In [2]:

```
import pandas as pd
import numpy as np
import csv

import scipy.stats as scs
import statsmodels.api as sm
import statsmodels.formula.api as sms
import scipy.stats as stats

from math import sqrt

from sklearn.preprocessing import OneHotEncoder
from sklearn.tree import DecisionTreeClassifier
from sklearn import tree
from sklearn.feature_selection import SelectKBest, chi2
from sklearn.metrics import accuracy_score, confusion_matrix, classification_

import matplotlib.pyplot as plt
import seaborn as sns
```

Question 1

What parts of Chicago have the most fatalities?

In [3]:

```
df = pd.read_csv(r'data\ChicagoCrashes.csv')
```

In [4]:

```
df.describe()
```

	Unnamed: 0	CRASH_DATE_x	OCCUPANT_CNT	POSTED_SPEED_LIMIT	BEAT_OF
count	1.134909e+06	1.134909e+06	1.134909e+06	1.134909e+06	1.134909e
mean	9.897487e+05	2.018056e+03	1.415067e+00	2.888418e+01	1.233346e
std	5.947572e+05	1.283893e+00	1.418414e+00	5.913001e+00	6.996664e
min	0.000000e+00	2.015000e+03	0.000000e+00	0.000000e+00	1.110000e
25%	4.751850e+05	2.017000e+03	1.000000e+00	3.000000e+01	7.250000e
50%	9.654550e+05	2.018000e+03	1.000000e+00	3.000000e+01	1.212000e
75%	1.493715e+06	2.019000e+03	2.000000e+00	3.000000e+01	1.821000e
max	2.115933e+06	2.020000e+03	6.000000e+01	9.900000e+01	2.535000e

