In [1]:

```
import numpy as np
import pandas as pd # pd is the universally-used abbreviation
```

In [2]:

```
raw_data = pd.read_csv('Motor_Vehicle_Collisions_-_Crashes.csv')
raw_data.head(10)
```

```
C:\Users\13475\anaconda3\lib\site-packages\IPython\core\interactiveshell.py:3
071: DtypeWarning: Columns (3) have mixed types.Specify dtype option on impor
t or set low_memory=False.
  has_raised = await self.run_ast_nodes(code_ast.body, cell_name,
```

Out[2]:

| | CRASH DATE | CRASH TIME | BOROUGH | ZIP CODE | LATITUDE | LONGITUDE | LOCATION | ON STREET NAME |
|---|---|---|---|---|---|---|---|---|
| **0** | 02/06/2020 | 9:59 | NaN | NaN | 40.772020 | -73.956024 | (40.77202, -73.956024) | EAST 77 STREET |
| **1** | 01/15/2020 | 19:00 | QUEENS | 11368 | 40.751064 | -73.854935 | (40.751064, -73.854935) | NaN |
| **2** | 02/10/2020 | 0:08 | BROOKLYN | 11223 | 40.598312 | -73.961190 | (40.598312, -73.96119) | AVENUE U |
| **3** | 01/18/2020 | 16:30 | NaN | NaN | 40.793198 | -73.824140 | (40.793198, -73.82414) | WHITESTONE EXPRESSWAY |
| **4** | 01/24/2020 | 6:55 | QUEENS | 11372 | 40.750717 | -73.872170 | (40.750717, -73.87217) | NaN |
| **5** | 02/10/2020 | 23:30 | MANHATTAN | 10029 | 40.792920 | -73.945790 | (40.79292, -73.94579) | LEXINGTON AVENUE |
| **6** | 01/13/2020 | 11:25 | MANHATTAN | 10003 | 40.731964 | -73.988160 | (40.731964, -73.98816) | EAST 12 STREET |
| **7** | 01/13/2020 | 21:30 | NaN | NaN | NaN | NaN | NaN | EAST 57 STREET |
| **8** | 02/12/2020 | 0:19 | NaN | NaN | 40.751940 | -73.832306 | (40.75194, -73.832306) | COLLEGE POINT BOULEVARD |
| **9** | 01/17/2020 | 16:00 | NaN | NaN | 40.667477 | -73.956230 | (40.667477, -73.95623) | BEDFORD AVENUE |

10 rows × 29 columns

In [3]:

```
# Remove the data not nessesary for this excercise

data1 = raw_data.drop(columns = ["LATITUDE", "LONGITUDE","LOCATION", "OFF STREET NAME", "NUMBER OF PEDESTRIANS INJURED", "NUMBER OF PEDESTRIANS KILLED", "NUMBER OF CYCLIST INJURED", "NUMBER OF CYCLIST KILLED", "NUMBER OF MOTORIST INJURED", "NUMBER OF MOTORIST KILLED", "CONTRIBUTING FACTOR VEHICLE 3", "CONTRIBUTING FACTOR VEHICLE 4", "CONTRIBUTING FACTOR VEHICLE 5", "VEHICLE TYPE CODE 3", "VEHICLE TYPE CODE 4", "VEHICLE TYPE CODE 5"])
```

In [4]:

```
# Drop all null values in borough and on street name

data1.dropna(axis=0, subset=['BOROUGH', 'ON STREET NAME'], inplace=True)
data1.head(10)
```

Out[4]:

| | CRASH DATE | CRASH TIME | BOROUGH | ZIP CODE | ON STREET NAME | CROSS STREET NAME | NUMBER OF PERSONS INJURED | NUMBER OF PERSONS KILLED |
|---|---|---|---|---|---|---|---|---|
| **2** | 02/10/2020 | 0:08 | BROOKLYN | 11223 | AVENUE U | CONEY ISLAND AVENUE | 3.0 | 0.0 |
| **5** | 02/10/2020 | 23:30 | MANHATTAN | 10029 | LEXINGTON AVENUE | EAST 107 STREET | 0.0 | 0.0 |
| **6** | 01/13/2020 | 11:25 | MANHATTAN | 10003 | EAST 12 STREET | 3 AVENUE | 0.0 | 0.0 |
| **11** | 02/03/2020 | 11:53 | BRONX | 10474 | SPOFFORD AVENUE | CASANOVA STREET | 3.0 | 0.0 |
| **13** | 01/18/2020 | 13:10 | BROOKLYN | 11209 | 79 STREET | 4 AVENUE | 0.0 | 0.0 |
| **15** | 01/19/2020 | 6:30 | BROOKLYN | 11230 | AVENUE I | OCEAN AVENUE | 0.0 | 0.0 |
| **16** | 01/20/2020 | 17:00 | QUEENS | 11416 | WOODHAVEN BOULEVARD | ATLANTIC AVENUE | 0.0 | 0.0 |
| **17** | 01/17/2020 | 20:20 | QUEENS | 11362 | 233 STREET | WEST ALLEY ROAD | 1.0 | 0.0 |
| **18** | 01/11/2020 | 16:25 | BROOKLYN | 11232 | 35 STREET | 4 AVENUE | 0.0 | 0.0 |
| **20** | 01/23/2020 | 9:10 | BROOKLYN | 11214 | BATH AVENUE | BAY PARKWAY | 0.0 | 0.0 |

In [5]:

```
# create a data frame that shows all the crashes in manhattan

filter1 = data1['BOROUGH'] == 'MANHATTAN'
manhattan = data1[filter1]
manhattan.head(10)
```

Out[5]:

| | CRASH DATE | CRASH TIME | BOROUGH | ZIP CODE | ON STREET NAME | CROSS STREET NAME | NUMBER OF PERSONS INJURED | NUMBER OF PERSONS KILLED | F |
|---|---|---|---|---|---|---|---|---|---|
| **5** | 02/10/2020 | 23:30 | MANHATTAN | 10029 | LEXINGTON AVENUE | EAST 107 STREET | 0.0 | 0.0 | Ir |
| **6** | 01/13/2020 | 11:25 | MANHATTAN | 10003 | EAST 12 STREET | 3 AVENUE | 0.0 | 0.0 | Ir |
| **27** | 01/30/2020 | 16:25 | MANHATTAN | 10009 | EAST 12 STREET | 1 AVENUE | 1.0 | 0.0 | Ir |
| **29** | 01/31/2020 | 9:32 | MANHATTAN | 10022 | 51 Street | 2 Avenue | 0.0 | 0.0 | Ir |
| **37** | 02/08/2020 | 10:20 | MANHATTAN | 10065 | YORK AVENUE | EAST 63 STREET | 0.0 | 0.0 | |
| **48** | 01/29/2020 | 18:10 | MANHATTAN | 10021 | 1 AVENUE | EAST 69 STREET | 0.0 | 0.0 | Ir |
| **51** | 02/09/2020 | 10:54 | MANHATTAN | 10026 | 5 AVENUE | EAST 116 STREET | 0.0 | 0.0 | |
| **61** | 01/29/2020 | 10:01 | MANHATTAN | 10023 | WEST 67 STREET | BROADWAY | 1.0 | 0.0 | Ir |
| **84** | 02/07/2020 | 17:35 | MANHATTAN | 10065 | YORK AVENUE | EAST 63 STREET | 0.0 | 0.0 | |
| **90** | 01/16/2020 | 6:10 | MANHATTAN | 10029 | EAST 99 STREET | PARK AVENUE | 0.0 | 0.0 | |

In [6]:

```
# number of people injured in manhattan

manhattan['NUMBER OF PERSONS INJURED'].sum()
```

Out[6]:

```
43848.0
```

In [7]:

```
# number of people dead in manhattan

manhattan['NUMBER OF PERSONS KILLED'].sum()
```

Out[7]:

```
241.0
```

In [8]:

```
# groups = manhattan.groupby('ON STREET NAME')
# # for street, group in groups:
# #     print("===============================================================
=============================== ")
# #     print(street + ":")
# #     print(group)
# df = pd.DataFrame(groups)
```

In [9]:

```
# 2. Split the data frame according to the zip code

groups1 = manhattan.groupby('ZIP CODE')
#for zip_code, group in groups1:
    #print("==================================================================
============================ ")
    #print(zip_code)
    #print(group)

# zip_code_df = pd.DataFrame(zip_code)
# zip_code_df['NUMBER OF PERSONS INJURED'] = manhattan['NUMBER OF PERSONS INJU
RED']
# zip_code_df.head(40)
```

In [10]:

```
# counts the number of crashes in each zip code
# returns the number of rows in each group

manhattan_zip_codes = groups1.size()
```

In [11]:

```
# 4. Organize the results as a data frame.

manhattan_zip_codes = manhattan_zip_codes.to_frame(name="Collisions per zip co
de")
manhattan_zip_codes.head()
```

Out[11]:

| | Collisions per zip code |
|---|---|
| **ZIP CODE** | |
| **10000.0** | 69 |
| **10001.0** | 6551 |
| **10002.0** | 5867 |
| **10003.0** | 3584 |
| **10004.0** | 969 |

In [12]:
```
# groups6 = manhattan['ZIP CODE'].groupby(['NUMBER OF PERSONS KILLED'])
# deaths = groups6.mean()
# deaths

# df_manhattan_deaths = deaths.to_frame(name="Number of deaths")
# df_manhattan_deaths.head()
# df_means_exam = df['Exam'].groupby(df['Course']).mean().to_frame(name="Average Exam Score")
```

In [13]:
```
manhattan_zip_codes
```

Out[13]:

| | Collisions per zip code |
|---|---|
| **ZIP CODE** | |
| **10000.0** | 69 |
| **10001.0** | 6551 |
| **10002.0** | 5867 |
| **10003.0** | 3584 |
| **10004.0** | 969 |
| **...** | ... |
| **10271** | 1 |
| **10278** | 3 |
| **10280** | 72 |
| **10281** | 18 |
| **10282** | 32 |

112 rows × 1 columns

In [14]:
```
manhattan_zip_codes.count()
```

Out[14]:
```
Collisions per zip code    112
dtype: int64
```

In [15]:
```
manhattan_zip_codes['Collisions per zip code'].sum()
```

Out[15]:
```
236981
```

In [16]:

```
# which zip codes have the most crashes
manhattan_zip_sorted = manhattan_zip_codes.sort_values(by="Collisions per zip
 code", ascending=False)
manhattan_zip_sorted.head(19)#33
```

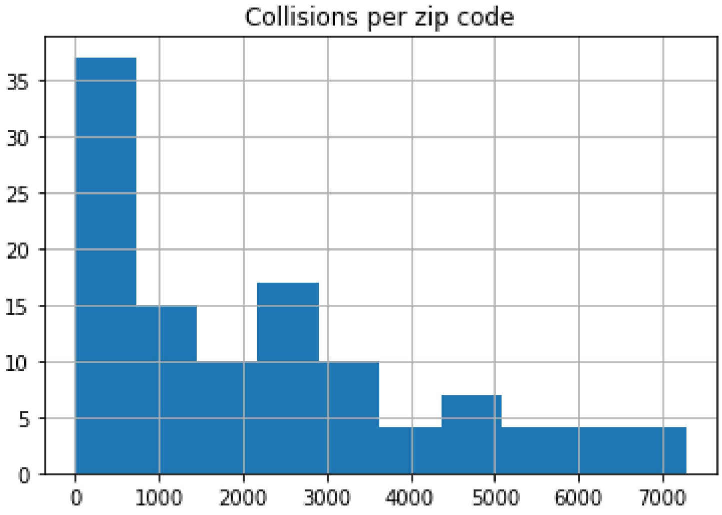
Out[16]:

| | Collisions per zip code |
|---|---|
| **ZIP CODE** | |
| **10016.0** | 7273 |
| **10022.0** | 7221 |
| **10019.0** | 7173 |
| **10001.0** | 6551 |
| **10036.0** | 6473 |
| **10013.0** | 6380 |
| **10022** | 5872 |
| **10002.0** | 5867 |
| **10016** | 5791 |
| **10019** | 5676 |
| **10036** | 5352 |
| **10001** | 5190 |
| **10018.0** | 4936 |
| **10011.0** | 4906 |
| **10065.0** | 4823 |
| **10013** | 4753 |
| **10029.0** | 4431 |
| **10002** | 4405 |
| **10017.0** | 4384 |

In [17]:
```
# The majority of zip codes in manhattan have few crashes and a few zip codes
 shave lots of crashes

manhattan_zip_codes.hist()
```

Out[17]:
```
array([[<AxesSubplot:title={'center':'Collisions per zip code'}>]],
      dtype=object)
```



In [18]:
```
filter2 = data1['BOROUGH'] == 'BRONX'
bronx = data1[filter2]
bronx.head(10)
```

Out[18]:

| | CRASH DATE | CRASH TIME | BOROUGH | ZIP CODE | ON STREET NAME | CROSS STREET NAME | NUMBER OF PERSONS INJURED | NUMBER OF PERSONS KILLED |
|---|---|---|---|---|---|---|---|---|
| **11** | 02/03/2020 | 11:53 | BRONX | 10474 | SPOFFORD AVENUE | CASANOVA STREET | 3.0 | 0.0 |
| **22** | 01/18/2020 | 13:00 | BRONX | 10453 | MAJOR DEEGAN EXPRESSWAY | WEST FORDHAM ROAD | 0.0 | 0.0 |
| **25** | 01/20/2020 | 17:45 | BRONX | 10466 | EAST 233 STREET | BRONX BOULEVARD | 1.0 | 0.0 |
| **33** | 02/04/2020 | 14:07 | BRONX | 10472 | WATSON AVENUE | MORRISON AVENUE | 1.0 | 0.0 |
| **36** | 01/21/2020 | 0:00 | BRONX | 10455 | EAST 149 STREET | BRUCKNER BOULEVARD | 0.0 | 0.0 |
| **40** | 01/19/2020 | 12:00 | BRONX | 10456 | SHERIDAN AVENUE | EAST 168 STREET | 0.0 | 0.0 |
| **60** | 01/31/2020 | 17:45 | BRONX | 10467 | BURKE AVENUE | WHITE PLAINS ROAD | 0.0 | 0.0 |
| **72** | 02/02/2020 | 14:34 | BRONX | 10458 | EAST 183 STREET | CROTONA AVENUE | 0.0 | 0.0 |
| **88** | 01/15/2020 | 5:00 | BRONX | 10451 | COURTLANDT AVENUE | EAST 148 STREET | 0.0 | 0.0 |
| **94** | 01/26/2020 | 17:23 | BRONX | 10463 | WEST 230 STREET | KINGSBRIDGE AVENUE | 0.0 | 0.0 |

In [19]:
```
# number of people injured in manhattan

bronx['NUMBER OF PERSONS INJURED'].sum()
```

Out[19]:
```
42908.0
```

In [20]:
```
bronx['NUMBER OF PERSONS KILLED'].sum()
```

Out[20]:
```
173.0
```

In [21]:

```
# 2. Split the data frame according to the zip code

groups2 = bronx.groupby('ZIP CODE')
#for zip_code, group in groups2:
    #print("==================================================================
============================ ")
    #print(zip_code)
    #print(group)
```

In [22]:

```
# counts the number of crashes in each zip code
# returns the number of rows in each group

bronx_zip_codes = groups2.size()
```

In [23]:

```
# 4. Organize the results as a data frame.

bronx_zip_codes = bronx_zip_codes.to_frame(name="Collisions per zip code")
bronx_zip_codes.head()
```

Out[23]:

| | Collisions per zip code |
|---|---|
| **ZIP CODE** | |
| **10451.0** | 4928 |
| **10452.0** | 3838 |
| **10453.0** | 4639 |
| **10454.0** | 3963 |
| **10455.0** | 3417 |

In [24]:

```
bronx_zip_codes.count()
```

Out[24]:

```
Collisions per zip code    51
dtype: int64
```

In [25]:

```
bronx_zip_codes['Collisions per zip code'].sum()
```

Out[25]:

```
133837
```

In [26]:

```
# which zip codes have the most crashes
bronx_zip_sorted = bronx_zip_codes.sort_values(by="Collisions per zip code", ascending=False)
bronx_zip_sorted.head(8)
```

Out[26]:

| ZIP CODE | Collisions per zip code |
|---|---|
| **10467.0** | 5467 |
| **10458.0** | 5051 |
| **10451.0** | 4928 |
| **10457.0** | 4817 |
| **10453.0** | 4639 |
| **10466.0** | 4523 |
| **10460.0** | 4187 |
| **10468.0** | 4000 |

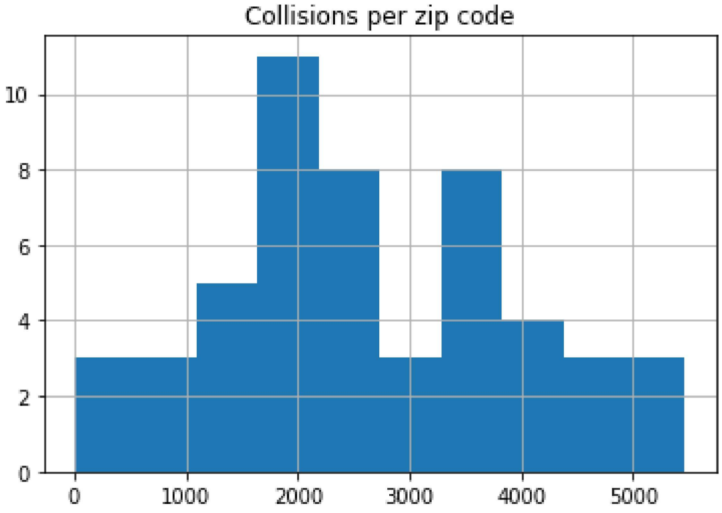
In [27]:

```
# greater number of zip codes with high number of crashes

bronx_zip_codes.hist()
```

Out[27]:

```
array([[<AxesSubplot:title={'center':'Collisions per zip code'}>]],
      dtype=object)
```



In [28]:
```
filter3 = data1['BOROUGH'] == 'BROOKLYN'
brooklyn = data1[filter3]
brooklyn.head(10)
```

Out[28]:

| | CRASH DATE | CRASH TIME | BOROUGH | ZIP CODE | ON STREET NAME | CROSS STREET NAME | NUMBER OF PERSONS INJURED | NUMBER OF PERSONS KILLED |
|---|---|---|---|---|---|---|---|---|
| **2** | 02/10/2020 | 0:08 | BROOKLYN | 11223 | AVENUE U | CONEY ISLAND AVENUE | 3.0 | 0.0 |
| **13** | 01/18/2020 | 13:10 | BROOKLYN | 11209 | 79 STREET | 4 AVENUE | 0.0 | 0.0 |
| **15** | 01/19/2020 | 6:30 | BROOKLYN | 11230 | AVENUE I | OCEAN AVENUE | 0.0 | 0.0 |
| **18** | 01/11/2020 | 16:25 | BROOKLYN | 11232 | 35 STREET | 4 AVENUE | 0.0 | 0.0 |
| **20** | 01/23/2020 | 9:10 | BROOKLYN | 11214 | BATH AVENUE | BAY PARKWAY | 0.0 | 0.0 |
| **21** | 01/30/2020 | 9:05 | BROOKLYN | 11221 | WEIRFIELD STREET | EVERGREEN AVENUE | 0.0 | 0.0 |
| **23** | 01/15/2020 | 9:45 | BROOKLYN | 11206 | HARRISON AVENUE | HEYWARD STREET | 2.0 | 0.0 |
| **28** | 01/14/2020 | 18:00 | BROOKLYN | 11223 | AVENUE W | STRYKER STREET | 1.0 | 0.0 |
| **43** | 02/07/2020 | 11:30 | BROOKLYN | 11201 | FLATBUSH AVENUE EXTENSION | WILLOUGHBY STREET | 1.0 | 0.0 |
| **44** | 01/20/2020 | 19:25 | BROOKLYN | 11206 | JOHNSON AVENUE | BUSHWICK PLACE | 1.0 | 0.0 |

In [29]:
```
# number of people injured in manhattan

brooklyn['NUMBER OF PERSONS INJURED'].sum()
```

Out[29]:
```
97885.0
```

In [30]:

```
# number of people injured in brooklyn

brooklyn['NUMBER OF PERSONS KILLED'].sum()
```

Out[30]:

```
419.0
```

In [31]:

```
# 2. Split the data frame according to the zip code

groups3 = brooklyn.groupby('ZIP CODE')
#for zip_code, group in groups3:
    #print("==================================================================
============================ ")
    #print(zip_code)
    #print(group)
```

In [32]:

```
# counts the number of crashes in each zip code
# returns the number of rows in each group

brooklyn_zip_codes = groups3.size()
```

In [33]:

```
# 4. Organize the results as a data frame.

brooklyn_zip_codes = brooklyn_zip_codes.to_frame(name="Collisions per zip cod
e")
brooklyn_zip_codes.head()
```

Out[33]:

| | Collisions per zip code |
|---|---|
| **ZIP CODE** | |
| **11201.0** | 8445 |
| **11203.0** | 7874 |
| **11204.0** | 4447 |
| **11205.0** | 2810 |
| **11206.0** | 4983 |

In [34]:

```
brooklyn_zip_codes.count()
```

Out[34]:

```
Collisions per zip code    81
dtype: int64
```

In [35]:

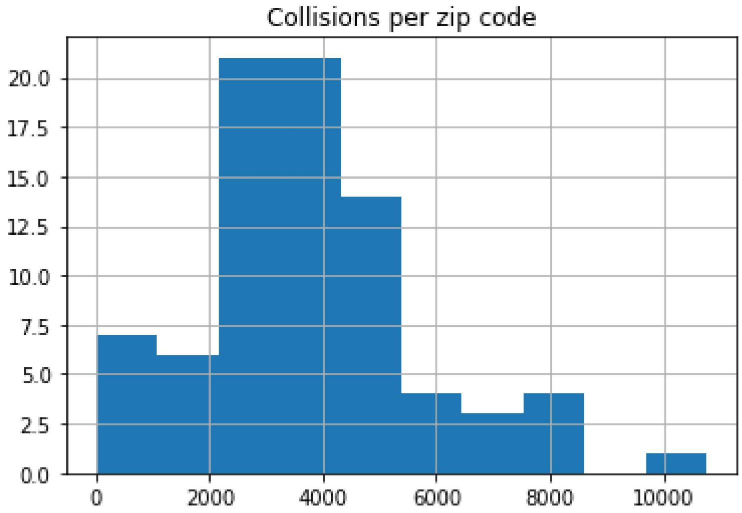
```
# which zip codes have the most crashes
brooklyn_zip_sorted = brooklyn_zip_codes.sort_values(by="Collisions per zip code", ascending=False)
brooklyn_zip_sorted.head(30)
```

Out[35]:

| | Collisions per zip code |
|---|---|
| **ZIP CODE** | |
| **11207.0** | 10762 |
| **11201.0** | 8445 |
| **11203.0** | 7874 |
| **11236.0** | 7639 |
| **11234.0** | 7550 |
| **11207** | 6981 |
| **11226.0** | 6815 |
| **11212.0** | 6804 |
| **11208.0** | 6266 |
| **11220.0** | 6216 |
| **11233.0** | 6164 |
| **11230.0** | 5929 |
| **11201** | 5260 |
| **11211.0** | 5110 |
| **11213.0** | 5085 |
| **11206.0** | 4983 |
| **11203** | 4967 |
| **11210.0** | 4905 |
| **11219.0** | 4592 |
| **11218.0** | 4575 |
| **11217.0** | 4545 |
| **11234** | 4540 |
| **11204.0** | 4447 |
| **11235.0** | 4424 |
| **11223.0** | 4418 |
| **11226** | 4361 |
| **11214.0** | 4298 |
| **11236** | 4287 |
| **11220** | 4085 |
| **11216.0** | 4031 |

```
In [36]: brooklyn_zip_codes.hist()
```

```
Out[36]: array([[<AxesSubplot:title={'center':'Collisions per zip code'}>]],
               dtype=object)
```



In [37]:

```
filter4 = data1['BOROUGH'] == 'QUEENS'
queens = data1[filter4]
queens
```

Out[37]:

| | CRASH DATE | CRASH TIME | BOROUGH | ZIP CODE | ON STREET NAME | CROSS STREET NAME | NUMBER OF PERSONS INJURED | NUM PERS KIL |
|---|---|---|---|---|---|---|---|---|
| **16** | 01/20/2020 | 17:00 | QUEENS | 11416 | WOODHAVEN BOULEVARD | ATLANTIC AVENUE | 0.0 | |
| **17** | 01/17/2020 | 20:20 | QUEENS | 11362 | 233 STREET | WEST ALLEY ROAD | 1.0 | |
| **26** | 02/10/2020 | 11:30 | QUEENS | 11101 | LONG ISLAND EXPRESSWAY | GREENPOINT AVENUE | 1.0 | |
| **35** | 02/04/2020 | 20:00 | QUEENS | 11411 | 207 STREET | MURDOCK AVENUE | 1.0 | |
| **47** | 01/18/2020 | 17:00 | QUEENS | NaN | 245 STREET | JAMAICA AVENUE | 0.0 | |
| **...** | ... | ... | ... | ... | ... | ... | ... | |
| **1734255** | 07/03/2012 | 22:25 | QUEENS | 11368 | 99 STREET | 38 AVENUE | 0.0 | |
| **1734258** | 07/05/2012 | 8:00 | QUEENS | 11103 | 30 AVENUE | 42 STREET | 0.0 | |
| **1734264** | 07/04/2012 | 4:17 | QUEENS | 11420 | SOUTH CONDUIT AVENUE | 130 STREET | 0.0 | |
| **1734269** | 07/03/2012 | 20:35 | QUEENS | 11413 | SOUTH CONDUIT AVENUE | 224 STREET | 0.0 | |
| **1734270** | 07/02/2012 | 17:15 | QUEENS | 11365 | FRANCIS LEWIS BOULEVARD | 48 AVENUE | 0.0 | |

263823 rows × 13 columns

In [38]:

```
# number of people injured in queens

queens['NUMBER OF PERSONS INJURED'].sum()
```

Out[38]:

```
77344.0
```

In [39]:
```
# number of people injured in queens

queens['NUMBER OF PERSONS KILLED'].sum()
```

Out[39]:
```
385.0
```

In [40]:
```
# 2. Split the data frame according to the zip code

groups4 = queens.groupby('ZIP CODE')
#for zip_code, group in groups4:
    #print("=================================================================
============================ ")
    #print(zip_code)
    #print(group)
```

In [41]:
```
# counts the number of crashes in each zip code
# returns the number of rows in each group

queens_zip_codes = groups4.size()
```

In [42]:
```
# 4. Organize the results as a data frame.

queens_zip_codes = queens_zip_codes.to_frame(name="Collisions per zip code")
queens_zip_codes.head()
```

Out[42]:

| | Collisions per zip code |
|---|---|
| **ZIP CODE** | |
| **11001.0** | 269 |
| **11004.0** | 1249 |
| **11005.0** | 12 |
| **11040.0** | 157 |
| **11101.0** | 9117 |

In [43]:
```
queens_zip_codes.count()
```

Out[43]:
```
Collisions per zip code    135
dtype: int64
```

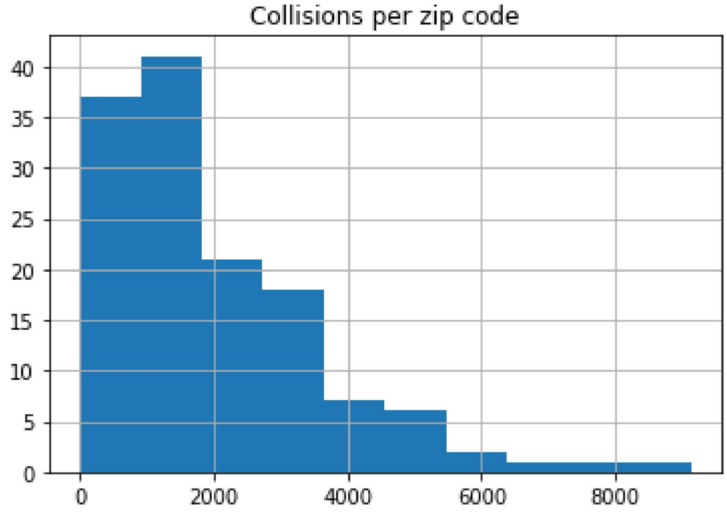
In [44]:
```
# which zip codes have the most crashes
queens_zip_sorted = queens_zip_codes.sort_values(by="Collisions per zip code",
ascending=False)
queens_zip_sorted.head(16)
```

Out[44]:

| ZIP CODE | Collisions per zip code |
|---|---|
| 11101.0 | 9117 |
| 11434.0 | 7940 |
| 11385.0 | 6834 |
| 11101 | 5922 |
| 11377.0 | 5911 |
| 11354.0 | 5452 |
| 11420.0 | 5249 |
| 11355.0 | 5000 |
| 11373.0 | 4762 |
| 11375.0 | 4761 |
| 11413.0 | 4717 |
| 11368.0 | 4498 |
| 11385 | 4437 |
| 11432.0 | 4175 |
| 11422.0 | 4053 |
| 11434 | 4037 |

In [45]:
```
queens_zip_codes.hist()
```

Out[45]:
```
array([[<AxesSubplot:title={'center':'Collisions per zip code'}>]],
      dtype=object)
```



In [46]:
```
filter5 = data1['BOROUGH'] == 'STATEN ISLAND'
staten_island = data1[filter5]
staten_island.head(10)
```

Out[46]:

| | CRASH DATE | CRASH TIME | BOROUGH | ZIP CODE | ON STREET NAME | CROSS STREET NAME | NUMBER OF PERSONS INJURED | NUMBER OF PERSONS KILLED |
|---|---|---|---|---|---|---|---|---|
| **381** | 01/23/2020 | 9:52 | STATEN ISLAND | 10306 | HYLAN BOULEVARD | CANNON BOULEVARD | 0.0 | 0.0 |
| **401** | 02/01/2020 | 23:04 | STATEN ISLAND | 10304 | FLAGG PLACE | FOUR CORNERS ROAD | 3.0 | 0.0 |
| **609** | 02/08/2020 | 9:08 | STATEN ISLAND | 10306 | ROCKLAND AVENUE | BURTON COURT | 1.0 | 0.0 |
| **921** | 02/13/2020 | 8:45 | STATEN ISLAND | 10312 | HUGUENOT AVENUE | SHORT PLACE | 0.0 | 0.0 |
| **970** | 01/23/2020 | 23:33 | STATEN ISLAND | 10302 | JEWETT AVENUE | WYGANT PLACE | 1.0 | 0.0 |
| **1164** | 02/02/2020 | 19:40 | STATEN ISLAND | 10306 | TYSENS LANE | AMBOY ROAD | 1.0 | 0.0 |
| **1175** | 01/25/2020 | 10:00 | STATEN ISLAND | 10304 | RICHMOND ROAD | NARROWS ROAD NORTH | 0.0 | 0.0 |
| **1277** | 01/14/2020 | 7:30 | STATEN ISLAND | 10306 | EDINBORO ROAD | RIGBY AVENUE | 0.0 | 0.0 |
| **1286** | 02/07/2020 | 4:30 | STATEN ISLAND | 10305 | SAINT JOHNS AVENUE | BEETHOVEN STREET | 0.0 | 0.0 |
| **1456** | 01/14/2020 | 22:20 | STATEN ISLAND | 10306 | HYLAN BOULEVARD | FAIRBANKS AVENUE | 2.0 | 0.0 |

In [47]:
```
# number of people injured in queens

staten_island['NUMBER OF PERSONS INJURED'].sum()
```

Out[47]:
```
12030.0
```

In [48]:
```
# number of people injured in brooklyn

staten_island['NUMBER OF PERSONS KILLED'].sum()
```

Out[48]:
```
70.0
```

In [49]:

```
# 2. Split the data frame according to the zip code

groups5 = staten_island.groupby('ZIP CODE')
#for zip_code, group in groups5:
    #print("==================================================================
============================ ")
    #print(zip_code)
    #print(group)
```

In [50]:

```
# counts the number of crashes in each zip code
# returns the number of rows in each group

staten_island_zip_codes = groups5.size()
```

In [51]:

```
# 4. Organize the results as a data frame.

staten_island_zip_codes = staten_island_zip_codes.to_frame(name="Collisions pe
r zip code")
staten_island_zip_codes.head()
```

Out[51]:

| | Collisions per zip code |
|---|---|
| **ZIP CODE** | |
| **10301.0** | 2928 |
| **10302.0** | 171 |
| **10303.0** | 106 |
| **10304.0** | 3447 |
| **10305.0** | 3033 |

In [52]:

```
staten_island_zip_codes.count()
```

Out[52]:

```
Collisions per zip code    24
dtype: int64
```

In [53]:

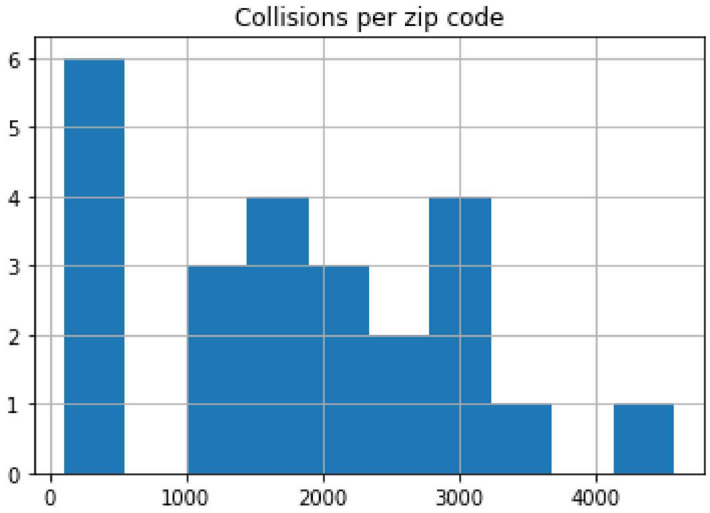
```
# which zip codes have the most crashes
staten_island_zip_codes.sort_values(by="Collisions per zip code", ascending=False)
```

Out[53]:

| | Collisions per zip code |
|---|---|
| **ZIP CODE** | |
| **10306.0** | 4573 |
| **10304.0** | 3447 |
| **10306** | 3190 |
| **10312.0** | 3106 |
| **10305.0** | 3033 |
| **10301.0** | 2928 |
| **10312** | 2436 |
| **10304** | 2397 |
| **10301** | 2140 |
| **10314** | 2113 |
| **10305** | 2100 |
| **10310.0** | 1836 |
| **10314.0** | 1574 |
| **10309.0** | 1555 |
| **10308.0** | 1484 |
| **10309** | 1316 |
| **10310** | 1248 |
| **10308** | 1015 |
| **10307.0** | 433 |
| **10303** | 397 |
| **10302** | 338 |
| **10307** | 332 |
| **10302.0** | 171 |
| **10303.0** | 106 |

```
In [54]: staten_island_zip_codes.hist()
```

```
Out[54]: array([[<AxesSubplot:title={'center':'Collisions per zip code'}>]],
               dtype=object)
```


Collisions per zip code