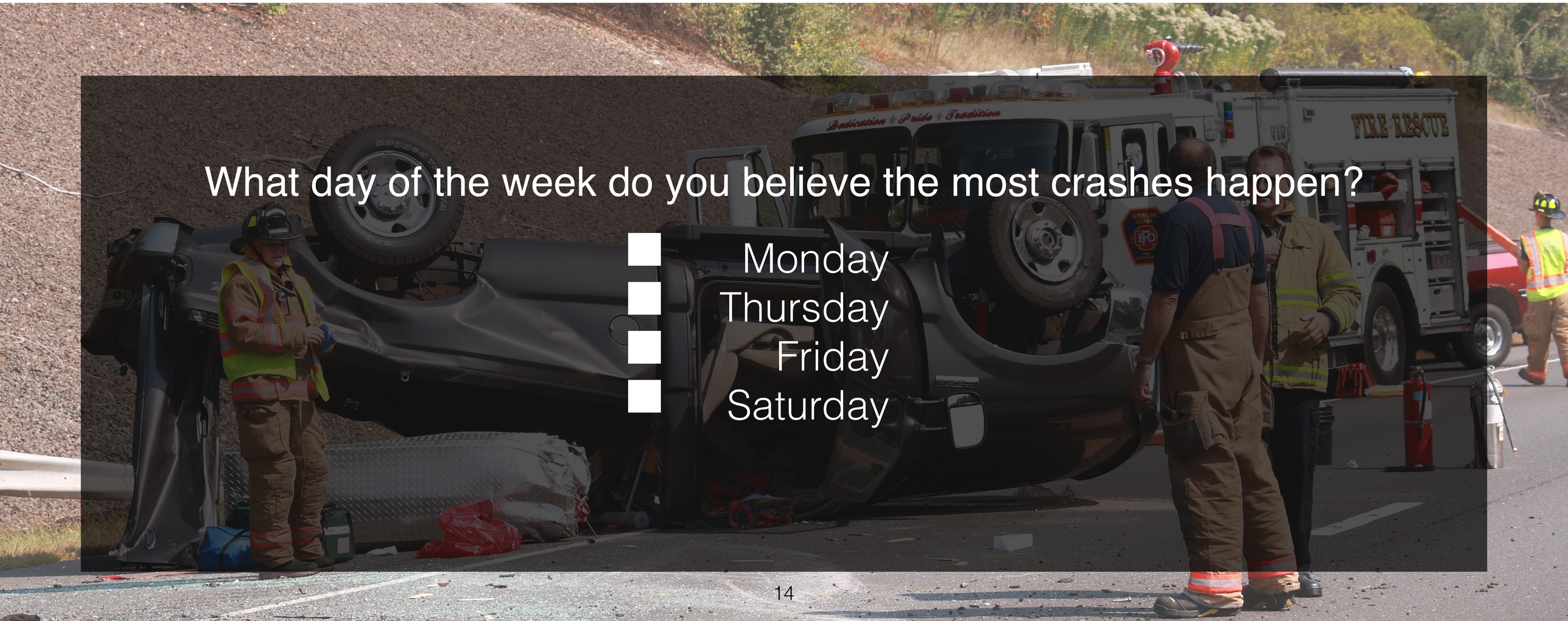


Note:

- After those unknown-time data were separated and removed properly, the final data now looks more reasonable!

PART 2: CRASHES VS. TIME

Analysis 2 - Crashes vs. Weekdays



What day of the week do you believe the most crashes happen?

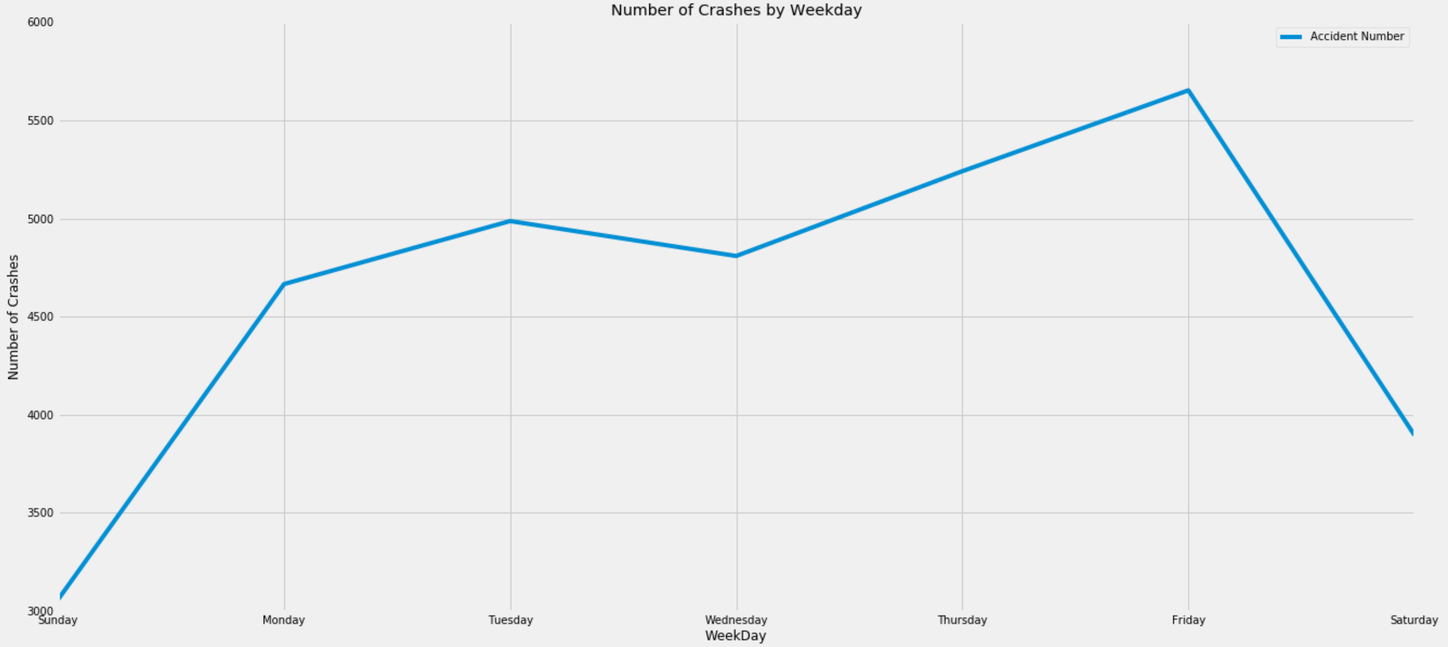
- Monday
- Thursday
- Friday
- Saturday

PART 2: CRASHES VS. TIME

Analysis 2 - Number of all Reported Crashes by Weekdays

Number of car accidents grouped by Weekdays

```
cats = ['Sunday','Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', ‘Saturday']
wdcs=notmidnight_df.groupby([‘WeekDay']).agg({"Accident Number":"count"}).reindex(cats)
```



Line Graph

```
wdcs.plot(kind=‘line',figsize=(20,10),rot=0)
plt.ylim(3000,6000)
plt.title("Number of Crashes by Weekday")
plt.ylabel("Number of Crashes")
plt.margins(3)
plt.savefig("Number of Crashes by Weekday")
plt.show()
```

Question:

Whether or not do our reported frequencies of traffic accidents occur equally frequently for the different weekdays?

PART 2: CRASHES VS. TIME

Analysis 2 - Number of all Reported Crashes by Weekdays: Statistics-Chi Square Test

Chi-Square test:

- Chi-square is used to test hypotheses about the distribution of observations in different categories.
- The larger the value of χ^2, the more likely it is that the distributions are significantly different.

$$\tilde{\chi}^2 = \sum_{k=1}^{n} \frac{(O_k - E_k)^2}{E_k}$$

Null Hypotheses:

The frequencies of traffic accidents occur equally frequently for the different weekdays

Creation of expected data:

```
wdcs['Expected Accident Number'] = int(wdcs["Accident Number"].sum()/7)
```

WeekDay	Accident Number	Expected Accident Number
Sunday	3330	4925
Monday	4989	4925
Tuesday	5300	4925
Wednesday	5062	4925
Thursday	5541	4925
Friday	6038	4925
Saturday	4219	4925

Calculation of the standard normal distribution for the probability 0.95

```
# The statistical module used to run chi square test
import scipy.stats as stats

# With a p-value of 0.05, the confidence level is 1.00-0.05 = 0.95.
critical_value = stats.chi2.ppf(q = 0.95, df = weekday_crashes.shape[0])
critical_value
```

```
14.067140449340169
```

Run the chi square test with stats.chisquare()

```
stats.chisquare(wdcs["Accident Number"],wdcs['Expected Accident Number'])
```

```
Power_divergenceResult(statistic = 979.5281218274112,
                       pvalue.   = 2.3926998428223567e-208)
```

Comparison:

```
statistic = 979.5281218274112 > critical_value = 14.067140449340169
```

We can therefore be relatively confident in concluding that our observed frequencies are significantly different from the frequencies that we would expect to obtain if all categories were equally distributed. In other words, weekday is related to the amount of road traffic accidents that occur.

PART 2: CRASHES VS. TIME

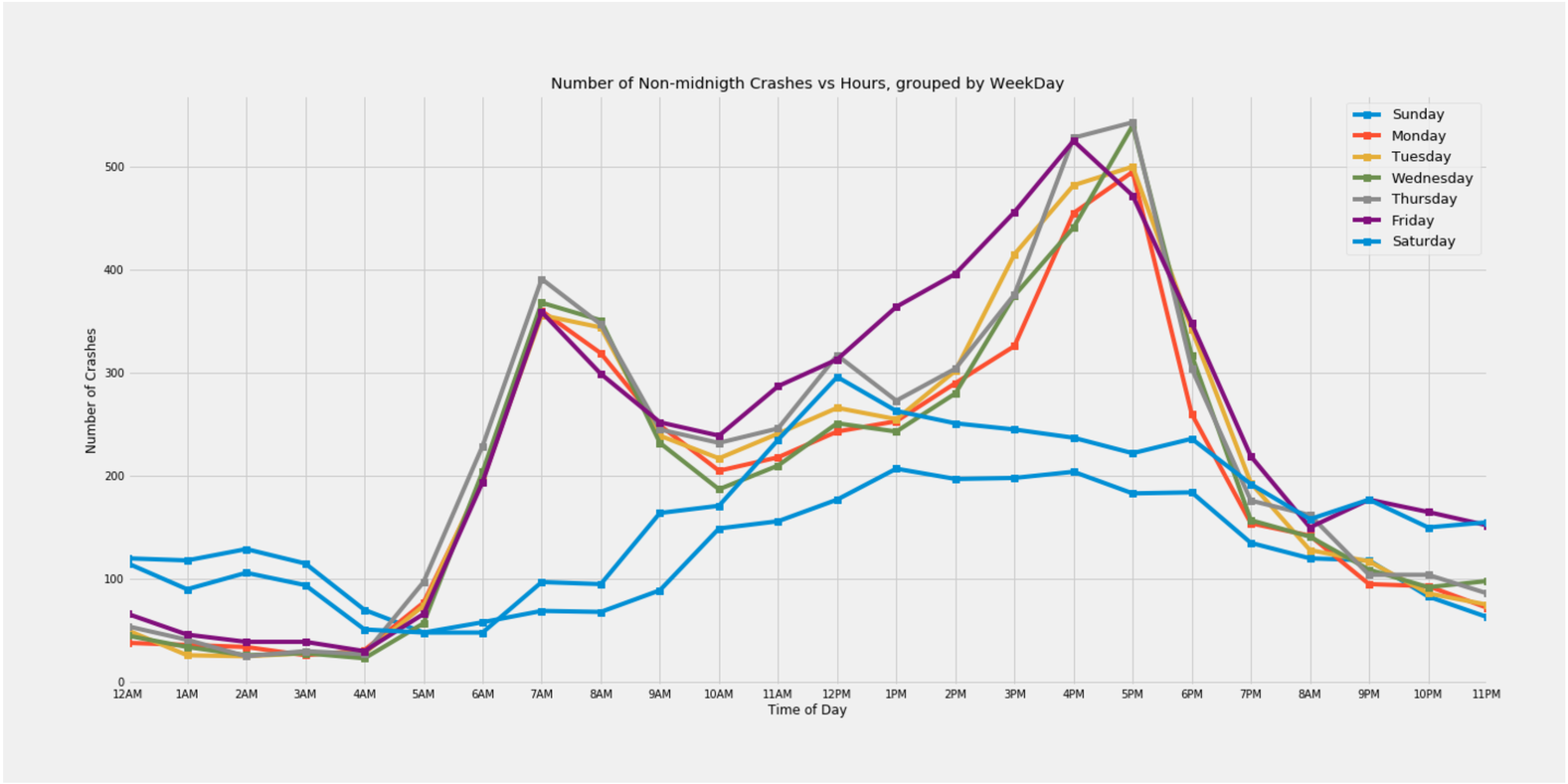
Analysis 3 - Number of all Reported Crashes vs. Weekdays by Time of Day

Number of car accidents grouped by Weekdays & Hour

```
wdhour_crashes=notmidnight_df.groupby(['WeekDay','Hour']).agg({"Accident Number”:”count"})
wdhour_crashes=wdhour_crashes.reset_index()
pivot_wdhour_crashes=wdhour_crashes.pivot(index='Hour',columns='WeekDay',values='Accident Number’)
cats = ['Sunday','Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday']
pivot_wdhour_crashes=pivot_wdhour_crashes[cats]
```

Line Graph

```
pivot_wdhour_crashes.plot(figsize=(20,10),marker='s')
plt.title("Number of Non-midnigth Crashes vs Hours,
grouped by WeekDay")
plt.xlabel("Time of Day")
plt.ylabel("Number of Crashes")
plt.xticks(np.arange(0,24,step=1),('12AM', '1AM',
'2AM','3AM','4AM','5AM','6AM','7AM','8AM','9AM','10AM',
'11AM',
                                   '12PM', '1PM',
'2PM','3PM','4PM','5PM','6PM','7PM','8AM','9PM','10PM',
'11PM'))
plt.legend(loc='best', prop={'size': 13})
plt.savefig("Number of Crashes VS Hour By Weekday")
plt.show()
```



SUMMARY

- The majority of accidents are clustered in the metro Nashville area and are less widespread as you leave the downtown radius. The exception is leaving Nashville going Eastbound/Westbound on I-24.

- Top 5 of Zip-code areas have the highest number of of accidents are: 37211, 37013, 37203, 37210, 37207

- The highest number of accidents occurred on Friday.
- The lowest number of accidents occurred on Sunday.

- The highest number of accidents occurred on Thursday at 5:00pm.
- The lowest number of accidents occurred on Monday at 12:00am (midnight).

- The data showed a peak at 7:00am of accidents on Monday-Friday.
- The data showed a peak at 5:00pm of accidents on Monday-Thursday and a peak of accidents at 4:00pm on Friday.

This project was completed by using pandas, numpy, matplotlib, json, spicy.stats & gmaps with Gmaps API

WHAT WE WOULD ANALYZE IF WE HAD MORE TIME

1 Open Weather:
We could use Open Weather API to see if there was any correlation between weather and accidents

2 Census Data:
We could used the US Census API data to review the total population change from 2008 vs. 2018 per zip code to see if a population change resulted in more crashes

3 Census Data:
We could use the US Census API for population by zip code in Davidson County to determine if more accidents occurred in areas with a larger population.

4 Location Analysis:
We could compare location data to see if any certain intersection has a higher crash risk

Thank You

Any Questions?