In [5]:

```
df2 = df.sample(frac=0.0005)
```

In [6]:

```
df.info()
```

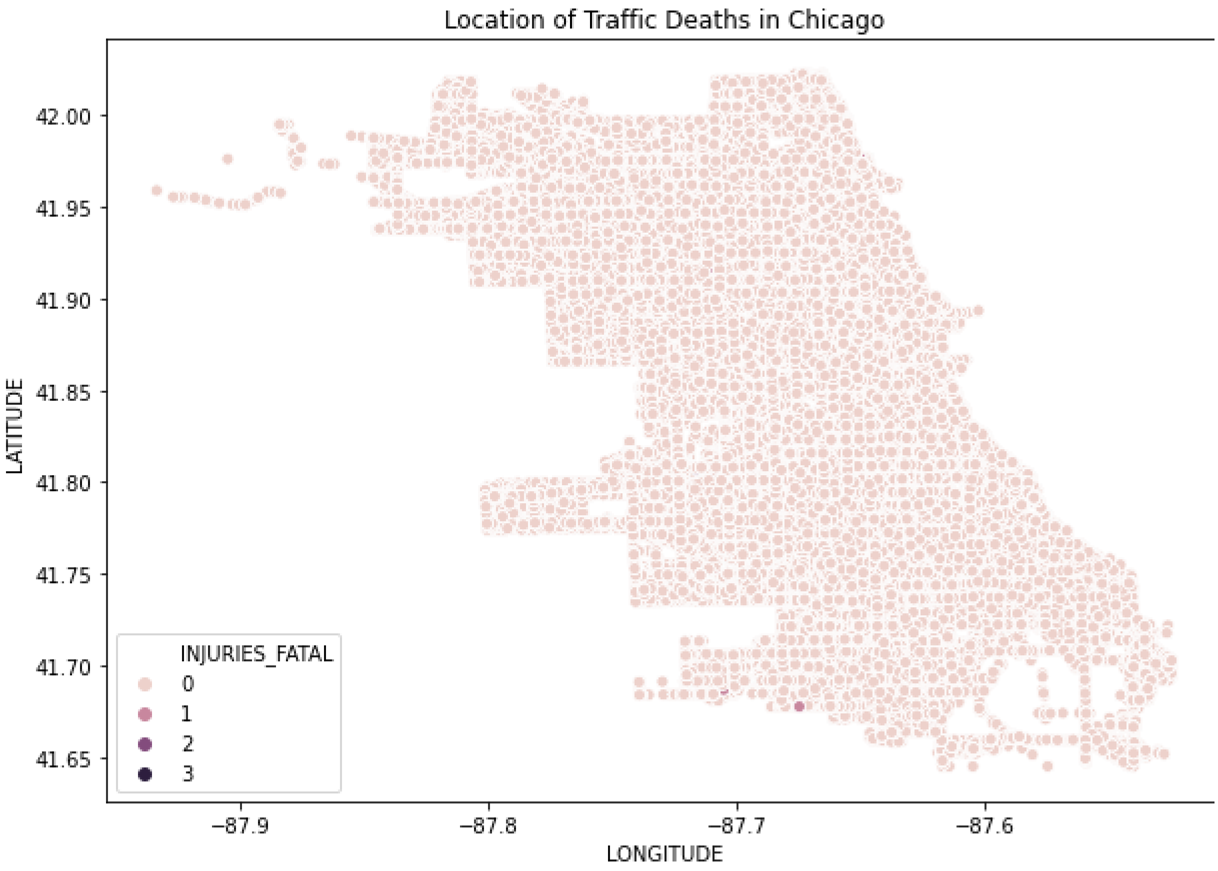
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1134909 entries, 0 to 1134908
Data columns (total 49 columns):
 #   Column                         Non-Null Count    Dtype
---  ------                         --------------    -----
 0   Unnamed: 0                     1134909 non-null  int64
 1   CRASH_DATE_x                   1134909 non-null  int64
 2   UNIT_TYPE                      1134909 non-null  object
 3   MAKE                           1134909 non-null  object
 4   MODEL                          1134909 non-null  object
 5   VEHICLE_DEFECT                 1134909 non-null  object
 6   VEHICLE_TYPE                   1134909 non-null  object
 7   VEHICLE_USE                    1134909 non-null  object
 8   MANEUVER                       1134909 non-null  object
 9   OCCUPANT_CNT                   1134909 non-null  float64
 10  CRASH_DATE_y                   1134909 non-null  object
 11  POSTED_SPEED_LIMIT             1134909 non-null  int64
 12  TRAFFIC_CONTROL_DEVICE         1134909 non-null  object
 13  DEVICE_CONDITION               1134909 non-null  object
 14  WEATHER_CONDITION              1134909 non-null  object
 15  LIGHTING_CONDITION             1134909 non-null  object
 16  FIRST_CRASH_TYPE               1134909 non-null  object
 17  TRAFFICWAY_TYPE                1134909 non-null  object
 18  ALIGNMENT                      1134909 non-null  object
 19  ROADWAY_SURFACE_COND           1134909 non-null  object
 20  ROAD_DEFECT                    1134909 non-null  object
 21  REPORT_TYPE                    1134909 non-null  object
 22  CRASH_TYPE                     1134909 non-null  object
 23  DAMAGE                         1134909 non-null  object
 24  PRIM_CONTRIBUTORY_CAUSE        1134909 non-null  object
 25  SEC_CONTRIBUTORY_CAUSE         1134909 non-null  object
 26  BEAT_OF_OCCURRENCE             1134909 non-null  float64
 27  NUM_UNITS                      1134909 non-null  int64
 28  MOST_SEVERE_INJURY             1134909 non-null  object
 29  INJURIES_TOTAL                 1134909 non-null  float64
 30  INJURIES_FATAL                 1134909 non-null  float64
 31  INJURIES_INCAPACITATING        1134909 non-null  float64
 32  INJURIES_NON_INCAPACITATING    1134909 non-null  float64
 33  INJURIES_REPORTED_NOT_EVIDENT  1134909 non-null  float64
 34  INJURIES_NO_INDICATION         1134909 non-null  float64
 35  INJURIES_UNKNOWN               1134909 non-null  float64
 36  CRASH_HOUR                     1134909 non-null  int64
 37  CRASH_DAY_OF_WEEK              1134909 non-null  int64
 38  CRASH_MONTH                    1134909 non-null  int64
 39  LATITUDE                       1134909 non-null  float64
 40  LONGITUDE                      1134909 non-null  float64
 41  PERSON_ID                      1134909 non-null  object
 42  PERSON_TYPE                    1134909 non-null  object
 43  CRASH_DATE                     1134909 non-null  object
 44  SEX                            1134909 non-null  object
 45  SAFETY_EQUIPMENT               1134909 non-null  object
 46  AIRBAG_DEPLOYED                1134909 non-null  object
 47  EJECTION                       1134909 non-null  object
 48  INJURY_CLASSIFICATION          1134909 non-null  object
dtypes: float64(11), int64(7), object(31)
memory usage: 424.3+ MB
```

In [7]:

```
df1 = df[df['INJURIES_FATAL'] > 0]
```

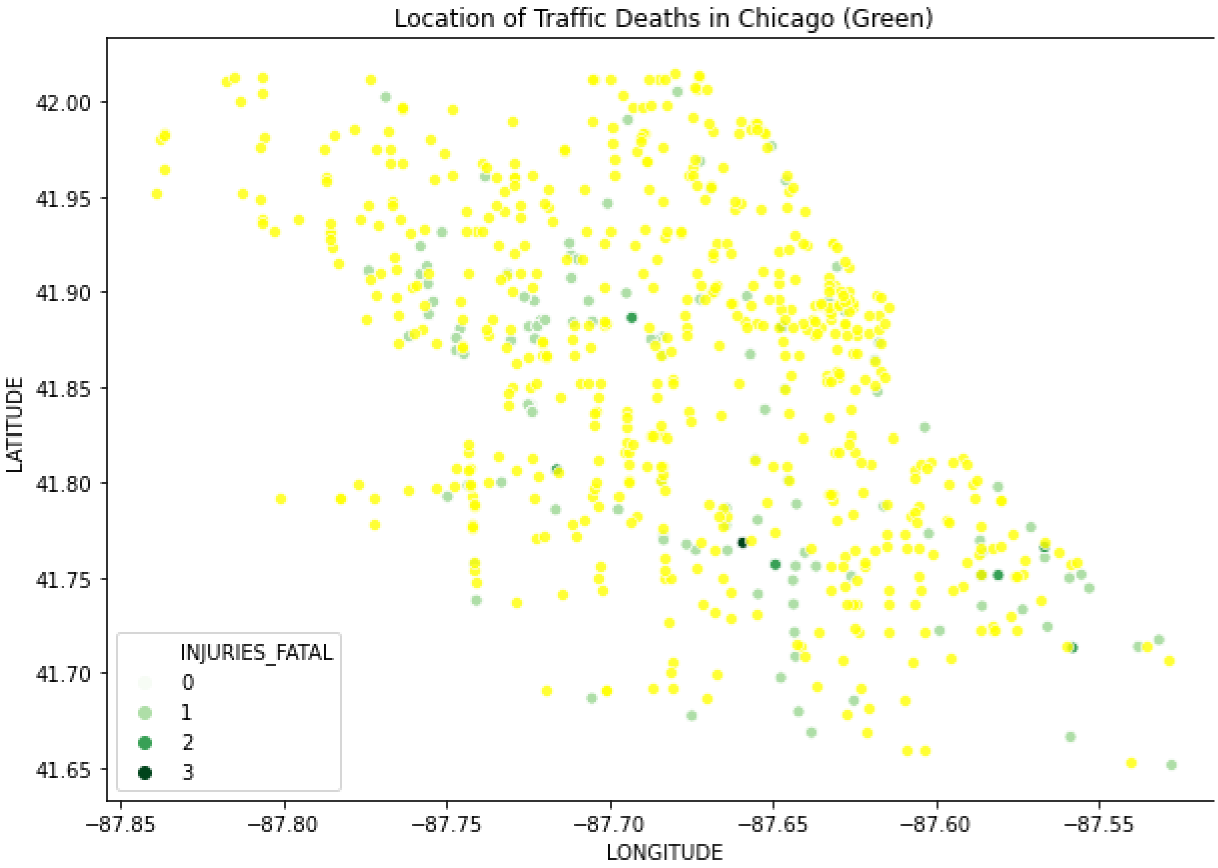
In [8]:

```
plt.figure(figsize=(10,7))
sns.scatterplot(x=df['LONGITUDE'],y=df['LATITUDE'],hue=df['INJURIES_FATAL'])
plt.legend(loc='lower left')
plt.title('Location of Traffic Deaths in Chicago')
plt.show()
```



In [11]:

```
plt.figure(figsize=(10,7))
sns.scatterplot(x=df1['LONGITUDE'],y=df1['LATITUDE'],hue=df['INJURIES_FATAL']
sns.scatterplot(x=df2['LONGITUDE'],y=df2['LATITUDE'],color='Yellow',legend='b
plt.legend(loc='lower left')
plt.title('Location of Traffic Deaths in Chicago (Green)')
plt.show()
```



Question 1 Insights

We can see that there is no discernible pattern to location. In the Graph abo
of Lake Michigan along the upper righthand side, with fatal accidents prese
around the Downtown Chicago Area.

In [ ]:

In [ ]:

In [8]:

```
# INJURIES_FATAL lists total fatalities in the incident
# df['MOST_SEVERE_INJURY'].unique()
# df.INJURIES_FATAL[df['INJURIES_FATAL']>1] = 1
# df.INJURIES_FATAL[df['INJURIES_FATAL']==0] = 0
df.INJURIES_FATAL.sum()
#
```

```
140.0
```

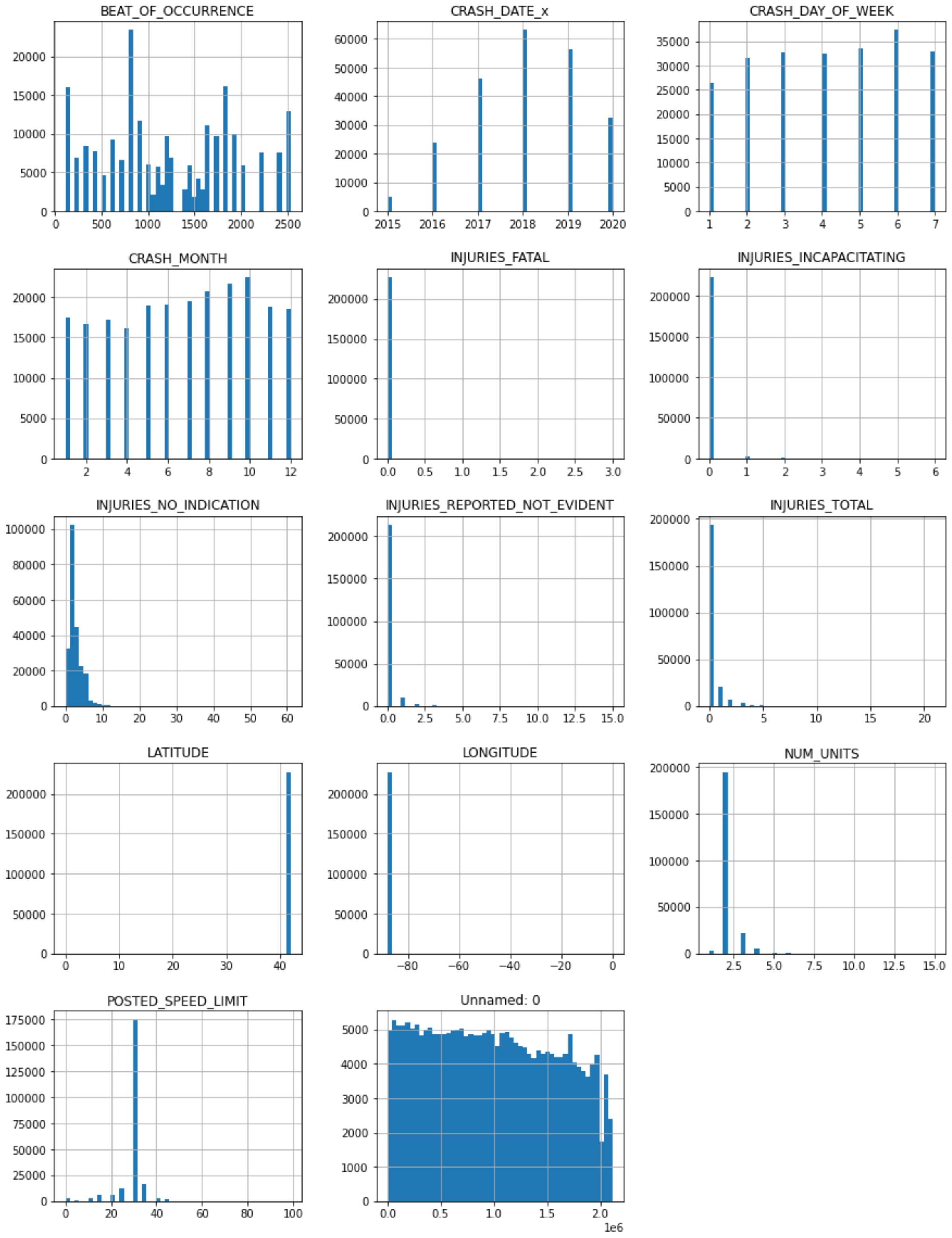
In [117]:

```
df['CRASH_DAY_OF_WEEK'].unique() # Sunday = 1
```

```
array([5, 2, 7, 4, 1, 6, 3], dtype=int64)
```

In [131]:

```
df.hist(figsize=(20,20),bins=50)
plt.show()
# quick observations - more likely to get in an accident on a Friday.
# after or around 3 PM to 5 PM (rush hour)
# October is most likely month in which to have an accident
# Speed limit in the Chicago city area is generally 35 MPH
# most accidents involve 1 person only.
```



Train Test Split and OneHotEncode

In [122]:

```
# create a map
# vehicle_defect_pairs = []

# for ix, row in enumerate(df.select("VEHICLE_DEFECT").distinct().collect())
#   pair = (ix, row.VEHICLE_DEFECT)
#   vehicle_defect_pairs.append(pair)
# vehicle_defect_pairs
```

In [123]:

```
# feature_list = []

# for col in df.columns:
#   if col in ("_c0", "CRASH_RECORD_ID", "RD_NO_x", "CRASH_DATE_x", "VEHICLE_
#     continue
#   else:
#     feature_list.append(col)

# assembler = VectorAssembler(inputCols=feature_list, outputCol="features")
```

In [100]:

```
# # # Remove "object"-type features from df
# cont_features = [col for col in df.columns if df[col].dtype in [np.float64,

# # # Remove "object"-type features from df
# df_cont = df.loc[:, cont_features]
```

In [101]:

```
# # Create df_cat which contains only the categorical variables
# features_cat = [col for col in df.columns if df[col].dtype in [np.object]]
# other_ind = []
# for col in features_cat:
#     others = list(df[df[col].str.contains("OTHER")].index)
#     for oth in others:
#         if oth in other_ind:
#             continue
#         else: other_ind.append(oth)

# df.drop(other_ind, inplace=True)
# df_cat = df.loc[:, features_cat]
# df_target = df.loc[:, ['INJURIES_FATAL']]
```

In [102]:

```
# df_target['INJURIES_FATAL'] = df_target['INJURIES_FATAL'].astype('category

# df = df.drop("INJURIES_FATAL", axis=1)
```

In [103]:

```
# X = df[['MAKE', 'MODEL', 'VEHICLE_DEFECT', 'VEHICLE_TYPE', 'OCCUPANT_CNT',
# X = df.drop(columns='INJURIES_FATAL')
# target = df['INJURIES_FATAL']
```

In [124]:

```
# create a map
# vehicle_defect_pairs = []

# for ix, row in enumerate(df.select("VEHICLE_DEFECT").distinct().collect())
#   pair = (ix, row.VEHICLE_DEFECT)
#   vehicle_defect_pairs.append(pair)
# vehicle_defect_pairs
```

In [27]:

In [ ]:

In [ ]: