# ParkingAnalysis

May 30, 2021

# 1 MELBOURNE PARKING DATA ANALYTICS

## 1.1 1.0 DATA PRE PROCESSING

### 1.1.1 1.1 Imports

In this section we'll import the important packages required

[41]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

parkingDB = pd.read_csv("parking_duration_of_parking_event_vs_street_ID.csv")
```

### 1.1.2 1.2 Column Modifications

**1.2.1 Column Names** Given that the column names are slightly tedious or complicated to read, these names will be simplified.

[42]:
```
#renaming code
parkingDB = parkingDB.rename(columns={'Area Name':'Area', 'Street Name':
 ↪'Street', 'Duration of Parking Event (in seconds)':'Parking Duration (s)',␣
 ↪'Between Street 1':'Street Limit 1','Between Street 2':'Street Limit 2', 'In␣
 ↪Violation?': 'Violation' })

#explanation
print("\u0332".join("Renamed Columns"))
print("Area Name --> Area")
print("Street Name --> Street")
print("Duration of Parking Event (in seconds) --> Parking Duration (s)")
print("Between Street 1 --> Street Limit 1")
print("Between Street 2 --> Street Limit 2")
print("In Violation? --> Violation")
print("\nnote: all other column names remained constant")
```

```
Renamed Columns
Area Name --> Area
Street Name --> Street
```

```
Duration of Parking Event (in seconds) --> Parking Duration (s)
Between Street 1 --> Street Limit 1
Between Street 2 --> Street Limit 2
In Violation? --> Violation

note: all other column names remained constant
```

### 1.2.2 Column Removal

The following types of columns will be deleted:

```
- ID Columns: This is because they have no use in analysis, since no patterns can be inferred
- Columns with >50% of data not available: This is to stop any sort of skewing of data since n
- Redundant Columns: columns that share the same information will most lkely be simplified to
```

[43]:
```
#Comparing 'Street' to 'Street ID'
print("\u0332".join("Checking Street Entries"))
print('Number of Entries in \'Column Street\': ', parkingDB['Street'].unique().
 ↪size)
print('Number of Entries in \'Street ID\':  ', parkingDB['Street ID'].unique().
 ↪size)
print("\nThe number of Street ID's against the Column Street entries are equal.␣
 ↪This means that the street ID will match up with one street name present in␣
 ↪the former column.")
print("\nBoth entries will be kept for the time being since the ID can be used␣
 ↪to determine the street name later on. This is useful as the the Street name␣
 ↪will not have to be transformed to a numerical value.\n")

print("\u0332".join("Removing Columns with <50% data entered:"))
print("Number of columns before clean: ", len(parkingDB.columns))
parkingDB.dropna(axis = 1, thresh = 6, inplace = True)
print("Number of columns after clean: ", len(parkingDB.columns))
```

```
Checking Street Entries
Number of Entries in 'Column Street':  75
Number of Entries in 'Street ID':   75

The number of Street ID's against the Column Street entries are equal. This
means that the street ID will match up with one street name present in the
former column.

Both entries will be kept for the time being since the ID can be used to
determine the street name later on. This is useful as the the Street name will
not have to be transformed to a numerical value.

Removing Columns with <50% data entered:
Number of columns before clean:  13
Number of columns after clean:  13
```

### 1.2.3 UNIQUE IDENTIFIER CHECK

[44]:
```
print("This section will check for unique values in columns of interest, find␣
 ↪outliers and possible mistakes. \n")
print('Unique Entries in \'Area\' ', parkingDB['Area'].unique(), '\n')
print('Unique Entries in \'Street\' ', parkingDB['Street'].unique(), '\n')
print('Unique Entries in \'Side of Street\' ', parkingDB['Side Of Street'].
 ↪unique(), '\n')
print('Unique Entries in \'Device ID\' ', parkingDB['Device ID'].unique().size,␣
 ↪'\n')
print('Unique Entries in \'Sign\' ', parkingDB['Sign'].unique().size, '\n')
print("The values for Device ID indicate that the dataset has multiple events␣
 ↪for the same car, and so using this to determine rates wtih repeating visits␣
 ↪will be useful in the analysis")
```

```
This section will check for unique values in columns of interest, find outliers
and possible mistakes.

Unique Entries in 'Area'  ['Banks' 'Chinatown' 'Courtney' 'Princes Theatre'
'Hyatt' 'County' 'RACV'
 'Spencer' 'City Square' 'The Mac' 'Titles' 'Magistrates' 'Rialto'
 'Queensberry' 'Victoria Market' 'Supreme' 'Hardware' 'Regency'
 'Docklands' 'Tavistock' 'Southbank' 'West Melbourne' 'Jolimont']

Unique Entries in 'Street'  ['MARKET STREET' 'RUSSELL STREET' 'ELIZABETH STREET'
'EXHIBITION STREET'
 'LONSDALE STREET' 'Lt COLLINS STREET' 'BOURKE STREET' 'FLINDERS LANE'
 'SPRING STREET' 'COLLINS STREET' 'Lt LONSDALE STREET' "A'BECKETT STREET"
 'Lt BOURKE STREET' 'KING STREET' 'ERROL STREET' 'FRANKLIN STREET'
 'Lt DRYBURGH STREET SOUTH' 'WILLIAM STREET' 'WILLS STREET'
 'SPENCER STREET' 'FLINDERS STREET' 'THERRY STREET' 'QUEEN STREET'
 'LEVESON STREET' "O'CONNELL STREET" 'CHETWYND STREET' 'WALSH STREET'
 'ANDERSON STREET' 'ROSSLYN STREET' 'LA TROBE STREET' 'BOND STREET'
 'CAPEL STREET' 'COBDEN STREET' 'FRANCIS STREET' 'Lt LA TROBE STREET'
 'QUEENSBERRY STREET' 'EADES PLACE' 'DRYBURGH STREET' 'CHURCH STREET'
 'SWANSTON STREET' 'PEEL STREET' 'DUDLEY STREET' 'CURZON STREET'
 'VICTORIA STREET' 'HOWARD STREET' 'ANTHONY STREET' 'RODEN STREET'
 'MACKENZIE STREET' 'ABBOTSFORD STREET' 'PRINCESS STREET' 'UNION STREET'
 'DODDS STREET' 'GRANT STREET' 'COVENTRY STREET' 'BALSTON STREET'
 'KAVANAGH STREET' 'MILES STREET' 'DORCAS STREET' 'SOUTHBANK BOULEVARD'
 'STURT STREET' 'WELLS STREET' 'FAWKNER STREET' 'BATMAN STREET'
 'LANSDOWNE STREET' 'ALBERT STREET' 'CATHEDRAL PLACE' 'GISBORNE STREET'
 'CLARENDON STREET' 'PARLIAMENT PLACE' 'ST ANDREWS PLACE' 'CITY ROAD'
 'ST KILDA ROAD' 'WELLINGTON PARADE' 'NICHOLSON STREET' 'JEFFCOTT STREET']

Unique Entries in 'Side of Street'  [5 2 4 3 1]

Unique Entries in 'Device ID'  7113
```

```
Unique Entries in 'Sign'  329
```

```
The values for Device ID indicate that the dataset has multiple events for the
same car, and so using this to determine rates wtih repeating visits will be
useful in the analysis
```

There seems to be no strange outliers present in any of the tested attributes. Therefore we do not have to delete rows containing any specific outliers

### 1.1.3 1.2.4 NULL ROWS CHECK

[45]:
```
print("This section will check for rows with less than 50% of columns filled.␣
 ↪Rows that qualify under this definition will be deleted so as to not tamper␣
 ↪with results.\n")
print('Number of Rows Before Row Deletion: ', parkingDB.shape[0])
parkingDB.dropna(axis = 0, thresh = 5, inplace = True)
print('Number of Rows After Row Deletion: ', parkingDB.shape[0])
```

```
This section will check for rows with less than 50% of columns filled. Rows that
qualify under this definition will be deleted so as to not tamper with results.

Number of Rows Before Row Deletion:  12208178
Number of Rows After Row Deletion:  12208178
```

Therefore no rows had more than 50% of data omitted from their entries.

[46]:
```
print("Now that we have removed problematic row entries in the database, we␣
 ↪will perform a null search to see if there are any other null values present.
 ↪")
print("\nNull Values Present in Each Column:")
print(parkingDB.isnull().sum())
print("\nThere are no null values within the database after cleaning columns␣
 ↪with more than 50% of data missing.")
```

```
Now that we have removed problematic row entries in the database, we will
perform a null search to see if there are any other null values present.

Null Values Present in Each Column:
Area                    0
Street                  0
Street Limit 1          0
Street Limit 2          0
Side Of Street          0
Street Marker           0
Arrival Time            0
Departure Time          0
Parking Duration (s)    0
Sign                    0
Violation               0
```

```
Street ID                0
Device ID                0
dtype: int64
```

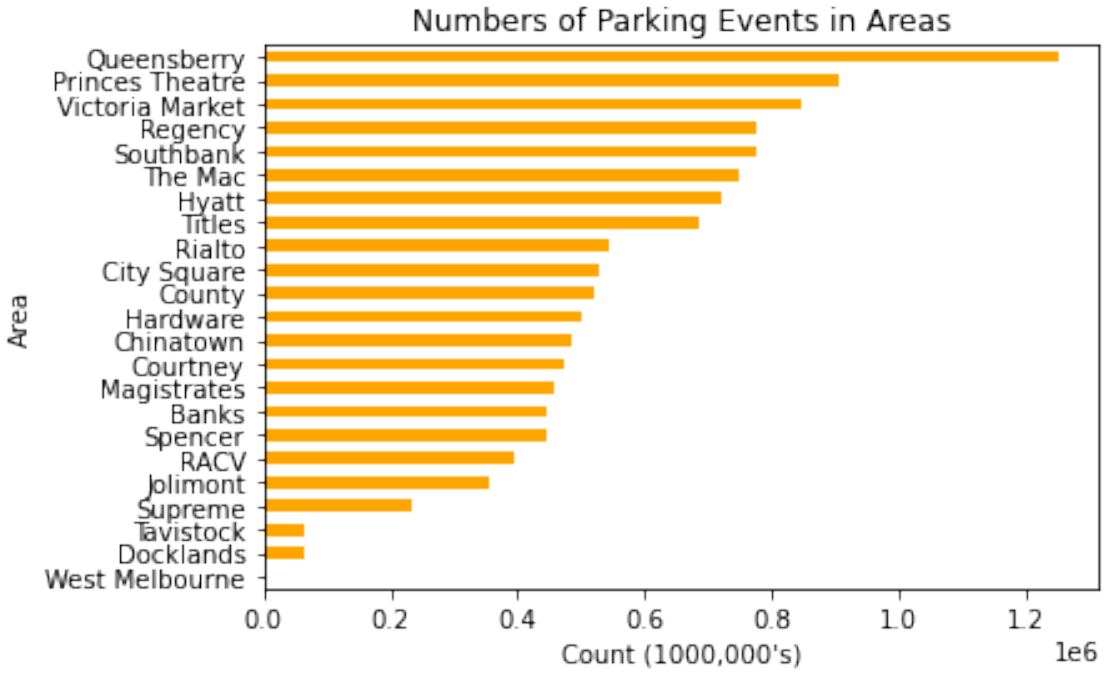
There are no null values within the database after cleaning columns with more than 50% of data missing.

## 1.2 2.0 DATA EXPLORATION

### 1.2.1 2.1 EXPLORING AREAS

[7]:
```
parkingDB['Area'].value_counts().sort_values().plot.barh(color="orange")
plt.title('Numbers of Parking Events in Areas')
plt.ylabel('Area')
plt.xlabel("Count (1000,000's)")
```

[7]:
```
Text(0.5, 0, "Count (1000,000's)")
```

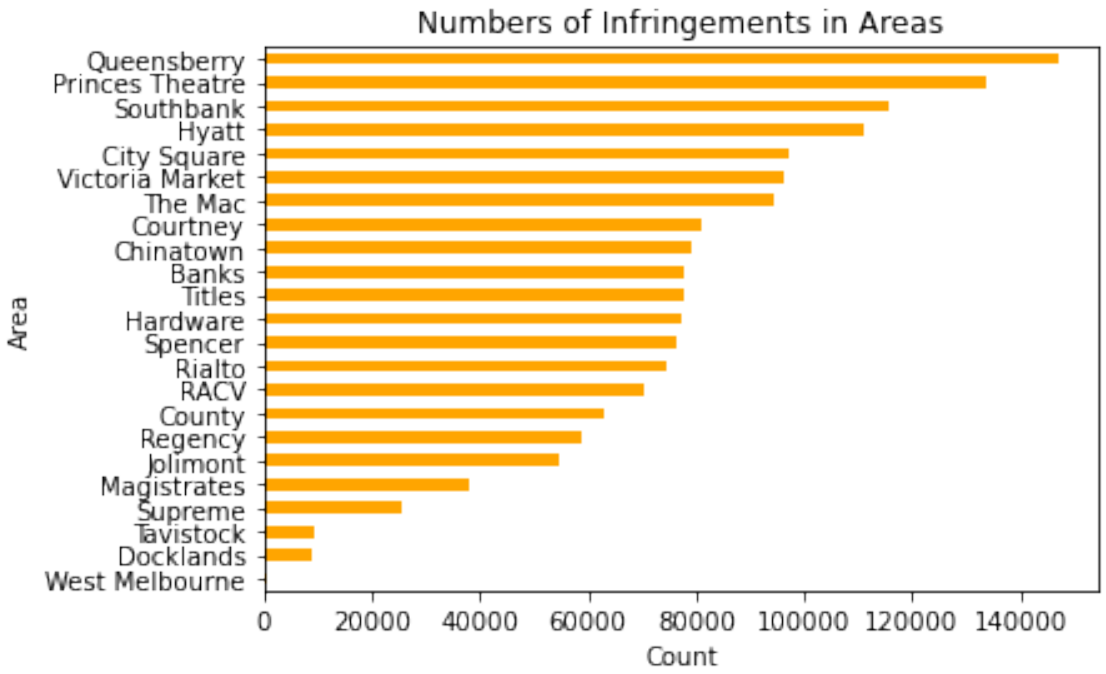



It's found that Queensberry street has the most number of parking events by a far margin in comparison with many of the other sites.

[9]:
```
mask_violation = parkingDB['Violation'] == 1
parkingDB.loc[mask_violation, 'Area'].value_counts().sort_values().plot.
 ↪barh(color="orange")
plt.title('Numbers of Infringements in Areas')
plt.ylabel('Area')
```

```
plt.xlabel("Count")
```

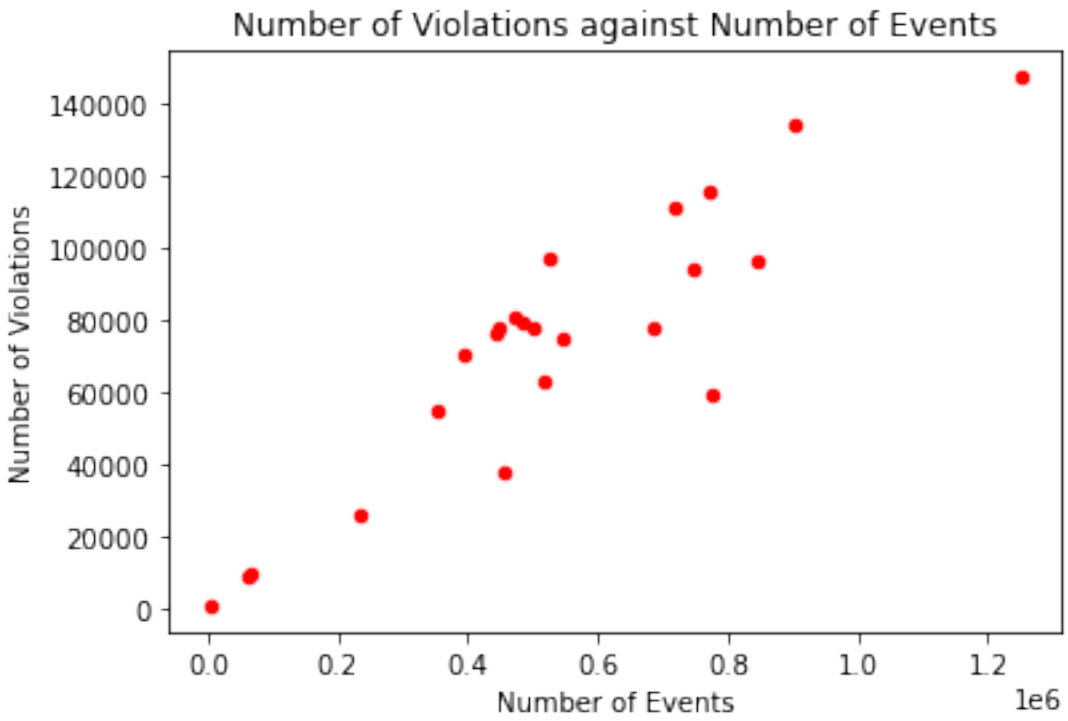
[9]: `Text(0.5, 0, 'Count')`




The number of parking infringements also seem to follow a similar pattern to the number of parking events in the same areas. To identify the relationship between these two variables, we will now plot a graph of the number of infringements based on the number of parking events of each section.

[10]:
```
Viol_Num = parkingDB.loc[mask_violation, 'Area'].value_counts()
Num_Events = parkingDB['Area'].value_counts()

data = {'Number of Violations':  Viol_Num,
        'Number of Events': Num_Events}

df = pd.DataFrame (data, columns = ['Number of Violations','Number of Events'])

ax2 = df.plot.scatter(x='Number of Events',
                      y='Number of Violations',
                      c='red')
plt.title("Number of Violations against Number of Events")
```

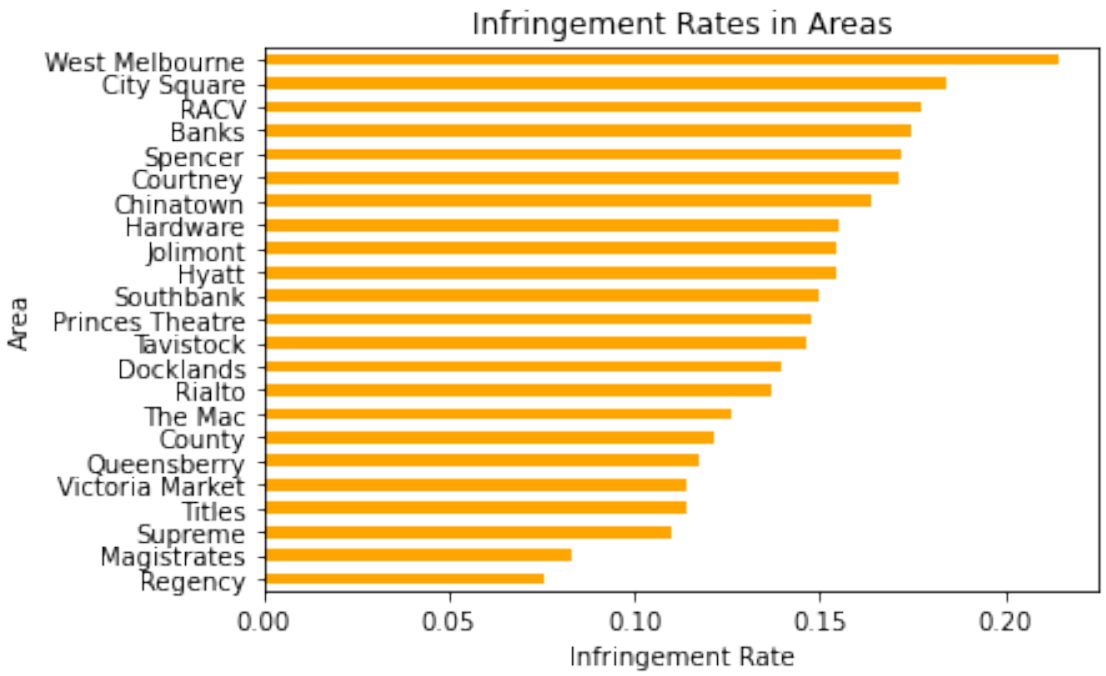
[10]: `Text(0.5, 1.0, 'Number of Violations against Number of Events')`




There seems to be a positive linear correlation between the number of parking events and the number of violations in the same area. Therefore linear regression will be attempted first before more categorical values.

[69]:
```
areas = parkingDB['Area'].value_counts()
violations = parkingDB.loc[mask_violation, 'Area'].value_counts()
areas = areas.astype(float)
violations = violations.astype(float)/areas
violations.sort_values().plot.barh(color="orange")
plt.title('Infringement Rates in Areas')
plt.ylabel('Area')
plt.xlabel("Infringement Rate")
```

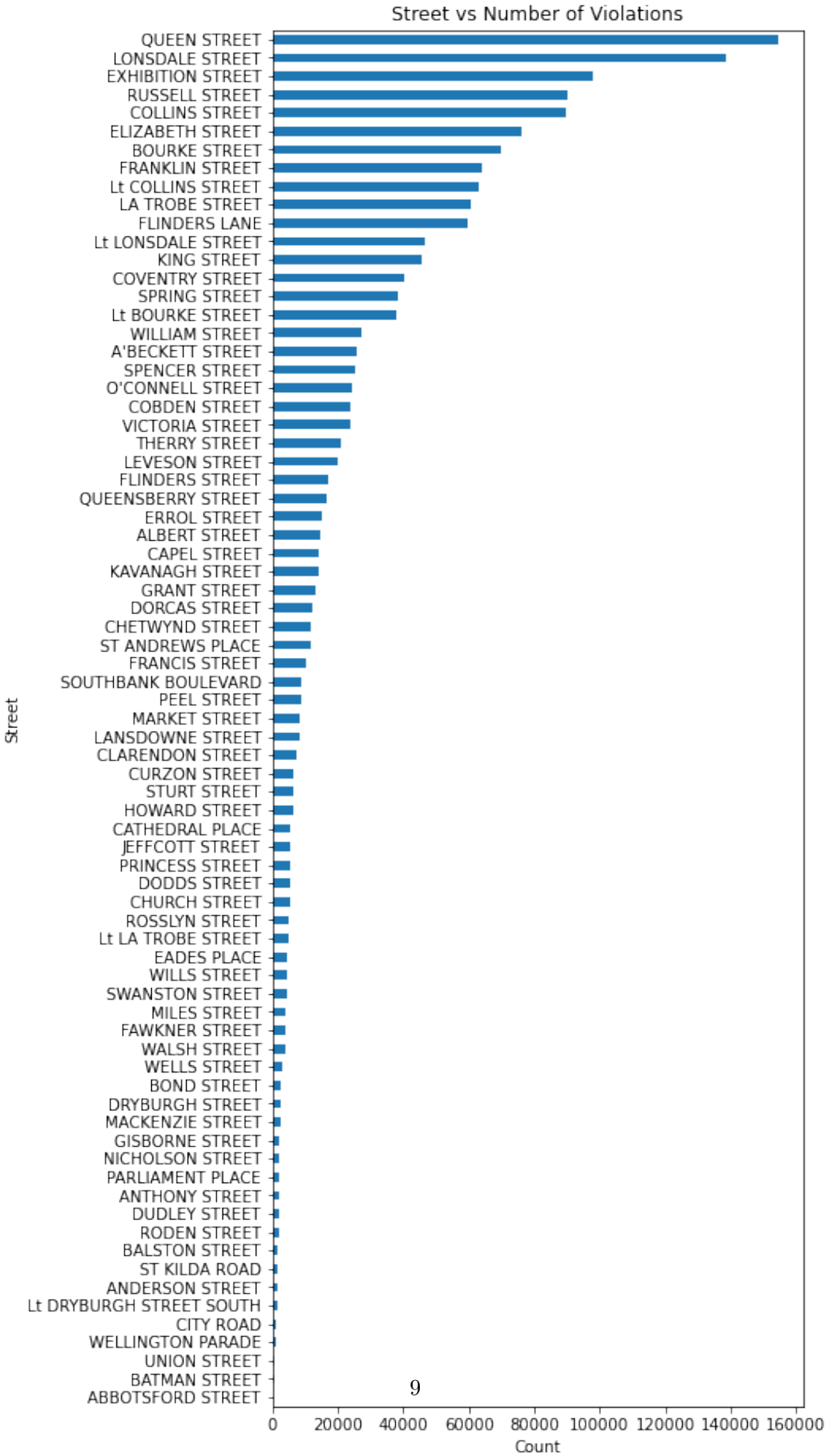
[69]:
```
Text(0.5, 0, 'Infringement Rate')
```




### 1.2.2 2.2 Street Exploration

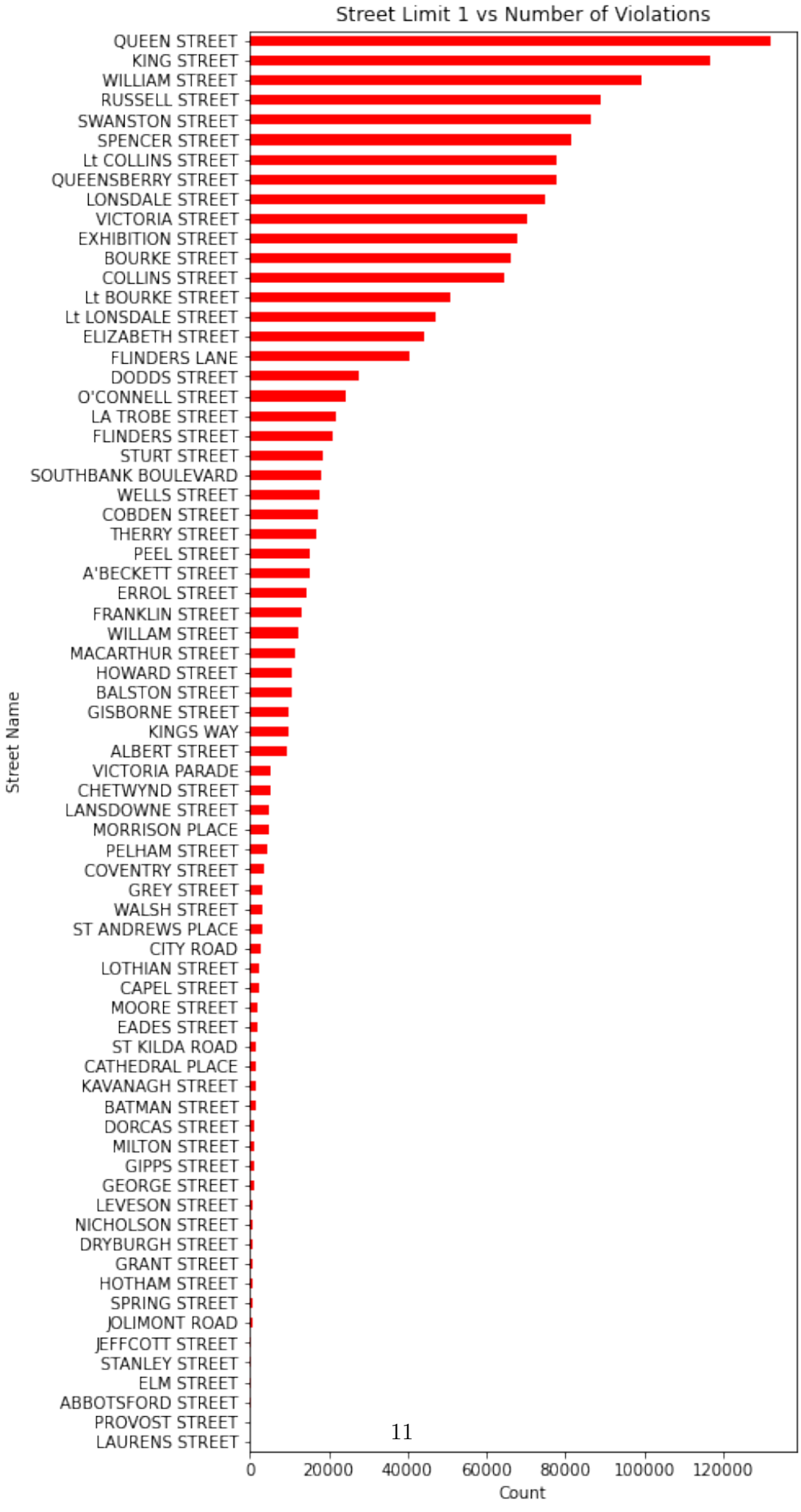
[70]:
```
mask_violation = parkingDB['Violation'] == 1
plt.figure(figsize=(6,16))
parkingDB.loc[mask_violation, 'Street'].value_counts().sort_values().plot.barh()
plt.title('Street vs Number of Violations')
plt.ylabel('Street')
plt.xlabel("Count")
```

[70]:
```
Text(0.5, 0, 'Count')
```


Street vs Number of Violations
QUEEN STREET
LONSDALE STREET
EXHIBITION STREET
RUSSELL STREET
COLLINS STREET
ELIZABETH STREET
BOURKE STREET
FRANKLIN STREET
Lt COLLINS STREET
LA TROBE STREET
FLINDERS LANE
Lt LONSDALE STREET
KING STREET
COVENTRY STREET
SPRING STREET
Lt BOURKE STREET
WILLIAM STREET
A'BECKETT STREET
SPENCER STREET
O'CONNELL STREET
COBDEN STREET
VICTORIA STREET
THERRY STREET
LEVESON STREET
FLINDERS STREET
QUEENSBERRY STREET
ERROL STREET
ALBERT STREET
CAPEL STREET
KAVANAGH STREET
GRANT STREET
DORCAS STREET
CHETWYND STREET
ST ANDREWS PLACE
FRANCIS STREET
SOUTHBANK BOULEVARD
PEEL STREET
MARKET STREET
LANSDOWNE STREET
CLARENDON STREET
CURZON STREET
STURT STREET
HOWARD STREET
CATHEDRAL PLACE
JEFFCOTT STREET
PRINCESS STREET
DODDS STREET
CHURCH STREET
ROSSLYN STREET
Lt LA TROBE STREET
EADES PLACE
WILLS STREET
SWANSTON STREET
MILES STREET
FAWKNER STREET
WALSH STREET
WELLS STREET
BOND STREET
DRYBURGH STREET
MACKENZIE STREET
GISBORNE STREET
NICHOLSON STREET
PARLIAMENT PLACE
ANTHONY STREET
DUDLEY STREET
RODEN STREET
BALSTON STREET
ST KILDA ROAD
ANDERSON STREET
Lt DRYBURGH STREET SOUTH
CITY ROAD
WELLINGTON PARADE
UNION STREET
BATMAN STREET
ABBOTSFORD STREET
Street
0
20000
40000
60000
80000
100000
120000
140000
160000
Count

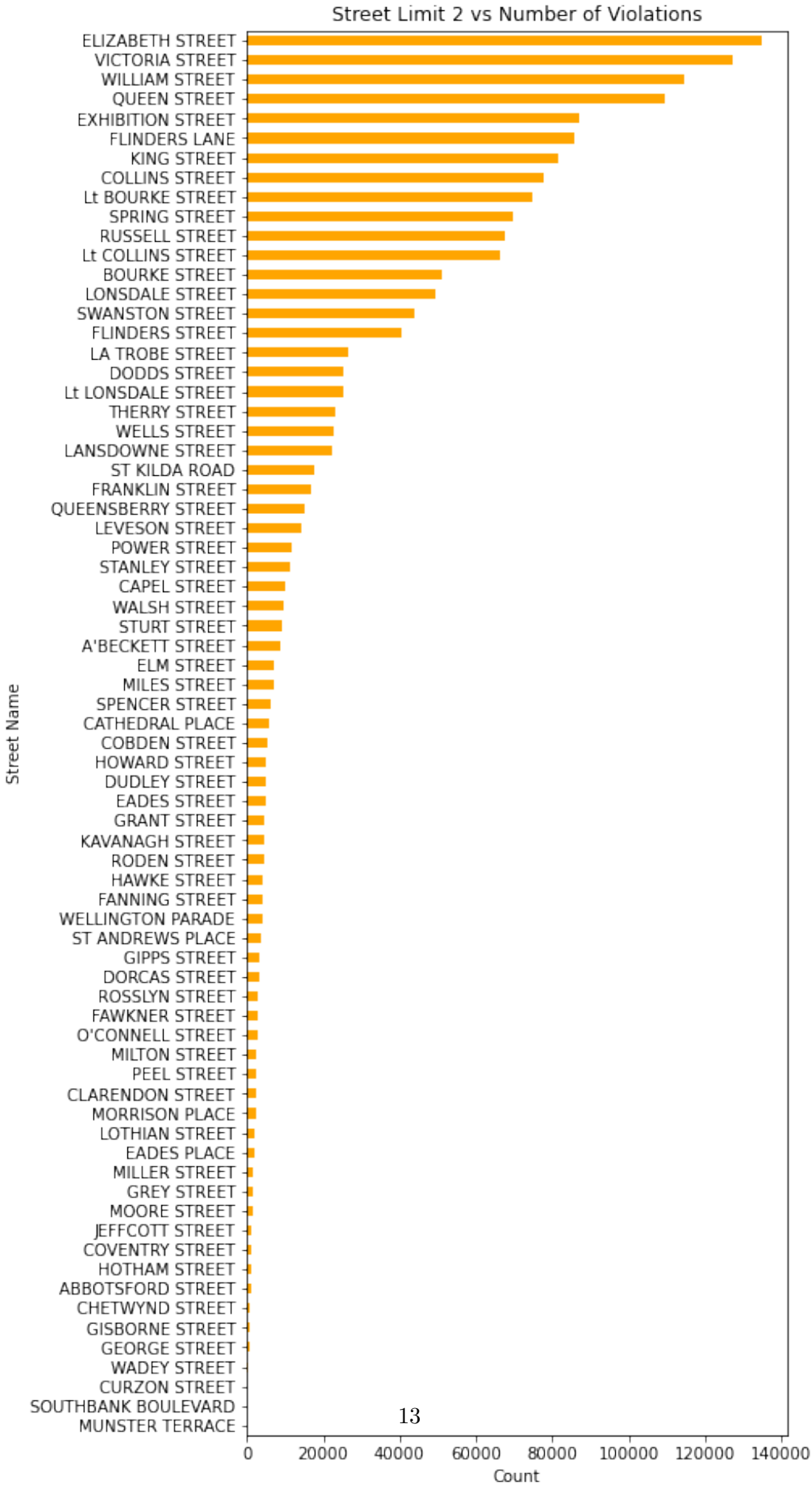
[71]:
```
mask_violation = parkingDB['Violation'] == 1
plt.figure(figsize=(6,16))
parkingDB.loc[mask_violation, 'Street Limit 1'].value_counts().sort_values().
 ↪plot.barh(color="red")
plt.title('Street Limit 1 vs Number of Violations')
plt.ylabel('Street Name')
plt.xlabel("Count")
```

[71]:
```
Text(0.5, 0, 'Count')
```


Street Limit 1 vs Number of Violations
QUEEN STREET
KING STREET
WILLIAM STREET
RUSSELL STREET
SWANSTON STREET
SPENCER STREET
Lt COLLINS STREET
QUEENSBERRY STREET
LONSDALE STREET
VICTORIA STREET
EXHIBITION STREET
BOURKE STREET
COLLINS STREET
Lt BOURKE STREET
Lt LONSDALE STREET
ELIZABETH STREET
FLINDERS LANE
DODDS STREET
O'CONNELL STREET
LA TROBE STREET
FLINDERS STREET
STURT STREET
SOUTHBANK BOULEVARD
WELLS STREET
COBDEN STREET
THERRY STREET
PEEL STREET
A'BECKETT STREET
ERROL STREET
FRANKLIN STREET
WILLAM STREET
MACARTHUR STREET
HOWARD STREET
BALSTON STREET
GISBORNE STREET
KINGS WAY
ALBERT STREET
VICTORIA PARADE
CHETWYND STREET
LANSDOWNE STREET
MORRISON PLACE
PELHAM STREET
COVENTRY STREET
GREY STREET
WALSH STREET
ST ANDREWS PLACE
CITY ROAD
LOTHIAN STREET
CAPEL STREET
MOORE STREET
EADES STREET
ST KILDA ROAD
CATHEDRAL PLACE
KAVANAGH STREET
BATMAN STREET
DORCAS STREET
MILTON STREET
GIPPS STREET
GEORGE STREET
LEVESON STREET
NICHOLSON STREET
DRYBURGH STREET
GRANT STREET
HOTHAM STREET
SPRING STREET
JOLIMONT ROAD
JEFFCOTT STREET
STANLEY STREET
ELM STREET
ABBOTSFORD STREET
PROVOST STREET
LAURENS STREET
Street Name
0
20000
40000
60000
80000
100000
120000
Count

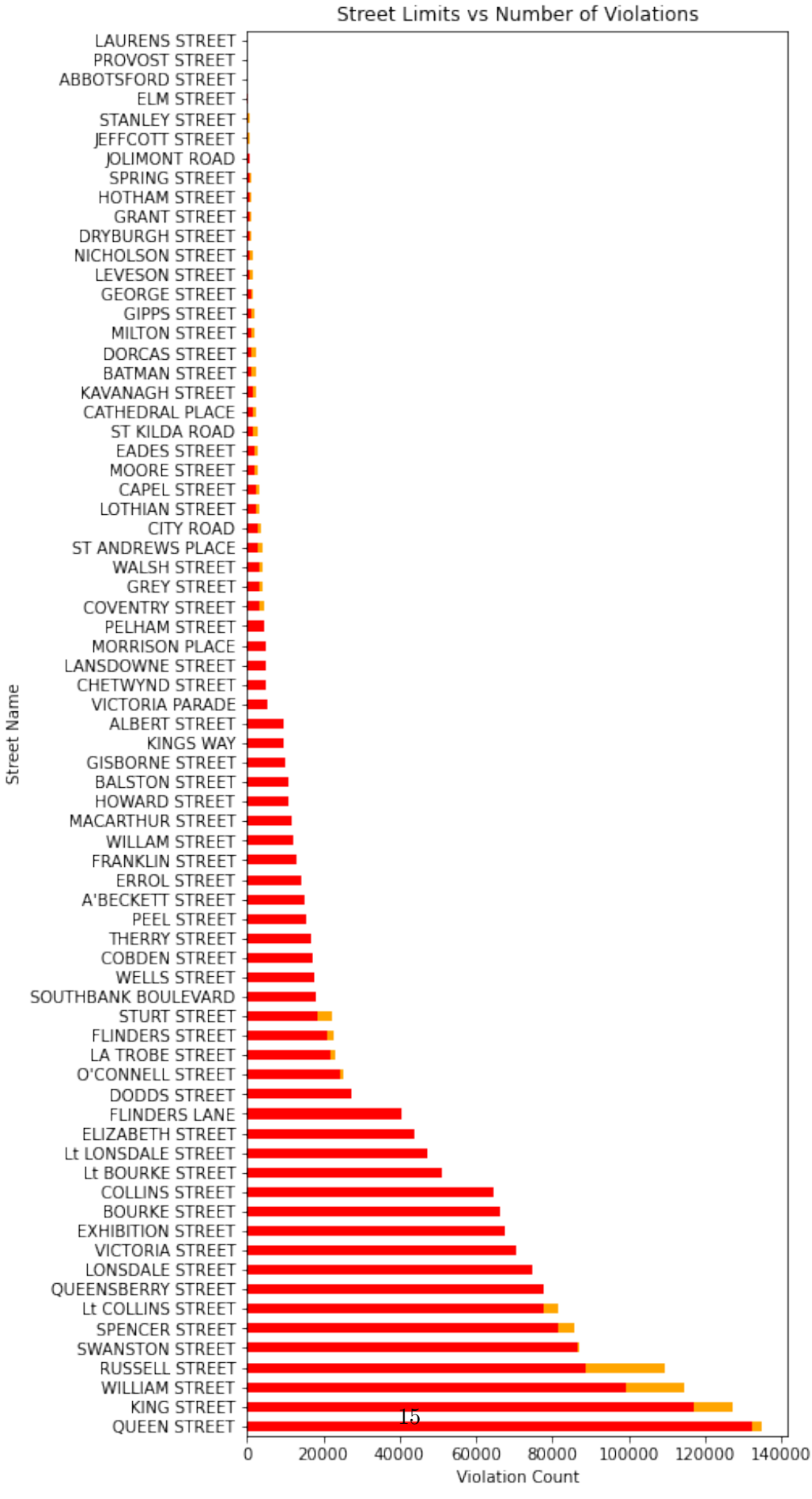
[5]:
```
mask_violation = parkingDB['Violation'] == 1
plt.figure(figsize=(6,16))
parkingDB.loc[mask_violation, 'Street Limit 2'].value_counts().sort_values().
 ↪plot.barh(color='orange')
plt.title('Street Limit 2 vs Number of Violations')
plt.ylabel('Street Name')
plt.xlabel("Count")
```

[5]:
```
Text(0.5, 0, 'Count')
```


Street Limit 2 vs Number of Violations
ELIZABETH STREET
VICTORIA STREET
WILLIAM STREET
QUEEN STREET
EXHIBITION STREET
FLINDERS LANE
KING STREET
COLLINS STREET
Lt BOURKE STREET
SPRING STREET
RUSSELL STREET
Lt COLLINS STREET
BOURKE STREET
LONSDALE STREET
SWANSTON STREET
FLINDERS STREET
LA TROBE STREET
DODDS STREET
Lt LONSDALE STREET
THERRY STREET
WELLS STREET
LANSDOWNE STREET
ST KILDA ROAD
FRANKLIN STREET
QUEENSBERRY STREET
LEVESON STREET
POWER STREET
STANLEY STREET
CAPEL STREET
WALSH STREET
STURT STREET
A'BECKETT STREET
ELM STREET
MILES STREET
SPENCER STREET
CATHEDRAL PLACE
COBDEN STREET
HOWARD STREET
DUDLEY STREET
EADES STREET
GRANT STREET
KAVANAGH STREET
RODEN STREET
HAWKE STREET
FANNING STREET
WELLINGTON PARADE
ST ANDREWS PLACE
GIPPS STREET
DORCAS STREET
ROSSLYN STREET
FAWKNER STREET
O'CONNELL STREET
MILTON STREET
PEEL STREET
CLARENDON STREET
MORRISON PLACE
LOTHIAN STREET
EADES PLACE
MILLER STREET
GREY STREET
MOORE STREET
JEFFCOTT STREET
COVENTRY STREET
HOTHAM STREET
ABBOTSFORD STREET
CHETWYND STREET
GISBORNE STREET
GEORGE STREET
WADEY STREET
CURZON STREET
SOUTHBANK BOULEVARD
MUNSTER TERRACE
Street Name
0
20000
40000
60000
80000
100000
120000
140000
Count

[73]:
```
mask_violation = parkingDB['Violation'] == 1
plt.figure(figsize=(6,16))
parkingDB.loc[mask_violation, 'Street Limit 2'].value_counts().plot.
 ↪barh(color="orange")
parkingDB.loc[mask_violation, 'Street Limit 1'].value_counts().plot.
 ↪barh(color="red")

plt.title('Street Limits vs Number of Violations')
plt.ylabel('Street Name')
plt.xlabel("Violation Count")
```

[73]:
```
Text(0.5, 0, 'Violation Count')
```


Street Limits vs Number of Violations
LAURENS STREET
PROVOST STREET
ABBOTSFORD STREET
ELM STREET
STANLEY STREET
JEFFCOTT STREET
JOLIMONT ROAD
SPRING STREET
HOTHAM STREET
GRANT STREET
DRYBURGH STREET
NICHOLSON STREET
LEVESON STREET
GEORGE STREET
GIPPS STREET
MILTON STREET
DORCAS STREET
BATMAN STREET
KAVANAGH STREET
CATHEDRAL PLACE
ST KILDA ROAD
EADES STREET
MOORE STREET
CAPEL STREET
LOTHIAN STREET
CITY ROAD
ST ANDREWS PLACE
WALSH STREET
GREY STREET
COVENTRY STREET
PELHAM STREET
MORRISON PLACE
LANSDOWNE STREET
CHETWYND STREET
VICTORIA PARADE
ALBERT STREET
KINGS WAY
GISBORNE STREET
BALSTON STREET
HOWARD STREET
MACARTHUR STREET
WILLAM STREET
FRANKLIN STREET
ERROL STREET
A'BECKETT STREET
PEEL STREET
THERRY STREET
COBDEN STREET
WELLS STREET
SOUTHBANK BOULEVARD
STURT STREET
FLINDERS STREET
LA TROBE STREET
O'CONNELL STREET
DODDS STREET
FLINDERS LANE
ELIZABETH STREET
Lt LONSDALE STREET
Lt BOURKE STREET
COLLINS STREET
BOURKE STREET
EXHIBITION STREET
VICTORIA STREET
LONSDALE STREET
QUEENSBERRY STREET
Lt COLLINS STREET
SPENCER STREET
SWANSTON STREET
RUSSELL STREET
WILLIAM STREET
KING STREET
QUEEN STREET
Street Name
0
20000
40000
60000
80000
100000
120000
140000
Violation Count

After changing which street limit was either front or back, the number of violations for each of these side streets are almost identical whether or not these streets are classified as limit 1 or 2.

The top 4 street limits with the most violations include:

- Queen Street
- King Street
- William Street
- Russell Street

This is applicable for both street limit 1 and 2.

The bottom 4 street limits with the most violations include:

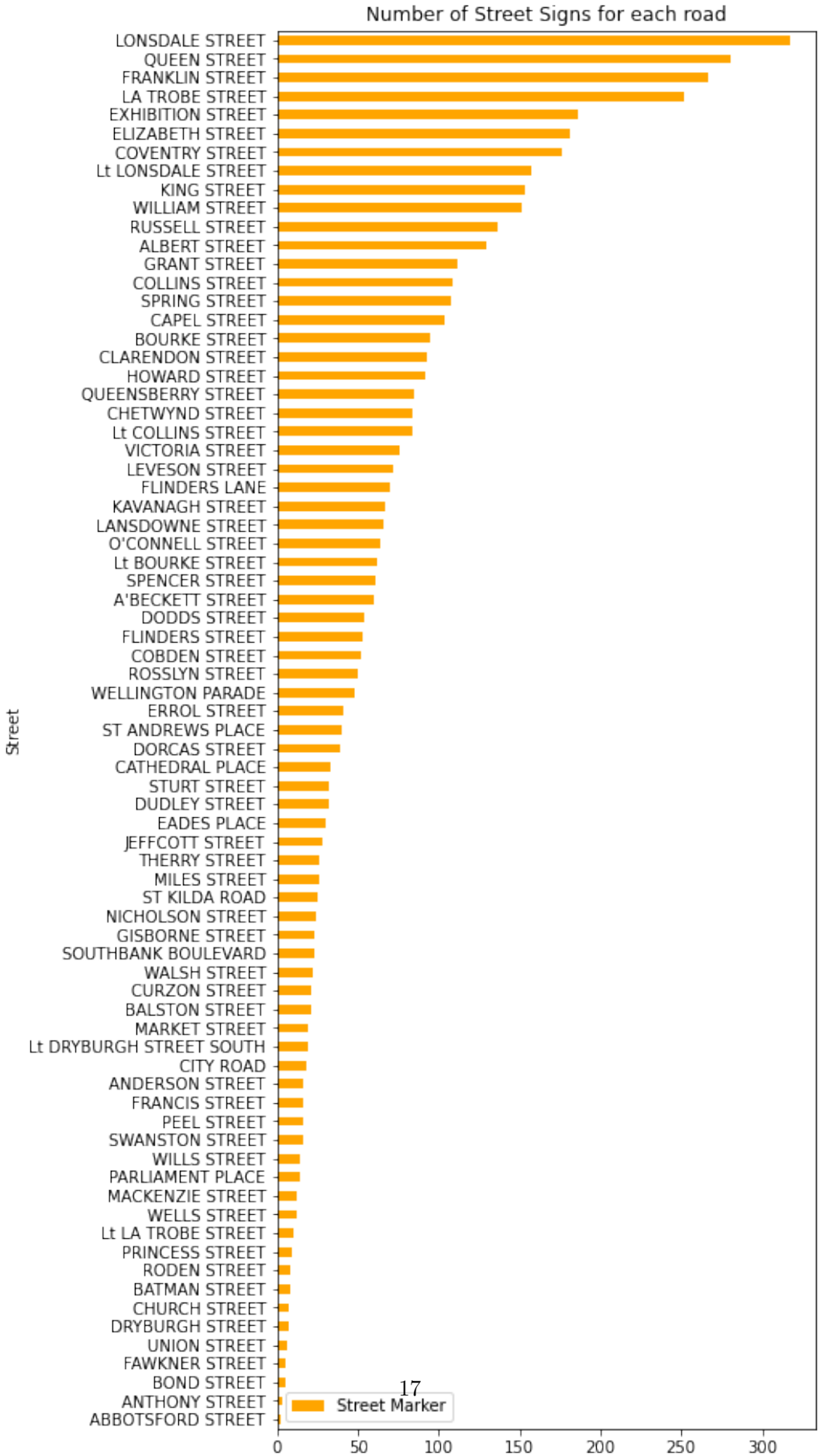
- Laurens Street
- Provost Street
- Abbotsford Street
- Elm Street

This is also applicable for both street limit 1 and 2.

It is therefore reasonable to predict that these side streets with similar violation counts are close to one another and surround the actual street that is being parked in.

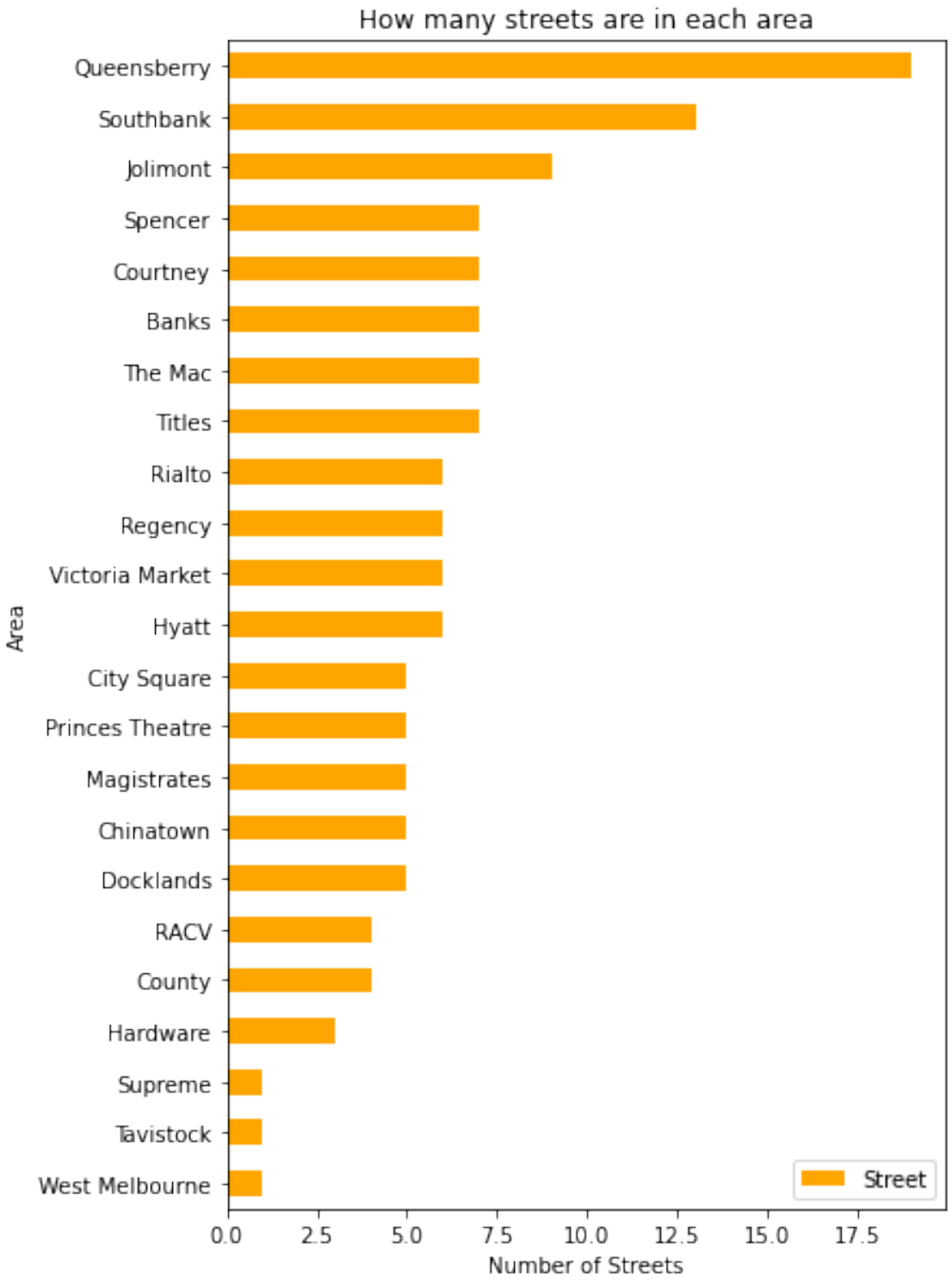
[74]:
```
df = parkingDB.groupby('Street')
df = df.agg({'Street Marker':'nunique'}).sort_values(by='Street Marker')
df.plot.barh(figsize=(6, 16), color="orange")
plt.title('Number of Street Signs for each road')
```

[74]:
```
Text(0.5, 1.0, 'Number of Street Signs for each road')
```


Number of Street Signs for each road
LONSDALE STREET
QUEEN STREET
FRANKLIN STREET
LA TROBE STREET
EXHIBITION STREET
ELIZABETH STREET
COVENTRY STREET
Lt LONSDALE STREET
KING STREET
WILLIAM STREET
RUSSELL STREET
ALBERT STREET
GRANT STREET
COLLINS STREET
SPRING STREET
CAPEL STREET
BOURKE STREET
CLARENDON STREET
HOWARD STREET
QUEENSBERRY STREET
CHETWYND STREET
Lt COLLINS STREET
VICTORIA STREET
LEVESON STREET
FLINDERS LANE
KAVANAGH STREET
LANSDOWNE STREET
O'CONNELL STREET
Lt BOURKE STREET
SPENCER STREET
A'BECKETT STREET
DODDS STREET
FLINDERS STREET
COBDEN STREET
ROSSLYN STREET
WELLINGTON PARADE
ERROL STREET
ST ANDREWS PLACE
DORCAS STREET
CATHEDRAL PLACE
STURT STREET
DUDLEY STREET
EADES PLACE
JEFFCOTT STREET
THERRY STREET
MILES STREET
ST KILDA ROAD
NICHOLSON STREET
GISBORNE STREET
SOUTHBANK BOULEVARD
WALSH STREET
CURZON STREET
BALSTON STREET
MARKET STREET
Lt DRYBURGH STREET SOUTH
CITY ROAD
ANDERSON STREET
FRANCIS STREET
PEEL STREET
SWANSTON STREET
WILLS STREET
PARLIAMENT PLACE
MACKENZIE STREET
WELLS STREET
Lt LA TROBE STREET
PRINCESS STREET
RODEN STREET
BATMAN STREET
CHURCH STREET
DRYBURGH STREET
UNION STREET
FAWKNER STREET
BOND STREET
ANTHONY STREET
ABBOTSFORD STREET
Street
Street Marker
0
50
100
150
200
250
300

[75]:
```
df = parkingDB.groupby('Area')
df = df.agg({'Street':'nunique'}).sort_values(by='Street')
df.plot.barh(figsize=(6, 10), color="orange")
plt.title('How many streets are in each area')
plt.xlabel("Number of Streets")
```

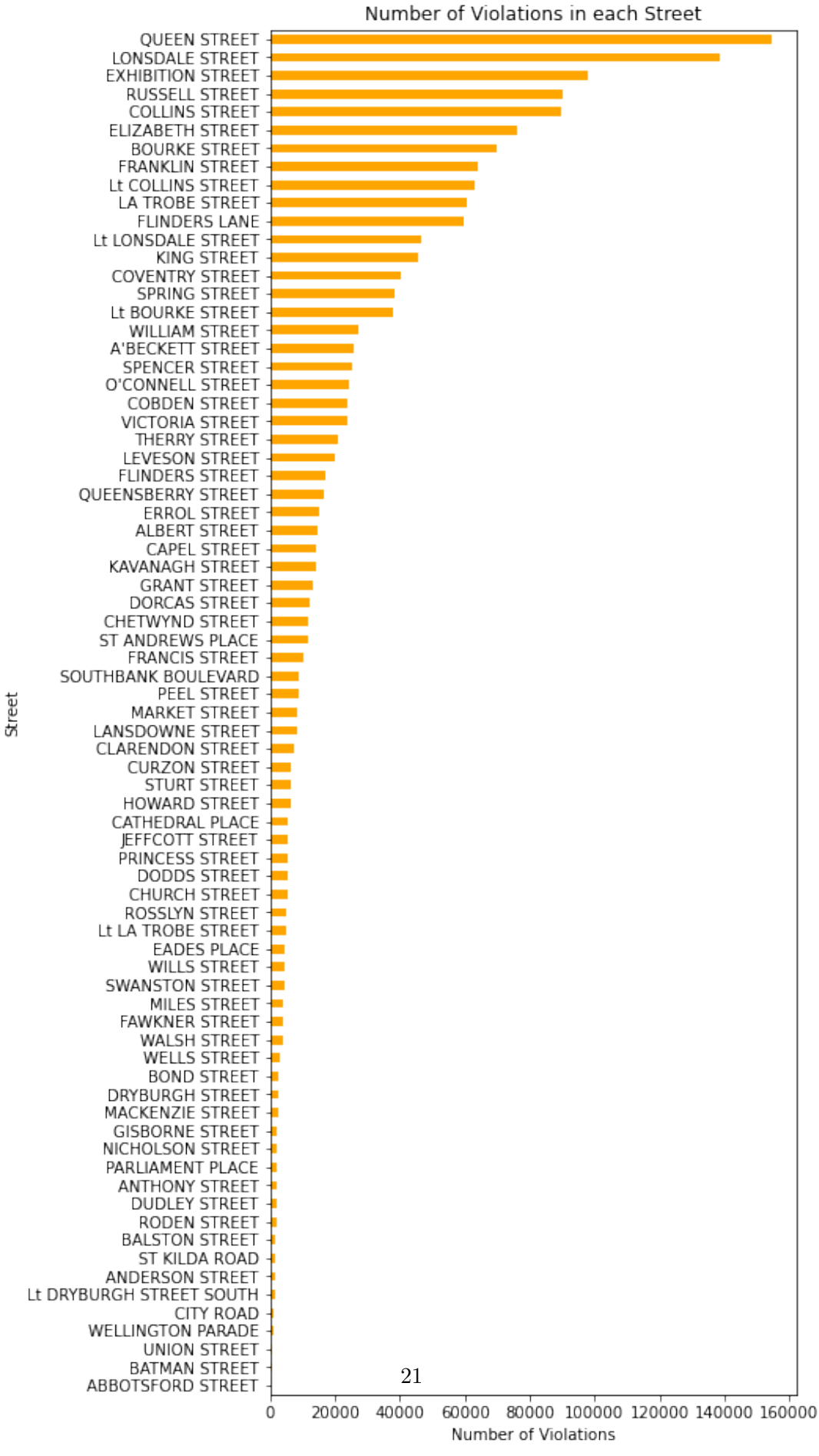
[75]:
```
Text(0.5, 0, 'Number of Streets')
```




[76]:
```
mask_violation = parkingDB['Violation'] == 1
plt.figure(figsize=(6,16))
```

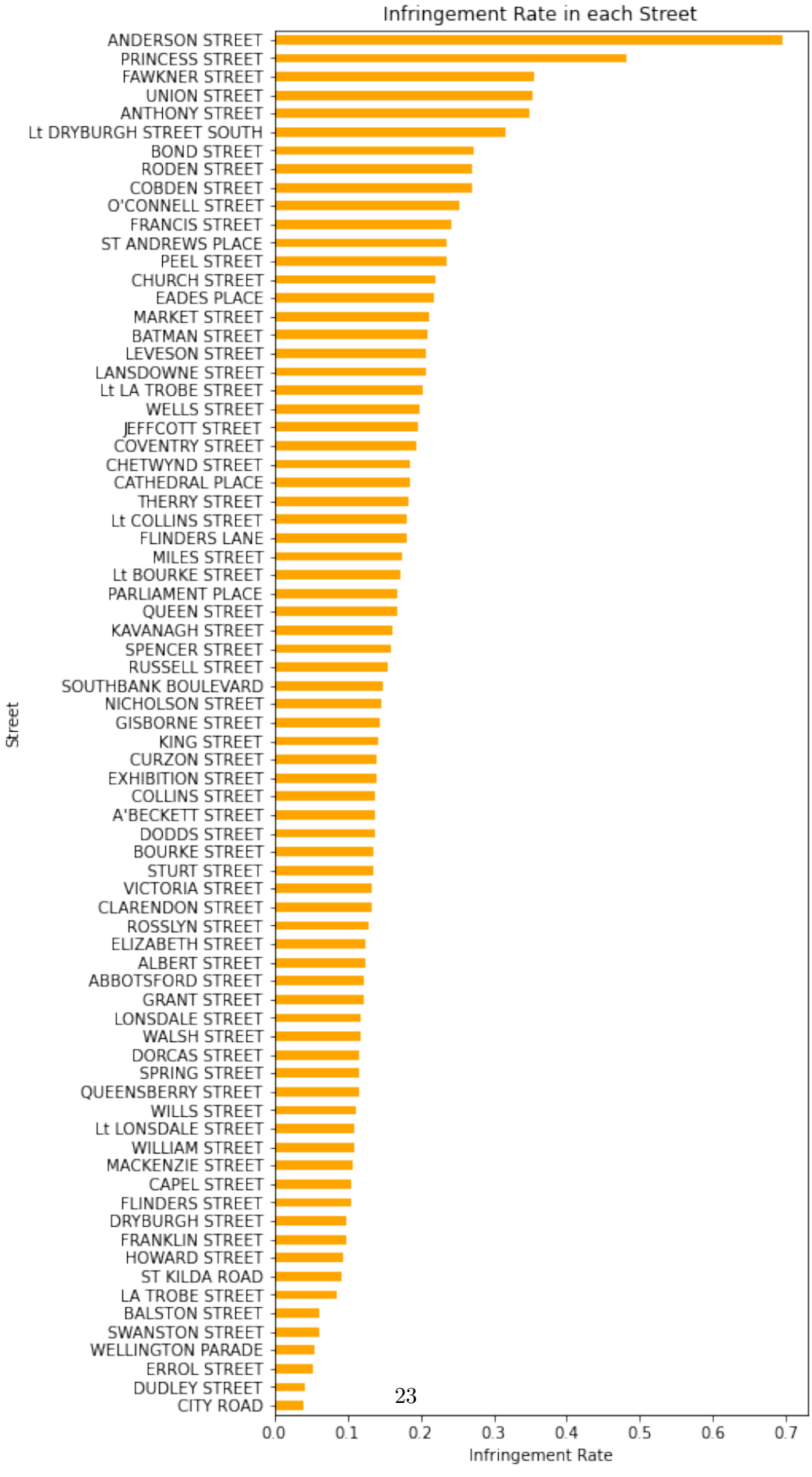
```
parkingDB.loc[mask_violation, 'Street'].value_counts().sort_values().plot.
 ↪barh(color="orange")
plt.title('Number of Violations in each Street')
plt.ylabel('Street')
plt.xlabel("Number of Violations")
```

[76]: Text(0.5, 0, 'Number of Violations')


Number of Violations in each Street
QUEEN STREET
LONSDALE STREET
EXHIBITION STREET
RUSSELL STREET
COLLINS STREET
ELIZABETH STREET
BOURKE STREET
FRANKLIN STREET
Lt COLLINS STREET
LA TROBE STREET
FLINDERS LANE
Lt LONSDALE STREET
KING STREET
COVENTRY STREET
SPRING STREET
Lt BOURKE STREET
WILLIAM STREET
A'BECKETT STREET
SPENCER STREET
O'CONNELL STREET
COBDEN STREET
VICTORIA STREET
THERRY STREET
LEVESON STREET
FLINDERS STREET
QUEENSBERRY STREET
ERROL STREET
ALBERT STREET
CAPEL STREET
KAVANAGH STREET
GRANT STREET
DORCAS STREET
CHETWYND STREET
ST ANDREWS PLACE
FRANCIS STREET
SOUTHBANK BOULEVARD
PEEL STREET
MARKET STREET
LANSDOWNE STREET
CLARENDON STREET
CURZON STREET
STURT STREET
HOWARD STREET
CATHEDRAL PLACE
JEFFCOTT STREET
PRINCESS STREET
DODDS STREET
CHURCH STREET
ROSSLYN STREET
Lt LA TROBE STREET
EADES PLACE
WILLS STREET
SWANSTON STREET
MILES STREET
FAWKNER STREET
WALSH STREET
WELLS STREET
BOND STREET
DRYBURGH STREET
MACKENZIE STREET
GISBORNE STREET
NICHOLSON STREET
PARLIAMENT PLACE
ANTHONY STREET
DUDLEY STREET
RODEN STREET
BALSTON STREET
ST KILDA ROAD
ANDERSON STREET
Lt DRYBURGH STREET SOUTH
CITY ROAD
WELLINGTON PARADE
UNION STREET
BATMAN STREET
ABBOTSFORD STREET
Street
0
20000
40000
60000
80000
100000
120000
140000
160000
Number of Violations

[77]:
```
df = parkingDB['Street'].value_counts()
violations = parkingDB.loc[mask_violation, 'Street'].value_counts()
df = df.astype(float)
plt.figure(figsize=(6,16))
violations = violations.astype(float)/df
violations.sort_values().plot.barh(plt.xlabel("Number of Streets"),␣
 ↪color="orange")
plt.title('Infringement Rate in each Street')
plt.ylabel('Street')
plt.xlabel("Infringement Rate")
```

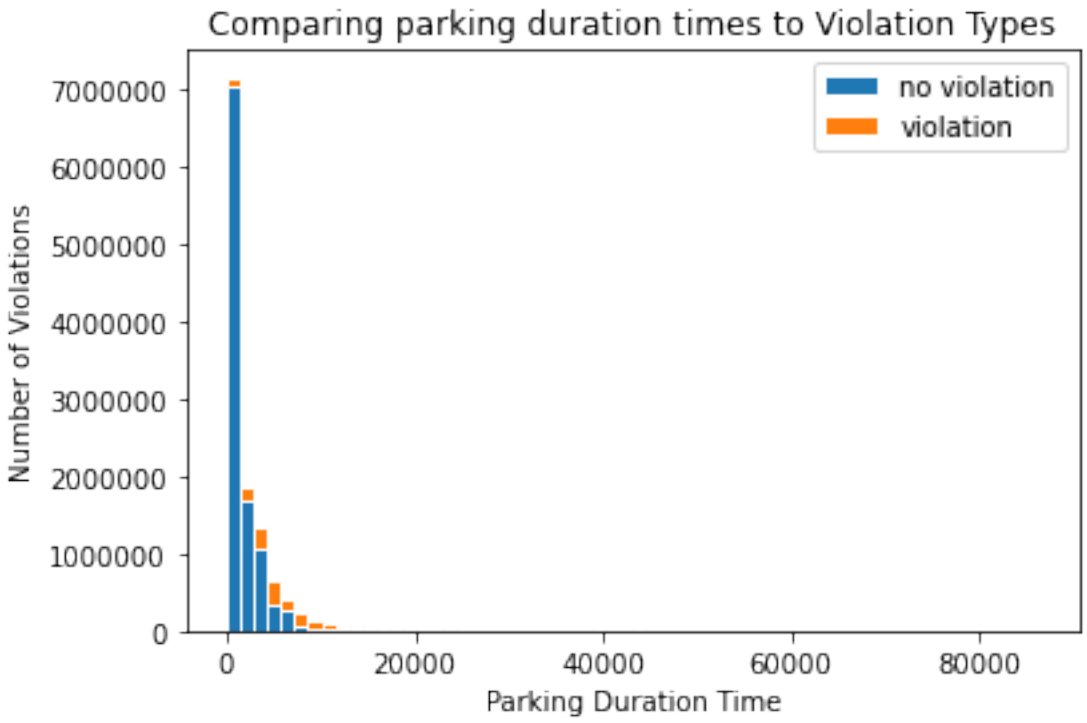
[77]:
```
Text(0.5, 0, 'Infringement Rate')
```


Infringement Rate in each Street
ANDERSON STREET
PRINCESS STREET
FAWKNER STREET
UNION STREET
ANTHONY STREET
Lt DRYBURGH STREET SOUTH
BOND STREET
RODEN STREET
COBDEN STREET
O'CONNELL STREET
FRANCIS STREET
ST ANDREWS PLACE
PEEL STREET
CHURCH STREET
EADES PLACE
MARKET STREET
BATMAN STREET
LEVESON STREET
LANSDOWNE STREET
Lt LA TROBE STREET
WELLS STREET
JEFFCOTT STREET
COVENTRY STREET
CHETWYND STREET
CATHEDRAL PLACE
THERRY STREET
Lt COLLINS STREET
FLINDERS LANE
MILES STREET
Lt BOURKE STREET
PARLIAMENT PLACE
QUEEN STREET
KAVANAGH STREET
SPENCER STREET
RUSSELL STREET
SOUTHBANK BOULEVARD
NICHOLSON STREET
GISBORNE STREET
KING STREET
CURZON STREET
EXHIBITION STREET
COLLINS STREET
A'BECKETT STREET
DODDS STREET
BOURKE STREET
STURT STREET
VICTORIA STREET
CLARENDON STREET
ROSSLYN STREET
ELIZABETH STREET
ALBERT STREET
ABBOTSFORD STREET
GRANT STREET
LONSDALE STREET
WALSH STREET
DORCAS STREET
SPRING STREET
QUEENSBERRY STREET
WILLS STREET
Lt LONSDALE STREET
WILLIAM STREET
MACKENZIE STREET
CAPEL STREET
FLINDERS STREET
DRYBURGH STREET
FRANKLIN STREET
HOWARD STREET
ST KILDA ROAD
LA TROBE STREET
BALSTON STREET
SWANSTON STREET
WELLINGTON PARADE
ERROL STREET
DUDLEY STREET
CITY ROAD
Street
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
Infringement Rate

### 1.2.3 2.3 Parking Duration Exploration

[94]:
```
plt.hist(
    [parkingDB.loc[parkingDB['Violation'] == 0, 'Parking Duration (s)'],
     parkingDB.loc[parkingDB['Violation'] == 1, 'Parking Duration (s)']],
    stacked=True,
    label=['no violation', 'violation'],
    edgecolor='white', bins=60)
plt.ticklabel_format(style="plain")
plt.title("Comparing parking duration times to Violation Types")
plt.xlabel("Parking Duration Time (s)")
plt.ylabel("Number of Violations")

plt.legend()
```

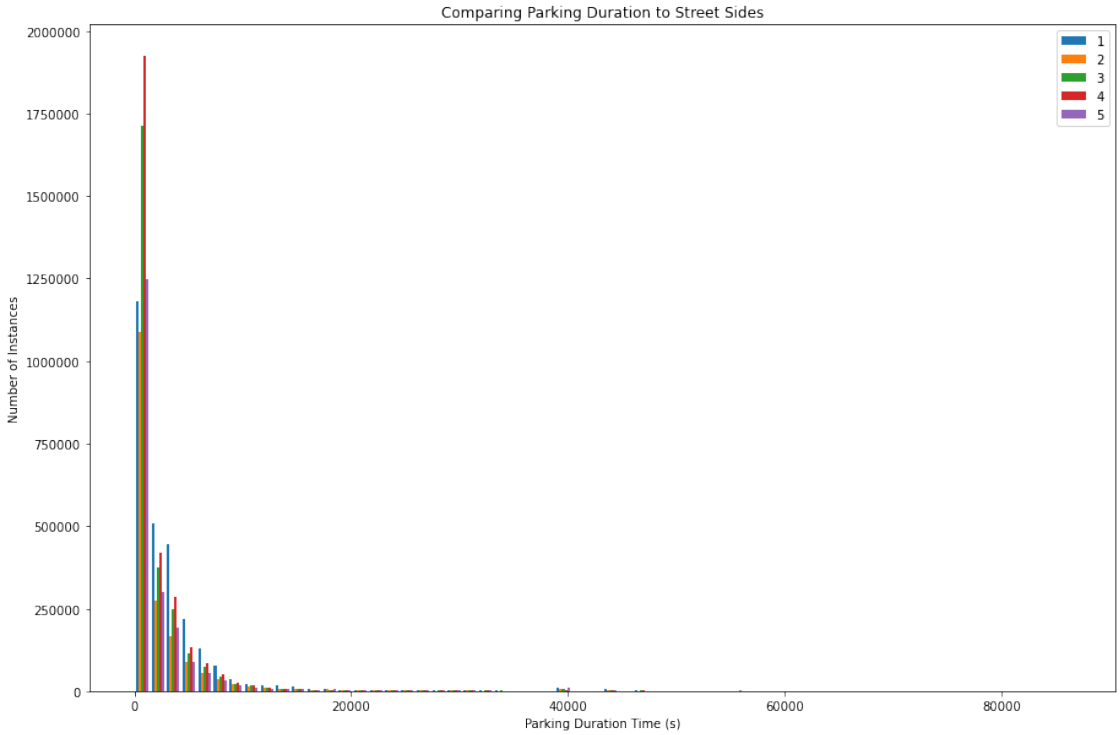
[94]:
```
<matplotlib.legend.Legend at 0x7fb412c219d0>
```



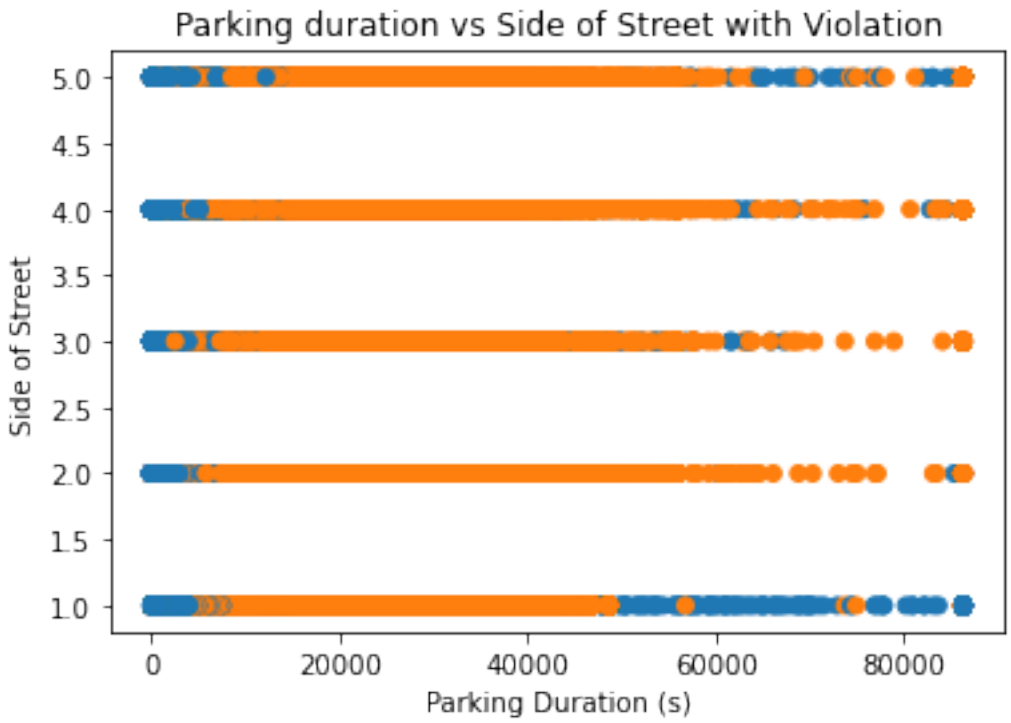

[104]:
```
plt.figure(figsize=(15,10)) #change your figure size as per your desire here
plt.hist(
    [parkingDB.loc[parkingDB['Side Of Street'] == 1, 'Parking Duration (s)'],
     parkingDB.loc[parkingDB['Side Of Street'] == 2, 'Parking Duration (s)'],
```

```
    parkingDB.loc[parkingDB['Side Of Street'] == 3, 'Parking Duration (s)'],
    parkingDB.loc[parkingDB['Side Of Street'] == 4, 'Parking Duration (s)'],
    parkingDB.loc[parkingDB['Side Of Street'] == 5, 'Parking Duration (s)']],
   label=['1', '2', '3', '4', '5'], bins=60)
plt.ticklabel_format(style="plain")
plt.title("Comparing Parking Duration to Street Sides")
plt.xlabel("Parking Duration Time (s)")
plt.ylabel("Number of Instances")
plt.legend()
```

[104]: <matplotlib.legend.Legend at 0x7fb439ff44f0>

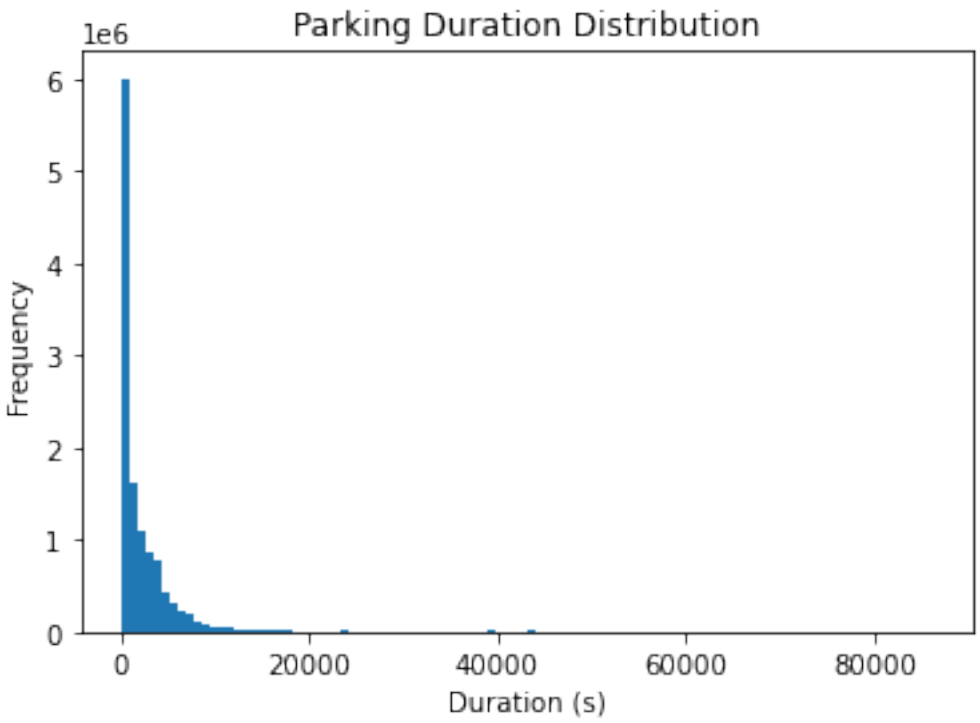



[3]:
```
fig, ax = plt.subplots()
colors = {0:'tab:blue', 1:'tab:orange'}
ax.scatter(parkingDB['Parking Duration (s)'], parkingDB['Side Of Street'],␣
 ↪c=parkingDB['Violation'].map(colors))
plt.title('Parking duration vs Side of Street with Violation')
plt.xlabel('Parking Duration (s)')
plt.ylabel('Side of Street')
plt.show()
```

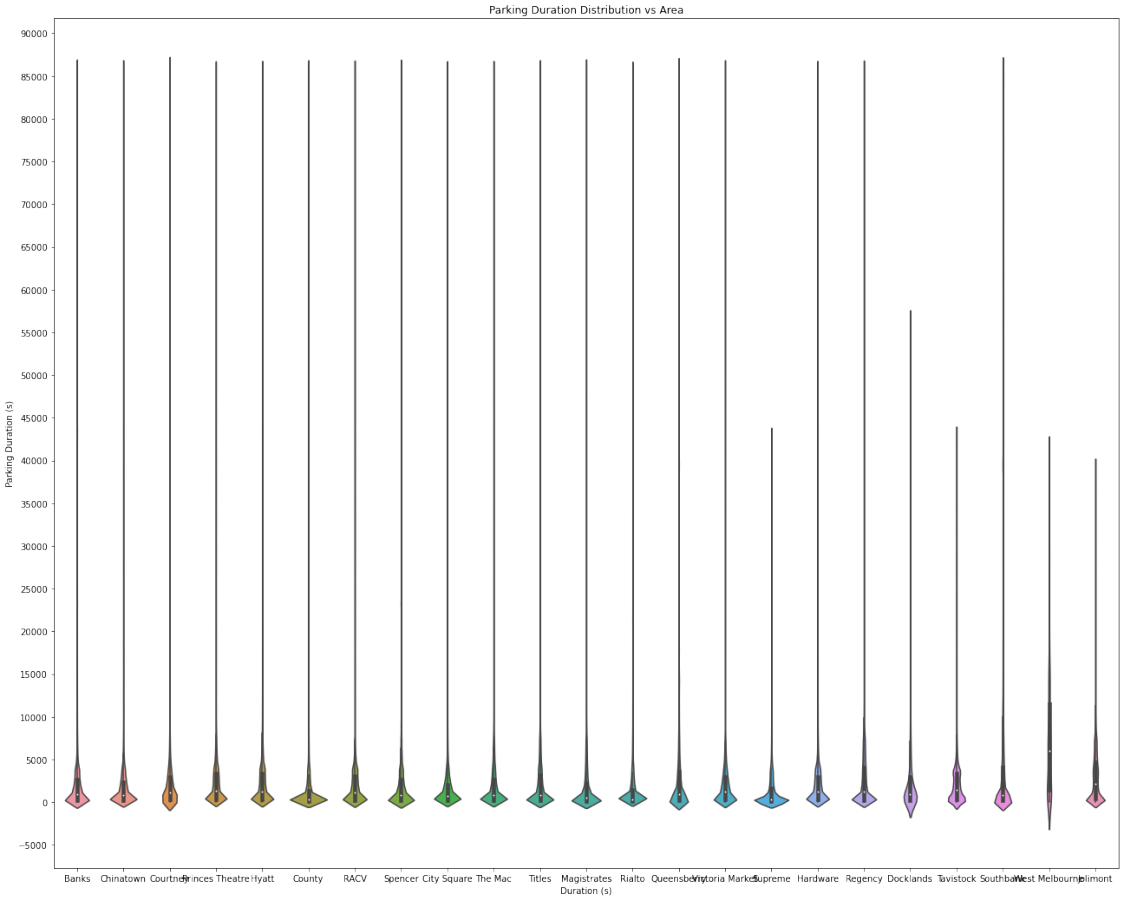



[15]:
```
parkingDB['Parking Duration (s)'].plot(kind='hist', bins=100)
plt.title('Parking Duration Distribution')
plt.xlabel('Duration (s)')
```

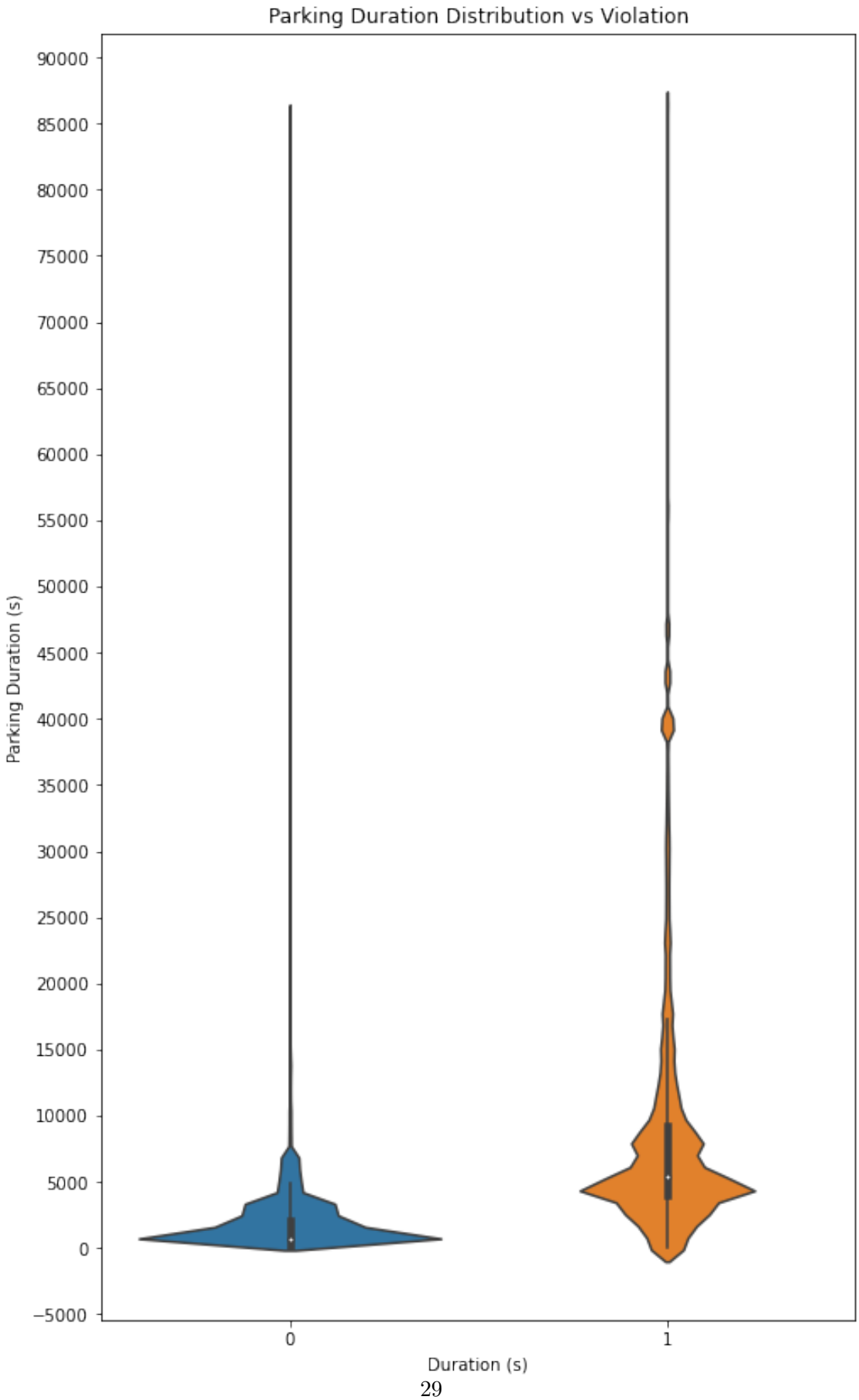
[15]:
```
Text(0.5, 0, 'Duration (s)')
```




[6]:
```
from matplotlib.pyplot import figure
plt.figure(figsize=(22,18))
sns.violinplot(x="Area", y="Parking Duration (s)", data=parkingDB)
#parkingDB.boxplot(figsize=(20, 18), column='Parking Duration (s)', by='Area',␣
↪grid=True, showmeans=True)
plt.title('Parking Duration Distribution vs Area')
plt.xlabel('Duration (s)')
plt.locator_params(axis="y", nbins=20)
```



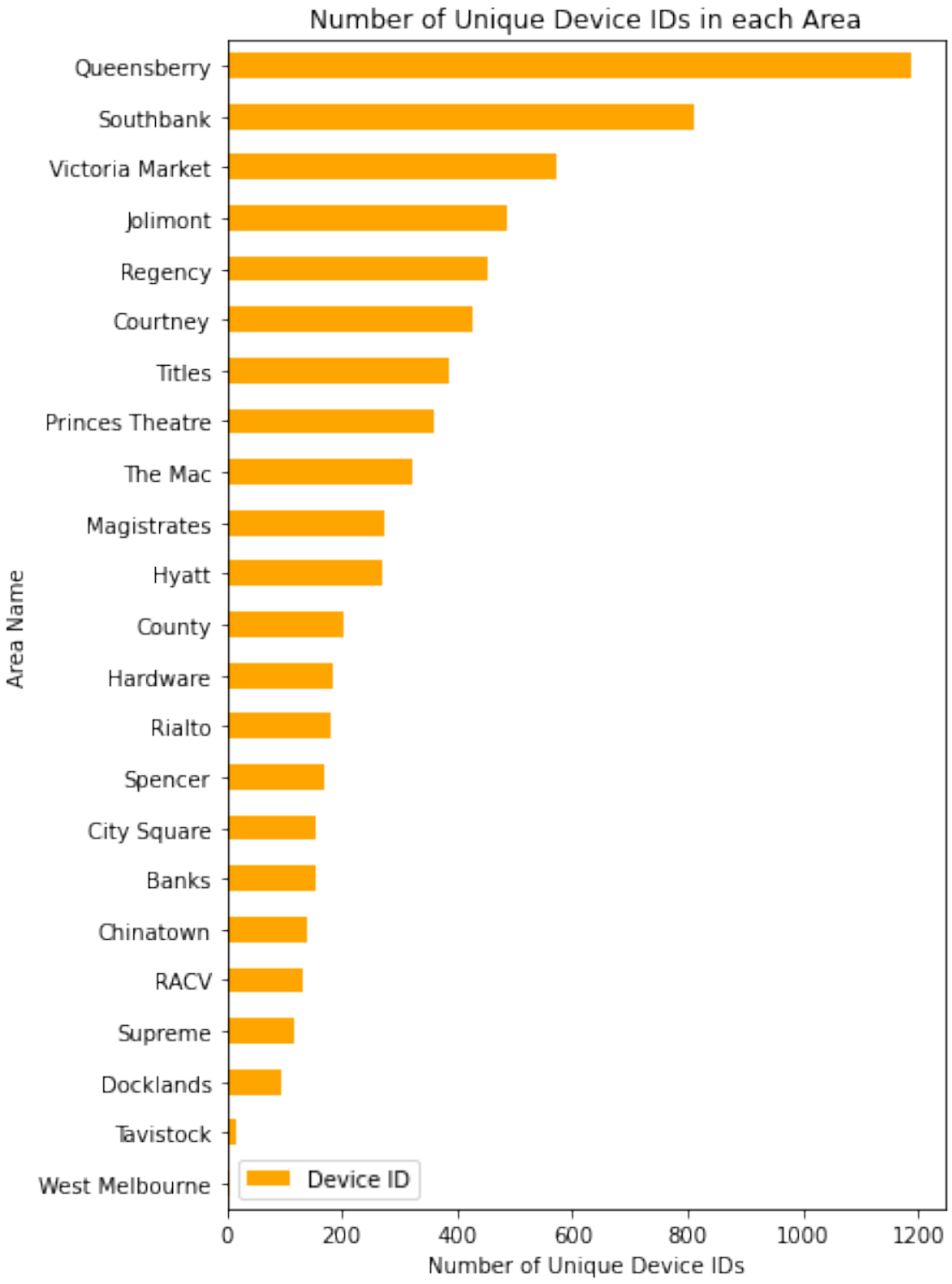

[7]:
```
#parkingDB.boxplot(figsize=(8, 14), column='Parking Duration (s)',␣
 ↪by='Violation', grid=True, showmeans=True)
plt.figure(figsize=(8,14))
sns.violinplot(x="Violation", y="Parking Duration (s)", data=parkingDB)
plt.title('Parking Duration Distribution vs Violation')
plt.xlabel('Duration (s)')
plt.locator_params(axis="y", nbins=20)
```


Parking Duration Distribution vs Violation
Parking Duration (s)
90000
85000
80000
75000
70000
65000
60000
55000
50000
45000
40000
35000
30000
25000
20000
15000
10000
5000
0
−5000
0
1
Duration (s)

### 1.2.4 2.4 Device ID Exploration

[107]:
```
df = parkingDB.groupby('Area')
df = df.agg({'Device ID':'nunique'}).sort_values(by='Device ID')
df.plot.barh(figsize=(6, 10), color="orange")
plt.xlabel("Number of Unique Device IDs")
plt.ylabel("Area Name")
plt.title('Number of Unique Device IDs in each Area')
```

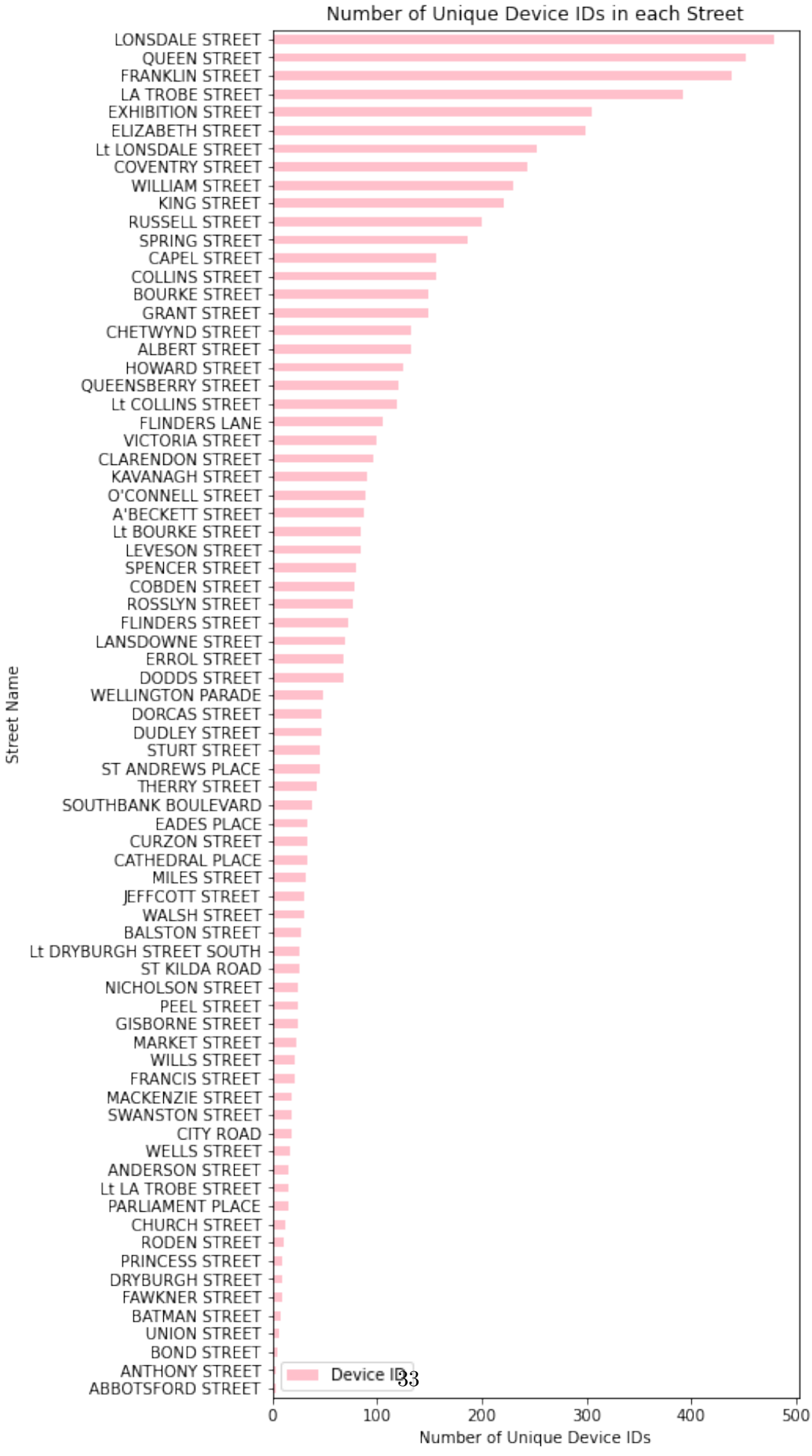
[107]:
```
Text(0.5, 1.0, 'Number of Unique Device IDs in each Area')
```




From this we see that the number of parking instances in these areas in comparison to the number of unique car IDs shows that these cars are regular consumers of carparking spaces in these areas.

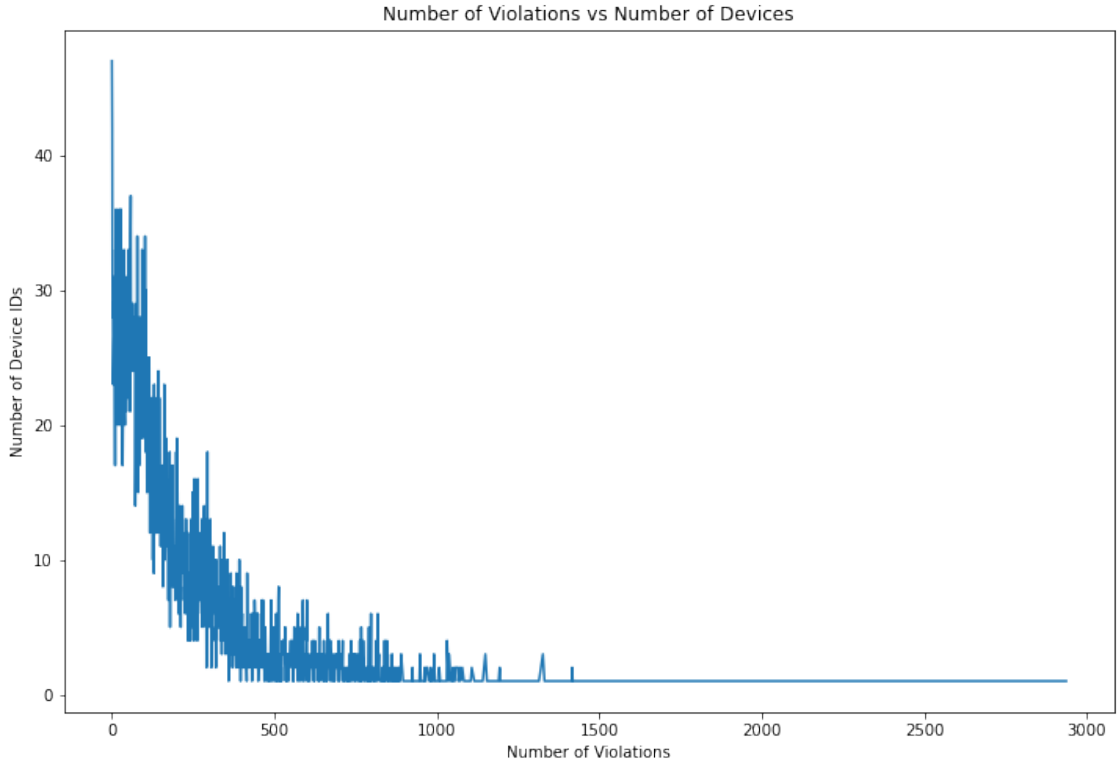
[108]:
```
df = parkingDB.groupby('Street')
df = df.agg({'Device ID':'nunique'}).sort_values(by='Device ID')
df.plot.barh(figsize=(6, 16), color="pink")
plt.xlabel("Number of Unique Device IDs")
plt.ylabel("Street Name")
plt.title('Number of Unique Device IDs in each Street')
```

[108]:
```
Text(0.5, 1.0, 'Number of Unique Device IDs in each Street')
```


Number of Unique Device IDs in each Street
LONSDALE STREET
QUEEN STREET
FRANKLIN STREET
LA TROBE STREET
EXHIBITION STREET
ELIZABETH STREET
Lt LONSDALE STREET
COVENTRY STREET
WILLIAM STREET
KING STREET
RUSSELL STREET
SPRING STREET
CAPEL STREET
COLLINS STREET
BOURKE STREET
GRANT STREET
CHETWYND STREET
ALBERT STREET
HOWARD STREET
QUEENSBERRY STREET
Lt COLLINS STREET
FLINDERS LANE
VICTORIA STREET
CLARENDON STREET
KAVANAGH STREET
O'CONNELL STREET
A'BECKETT STREET
Lt BOURKE STREET
LEVESON STREET
SPENCER STREET
COBDEN STREET
ROSSLYN STREET
FLINDERS STREET
LANSDOWNE STREET
ERROL STREET
DODDS STREET
WELLINGTON PARADE
DORCAS STREET
DUDLEY STREET
STURT STREET
ST ANDREWS PLACE
THERRY STREET
SOUTHBANK BOULEVARD
EADES PLACE
CURZON STREET
CATHEDRAL PLACE
MILES STREET
JEFFCOTT STREET
WALSH STREET
BALSTON STREET
Lt DRYBURGH STREET SOUTH
ST KILDA ROAD
NICHOLSON STREET
PEEL STREET
GISBORNE STREET
MARKET STREET
WILLS STREET
FRANCIS STREET
MACKENZIE STREET
SWANSTON STREET
CITY ROAD
WELLS STREET
ANDERSON STREET
Lt LA TROBE STREET
PARLIAMENT PLACE
CHURCH STREET
RODEN STREET
PRINCESS STREET
DRYBURGH STREET
FAWKNER STREET
BATMAN STREET
UNION STREET
BOND STREET
ANTHONY STREET
ABBOTSFORD STREET
Street Name
Device ID
0
100
200
300
400
500
Number of Unique Device IDs

[6]:
```
#e.g. There are 47 devices committed 1 violation each, meanwhile there is one␣
 ↪device committed 2937 violations.
mask_violation = parkingDB['Violation'] == 1
df = parkingDB.loc[mask_violation, 'Device ID'].value_counts().sort_values()
#parkingDB.loc[mask_violation, 'Street'].value_counts().sort_values().plot.
 ↪barh()
plt.title('Number of Violations vs Number of Devices')
plt.ylabel('Number of Device IDs')
plt.xlabel('Number of Violations')
df.value_counts().sort_index().plot.line(figsize=(12,8))
```

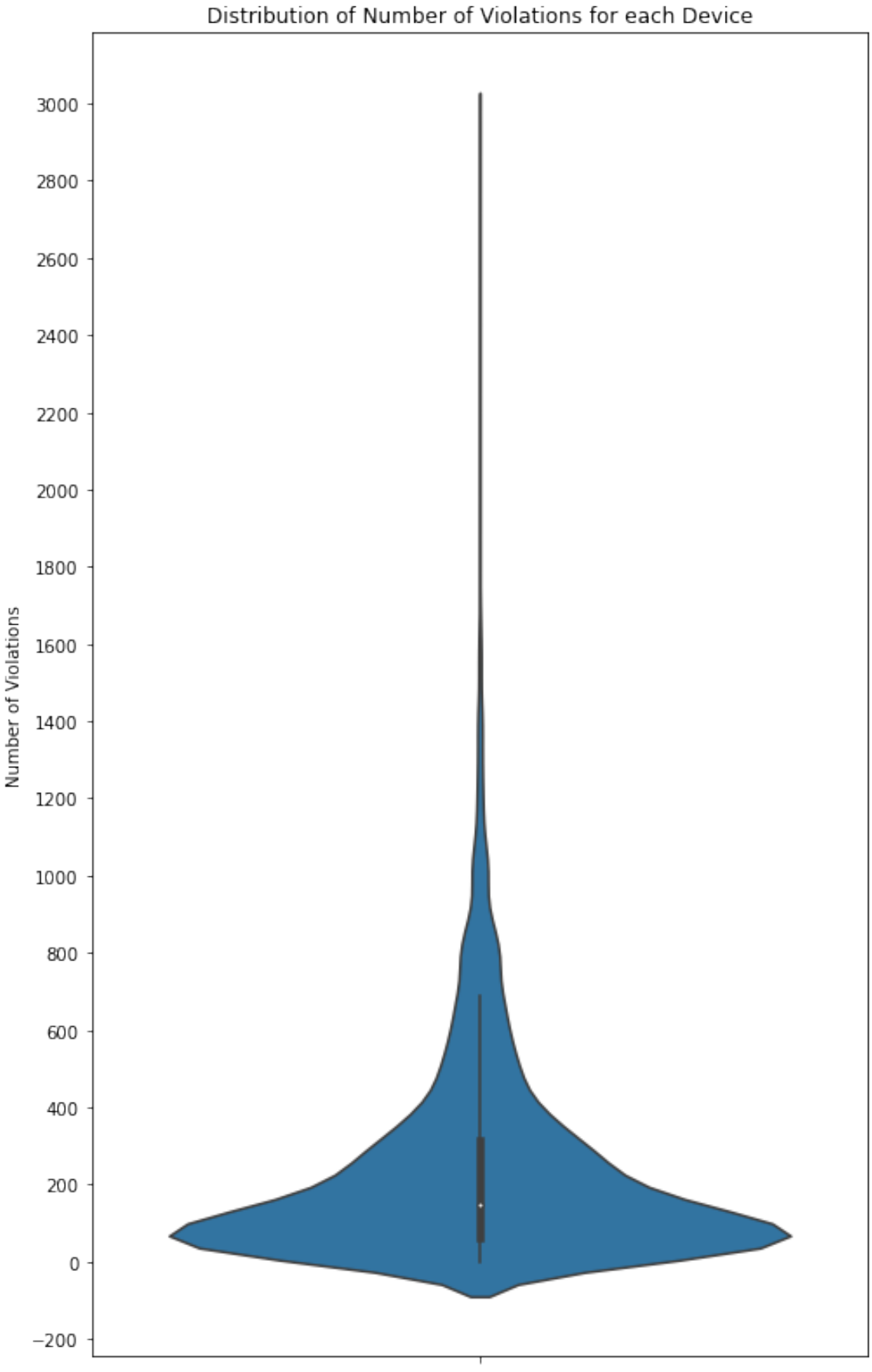
[6]:
```
<AxesSubplot:title={'center':'Number of Violations vs Number of Devices'},
xlabel='Number of Violations', ylabel='Number of Device IDs'>
```




[10]:
```
# mask_violation = parkingDB['Violation'] == 1
# df = parkingDB.loc[mask_violation, 'Device ID'].value_counts()
# #df.plot.box(figsize=(8, 14), grid=True, showmeans=True)
# plt.figure(figsize=(8,14))
# sns.violinplot(y=df, data=df)
# plt.locator_params(axis="y", nbins=20)
```

```
# plt.title('Distribution of Number of Violations for each Device')
# plt.ylabel('Number of Violations')
# MAKES NO SENSE
```

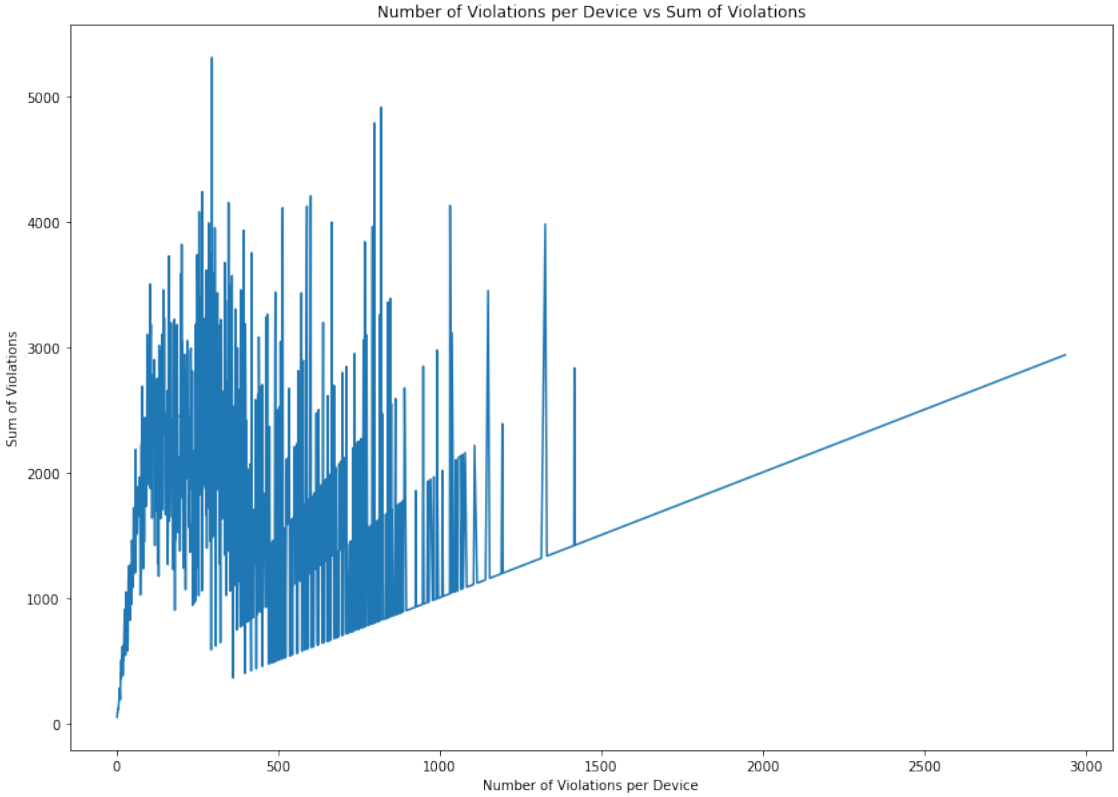
[10]: Text(0, 0.5, 'Number of Violations')


Distribution of Number of Violations for each Device
Number of Violations
3000
2800
2600
2400
2200
2000
1800
1600
1400
1200
1000
800
600
400
200
0
−200

### 1.2.5 2.5 Multivariate Exploration

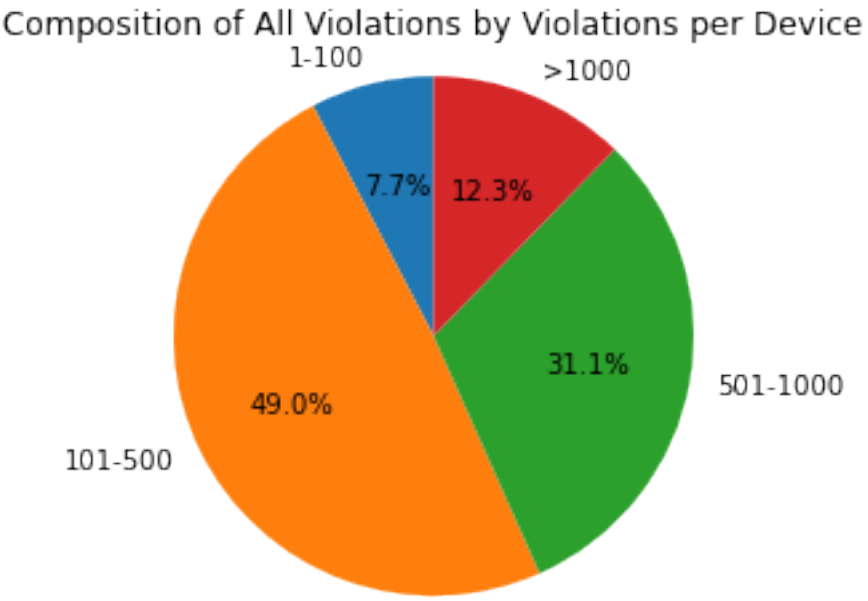
[99]:
```
# If there are 5 devices committed 500 violations each, the sum of violations␣
↪would be 5*500 = 2500, (500, 2500) on the graph
mask_violation = parkingDB['Violation'] == 1
df = parkingDB.loc[mask_violation, 'Device ID'].value_counts()
df_counts = df.value_counts().sort_index()
df_values = df_counts*df_counts.index
df_values.plot.line(figsize=(14, 10))
plt.title('Number of Violations per Device vs Sum of Violations')
plt.ylabel('Sum of Violations')
plt.xlabel('Number of Violations per Device')
```

[99]:
```
Text(0.5, 0, 'Number of Violations per Device')
```



### 1.2.6 2.22 Composition of All Violations by Violations per Device

[102]:
```
# 49% of all violations are committed by devices which committed 101-500 violations each
total_1_100 = df_values.loc[df_values.index <= 100].sum()
total_101_500 = df_values.loc[(101 <= df_values.index) & (df_values.index <= 500)].sum()
total_501_1000 = df_values.loc[(501 <= df_values.index) & (df_values.index <= 1000)].sum()
total_1001 = df_values.loc[1001 <= df_values.index].sum()
labels = '1-100', '101-500', '501-1000', '>1000'
sizes = [total_1_100, total_101_500, total_501_1000, total_1001]

fig1, ax1 = plt.subplots()
ax1.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=90)
ax1.axis('equal')
plt.title('Composition of All Violations by Violations per Device')
plt.show()
```




### 1.2.7 2.3 Individual Area Analysis

From this, I will be looking at all areas of melbourne that have violations, and determine what are the common factors in these violations.

#### 2.3.1 Queensberry

[36]:
```
queensberry = parkingDB[parkingDB["Area"] == "Queensberry"]
queensberry = queensberry[queensberry["Violation"] == 1]
queensberry.head()
```

[36]:

| | Area | Street | Street Limit 1 | Street Limit 2 | \ |
|---|---|---|---|---|---|
| 45 | Queensberry | ERROL STREET | QUEENSBERRY STREET | VICTORIA STREET | |
| 661 | Queensberry | LEVESON STREET | QUEENSBERRY STREET | VICTORIA STREET | |
| 672 | Queensberry | CHETWYND STREET | VICTORIA STREET | STANLEY STREET | |
| 673 | Queensberry | CHETWYND STREET | VICTORIA STREET | STANLEY STREET | |
| 674 | Queensberry | CHETWYND STREET | VICTORIA STREET | STANLEY STREET | |

| | Side Of Street | Street Marker | Arrival Time | \ |
|---|---|---|---|---|
| 45 | 2 | 10174E | 01/10/2011 12:01:29 AM | |
| 661 | 1 | C6340 | 01/10/2011 07:30:00 AM | |
| 672 | 2 | 6034E | 01/10/2011 07:30:07 AM | |
| 673 | 2 | 6044E | 01/10/2011 07:30:07 AM | |
| 674 | 5 | 6047W | 01/10/2011 07:30:07 AM | |

| | Departure Time | Parking Duration (s) | Sign | \ |
|---|---|---|---|---|
| 45 | 01/10/2011 01:33:36 AM | 5527 | LZ 30M M-SUN 00:00-23:59 | |
| 661 | 01/10/2011 12:30:00 PM | 18000 | 4P MTR SAT 7:30-12:30 | |
| 672 | 01/10/2011 12:30:00 PM | 17993 | 1/2P RPA SAT 7:30-12:30 | |
| 673 | 01/10/2011 12:30:00 PM | 17993 | 2P RPA SAT 7:30-12:30 | |
| 674 | 01/10/2011 12:30:00 PM | 17993 | 1/2P RPA SAT 7:30-12:30 | |

| | Violation | Street ID | Device ID |
|---|---|---|---|
| 45 | 1 | 641 | 3215 |
| 661 | 1 | 881 | 3525 |
| 672 | 1 | 192 | 2738 |
| 673 | 1 | 192 | 2733 |
| 674 | 1 | 192 | 2720 |

Queensberry is currently the place with the most violations, and the most parking, so it's important to have a look at the statistics of what is happening to determine the best course of action.

We will start by simplifying the issue down to the the streets with the highest number of infringements.

[50]:
```
print("Streets in this area: ", queensberry['Street'].unique().size)
queensberry['Street'].unique()

#violation count for street
print("Finding the number of times a street has a violation:\n",␣
 ↪queensberry['Street'].value_counts().to_frame())
```

```
Streets in this area:  19
Finding the number of times a street has a violation:
                          Street
```

```
VICTORIA STREET            21761
LEVESON STREET             19703
ERROL STREET               15151
CAPEL STREET               14241
QUEENSBERRY STREET         13056
WILLIAM STREET             12591
CHETWYND STREET            11787
CURZON STREET               6308
HOWARD STREET               6254
ROSSLYN STREET              4911
KING STREET                 4654
EADES PLACE                 4546
WALSH STREET                3831
DRYBURGH STREET             2386
DUDLEY STREET               1885
RODEN STREET                1837
Lt DRYBURGH STREET SOUTH    1411
UNION STREET                 769
ABBOTSFORD STREET            163
```

The street importance will be ranked as the following:

- High Priority
- Medium Priority
- Low Priority

High priority being that these streets that contain more than 10,000 violations, medium priority having between 4,000-10,000 violations and low priority having under 5,000 violations.

This will mean that funding for recommendations will be based on how high priority the streets in these specific areas are (also in comparison to the violation numbers in other areas).

For now, let's only focus on high priority areas:

[60]:
```
queensberryHIGH = queensberry[queensberry["Street"].isin(['VICTORIA STREET',␣
↪'LEVESON STREET', 'ERROL STREET', 'CAPEL STREET', 'QUEENSBERRY STREET',␣
↪'WILLIAM STREET', 'CHETWYND STREET'])]
```

Now based on the high priority streets, let's look at the sides of the street that require attention:

[73]:
```
print("Size of Street ", queensberryHIGH['Side Of Street'].unique().size, "\n")
queensberryHIGH['Side Of Street'].unique()

#violation count for Side of street
print("Finding the number of times a street has a violation:\n",␣
↪queensberryHIGH['Side Of Street'].value_counts().to_frame(), "\n")
print("The Percentages of each Side of Street in a Violation:", "\n",␣
↪queensberryHIGH['Side Of Street'].value_counts(normalize=True)*100)
```

```
Size of Street  5
```

```
Finding the number of times a street has a violation:
    Side Of Street
1           31962
2           30457
5           17519
3           15341
4           13011

The Percentages of each Side of Street in a Violation:
 1    29.515191
2    28.125404
5    16.177856
3    14.166590
4    12.014960
Name: Side Of Street, dtype: float64
```

From this we can therefore see that the violation types are mostly to do with side 1 and 2. Therefore if budget is an issue, recommendations to this budget will pertain to mostly side 1 and 2. If budget will allow, side 5 will also be applicable.

We will now observe the parking signs to determine whether the signs should allow for more time, or if other parking solutions should be employed.

[94]:
```
#make separate dataframe
queensberryHIGH12 = queensberryHIGH[queensberryHIGH["Side Of Street"] <= 2]
```

[146]:
```
print("Counting Signs in Violations \n", queensberryHIGH['Sign'].value_counts().
↪to_frame(), "\n")
```

```
Counting Signs in Violations
                                    Sign
1P TKT A M-F 7:30-18:30            10792
4P MTR M-F 7:30-18:30              10474
1P MTR M-F 7:30-18:30               4684
1P M-F 18:30-23:30                  3904
4P MTR M-SAT 7:30-18:30             3623
2P MTR M-SAT 7:30-18:30             3469
2P MTR RPA M-F 7:30-18:30           3096
2P SUN 7:30-18:30                   2597
1P MTR RPA M-SUN 7:30-23:00         2458
1/2P RPA M-F 7:30-18:30             2114
LZ 15M M-F 7:30-18:30               1474
4P MTR RPA M-F 7:30-18:30           1472
1P RPA M-SUN 7:30-23:00             1445
2P TKT A M-F 7:30-18:30             1424
1/2P RPA M-SAT 7:30-18:30           1286
1P SAT 12:30-23:30                  1246
1P TKT A SAT 7:30-12:30             1218
LZ 30M M-SUN 00:00-23:59            1036
```

```
4P MTR SAT 7:30-12:30                877
1P MTR SAT 7:30-12:30                608
2P SAT 7:30-12:30                    529
1P DIS M-SUN 0:00-23:59              462
2P RPA M-F 7:30-18:30                441
2P MTR RPA SAT 7:30-12:30            377
LZ 30M M-F 7:30-18:30                354
1/2P RPA SAT 7:30-12:30              271
3P A RPE M-F 7:30-18:30              224
4P MTR RPA SAT 7:30-12:30            159
LZ 15M SAT 7:30-12:30                 94
4P DIS M-SUN 7:30-18:30               71
LZ 30M SAT 7:30-12:30                 45
3P A RPE SAT 7:30-12:30               36
S/ No Stop M-Sun 0:00-23-59           30
Temp Sign Plate Sun-Sun 4:00-5:00     20
2P RPA SAT 7:30-12:30                  9
```

Let's use the cutoff measure of 1000 violations. It seems here that the issues with the most violations comes from having parking spaces for only 1 hour. Ocassionally a 2 hour parking space also has violations of over a count of 1000, and 4 hour parking spaces have the second highest infraction rate. Most of these violations come from limits throughout the working day on a Monday to Friday, with the occasional on a Saturday or Sunday.

There is also a fairly normal spread of RPA, ticket and meter signs within this spread. I'll test the parking duration on these signs to determine whether it was a minor overstay or a considerable overstay:

[113]:
```
print("Testing the Worst Ranked Signs\n")

#sign 1
print("SIGN ONE: 1P TKT A M-F 7:30-18:30")
queensberryLOS = queensberryHIGH[queensberryHIGH["Sign"] == "1P TKT A M-F 7:
↪30-18:30"]
print("Value Counts for Parking Duration: ", queensberryLOS['Parking Duration␣
↪(s)'].value_counts().to_frame())
print("The time (hours) of the max length of stay is: ",␣
↪(round(queensberryLOS['Parking Duration (s)'].max()/60)/60), "Hours")
#sign 2
print("SIGN TWO: 4P MTR M-F 7:30-18:30")
queensberryLOS = queensberryHIGH[queensberryHIGH["Sign"] == "4P MTR M-F 7:30-18:
↪30"]
print("Value Counts for Parking Duration: ", queensberryLOS['Parking Duration␣
↪(s)'].value_counts().to_frame())
print("The time (hours) of the max length of stay is: ",␣
↪(round(queensberryLOS['Parking Duration (s)'].max()/60)/60), "Hours")
#sign 3
```

```
print("SIGN THREE: 1/4P M-F 7:30-18:30")
queensberryLOS = queensberryHIGH[queensberryHIGH["Sign"] == "1/4P M-F 7:30-18:
↪30"]
print("Value Counts for Parking Duration: ", queensberryLOS['Parking Duration␣
↪(s)'].value_counts().to_frame())
print("The time (hours) of the max length of stay is: ",␣
↪(round(queensberryLOS['Parking Duration (s)'].max()/60)/60), "Hours")
#sign 4
print("SIGN FOUR: 1/4P 1P MTR RPA M-SUN 7:30-23:00")
queensberryLOS = queensberryHIGH[queensberryHIGH["Sign"] == "1P MTR RPA M-SUN 7:
↪30-23:00"]
print("Value Counts for Parking Duration: ", queensberryLOS['Parking Duration␣
↪(s)'].value_counts().to_frame())
print("The time (hours) of the max length of stay is: ",␣
↪(round(queensberryLOS['Parking Duration (s)'].max()/60)/60), "Hours")
#sign 5
print("SIGN FIVE: 1P MTR M-SAT 7:30-18:30")
queensberryLOS = queensberryHIGH[queensberryHIGH["Sign"] == "1P MTR M-SAT 7:
↪30-18:30"]
print("Value Counts for Parking Duration: ", queensberryLOS['Parking Duration␣
↪(s)'].value_counts().to_frame())
print("The time (hours) of the max length of stay is: ",␣
↪(round(queensberryLOS['Parking Duration (s)'].max()/60)/60), "Hours")
#sign 6
print("SIGN SIX: 1P MTR M-F 7:30-18:30")
queensberryLOS = queensberryHIGH[queensberryHIGH["Sign"] == "1P MTR M-F 7:30-18:
↪30"]
print("Value Counts for Parking Duration: ", queensberryLOS['Parking Duration␣
↪(s)'].value_counts().to_frame())
print("The time (hours) of the max length of stay is: ",␣
↪(round(queensberryLOS['Parking Duration (s)'].max()/60)/60), "Hours")
#sign 7
print("SIGN SEVEN: 1/2P RPA M-F 7:30-18:30")
queensberryLOS = queensberryHIGH[queensberryHIGH["Sign"] == "1/2P RPA M-F 7:
↪30-18:30"]
print("Value Counts for Parking Duration: ", queensberryLOS['Parking Duration␣
↪(s)'].value_counts().to_frame())
print("The time (hours) of the max length of stay is: ",␣
↪(round(queensberryLOS['Parking Duration (s)'].max()/60)/60), "Hours")
#sign 8
print("SIGN EIGHT: 1P M-F 7:30-18:30")
queensberryLOS = queensberryHIGH[queensberryHIGH["Sign"] == "1P M-F 7:30-18:30"]
print("Value Counts for Parking Duration: ", queensberryLOS['Parking Duration␣
↪(s)'].value_counts().to_frame())
print("The time (hours) of the max length of stay is: ",␣
↪(round(queensberryLOS['Parking Duration (s)'].max()/60)/60), "Hours")
```

```
#sign 9
print("SIGN NINE: 2P MTR RPA M-F 7:30-18:30")
queensberryLOS = queensberryHIGH[queensberryHIGH["Sign"] == "2P MTR RPA M-F 7:
 ↪30-18:30"]
print("Value Counts for Parking Duration: ", queensberryLOS['Parking Duration␣
 ↪(s)'].value_counts().to_frame())
print("The time (hours) of the max length of stay is: ",␣
 ↪(round(queensberryLOS['Parking Duration (s)'].max()/60)/60), "Hours")
#sign 10
print("SIGN TEN: 1/4P M-SAT 7:30-18:30")
queensberryLOS = queensberryHIGH[queensberryHIGH["Sign"] == "1/4P M-SAT 7:30-18:
 ↪30"]
print("Value Counts for Parking Duration: ", queensberryLOS['Parking Duration␣
 ↪(s)'].value_counts().to_frame())
print("The time (hours) of the max length of stay is: ",␣
 ↪(round(queensberryLOS['Parking Duration (s)'].max()/60)/60), "Hours")
```

```
Testing the Worst Ranked Signs

SIGN ONE: 1P TKT A M-F 7:30-18:30
Value Counts for Parking Duration:         Parking Duration (s)
39592                     998
39593                     560
39591                      86
3965                       18
3941                       17
…                         …
23592                       1
7224                        1
17467                       1
15420                       1
25613                       1

[6307 rows x 1 columns]
The time (hours) of the max length of stay is:  11.0 Hours
SIGN TWO: 4P MTR M-F 7:30-18:30
Value Counts for Parking Duration:         Parking Duration (s)
39593                    1798
39592                     888
39591                     107
39590                      32
39589                      29
…                         …
16484                       1
18533                       1
30827                       1
16492                       1
```

```
22525                     1

[4936 rows x 1 columns]
The time (hours) of the max length of stay is:  11.0 Hours
SIGN THREE: 1/4P M-F 7:30-18:30
Value Counts for Parking Duration:         Parking Duration (s)
39593                    58
39592                    39
1246                     20
1321                     19
1237                     18
…                       …
10481                     1
2285                      1
6375                      1
4326                      1
5546                      1

[3409 rows x 1 columns]
The time (hours) of the max length of stay is:  11.0 Hours
SIGN FOUR: 1/4P 1P MTR RPA M-SUN 7:30-23:00
Value Counts for Parking Duration:         Parking Duration (s)
55792                   963
55793                   398
55794                     8
6479                      4
4081                      4
…                       …
6986                      1
19276                     1
25421                     1
25425                     1
12286                     1

[3969 rows x 1 columns]
The time (hours) of the max length of stay is:  15.5 Hours
SIGN FIVE: 1P MTR M-SAT 7:30-18:30
Value Counts for Parking Duration:        Parking Duration (s)
4183                    10
3953                     9
3956                     8
4009                     8
4162                     8
…                       …
9469                     1
9477                     1
5383                     1
5387                     1
```

```
4152                     1

[3024 rows x 1 columns]
The time (hours) of the max length of stay is:  11.0 Hours
SIGN SIX: 1P MTR M-F 7:30-18:30
Value Counts for Parking Duration:         Parking Duration (s)
4100                     10
39592                    10
3985                      8
4089                      8
4095                      8
…                        …
5758                      1
10667                     1
7807                      1
5762                      1
5350                      1

[2714 rows x 1 columns]
The time (hours) of the max length of stay is:  11.0 Hours
SIGN SEVEN: 1/2P RPA M-F 7:30-18:30
Value Counts for Parking Duration:         Parking Duration (s)
39593                   718
39592                   349
39591                    30
39590                    11
39589                     8
…                        …
2868                      1
25397                     1
6966                      1
9017                      1
26623                     1

[2815 rows x 1 columns]
The time (hours) of the max length of stay is:  11.0 Hours
SIGN EIGHT: 1P M-F 7:30-18:30
Value Counts for Parking Duration:         Parking Duration (s)
39593                    69
39592                    42
4240                      7
4419                      6
39591                     6
…                        …
5935                      1
7990                      1
5943                      1
5947                      1
```

```
6147                      1

[3099 rows x 1 columns]
The time (hours) of the max length of stay is:  11.0 Hours
SIGN NINE: 2P MTR RPA M-F 7:30-18:30
Value Counts for Parking Duration:         Parking Duration (s)
39593                    985
39592                    427
39591                     43
39590                     15
12496                      3
…                        …
23318                      1
19224                      1
21279                      1
11040                      1
12282                      1

[2523 rows x 1 columns]
The time (hours) of the max length of stay is:  11.0 Hours
SIGN TEN: 1/4P M-SAT 7:30-18:30
Value Counts for Parking Duration:         Parking Duration (s)
39592                     17
1251                      10
1249                       9
1248                       9
39593                      9
…                        …
2302                       1
4349                       1
2951                       1
2298                       1
2047                       1

[2329 rows x 1 columns]
The time (hours) of the max length of stay is:  11.0 Hours
```

A lot of the signs tend to have a really high margin for parking durations, where people stay for almost an entire day and subsequently get fined. This is irrelevant to the types of ticketing machine or charge used.

This is considerable findings. This means that people, on average, during the working day do not stay for only 1 hour. Instead they stay parked in these areas for more than 10 hours. This indicates that possible these individuals park their car for the work day (let's assume a 9-5 role). This takes away 8 hours already. Then let's assume a buffer time of 30 minutes to park and travel to work, and 30 minutes after work to get to the car. This adds an extra hour. so 9/11 hours are accounted for for workers. Another 2 hours can include post-work outings, drinks or shopping.

Now this area is ticketed, which means they are paying for a spot, but are unable to continue paying

every hour or move their car, furthering the point that they are probably working people. Another factor, as seen by graphs previous to this, is that the car IDs are constantly reoccuring in all areas. This inidicates furthermore that these drivers are regulars in the area, and that they park during the morning and afternoon of a work week.

Solutions to this would include:

- Removing the 1 and 1/2 hour limits from Monday to Fridays, but keeping the ticketing system. This allows workers to use this space for their entire shift but allows for the owners of the parking spaces to still make money from the ticketing system. There can be instances where not all issues of this sign have to change, as keeping a few will allow others coming into the city to drop off packages or to complete other quick errands. This will also allow people to use more sustainable, eco friendly options like cycling, taking public transport or using ride-share apps.
- Removing this limit also allows for people to park safely when they are attending a night out in the city, without worrying about parking fines.
- Keeping limits for Sundays and Saturdays, since 'day out' attendance can be supplemented by using public transport.

Now let's introduce medium priority streets around the area of Queensberry:

[137]:
```
queensberryMED = queensberry[queensberry["Street"].isin(['CURZON STREET',␣
↪'HOWARD STREET', 'ROSSLYN STREET', 'KING STREET', 'EADES STREET'])]
print("Counting Signs in Violations \n", queensberryMED['Sign'].value_counts().
↪to_frame(), "\n")
```

```
Counting Signs in Violations
                                    Sign
4P MTR M-F 7:30-18:30               6334
1P A RPE M-F 7:30-18:30             4218
2P M-F 7:30-18:30                   2151
2P M-SUN 7:30-18:30                 2047
4P TKT A M-F 7:30-18:30             1744
1P M-F 7:30-18:30                   1702
1/4P M-F 7:30-18:30                 1199
1P MTR M-F 7:30-18:30                889
2P SAT 7:30-12:30                    450
1P A RPE SAT 7:30-12:30              436
4P MTR SAT 7:30-12:30                382
1P SAT 7:30-12:30                    194
1P RPA M-F 7:30-18:30                162
2P DIS M-SUN 0:00-23:59               81
1P MTR SAT 7:30-12:30                 55
4P TKT A SAT 7:30-12:30               48
1P RPA SAT 7:30-12:30                 19
1/4P SAT 7:30-12:30                   11
Temp Sign Plate Sun-Sun 4:00-5:00      5
```

Let's use the cutoff measure of 1000 violations. This spread is smaller, and the top 7 have a

relatively even mix of parking signs of 1, 2 and 4 hour lengths. All also have a spread of Monday-Friday during the working day, with the exception of the 4th highest sign, 2P M-SUN 7:30-18:30 (with a count of ~2000 violations).

I will go through the same process with these to compare results to streets of the highest priority.

[138]:
```
print("Testing the Worst Ranked Signs\n")
#sign 1
print("SIGN ONE: 4P MTR M-F 7:30-18:30")
queensberryMED = queensberryMED[queensberryMED["Sign"] == "4P MTR M-F 7:30-18:
↪30"]
print("Value Counts for Parking Duration: ", queensberryMED['Parking Duration␣
↪(s)'].value_counts().to_frame())
print("The time (hours) of the max length of stay is: ",␣
↪(round(queensberryMED['Parking Duration (s)'].max()/60)/60), "Hours")
#sign 2-7 have NAs and null values in them
```

```
Testing the Worst Ranked Signs

SIGN ONE: 4P MTR M-F 7:30-18:30
Value Counts for Parking Duration:         Parking Duration (s)
39593                     783
39592                     376
39591                      45
14746                       7
39590                       7
…                         …
23534                       1
29679                       1
17393                       1
15346                       1
28670                       1

[3862 rows x 1 columns]
The time (hours) of the max length of stay is:  11.0 Hours
```

In this instance, many of the top violations had null values in them, and so the time wasn't computed. Going off of the most highly ranked sign (in terms of violations), the outcome is still the same as that of the high priority streets within Queensberry.

### 2.3.2 Princes Theatre

[3]:
```
princesTheatre = parkingDB[parkingDB["Area"] == "Princes Theatre"]
princesTheatre = princesTheatre[princesTheatre["Violation"] == 1]
print(princesTheatre.head())
print("The exact number of infringements in Princes Theatre: ",␣
↪len(princesTheatre))
```

```
                Area            Street  Street Limit 1     Street Limit 2  \
27  Princes Theatre     BOURKE STREET  RUSSELL STREET  EXHIBITION STREET
```

```
32  Princes Theatre  Lt BOURKE STREET  RUSSELL STREET  EXHIBITION STREET
40  Princes Theatre  Lt BOURKE STREET  RUSSELL STREET  EXHIBITION STREET
43  Princes Theatre  Lt BOURKE STREET  RUSSELL STREET  EXHIBITION STREET
60  Princes Theatre     BOURKE STREET  RUSSELL STREET  EXHIBITION STREET

    Side Of Street Street Marker            Arrival Time  \
27               4         2381S  01/10/2011 12:01:24 AM
32               4         2531S  01/10/2011 12:01:25 AM
40               4         2529S  01/10/2011 12:01:28 AM
43               4         2527S  01/10/2011 12:01:29 AM
60               4         2383S  01/10/2011 12:01:56 AM

            Departure Time  Parking Duration (s)                    Sign  \
27  01/10/2011 03:36:42 AM                 12918    P10 M-SUN 0:00-23:59
32  01/10/2011 02:19:48 AM                  8303  1P AOT M-SAT 0:00-7:30
40  01/10/2011 06:21:31 AM                 22803  1P AOT M-SAT 0:00-7:30
43  01/10/2011 02:35:09 AM                  9220  1P AOT M-SAT 0:00-7:30
60  01/10/2011 11:59:00 PM                 86224    P10 M-SUN 0:00-23:59

    Violation  Street ID  Device ID
27          1        123       1401
32          1        907       1294
40          1        907       1305
43          1        907       1291
60          1        123       1392
The exact number of infringements in Princes Theatre:  133812
```

Princess Theatre is the area with the second highest rate of infringements, closely behind Queensberry. The same methodology for Queensberry will be applied to Princes Theatre below:

[9]:
```
print("Streets in this area: ", princesTheatre['Street'].unique().size)
princesTheatre['Street'].unique()

#violation count for street
print("Finding the number of times a street has a violation:\n",␣
 ↪princesTheatre['Street'].value_counts().to_frame())
```

```
Streets in this area:  5
Finding the number of times a street has a violation:
                    Street
LONSDALE STREET     53128
EXHIBITION STREET   33090
BOURKE STREET       26660
Lt BOURKE STREET    12836
SPRING STREET        8098
```

All of these streets have significant levels of violations. The high priority violations will be assessed first. This is inclusive of 4 streets in total.

[10]:
```
princesTheatreHIGH = princesTheatre[princesTheatre["Street"].isin(['LONSDALE␣
↪STREET', 'EXHIBITION STREET', 'BOURKE STREET', 'Lt BOURKE STREET'])]
```

Now let's observe side of streets

[11]:
```
print("Size of Street ", princesTheatreHIGH['Side Of Street'].unique().size,␣
↪"\n")
princesTheatreHIGH['Side Of Street'].unique()

#violation count for Side of street
print("Finding the number of times a street has a violation:\n",␣
↪princesTheatreHIGH['Side Of Street'].value_counts().to_frame(), "\n")
print("The Percentages of each Side of Street in a Violation:", "\n",␣
↪princesTheatreHIGH['Side Of Street'].value_counts(normalize=True)*100)
```

```
Size of Street  5

Finding the number of times a street has a violation:
    Side Of Street
1           62004
4           32988
3           19263
5            6495
2            4964

The Percentages of each Side of Street in a Violation:
 1    49.321476
4    26.240514
3    15.322876
5     5.166489
2     3.948645
Name: Side Of Street, dtype: float64
```

Therefore the biggest worry is side number 1, with an almost 50% share in violations. Side 4 is also considerable, covering 25% of the total violations. Both of these sides will be looked at, and sides 3,5 and 2 will not be considered.

[12]:
```
print("Counting Signs in Violations \n", princesTheatreHIGH['Sign'].
↪value_counts().to_frame(), "\n")
```

```
Counting Signs in Violations
                            Sign
1P MTR M-SAT 7:30-19:30    55692
1P SUN 7:30-18:30          14567
1P TKT A M-SAT 7:30-19:30  11528
2P MTR M-SAT 7:30-20:30     9900
1/2P MTR M-SAT 7:30-19:30   7051
P10 M-SUN 0:00-23:59        5050
1/2P MTR M-F 9:30-19:30     3305
```

```
1P MTR M-F 7:30-16:00       3052
2P SUN 7:30-18:30           2010
1P M-SAT 7:30-19:30         1915
1/4P M-SUN 7:30-18:30       1903
2P DIS M-SUN 0:00-23:59     1170
1P AOT M-SAT 19:30-23:59    1102
1P MTR SAT 7:30-19:30        882
1/2P M-SAT 7:30-19:30        800
1/2P MTR SAT 7:30-1930       727
1P AOT SUN 0:00-23:59        723
1P AOT M-SAT 0:00-7:30       682
1P MTR M-F 9:30-19:30        628
1P SUN 7:30-19:30            516
CW TOW M-F 7:00-9:30         511
CW TOW M-F 16:00-18:30       450
2P DIS AOT 9:30-23:59        363
2P DIS M-SUN 7:30-19:30      319
3P DIS M-SUN 7:30-19:30      291
2P DIS AOT 0:00-23:59        246
2P DIS M-F 7:30-16:00        183
2P DIS AOT 0:00-7:00          62
2P DIS SUN 7:30-18:30         47
2P DIS SAT 7:30-19:30         39
```

Let's use the cutoff measure of 3000 violations. It seems here that the issues with the most violations comes from having parking spaces for only 1 hour. Ocassionally a 2 hour or half hour parking space also has violations of over a count of 1000.

The days are also more spread out, to include Saturdays in most cases, and in some, Sundays also.

[13]:
```
print("Testing the Worst Ranked Signs\n")

#sign 1
print("SIGN ONE: 1P MTR M-SAT 7:30-19:30")
princessLOS = princesTheatreHIGH[princesTheatreHIGH["Sign"] == "1P MTR M-SAT 7:
↪30-19:30"]
print("Value Counts for Parking Duration: ", princessLOS['Parking Duration␣
↪(s)'].value_counts().to_frame())
print("The time (hours) of the max length of stay is: ",␣
↪(round(princessLOS['Parking Duration (s)'].max()/60)/60), "Hours")

#sign 2
print("SIGN TWO: 1P SUN 7:30-18:30")
princessLOS = princesTheatreHIGH[princesTheatreHIGH["Sign"] == "1P SUN 7:30-18:
↪30"]
print("Value Counts for Parking Duration: ", princessLOS['Parking Duration␣
↪(s)'].value_counts().to_frame())
```

```
print("The time (hours) of the max length of stay is: ",␣
↪(round(princessLOS['Parking Duration (s)'].max()/60)/60), "Hours")

#sign 3
print("SIGN THREE: 1P TKT A M-SAT 7:30-19:30")
princessLOS = princesTheatreHIGH[princesTheatreHIGH["Sign"] == "1P TKT A M-SAT␣
↪7:30-19:30"]
print("Value Counts for Parking Duration: ", princessLOS['Parking Duration␣
↪(s)'].value_counts().to_frame())
print("The time (hours) of the max length of stay is: ",␣
↪(round(princessLOS['Parking Duration (s)'].max()/60)/60), "Hours")

#sign 4
print("SIGN FOUR: 2P MTR M-SAT 7:30-20:30")
princessLOS = princesTheatreHIGH[princesTheatreHIGH["Sign"] == "2P MTR M-SAT 7:
↪30-20:30"]
print("Value Counts for Parking Duration: ", princessLOS['Parking Duration␣
↪(s)'].value_counts().to_frame())
print("The time (hours) of the max length of stay is: ",␣
↪(round(princessLOS['Parking Duration (s)'].max()/60)/60), "Hours")

#sign 5
print("SIGN FIVE: 1/2P MTR M-SAT 7:30-19:30")
princessLOS = princesTheatreHIGH[princesTheatreHIGH["Sign"] == "1/2P MTR M-SAT␣
↪7:30-19:30"]
print("Value Counts for Parking Duration: ", princessLOS['Parking Duration␣
↪(s)'].value_counts().to_frame())
print("The time (hours) of the max length of stay is: ",␣
↪(round(princessLOS['Parking Duration (s)'].max()/60)/60), "Hours")

#sign 6
print("SIGN SIX: P10 M-SUN 0:00-23:59")
princessLOS = princesTheatreHIGH[princesTheatreHIGH["Sign"] == "P10 M-SUN 0:
↪00-23:59"]
print("Value Counts for Parking Duration: ", princessLOS['Parking Duration␣
↪(s)'].value_counts().to_frame())
print("The time (hours) of the max length of stay is: ",␣
↪(round(princessLOS['Parking Duration (s)'].max()/60)/60), "Hours")

#sign 7
print("SIGN SEVEN: 1/2P MTR M-F 9:30-19:30")
princessLOS = princesTheatreHIGH[princesTheatreHIGH["Sign"] == "1/2P MTR M-F 9:
↪30-19:30"]
print("Value Counts for Parking Duration: ", princessLOS['Parking Duration␣
↪(s)'].value_counts().to_frame())
```

```
print("The time (hours) of the max length of stay is: ",␣
 ↪(round(princessLOS['Parking Duration (s)'].max()/60)/60), "Hours")

#sign 8
print("SIGN EIGHT: 1P MTR M-F 7:30-16:00")
princessLOS = princesTheatreHIGH[princesTheatreHIGH["Sign"] == "1P MTR M-F 7:
 ↪30-16:00"]
print("Value Counts for Parking Duration: ", princessLOS['Parking Duration␣
 ↪(s)'].value_counts().to_frame())
print("The time (hours) of the max length of stay is: ",␣
 ↪(round(princessLOS['Parking Duration (s)'].max()/60)/60), "Hours")
```

```
Testing the Worst Ranked Signs

SIGN ONE: 1P MTR M-SAT 7:30-19:30
Value Counts for Parking Duration:         Parking Duration (s)
43192                    336
43193                    304
3941                      52
3911                      51
3907                      50
…                        …
17287                      1
27522                      1
11050                      1
23332                      1
10059                      1

[9439 rows x 1 columns]
The time (hours) of the max length of stay is:  12.0 Hours
SIGN TWO: 1P SUN 7:30-18:30
Value Counts for Parking Duration:         Parking Duration (s)
39592                     97
39593                     71
3973                      20
4036                      18
3960                      17
…                        …
13194                      1
23439                      1
9104                       1
11153                      1
9568                       1

[5169 rows x 1 columns]
The time (hours) of the max length of stay is:  11.0 Hours
SIGN THREE: 1P TKT A M-SAT 7:30-19:30
```

```
Value Counts for Parking Duration:         Parking Duration (s)
43192                     66
43193                     32
3903                      17
4078                      15
3948                      15
…                         …
5779                       1
7455                       1
5414                       1
7826                       1
16384                      1

[5200 rows x 1 columns]
The time (hours) of the max length of stay is:  12.0 Hours
SIGN FOUR: 2P MTR M-SAT 7:30-20:30
Value Counts for Parking Duration:         Parking Duration (s)
46792                     89
46793                     42
46791                     15
7578                      12
7538                      12
…                         …
15156                      1
11086                      1
9039                       1
15188                      1
12286                      1

[4920 rows x 1 columns]
The time (hours) of the max length of stay is:  13.0 Hours
SIGN FIVE: 1/2P MTR M-SAT 7:30-19:30
Value Counts for Parking Duration:         Parking Duration (s)
43200                     48
2109                      11
2123                      11
2171                      10
2174                      10
…                         …
10904                      1
2859                       1
8131                       1
8135                       1
2423                       1

[3451 rows x 1 columns]
The time (hours) of the max length of stay is:  12.0 Hours
SIGN SIX: P10 M-SUN 0:00-23:59
```

```
Value Counts for Parking Duration:        Parking Duration (s)
921                      14
938                      13
926                      11
939                      11
942                      10
…                        …
9469                      1
7418                      1
4083                      1
1269                      1
9600                      1

[2691 rows x 1 columns]
The time (hours) of the max length of stay is:  23.966666666666665 Hours
SIGN SEVEN: 1/2P MTR M-F 9:30-19:30
Value Counts for Parking Duration:         Parking Duration (s)
2220                       9
2179                       9
2173                       8
2109                       8
2414                       8
…                          …
3561                       1
3557                       1
5604                       1
3555                       1
22525                      1

[1927 rows x 1 columns]
The time (hours) of the max length of stay is:  10.0 Hours
SIGN EIGHT: 1P MTR M-F 7:30-16:00
Value Counts for Parking Duration:         Parking Duration (s)
3909                       8
3961                       7
3990                       6
30592                      6
4139                       6
…                          …
6372                       1
9758                       1
5656                       1
7701                       1
6143                       1

[2022 rows x 1 columns]
The time (hours) of the max length of stay is:  8.5 Hours
```

These results somewhat mimic what was happening in the queensberry instance, where working class individuals were parking for long periods of time during the working day. However another interesting observation has been made, specifically in the case of 'P10 M-SUN 0:00-23:59', where cases stayed for almost a full day. This could be indicative of those during the night life of the city, staying for longer periods of time, drinking and subsequently not being able to drive back home.

This is one of multiple scenarios that could occur. But this is deduced given that this sign also applies to the Sunday, whereas most other signs do not. I'll compare these broad values with that of the

### 2.3.3 Southbank

[4]:
```
southbank = parkingDB[parkingDB["Area"] == "Southbank"]
southbank = southbank[southbank["Violation"] == 1]
print(southbank.head())
print("The exact number of infringements in Southbank: ", len(southbank))
```

```
             Area        Street       Street Limit 1 Street Limit 2  \
453929  Southbank  DODDS STREET  SOUTHBANK BOULEVARD   GRANT STREET
455210  Southbank  DODDS STREET  SOUTHBANK BOULEVARD   GRANT STREET
456678  Southbank  DODDS STREET  SOUTHBANK BOULEVARD   GRANT STREET
457764  Southbank  DODDS STREET  SOUTHBANK BOULEVARD   GRANT STREET
458316  Southbank  DODDS STREET  SOUTHBANK BOULEVARD   GRANT STREET

        Side Of Street Street Marker             Arrival Time  \
453929               5         8311W  14/10/2011 08:42:18 AM
455210               2         8342E  14/10/2011 09:07:26 AM
456678               2         8340E  14/10/2011 09:31:00 AM
457764               2         8332E  14/10/2011 09:49:14 AM
458316               2         8338E  14/10/2011 09:57:27 AM

                Departure Time  Parking Duration (s)                     Sign  \
453929  14/10/2011 12:35:33 PM                 13995  2P TKT A M-F 7:30-18:30
455210  14/10/2011 06:30:00 PM                 33754  3P TKT A M-F 7:30-18:30
456678  14/10/2011 12:53:12 PM                 12132  3P TKT A M-F 7:30-18:30
457764  14/10/2011 01:09:33 PM                 12019  3P TKT A M-F 7:30-18:30
458316  14/10/2011 02:00:43 PM                 14596  3P TKT A M-F 7:30-18:30

        Violation  Street ID  Device ID
453929          1        591       4465
455210          1        591       4661
456678          1        591       4639
457764          1        591       4705
458316          1        591       4623
The exact number of infringements in Southbank:  115754
```

[16]:
```
print("Streets in this area: ", southbank['Street'].unique().size)
southbank['Street'].unique()
```

```
#violation count for street
print("Finding the number of times a street has a violation:\n",␣
 ↪southbank['Street'].value_counts().to_frame())
```

```
Streets in this area:  13
Finding the number of times a street has a violation:
                     Street
COVENTRY STREET       40408
KAVANAGH STREET       14051
GRANT STREET          13287
DORCAS STREET         12146
SOUTHBANK BOULEVARD    8921
STURT STREET           6289
DODDS STREET           5340
MILES STREET           4054
FAWKNER STREET         3949
WELLS STREET           3123
BALSTON STREET         1564
ST KILDA ROAD          1547
CITY ROAD              1075
```

The top 4 streets with violations are: Coventry, Kavanagh, Grant and Dorcas street. These will be teh top analysed streets to reduce violation numbers.

[29]:
```
southbankHIGH = southbank[southbank["Street"].isin(['CONVENTRY STREET',␣
 ↪'KAVANAGH STREET', 'GRANT STREET', 'DORCAS STREET'])]

print("Size of Street ", southbankHIGH['Side Of Street'].unique().size, "\n")
southbankHIGH['Side Of Street'].unique()

#violation count for Side of street
print("Finding the number of times a street has a violation:\n",␣
 ↪southbankHIGH['Side Of Street'].value_counts().to_frame(), "\n")
print("The Percentages of each Side of Street in a Violation:", "\n",␣
 ↪southbankHIGH['Side Of Street'].value_counts(normalize=True)*100)

southbankHIGH35 = southbankHIGH[southbankHIGH["Side Of Street"].isin([3, 5])]
```

```
Size of Street  4

Finding the number of times a street has a violation:
    Side Of Street
3           22214
5            8060
2            5991
4            3219

The Percentages of each Side of Street in a Violation:
```

```
 3    56.260764
5    20.413332
2    15.173235
4     8.152669
Name: Side Of Street, dtype: float64
```

It appears as there are only 4 counts in side of street. The main areas of focus include sides 3 and 5, given side 3 counts for more than 50% of the violations. When incorporating side 5, we pass a violation rate of 75% and so this will also be included.

[30]:
```
print("Counting Signs in Violations \n", southbankHIGH35['Sign'].value_counts().
↪to_frame(), "\n")
```

```
Counting Signs in Violations
                                        Sign
1P TKT A RPA M-F 7:30-18:30            18326
2P TKT A M-F 7:30-18:30                 8570
1P TKT A M-F 7:30-18:30                 7981
3P TKT A M-SAT 7:30-18:30               5162
2P RPA M-F 7:30-18:30                   3305
2P M-F 7:30-18:30                       2645
2P TKT A M-SAT 7:30-18:30               2244
1P MTR M-F 7:30-19:30                   2205
2P TKT A RPA M-F 7:30-18:30             2157
LZ 15M M-F 7:30-18:30                   2114
2P RPA S-S 7:30-23:00                   1826
1P MTR M-SAT 7:30-19:30                 1780
LZ 30M M-F 7:30-18:30                   1717
2P MTR RPA M-F 7:30-18:30               1591
P/ 15 M-SUN 00:00-23:59                 1064
2P SUN 7:30-18:30                       1014
4P DIS ONLY M-SUN                        967
3P SUN 7:30-18:30                        949
2P MTR M-F 7:30-18:30                    815
P/ 10 M-SUN 0:00-11:59                   781
3P TKT A M-F 7:30-18:30                  757
LZ 15M M-SUN 0:00-23:59                  679
S/ No Stop M-Sun 0:00-23-59              527
P5 M-F 7:30-18:30                        466
P/15 M-F 7:30-9:00                       454
2P TKT A M-SAT 7:30-20:30                396
2P MTR M-F 9:00-16:30                    370
2P A PRE SAT 7:30-12:30                  346
P/ 15 M-F 16:30-18:30                    333
1P MTR SAT 7:30-19:30                    253
2P TKT A SAT 7:30-12:30                  221
1P MTR M-F 7:30-16:00                    181
2P SUN 7:30-23:00                        151
```

```
2P DIS M-F 7:30-19:30                151
1P TKT A SAT 7:30-12:30              120
LZ 30M SAT 7:30-12:30                119
1P SAT 7:30-19:30                    118
2P M-F 20:30-23:00                    84
Temp Sign Plate Sun-Sun 4:00-5:00     38
2P DIS M-SAT 7:30-18:30               29
2P SUN 7:30-18:30 - old               29
Temp Sign Plate SUN 8:30-5:30          8
2P RPA SAT 7:30-23:00                  6
```

Again, it seems like the majority of high violation cases are those that are between normal working hours and have a limit of 2 hours or less.

[31]:
```
print("Testing the Worst Ranked Signs\n")

#sign 1
print("SIGN ONE: 1P TKT A RPA M-F 7:30-18:30")
southbankLOS = southbankHIGH35[southbankHIGH35["Sign"] == "1P TKT A RPA M-F 7:
 ↪30-18:30"]
print("Value Counts for Parking Duration: ", southbankLOS['Parking Duration␣
 ↪(s)'].value_counts().to_frame())
print("The time (hours) of the max length of stay is: ",␣
 ↪(round(southbankLOS['Parking Duration (s)'].max()/60)/60), "Hours")
```

```
Testing the Worst Ranked Signs

SIGN ONE: 1P TKT A RPA M-F 7:30-18:30
Value Counts for Parking Duration:         Parking Duration (s)
39593                   2369
39592                   1123
39591                    107
39590                     36
39589                     12
…                        …
14401                      1
6213                       1
32838                      1
30793                      1
4094                       1

[11020 rows x 1 columns]
The time (hours) of the max length of stay is:  11.0 Hours
```

This also confirms like in the other cases, where the violations are for a full work day give or take another couple of hours. This will only be tested on teh above 10,000 violation figures as it is more of a confirmation about the pattern being found in many of these areas.

### 2.3.4 Hyatt

[5]:
```
hyatt = parkingDB[parkingDB["Area"] == "Hyatt"]
hyatt = hyatt[hyatt["Violation"] == 1]
print(hyatt.head())
print("The exact number of infringements around the Hyatt: ", len(hyatt))
```

```
      Area             Street     Street Limit 1   Street Limit 2  \
19   Hyatt      SPRING STREET      FLINDERS LANE  FLINDERS STREET
35   Hyatt  Lt COLLINS STREET  EXHIBITION STREET    SPRING STREET
98   Hyatt    FLINDERS STREET  EXHIBITION STREET    SPRING STREET
106  Hyatt    FLINDERS STREET  EXHIBITION STREET    SPRING STREET
302  Hyatt    FLINDERS STREET  EXHIBITION STREET    SPRING STREET

     Side Of Street Street Marker            Arrival Time  \
19                2            2E  01/10/2011 12:01:20 AM
35                4         2133S  01/10/2011 12:01:25 AM
98                3         1596N  01/10/2011 12:36:28 AM
106               3         1596N  01/10/2011 12:44:02 AM
302               3         1596N  01/10/2011 04:50:52 AM

             Departure Time  Parking Duration (s)  \
19   01/10/2011 11:59:00 PM                 86260
35   01/10/2011 05:43:12 AM                 20507
98   01/10/2011 12:39:33 AM                   185
106  01/10/2011 12:50:16 AM                   374
302  01/10/2011 04:55:55 AM                   303

                           Sign  Violation  Street ID  Device ID
19       2P DIS M-SUN 0:00-23:59          1       1288          7
35       2P DIS M-SUN 0:00-23:59          1        911       1203
98   S/ No Stop M-Sun 0:00-23-59          1        670       1237
106  S/ No Stop M-Sun 0:00-23-59          1        670       1237
302  S/ No Stop M-Sun 0:00-23-59          1        670       1237
The exact number of infringements around the Hyatt:  110799
```

[43]:
```
print("Streets in this area: ", hyatt['Street'].unique().size)
hyatt['Street'].unique()

#violation count for street
print("Finding the number of times a street has a violation:\n",␣
 ↪hyatt['Street'].value_counts().to_frame())
```

```
Streets in this area:  6
Finding the number of times a street has a violation:
                    Street
EXHIBITION STREET   47398
SPRING STREET       16587
COLLINS STREET      14656
```

```
Lt COLLINS STREET   14552
FLINDERS LANE       11729
FLINDERS STREET      5877
```

Given this, the high priority streets include:

- Exhibition
- Sprint
- Collins
- Lt Collins Street
- Flinders Lane

[45]:
```
hyattHIGH = hyatt[hyatt["Street"].isin(['EXHIBITION STREET', 'SPRING STREET',␣
 ↪'COLLINS STREET', 'Lt COLLINS STREET', 'FLINDERS LANE'])]

print("Size of Street ", hyattHIGH['Side Of Street'].unique().size, "\n")
hyattHIGH['Side Of Street'].unique()

#violation count for Side of street
print("Finding the number of times a street has a violation:\n",␣
 ↪hyattHIGH['Side Of Street'].value_counts().to_frame(), "\n")
print("The Percentages of each Side of Street in a Violation:", "\n",␣
 ↪hyattHIGH['Side Of Street'].value_counts(normalize=True)*100)
```

```
Size of Street  5

Finding the number of times a street has a violation:
    Side Of Street
1           35676
4           23874
5           17914
3           17063
2           10395

The Percentages of each Side of Street in a Violation:
 1    34.002402
4    22.754046
5    17.073636
3    16.262557
2     9.907360
Name: Side Of Street, dtype: float64
```

Given these results, we will be using the results from areas 1, 4 and 5

[46]:
```
hyattHIGH145 = hyattHIGH[hyattHIGH["Side Of Street"].isin([1, 4, 5])]


print("Counting Signs in Violations \n", hyattHIGH145['Sign'].value_counts().
 ↪to_frame(), "\n")
```

```
Counting Signs in Violations
                                Sign
1P MTR M-SAT 7:30-19:30        47566
1P SUN 7:30-18:30               9191
1P TKT A M-SAT 7:30-19:30       7865
1/2P MTR M-SAT 7:30-19:30       3720
LZ 15M M-F 7:30-19:30           2712
2P DIS M-SUN 0:00-23:59         2347
1P M-SAT 7:30-19:30             2020
LZ 15M M-SUN 7:30-19:30          870
LZ 30M M-F 7:30-19:30            508
1P SAT 7:30-19:30                411
4P DIS ONLY M-SUN                198
1P SUN 7:00-18:30                 56
```

[47]:
```
print("Testing the Worst Ranked Signs\n")

#sign 1
print("SIGN ONE: 1P MTR M-SAT 7:30-19:30")
hyattLOS = hyattHIGH145[hyattHIGH145["Sign"] == "1P MTR M-SAT 7:30-19:30"]
print("Value Counts for Parking Duration: ", hyattLOS['Parking Duration (s)'].
 ↪value_counts().to_frame())
print("The time (hours) of the max length of stay is: ",␣
 ↪(round(hyattLOS['Parking Duration (s)'].max()/60)/60), "Hours")
```

```
Testing the Worst Ranked Signs

SIGN ONE: 1P MTR M-SAT 7:30-19:30
Value Counts for Parking Duration:         Parking Duration (s)
43193                                  296
43192                                  215
3941                                    49
4057                                    46
3985                                    45
…                                      …
28418                                    1
24292                                    1
26339                                    1
32480                                    1
11653                                    1

[8998 rows x 1 columns]
The time (hours) of the max length of stay is:  12.0 Hours
```

From this we can see that the most problematic carparking spot is again a “weekday” slot of 1 hour, with an average stay of around 12 hours. This further proves the same explanation of the previous areas.

### 2.3.5 City Square

[6]:
```
citySquare = parkingDB[parkingDB["Area"] == "City Square"]
citySquare = citySquare[citySquare["Violation"] == 1]
print(citySquare.head())
print("The exact number of infringements around the City Square: ",␣
↪len(citySquare))
print("")

#STREETS
print("Streets in this area: ", citySquare['Street'].unique().size)
citySquare['Street'].unique()

print("Finding the number of times a street has a violation:\n",␣
↪citySquare['Street'].value_counts().to_frame())
citySquareHIGH = citySquare[citySquare["Street"].isin(['RUSSELL STREET',␣
↪'COLLINS STREET', 'ELIZABETH STREET'])]
print("")

#STREET SIDES
print("Counts of Side of Street:", citySquareHIGH['Side Of Street'].unique().
↪size, "\n")
citySquareHIGH['Side Of Street'].unique()

print("Finding the number of times a street has a violation:\n",␣
↪citySquareHIGH['Side Of Street'].value_counts().to_frame(), "\n")
print("The Percentages of each Side of Street in a Violation:", "\n",␣
↪citySquareHIGH['Side Of Street'].value_counts(normalize=True)*100)

citySquareHIGH235 = citySquareHIGH[citySquareHIGH["Side Of Street"].isin([2, 3,␣
↪5])]
print("")

#SIGNS
print("Counting Signs in Violations \n", citySquareHIGH235['Sign'].
↪value_counts().to_frame(), "\n")

print("Testing the Worst Ranked Signs\n")
#sign 1
print("SIGN ONE: 1/2P MTR M-SAT 7:30-19:30")
citySquareLOS = citySquareHIGH235[citySquareHIGH235["Sign"] == "1/2P MTR M-SAT␣
↪7:30-19:30"]
print("Value Counts for Parking Duration: ", citySquareLOS['Parking Duration␣
↪(s)'].value_counts().to_frame())
print("The time (hours) of the max length of stay is: ",␣
↪(round(citySquareLOS['Parking Duration (s)'].max()/60)/60), "Hours")
print("")
```

```
#sign 2
print("SIGN ONE: 1/2P M-SAT 7:30-19:30")
citySquareLOS = citySquareHIGH235[citySquareHIGH235["Sign"] == "1/2P M-SAT 7:
 ↪30-19:30"]
print("Value Counts for Parking Duration: ", citySquareLOS['Parking Duration␣
 ↪(s)'].value_counts().to_frame())
print("The time (hours) of the max length of stay is: ",␣
 ↪(round(citySquareLOS['Parking Duration (s)'].max()/60)/60), "Hours")
print("")
```

```
           Area          Street    Street Limit 1   Street Limit 2  \
30  City Square  RUSSELL STREET    COLLINS STREET    FLINDERS LANE
34  City Square   FLINDERS LANE   SWANSTON STREET   RUSSELL STREET
56  City Square   FLINDERS LANE  ELIZABETH STREET  SWANSTON STREET
61  City Square  RUSSELL STREET    COLLINS STREET    FLINDERS LANE
62  City Square   FLINDERS LANE  ELIZABETH STREET  SWANSTON STREET

    Side Of Street Street Marker            Arrival Time  \
30               1          C650  01/10/2011 12:01:24 AM
34               4         1767S  01/10/2011 12:01:25 AM
56               4         1773S  01/10/2011 12:01:35 AM
61               1          C652  01/10/2011 12:02:02 AM
62               4         1777S  01/10/2011 12:04:46 AM

            Departure Time  Parking Duration (s)                      Sign  \
30  01/10/2011 12:03:07 AM                   103  S/ No Stop AOT 0:00-7:30
34  01/10/2011 01:10:28 AM                  4143      P10 M-SUN 0:00-23:59
56  01/10/2011 05:38:30 AM                 20215   2P DIS M-SUN 0:00-23:59
61  01/10/2011 12:04:28 AM                   146  S/ No Stop AOT 0:00-7:30
62  01/10/2011 10:54:39 AM                 38993   2P DIS M-SUN 0:00-23:59

    Violation  Street ID  Device ID
30          1       1221        479
34          1        669       1417
56          1        669       1366
61          1       1221        478
62          1        669       1419
The exact number of infringements around the City Square:  97001

Streets in this area:  5
Finding the number of times a street has a violation:
                  Street
RUSSELL STREET     32595
COLLINS STREET     31249
ELIZABETH STREET   25455
FLINDERS LANE       4835
FLINDERS STREET     2867
```

```
Counts of Side of Street: 5

Finding the number of times a street has a violation:
    Side Of Street
3           25794
2           21640
5           19614
1           16796
4            5455

The Percentages of each Side of Street in a Violation:
 3    28.884982
2    24.233194
5    21.964412
1    18.808721
4     6.108691
Name: Side Of Street, dtype: float64

Counting Signs in Violations
                                     Sign
1/2P MTR M-SAT 7:30-19:30           26751
1/2P M-SAT 7:30-19:30               24770
1P SUN 7:30-18:30                    5507
P/(No Parking) M-SUN 0:00 - 23:59    4105
1P MTR M-SAT 7:30-19:30              2847
P/ 5 M-SUN 0:00-23:59                1622
1/2 DIS M-F 7:30-1930                 772
2P DIS M-SUN 0:00-23:59               398
4P DIS ONLY M-SUN                     181
2P DIS AOT 19:30-23:59                 49
2P DIS AOT 12:00-7:30                  25
2P DIS AOT 0:00-23:59                  15
Temp Sign Plate Sun-Sun 4:00-5:00       3
2P DIS 00:00-7.30                       2
P5 M-SUN 0:00-23:59                     1

Testing the Worst Ranked Signs

SIGN ONE: 1/2P MTR M-SAT 7:30-19:30
Value Counts for Parking Duration:         Parking Duration (s)
43193                    232
43192                     94
2121                      32
2123                      31
2169                      31
…                        …
4484                       1
```

```
10627                      1
8578                       1
14721                      1
10259                      1

[6697 rows x 1 columns]
The time (hours) of the max length of stay is:  12.0 Hours

SIGN ONE: 1/2P M-SAT 7:30-19:30
Value Counts for Parking Duration:         Parking Duration (s)
43193                    100
43192                     62
2166                      39
2108                      33
2174                      32
…                        …
7974                       1
5911                       1
12682                      1
16146                      1
5862                       1

[5854 rows x 1 columns]
The time (hours) of the max length of stay is:  12.0 Hours
```

### 2.4.5 Victoria Market

[7]:
```
vicMarket = parkingDB[parkingDB["Area"] == "Victoria Market"]
vicMarket = vicMarket[vicMarket["Violation"] == 1]
print(vicMarket.head())
print("The exact number of infringements around Victoria Market: ",␣
 ↪len(vicMarket))
print("")

#STREETS
print("Streets in this area: ", vicMarket['Street'].unique().size)
vicMarket['Street'].unique()

print("Finding the number of times a street has a violation:\n",␣
 ↪vicMarket['Street'].value_counts().to_frame())
vicMarketHIGH = vicMarket[vicMarket["Street"].isin(['QUEEN STREET', 'FRANKLIN␣
 ↪STREET', 'THERRY STREET'])]
print("")

#STREET SIDES
print("Counts of Side of Street:", vicMarketHIGH['Side Of Street'].unique().
 ↪size, "\n")
```

```
vicMarketHIGH['Side Of Street'].unique()

print("Finding the number of times a street has a violation:\n",␣
 ↪vicMarketHIGH['Side Of Street'].value_counts().to_frame(), "\n")
print("The Percentages of each Side of Street in a Violation:", "\n",␣
 ↪vicMarketHIGH['Side Of Street'].value_counts(normalize=True)*100)

vicMarketHIGH145 = vicMarketHIGH[vicMarketHIGH["Side Of Street"].isin([1, 4,␣
 ↪5])]
print("")

#SIGNS
print("Counting Signs in Violations \n", vicMarketHIGH145['Sign'].
 ↪value_counts().to_frame(), "\n")

print("Testing the Worst Ranked Signs\n")
#sign 1
print("SIGN ONE: 2P MTR M-SAT 7:30-20:30")
vicMarketLOS = vicMarketHIGH145[vicMarketHIGH145["Sign"] == "2P MTR M-SAT 7:
 ↪30-20:30"]
print("Value Counts for Parking Duration: ", vicMarketLOS['Parking Duration␣
 ↪(s)'].value_counts().to_frame())
print("The time (hours) of the max length of stay is: ",␣
 ↪(round(vicMarketLOS['Parking Duration (s)'].max()/60)/60), "Hours")
print("")
#sign 2
print("SIGN TWO: 1/4P M-SAT 7:30-19:30")
vicMarketLOS = vicMarketHIGH145[vicMarketHIGH145["Sign"] == "1/4P M-SAT 7:30-19:
 ↪30"]
print("Value Counts for Parking Duration: ", vicMarketLOS['Parking Duration␣
 ↪(s)'].value_counts().to_frame())
print("The time (hours) of the max length of stay is: ",␣
 ↪(round(vicMarketLOS['Parking Duration (s)'].max()/60)/60), "Hours")
print("")
#sign 2
print("SIGN THREE: 1P MTR M-SAT 6:00-19:30")
vicMarketLOS = vicMarketHIGH145[vicMarketHIGH145["Sign"] == "1P MTR M-SAT 6:
 ↪00-19:30"]
print("Value Counts for Parking Duration: ", vicMarketLOS['Parking Duration␣
 ↪(s)'].value_counts().to_frame())
print("The time (hours) of the max length of stay is: ",␣
 ↪(round(vicMarketLOS['Parking Duration (s)'].max()/60)/60), "Hours")
print("")
```

```
                Area        Street   Street Limit 1   Street Limit 2  \
301  Victoria Market  QUEEN STREET  VICTORIA STREET    THERRY STREET
```

```
330  Victoria Market  QUEEN STREET  VICTORIA STREET    THERRY STREET
359  Victoria Market  QUEEN STREET    THERRY STREET  FRANKLIN STREET
362  Victoria Market  QUEEN STREET  VICTORIA STREET    THERRY STREET
365  Victoria Market  QUEEN STREET  VICTORIA STREET    THERRY STREET

     Side Of Street Street Marker            Arrival Time  \
301               2         4982E  01/10/2011 04:50:24 AM
330               2         4986E  01/10/2011 05:30:27 AM
359               5         4909W  01/10/2011 06:00:07 AM
362               1         C4940  01/10/2011 06:00:07 AM
365               5         5011W  01/10/2011 06:00:07 AM

             Departure Time  Parking Duration (s)  \
301  01/10/2011 05:29:19 AM                  2335
330  01/10/2011 05:34:16 AM                   229
359  01/10/2011 07:42:19 AM                  6132
362  01/10/2011 09:56:38 AM                 14191
365  01/10/2011 09:21:28 AM                 12081

                            Sign  Violation  Street ID  Device ID
301  S/ No Stop M-Sun 0:00-23-59          1       1171       3071
330  S/ No Stop M-Sun 0:00-23-59          1       1171       2957
359    1P TKT A M-SAT 6:00-19:30          1       1171       3105
362      1P MTR M-SAT 6:00-19:30          1       1171       3128
365        1P MTR SAT 6:00-16:00          1       1171       3140
The exact number of infringements around Victoria Market:  96227

Streets in this area:  6
Finding the number of times a street has a violation:
                  Street
QUEEN STREET       36746
FRANKLIN STREET    31228
THERRY STREET      20913
ELIZABETH STREET    3469
VICTORIA STREET     1949
PEEL STREET         1922

Counts of Side of Street: 5

Finding the number of times a street has a violation:
    Side Of Street
1           37130
4           23369
5           18531
2            6215
3            3642

The Percentages of each Side of Street in a Violation:
```

```
 1    41.772138
4    26.290684
5    20.847818
2     6.992024
3     4.097337
Name: Side Of Street, dtype: float64

Counting Signs in Violations
                                          Sign
2P MTR M-SAT 7:30-20:30                  13518
1/4P M-SAT 7:30-19:30                    12364
1P MTR M-SAT 6:00-19:30                  11549
1P TKT A M-SAT 6:00-19:30                 8489
1P MTR M-SAT 7:30-19:30                   5670
2P SUN 7:30-18:30                         5401
LZ 15M M-SAT 7:30-19:30                   3782
S/ No Stop SUN 6:00-18:30                 3620
2P TKT A M-SAT 7:30-20:30                 3155
1P MTR M-THU 8:00-14:30                   2809
1/2P MTR M-SAT 7:30-19:30                 2252
1P MTR SAT 6:00-16:00                     1670
P10 M-SAT 7:30-19:30                       901
1P TKT A M-SAT 6:30-19:30                  892
2P SUN 6:00-18:30                          841
2P DIS M-SUN 0:00-23:59                    515
LZ 30M M-SUN 00:00-23:59                   402
LZ 30M M-F 7:30-19:30                      305
Permit Zone Auth Veh M-THU 14:30-23:59     277
4P DIS ONLY M-SUN                          205
Permit Zone Auth Veh M-F 0:00-8:00         159
Permit Zone Auth Veh SAT 16:00-23:59       129
Permit Zone Auth Veh Fri 18:00-23:59        53
Permit Zone Auth Veh S-S 0:00-6:00          40
2P SAT 7:30-19:30                           29
Permit Zone Auth Veh SUN 18:30-23:59         3

Testing the Worst Ranked Signs

SIGN ONE: 2P MTR M-SAT 7:30-20:30
Value Counts for Parking Duration:         Parking Duration (s)
46792                   1494
46793                    515
46791                    132
46790                     52
46789                     34
…                        …
20828                      1
40332                      1
```

```
10838                     1
12640                     1
8196                      1

[5701 rows x 1 columns]
The time (hours) of the max length of stay is:  13.0 Hours

SIGN TWO: 1/4P M-SAT 7:30-19:30
Value Counts for Parking Duration:         Parking Duration (s)
43193                    34
1326                     24
1240                     23
1246                     23
1319                     23
…                       …
3475                      1
7577                      1
23470                     1
19867                     1
12238                     1

[3395 rows x 1 columns]
The time (hours) of the max length of stay is:  12.0 Hours

SIGN THREE: 1P MTR M-SAT 6:00-19:30
Value Counts for Parking Duration:         Parking Duration (s)
48593                   123
48592                   102
4042                     12
3951                     11
4372                     11
…                       …
21737                     1
18068                     1
5782                      1
11929                     1
16384                     1

[6848 rows x 1 columns]
The time (hours) of the max length of stay is:  13.5 Hours
```

Again, the same pattern goes in terms of timing, where drivers are parking during working hours in timed, mostly paid spots that have a limit of maximum 2 hours.

The street sides that also have the most violations are 1, 4 and 5.

### 2.4.5 The Mac

[8]:
```
theMac = parkingDB[parkingDB["Area"] == "The Mac"]
theMac = theMac[theMac["Violation"] == 1]
print(theMac.head())
print("The exact number of infringements around the Mac: ", len(theMac))
print("")

#STREETS
print("Streets in this area: ", theMac['Street'].unique().size)
theMac['Street'].unique()

print("Finding the number of times a street has a violation:\n",␣
 ↪theMac['Street'].value_counts().to_frame())
theMacHIGH = theMac[theMac["Street"].isin(['FRANKLIN STREET', 'LA TROBE␣
 ↪STREET', 'ELIZABETH STREET', "A'BECKETT STREET", 'Lt LONSDALE STREET'])]
print("")

#STREET SIDES
print("Counts of Side of Street:", theMacHIGH['Side Of Street'].unique().size,␣
 ↪"\n")
theMacHIGH['Side Of Street'].unique()

print("Finding the number of times a street has a violation:\n",␣
 ↪theMacHIGH['Side Of Street'].value_counts().to_frame(), "\n")
print("The Percentages of each Side of Street in a Violation:", "\n",␣
 ↪theMacHIGH['Side Of Street'].value_counts(normalize=True)*100)

theMacHIGH134 = theMacHIGH[theMacHIGH["Side Of Street"].isin([1, 3, 4])]
print("")

#SIGNS
print("Counting Signs in Violations \n", theMacHIGH134['Sign'].value_counts().
 ↪to_frame(), "\n")

print("Testing the Worst Ranked Signs\n")
#sign 1
print("SIGN ONE: 1P MTR M-SAT 7:30-19:30")
THEmacLOS = theMacHIGH134[theMacHIGH134["Sign"] == "1P MTR M-SAT 7:30-19:30"]
print("Value Counts for Parking Duration: ", THEmacLOS['Parking Duration (s)'].
 ↪value_counts().to_frame())
print("The time (hours) of the max length of stay is: ",␣
 ↪(round(THEmacLOS['Parking Duration (s)'].max()/60)/60), "Hours")
print("")
#sign 2
print("SIGN TWO: 2P MTR M-SAT 7:30-20:30")
THEmacLOS = theMacHIGH134[theMacHIGH134["Sign"] == "2P MTR M-SAT 7:30-20:30"]
```

```
print("Value Counts for Parking Duration: ", THEmacLOS['Parking Duration (s)'].
 ↪value_counts().to_frame())
print("The time (hours) of the max length of stay is: ",␣
 ↪(round(THEmacLOS['Parking Duration (s)'].max()/60)/60), "Hours")
print("")
```

```
        Area             Street    Street Limit 1      Street Limit 2  \
21    The Mac  ELIZABETH STREET   LA TROBE STREET  Lt LONSDALE STREET
26    The Mac  A'BECKETT STREET  ELIZABETH STREET     SWANSTON STREET
91    The Mac  A'BECKETT STREET  ELIZABETH STREET     SWANSTON STREET
175   The Mac  ELIZABETH STREET   LA TROBE STREET  Lt LONSDALE STREET
222   The Mac  A'BECKETT STREET  ELIZABETH STREET     SWANSTON STREET

     Side Of Street Street Marker            Arrival Time  \
21                2          972E  01/10/2011 12:01:21 AM
26                4         5975S  01/10/2011 12:01:23 AM
91                4         5981S  01/10/2011 12:31:45 AM
175               2          972E  01/10/2011 02:01:57 AM
222               4         5981S  01/10/2011 02:56:20 AM

             Departure Time  Parking Duration (s)                   Sign  \
21   01/10/2011 12:27:44 AM                  1583  1/4P M-SUN 0:00-23:59
26   01/10/2011 02:20:45 PM                 51562      4P DIS ONLY M-SUN
91   01/10/2011 12:51:04 AM                  1159   P10 M-SUN 0:00-23:59
175  01/10/2011 02:27:57 AM                  1560  1/4P M-SUN 0:00-23:59
222  01/10/2011 03:47:13 AM                  3053   P10 M-SUN 0:00-23:59

     Violation  Street ID  Device ID
21           1        627        596
26           1          5       2455
91           1          5       2440
175          1        627        596
222          1          5       2440
The exact number of infringements around the Mac:  94403

Streets in this area:  7
Finding the number of times a street has a violation:
                     Street
FRANKLIN STREET      32577
LA TROBE STREET      15379
ELIZABETH STREET     15132
A'BECKETT STREET     11814
Lt LONSDALE STREET   10515
Lt LA TROBE STREET    4764
SWANSTON STREET       4222

Counts of Side of Street: 5
```

```
Finding the number of times a street has a violation:
    Side Of Street
4           38124
3           19120
1           13041
2            8490
5            6642

The Percentages of each Side of Street in a Violation:
 4    44.632801
3    22.384303
1    15.267453
2     9.939473
5     7.775970
Name: Side Of Street, dtype: float64

Counting Signs in Violations
                                  Sign
1P MTR M-SAT 7:30-19:30          17226
2P MTR M-SAT 7:30-20:30          11997
P/ 5 M-SAT 7:30-19:30             8521
1/2P MTR M-SAT 7:30-19:30         6840
1P TKT A M-F 9:30-19:30           2614
2P SUN 7:30-18:30                 2539
2P MTR M-F 9:30-20:30             2346
1/2P TKT A M-F  7:30-16:00        2221
2P DIS M-SUN 0:00-23:59           2217
P15 M-SAT 7:30-19:30              2178
P/ 10 M-SUN 0:00-11:59            1891
1P SUN 7:30-18:30                 1257
LZ 30M M-F 7:30-19:30             1195
2P DIS AOT 9:30-23:59             1182
1/2P TKT A SAT 7:30-19:30          743
1/2P MTR M-F 9:30-19:30            658
1/2P TKT A M-F 18:30-19:30         651
1P TKT A SAT 7:30-19:30            622
CW TOW M-F 7:00-9:30               563
2P MTR SAT 7:30-20:30              464
2P DIS AOT 00:00-23:59             315
4P DIS ONLY M-SUN                  310
S/ No Stop MCCV M-F 7:30-19:30     274
LZ 30M M-F 9:30-19:30              220
2P SAT 7:30-19:30                  188
1/4P M-SAT 16:00-19:30             174
P10 M-SUN 0:00-23:59               160
1/2P MTR SAT 7:30-1930             154
2P DIS AOT 0:00-7:00               141
```

```
CW TOW M-F 16:00-18:30            122
1P MTR M-SAT 9:30-16:00            62
4P DIS SUN 7:30-18:30              52
3P MTR M-F 7:30-16:00              51
LZ 30M SAT 7:30-19:30              36
2P DIS M-SAT 7:30-19:30            34
2P DIS M-SAT 7:30-18:30            34
LZ 30M M-SUN 7:30-18:30             9
3P MTR Sat 7:30-20:30               9
1/4P M-SAT 7:30-9:30                8
P 5 Mon - Sat 7.30 - 19.30          5
CW M-F 7:00-9:30                    1
P5 THU 7:00-14:00                   1

Testing the Worst Ranked Signs

SIGN ONE: 1P MTR M-SAT 7:30-19:30
Value Counts for Parking Duration:         Parking Duration (s)
43192                     79
43193                     57
3919                      22
4056                      20
3928                      17
…                        …
8812                       1
29286                      1
8796                       1
6747                       1
14383                      1

[6498 rows x 1 columns]
The time (hours) of the max length of stay is:  12.0 Hours

SIGN TWO: 2P MTR M-SAT 7:30-20:30
Value Counts for Parking Duration:         Parking Duration (s)
46792                    130
46793                     41
7511                      15
7572                      14
46791                     14
…                        …
15988                      1
16209                      1
9839                       1
13933                      1
14572                      1

[5850 rows x 1 columns]
```

```
The time (hours) of the max length of stay is:  13.0 Hours
```

Again, the same pattern goes in terms of timing, where drivers are parking during working hours in timed, mostly paid spots that have a limit of maximum 2 hours.

The street sides that also have the most violations are 1, 4 and 5.

### 2.4.6 Courtney

[9]:
```
courtney = parkingDB[parkingDB["Area"] == "Courtney"]
courtney = courtney[courtney["Violation"] == 1]
print(courtney.head())
print("The exact number of infringements around Courtney: ", len(courtney))
print("")

#STREETS
print("Streets in this area: ", courtney['Street'].unique().size)
courtney['Street'].unique()

print("Finding the number of times a street has a violation:\n",␣
↪courtney['Street'].value_counts().to_frame())
courtneyHIGH = courtney[courtney["Street"].isin(['COBDEN STREET', 'ELIZABETH␣
↪STREET', "O'CONNELL STREET"])]
print("")

#STREET SIDES
print("Counts of Side of Street:", courtneyHIGH['Side Of Street'].unique().
↪size, "\n")
courtneyHIGH['Side Of Street'].unique()

print("Finding the number of times a street has a violation:\n",␣
↪courtneyHIGH['Side Of Street'].value_counts().to_frame(), "\n")
print("The Percentages of each Side of Street in a Violation:", "\n",␣
↪courtneyHIGH['Side Of Street'].value_counts(normalize=True)*100)

print("")

#SIGNS
print("Counting Signs in Violations \n", courtneyHIGH['Sign'].value_counts().
↪to_frame(), "\n")

print("Testing the Worst Ranked Signs\n")
#sign 1
print("SIGN ONE: 1/2P M-F 7:30-18:30")
COURNTEYlOss = courtneyHIGH[courtneyHIGH["Sign"] == "1/2P M-F 7:30-18:30"]
print("Value Counts for Parking Duration: ", COURNTEYlOss['Parking Duration␣
↪(s)'].value_counts().to_frame())
```

```
print("The time (hours) of the max length of stay is: ",␣
 ↪(round(COURNTEYlOss['Parking Duration (s)'].max()/60)/60), "Hours")
print("")
```

```
         Area             Street      Street Limit 1      Street Limit 2  \
2     Courtney  ELIZABETH STREET       PELHAM STREET  QUEENSBERRY STREET
669   Courtney  ELIZABETH STREET       PELHAM STREET  QUEENSBERRY STREET
670   Courtney  ELIZABETH STREET       PELHAM STREET  QUEENSBERRY STREET
671   Courtney  O'CONNELL STREET         PEEL STREET  QUEENSBERRY STREET
705   Courtney  ELIZABETH STREET  QUEENSBERRY STREET     VICTORIA STREET

     Side Of Street Street Marker            Arrival Time  \
2                 2         4744E  01/10/2011 12:01:11 AM
669               2         4760E  01/10/2011 07:30:07 AM
670               2         4764E  01/10/2011 07:30:07 AM
671               5         5147W  01/10/2011 07:30:07 AM
705               5         4639W  01/10/2011 07:30:07 AM

              Departure Time  Parking Duration (s)                        Sign  \
2     01/10/2011 11:00:19 AM                 39548     2P DIS M-SUN 0:00-23:59
669   01/10/2011 12:30:00 PM                 17993     2P TKT A SAT 7:30-12:30
670   01/10/2011 12:18:15 PM                 17288     2P TKT A SAT 7:30-12:30
671   01/10/2011 08:54:20 AM                  5053  1/2P A RPE SAT 7:30-12:30
705   01/10/2011 12:30:00 PM                 17993           2P SAT 7:30-12:30

     Violation  Street ID  Device ID
2            1        627       3202
669          1        627       3308
670          1        627       3453
671          1       1062       3923
705          1        627       3464
The exact number of infringements around Courtney:  80880

Streets in this area:  7
Finding the number of times a street has a violation:
                     Street
O'CONNELL STREET     24507
COBDEN STREET        23904
ELIZABETH STREET     16590
PEEL STREET           6878
PRINCESS STREET       5418
QUEENSBERRY STREET    3551
CURZON STREET           32

Counts of Side of Street: 2

Finding the number of times a street has a violation:
```

```
    Side Of Street
2           44206
5           20795

The Percentages of each Side of Street in a Violation:
 2    68.008184
5    31.991816
Name: Side Of Street, dtype: float64

Counting Signs in Violations
                                        Sign
1/2P M-F 7:30-18:30                    9297
1P M-F 7:30-18:30                      8253
1P TKT A M-F 7:30-18:30                6454
1P RPA M-SUN 7:30-23:00                6254
1/2P RPA M-SUN 7:30-18:30              5327
1P TKT A M-SAT 7:30-18:30              3343
2P TKT A M-F 7:30-18:30                3329
1P RPA M-F 7:30-18:30                  3328
1/2P A RPE M-F 7:30-18:30              2479
4P TKT A M-F 7:30-18:30                1930
S/ No Stop Auth Veh M-Sun 00:00-23:59  1773
1P A RPE M-F 7:30-18:30                1708
1/2P RPA M-F 7:30-18:30                1573
P/ 15 M-SUN 00:00-23:59                1439
1/2P A RPE M-SUN 7:30-23:00            1387
1P SUN 7:30-18:30                      1223
2P S-S 7:30-18:30                      1222
2P SAT 7:30-12:30                       885
2P TKT A SAT 7:30-12:30                 564
1/2P RPA S-S 7:30-12:30                 560
LZ 30M M-F 7:30-18:30                   460
1/2P A RPE SAT 7:30-12:30               435
1P RPA SAT 7:30-12:30                   433
2P DIS M-SUN 0:00-23:59                 366
4P MTR M-F 7:30-18:30                   256
1/2P SAT 7:30-12:30                     209
1P A RPE SAT 7:30-12:30                 205
4P TKT A SAT 7:30-12:30                 183
P5 THU 7:00-14:00                        65
LZ 30M SAT 7:30-12:30                    33
4P MTR SAT 7:30-12:30                    26
Temp Sign Plate Sun-Sun 4:00-5:00         2

Testing the Worst Ranked Signs

SIGN ONE: 1/2P M-F 7:30-18:30
Value Counts for Parking Duration:         Parking Duration (s)
```

```
39593                    59
39592                    20
2185                     17
2104                     15
2178                     14
…                        …
4529                      1
6576                      1
4076                      1
16795                     1
6141                      1

[3989 rows x 1 columns]
The time (hours) of the max length of stay is:  11.0 Hours
```

### 2.4.7 Chinatown

[10]:
```
chinatown = parkingDB[parkingDB["Area"] == "Chinatown"]
chinatown = chinatown[chinatown["Violation"] == 1]
print(chinatown.head())
print("The exact number of infringements around the Chinatown: ",␣
 ↪len(chinatown))
print("")

#STREETS
print("Streets in this area: ", chinatown['Street'].unique().size)
chinatown['Street'].unique()

print("Finding the number of times a street has a violation:\n",␣
 ↪chinatown['Street'].value_counts().to_frame())
chinatownHIGH = chinatown[chinatown["Street"].isin(['RUSSELL STREET',␣
 ↪'ELIZABETH STREET', "LONSDALE STREET"])]
print("")

#STREET SIDES
print("Counts of Side of Street:", chinatownHIGH['Side Of Street'].unique().
 ↪size, "\n")
chinatownHIGH['Side Of Street'].unique()

print("Finding the number of times a street has a violation:\n",␣
 ↪chinatownHIGH['Side Of Street'].value_counts().to_frame(), "\n")
print("The Percentages of each Side of Street in a Violation:", "\n",␣
 ↪chinatownHIGH['Side Of Street'].value_counts(normalize=True)*100)

chinatownHIGH125 = chinatownHIGH[chinatownHIGH["Side Of Street"].isin([1, 2,␣
 ↪5])]
```

```
print("")

#SIGNS
print("Counting Signs in Violations \n", chinatownHIGH125['Sign'].
 ↪value_counts().to_frame(), "\n")

print("Testing the Worst Ranked Signs\n")
#sign 1
print("SIGN ONE: 1P MTR M-SAT 7:30-19:30")
CHINATOWNlOSS = chinatownHIGH125[chinatownHIGH125["Sign"] == "1P MTR M-SAT 7:
 ↪30-19:30"]
print("Value Counts for Parking Duration: ", CHINATOWNlOSS['Parking Duration␣
 ↪(s)'].value_counts().to_frame())
print("The time (hours) of the max length of stay is: ",␣
 ↪(round(CHINATOWNlOSS['Parking Duration (s)'].max()/60)/60), "Hours")
print("")
#sign 2
print("SIGN TWO: 1/2P MTR M-SAT 7:30-19:30")
CHINATOWNlOSS = chinatownHIGH125[chinatownHIGH125["Sign"] == "1/2P MTR M-SAT 7:
 ↪30-19:30"]
print("Value Counts for Parking Duration: ", CHINATOWNlOSS['Parking Duration␣
 ↪(s)'].value_counts().to_frame())
print("The time (hours) of the max length of stay is: ",␣
 ↪(round(CHINATOWNlOSS['Parking Duration (s)'].max()/60)/60), "Hours")
print("")
```

```
         Area             Street    Street Limit 1     Street Limit 2  \
13   Chinatown  ELIZABETH STREET  Lt BOURKE STREET      BOURKE STREET
38   Chinatown    RUSSELL STREET     BOURKE STREET  Lt COLLINS STREET
39   Chinatown    RUSSELL STREET     BOURKE STREET  Lt COLLINS STREET
46   Chinatown    RUSSELL STREET  Lt BOURKE STREET      BOURKE STREET
55   Chinatown  ELIZABETH STREET  Lt BOURKE STREET      BOURKE STREET

    Side Of Street Street Marker            Arrival Time  \
13               2          946E  01/10/2011 12:01:18 AM
38               5          765W  01/10/2011 12:01:28 AM
39               5          767W  01/10/2011 12:01:28 AM
46               2          770E  01/10/2011 12:01:29 AM
55               2          944E  01/10/2011 12:01:34 AM

            Departure Time  Parking Duration (s)  \
13  01/10/2011 06:13:22 AM                 22324
38  01/10/2011 03:50:49 AM                 13761
39  01/10/2011 12:58:51 AM                  3443
46  01/10/2011 02:42:12 AM                  9643
55  01/10/2011 02:48:02 AM                  9988
```

```
                                    Sign  Violation  Street ID  Device ID
13  P/ (No Parking) AOT M-SUN 12:00-7:30          1        627        565
38               2P DIS M-SUN 0:00-23:59          1       1221        689
39                 1/4P M-SUN 0:00-23:59          1       1221        494
46               2P DIS M-SUN 0:00-23:59          1       1221        504
55  P/ (No Parking) AOT M-SUN 12:00-7:30          1        627        575
The exact number of infringements around the Chinatown:  79110

Streets in this area:  5
Finding the number of times a street has a violation:
                    Street
RUSSELL STREET      47229
ELIZABETH STREET    15316
LONSDALE STREET     10043
Lt COLLINS STREET    5501
BOURKE STREET        1021

Counts of Side of Street: 5

Finding the number of times a street has a violation:
    Side Of Street
1           24450
5           22874
2           15221
4            5298
3            4745

The Percentages of each Side of Street in a Violation:
 1    33.683253
5    31.512096
2    20.969031
4     7.298727
3     6.536893
Name: Side Of Street, dtype: float64

Counting Signs in Violations
                                      Sign
1P MTR M-SAT 7:30-19:30              20869
1/2P MTR M-SAT 7:30-19:30            20110
1/2P M-SAT 7:30-19:30                 8971
1P SUN 7:30-18:30                     6291
2P DIS M-SUN 0:00-23:59               1950
1/4P M-SUN 0:00-23:59                 1584
LZ 15M M-F 7:30-19:30                 1344
2P DIS M-SUN 7:30-19:30                516
P/ (No Parking) AOT M-SAT 19:30-23:59  265
P/ (No Parking) AOT M-SUN 12:00-7:30   200
S/ No Stop M-F 6:00-7:30               129
```

```
1P SAT 7:30-19:30                         117
4P DIS AOT 12:00-7:30                      76
4P DIS AOT 19:30-23:59                     67
P/ (No Parking) AOT SUN 18:30-23:59        56

Testing the Worst Ranked Signs

SIGN ONE: 1P MTR M-SAT 7:30-19:30
Value Counts for Parking Duration:         Parking Duration (s)
43200                      89
43193                      83
43192                      46
4047                       26
3930                       24
…                         …
12817                       1
21005                       1
24947                       1
10754                       1
9018                        1

[5984 rows x 1 columns]
The time (hours) of the max length of stay is:  12.0 Hours

SIGN TWO: 1/2P MTR M-SAT 7:30-19:30
Value Counts for Parking Duration:         Parking Duration (s)
43193                     281
43192                     111
43200                      72
2114                       25
2177                       25
…                         …
7691                        1
19969                       1
17920                       1
5626                        1
4098                        1

[5486 rows x 1 columns]
The time (hours) of the max length of stay is:  12.0 Hours
```

### 2.4.8 Banks

[11]:
```
banks = parkingDB[parkingDB["Area"] == "Banks"]
banks = banks[banks["Violation"] == 1]
print(banks.head())
print("The exact number of infringements around Banks: ", len(banks))
```

```
print("")

#STREETS
print("Streets in this area: ", banks['Street'].unique().size)
banks['Street'].unique()

print("Finding the number of times a street has a violation:\n",␣
 ↪banks['Street'].value_counts().to_frame())
banksHIGH = banks[banks["Street"].isin(['QUEEN STREET', 'FLINDERS STREET',␣
 ↪"COLLINS STREET"])]
print("")

#STREET SIDES
print("Counts of Side of Street:", banksHIGH['Side Of Street'].unique().size,␣
 ↪"\n")
banksHIGH['Side Of Street'].unique()

print("Finding the number of times a street has a violation:\n",␣
 ↪banksHIGH['Side Of Street'].value_counts().to_frame(), "\n")
print("The Percentages of each Side of Street in a Violation:", "\n",␣
 ↪banksHIGH['Side Of Street'].value_counts(normalize=True)*100)
banksHIGH13 = chinatownHIGH[chinatownHIGH["Side Of Street"].isin([1, 3])]

print("")

#SIGNS
print("Counting Signs in Violations \n", banksHIGH13['Sign'].value_counts().
 ↪to_frame(), "\n")

print("Testing the Worst Ranked Signs\n")
#sign 1
print("SIGN ONE: 1P MTR M-SAT 7:30-19:30")
banksLoss = banksHIGH13[banksHIGH13["Sign"] == "1P MTR M-SAT 7:30-19:30"]
print("Value Counts for Parking Duration: ", banksLoss['Parking Duration (s)'].
 ↪value_counts().to_frame())
print("The time (hours) of the max length of stay is: ",␣
 ↪(round(banksLoss['Parking Duration (s)'].max()/60)/60), "Hours")
print("")
```

```
       Area         Street   Street Limit 1   Street Limit 2  Side Of Street  \
44    Banks  MARKET STREET    FLINDERS LANE  FLINDERS STREET               5
658   Banks   QUEEN STREET  FLINDERS STREET    FLINDERS LANE               1
693   Banks  MARKET STREET   COLLINS STREET    FLINDERS LANE               1
940   Banks  MARKET STREET   COLLINS STREET    FLINDERS LANE               1
1002  Banks   QUEEN STREET   COLLINS STREET    FLINDERS LANE               1

     Street Marker            Arrival Time          Departure Time  \
```

```
44           1265W  01/10/2011 12:01:29 AM  01/10/2011 02:15:40 AM
658           C988  01/10/2011 07:30:00 AM  01/10/2011 10:26:30 AM
693          C1832  01/10/2011 07:30:07 AM  01/10/2011 10:15:58 AM
940          C1826  01/10/2011 07:30:07 AM  01/10/2011 07:30:00 PM
1002         C1058  01/10/2011 07:30:08 AM  01/10/2011 10:03:07 AM

      Parking Duration (s)                    Sign  Violation  Street ID  \
44                    8051  2P DIS M-SUN 0:00-23:59          1        957
658                  10590  1P MTR M-SAT 7:30-19:30          1       1171
693                   9951  1P MTR M-SAT 7:30-19:30          1        957
940                  43193  1P MTR M-SAT 7:30-19:30          1        957
1002                  9179  1P MTR M-SAT 7:30-19:30          1       1171

      Device ID
44         1673
658         602
693        1707
940        1713
1002        623
The exact number of infringements around Banks:  77995

Streets in this area:  7
Finding the number of times a street has a violation:
                  Street
QUEEN STREET      28417
FLINDERS LANE     18759
COLLINS STREET    11985
MARKET STREET      8417
FLINDERS STREET    4074
WILLIAM STREET     3744
BOND STREET        2599

Counts of Side of Street: 5

Finding the number of times a street has a violation:
    Side Of Street
1           22547
3           11990
5            4705
4            4069
2            1165

The Percentages of each Side of Street in a Violation:
 1    50.694757
3    26.958360
5    10.578739
4     9.148754
2     2.619390
```

```
Name: Side Of Street, dtype: float64

Counting Signs in Violations
                             Sign
1P MTR M-SAT 7:30-19:30   20869
1P SUN 7:30-18:30          4172
CW TOW M-F 16:00-18:30     2009
1P MTR M-F 7:30-16:00      1509
1P MTR SAT 7:30-19:30       507
S/ No Stop M-F 6:00-7:30    129

Testing the Worst Ranked Signs

SIGN ONE: 1P MTR M-SAT 7:30-19:30
Value Counts for Parking Duration:         Parking Duration (s)
43200                      89
43193                      83
43192                      46
4047                       26
3930                       24
…                         …
12817                       1
21005                       1
24947                       1
10754                       1
9018                        1

[5984 rows x 1 columns]
The time (hours) of the max length of stay is:  12.0 Hours
```

### 2.4.9 Titles

[12]:
```
titles = parkingDB[parkingDB["Area"] == "Titles"]
titles = titles[titles["Violation"] == 1]
print(titles.head())
print("The exact number of infringements around Titles: ", len(titles))
print("")

#STREETS
print("Streets in this area: ", titles['Street'].unique().size)
titles['Street'].unique()

print("Finding the number of times a street has a violation:\n",␣
 ↪titles['Street'].value_counts().to_frame())
titlesHIGH = titles[titles["Street"].isin(['QUEEN STREET', 'LA TROBE STREET',␣
 ↪"A'BECKETT STREET", "Lt LONSDALE STREET"])]
print("")
```

```
#STREET SIDES
print("Counts of Side of Street:", titlesHIGH['Side Of Street'].unique().size,␣
 ↪"\n")
titlesHIGH['Side Of Street'].unique()

print("Finding the number of times a street has a violation:\n",␣
 ↪titlesHIGH['Side Of Street'].value_counts().to_frame(), "\n")
print("The Percentages of each Side of Street in a Violation:", "\n",␣
 ↪titlesHIGH['Side Of Street'].value_counts(normalize=True)*100)
titles34 = titlesHIGH[titlesHIGH["Side Of Street"].isin([4, 3])]

print("")

#SIGNS
print("Counting Signs in Violations \n", titles34['Sign'].value_counts().
 ↪to_frame(), "\n")

print("Testing the Worst Ranked Signs\n")
#sign 1
print("SIGN ONE: 1P MTR M-SAT 7:30-19:30")
titlesLOSS = titles34[titles34["Sign"] == "1P MTR M-SAT 7:30-19:30"]
print("Value Counts for Parking Duration: ", titlesLOSS['Parking Duration (s)'].
 ↪value_counts().to_frame())
print("The time (hours) of the max length of stay is: ",␣
 ↪(round(titlesLOSS['Parking Duration (s)'].max()/60)/60), "Hours")
print("")
```

```
       Area            Street    Street Limit 1    Street Limit 2  \
688   Titles   LA TROBE STREET    WILLIAM STREET      QUEEN STREET
763   Titles      WILLS STREET  A'BECKETT STREET   LA TROBE STREET
764   Titles  A'BECKETT STREET    WILLIAM STREET      QUEEN STREET
853   Titles  A'BECKETT STREET      QUEEN STREET  ELIZABETH STREET
1069  Titles   LA TROBE STREET      QUEEN STREET  ELIZABETH STREET

      Side Of Street Street Marker            Arrival Time  \
688                3         3886N  01/10/2011 07:30:07 AM
763                2         5186E  01/10/2011 07:30:07 AM
764                4         6043S  01/10/2011 07:30:07 AM
853                3         5994N  01/10/2011 07:30:07 AM
1069               3         3804N  01/10/2011 07:30:08 AM

              Departure Time  Parking Duration (s)                      Sign  \
688   01/10/2011 07:30:00 PM                 43193     1/2P MTR SAT 7:30-1930
763   01/10/2011 01:58:15 PM                 23288    2P MTR M-SAT 7:30-20:30
764   01/10/2011 10:31:29 AM                 10882        2P M-SAT 7:30-20:30
853   01/10/2011 03:05:40 PM                 27333    1P MTR M-SAT 7:30-19:30
```

```
1069  01/10/2011 09:44:01 AM                  8033  1/2P TKT A SAT 7:30-19:30

      Violation  Street ID  Device ID
688           1        856       2277
763           1       1433       2349
764           1          5       2401
853           1          5       2374
1069          1        856       2610
The exact number of infringements around Titles:  77814

Streets in this area:  7
Finding the number of times a street has a violation:
                     Street
LA TROBE STREET      25419
QUEEN STREET         15107
A'BECKETT STREET     14159
Lt LONSDALE STREET   12004
WILLIAM STREET        4907
WILLS STREET          4284
ANTHONY STREET        1934

Counts of Side of Street: 5

Finding the number of times a street has a violation:
    Side Of Street
3           26550
4           25032
1            6248
5            4986
2            3873

The Percentages of each Side of Street in a Violation:
 3    39.811663
4    37.535426
1     9.368861
5     7.476495
2     5.807554
Name: Side Of Street, dtype: float64

Counting Signs in Violations
                               Sign
1P MTR M-SAT 7:30-19:30      10174
1P TKT A M-F 9:30-19:30       5545
2P SUN 7:30-18:30             4138
2P M-SAT 7:30-20:30           3346
CW TOW M-F 7:00-9:30          3187
2P MTR M-SAT 7:30-20:30       3097
2P TKT A M-SAT 7:30-20:30     2724
```

```
1/2P TKT A M-SAT 7:30-19:30   2386
P10 M-F 9:30-19:30            1945
1/2P TKT A M-F  7:30-16:00    1752
1P TKT A SAT 7:30-19:30       1391
1/2P MTR M-F 7:30-16:00       1179
1P MTR M-F 9:30-19:30          982
1P SUN 7:30-18:30              927
2P MTR M-F 9:30-20:30          848
CW TOW M-F 16:00-18:30         841
2P MTR M-F 7:30-16:00          839
2P DIS M-SUN 0:00-23:59        827
1P TKT A M-F 7:30-16:00        719
LZ 30M M-SAT 7:30-19:30        593
2P TKT A M-F 9:30-20:30        552
1/2P TKT A SAT 7:30-19:30      458
LZ 15M M-F 7:30-16:00          439
LZ 15M M-SAT 7:30-19:30        430
1/2P TKT A M-F 18:30-19:30     352
1/4P M-F 7:30-16:00            308
P10 SAT 7:30-19:30             289
1P MTR M-F 7:30-16:00          234
1P MTR SAT 7:30-19:30          219
2P MTR SAT 7:30-20:30          189
2P TKT A SAT 7:30-20:30        109
1/2P MTR M-F 18:30-19:30        92
1P M-SAT 7:30-19:30             86
2P DIS AOT 0:00-23:59           85
2P DIS AOT 18:30-23:59          84
2P DIS AOT 00:00-16:00          76
1/2P MTR SAT 7:30-1930          59
LZ 15M SAT 7:30-19:30           32
LZ 15M M-F 18:30-19:30          22
2P SAT 7:30-19:30               14
1/4P M-F 18:30-19:30            13

Testing the Worst Ranked Signs

SIGN ONE: 1P MTR M-SAT 7:30-19:30
Value Counts for Parking Duration:         Parking Duration (s)
43193                     63
43192                     32
3987                      15
4135                      14
4203                      12
…                         …
12874                      1
6727                       1
10817                      1
```

```
27979                    1
7648                     1

[4597 rows x 1 columns]
The time (hours) of the max length of stay is:  12.0 Hours
```

### 2.4.10 Hardware

[13]:
```
hardware = parkingDB[parkingDB["Area"] == "Hardware"]
hardware = hardware[hardware["Violation"] == 1]
print(hardware.head())
print("The exact number of infringements around Hardware: ", len(hardware))
print("")

#STREETS
print("Streets in this area: ", hardware['Street'].unique().size)
hardware['Street'].unique()

print("Finding the number of times a street has a violation:\n",␣
↪hardware['Street'].value_counts().to_frame())
hardwareHIGH = hardware[hardware["Street"].isin(['QUEEN STREET', 'Lt BOURKE␣
↪STREET', "LONSDALE STREET"])]
print("")

#STREET SIDES
print("Counts of Side of Street:", hardwareHIGH['Side Of Street'].unique().
↪size, "\n")
hardwareHIGH['Side Of Street'].unique()

print("Finding the number of times a street side has a violation:\n",␣
↪hardwareHIGH['Side Of Street'].value_counts().to_frame(), "\n")
print("The Percentages of each Side of Street in a Violation:", "\n",␣
↪hardwareHIGH['Side Of Street'].value_counts(normalize=True)*100)
hardwareHIGH14 = hardwareHIGH[hardwareHIGH["Side Of Street"].isin([4, 1])]

print("")

#SIGNS
print("Counting Signs in Violations \n", hardwareHIGH14['Sign'].value_counts().
↪to_frame(), "\n")

print("Testing the Worst Ranked Signs\n")
#sign 1
print("SIGN ONE: 1P MTR M-SAT 7:30-19:30")
hardwareLOSS = hardwareHIGH14[hardwareHIGH14["Sign"] == "1P MTR M-SAT 7:30-19:
↪30"]
```

```
print("Value Counts for Parking Duration: ", hardwareLOSS['Parking Duration␣
 ↪(s)'].value_counts().to_frame())
print("The time (hours) of the max length of stay is: ",␣
 ↪(round(hardwareLOSS['Parking Duration (s)'].max()/60)/60), "Hours")
print("")
```

```
          Area             Street      Street Limit 1    Street Limit 2  \
74    Hardware  Lt BOURKE STREET      WILLIAM STREET      QUEEN STREET
271   Hardware  Lt BOURKE STREET      WILLIAM STREET      QUEEN STREET
975   Hardware      QUEEN STREET     LONSDALE STREET  Lt BOURKE STREET
1000  Hardware      QUEEN STREET    Lt BOURKE STREET     BOURKE STREET
1060  Hardware      QUEEN STREET  Lt LONSDALE STREET   LONSDALE STREET

      Side Of Street Street Marker            Arrival Time  \
74                 4         2579S  01/10/2011 12:14:38 AM
271                4         2571S  01/10/2011 04:07:48 AM
975                5         1175W  01/10/2011 07:30:08 AM
1000               1         C1140  01/10/2011 07:30:08 AM
1060               1         C1218  01/10/2011 07:30:08 AM

              Departure Time  Parking Duration (s)                        Sign  \
74    01/10/2011 02:30:17 AM                  8139    2P DIS M-SUN 0:00-23:59
271   01/10/2011 06:34:50 AM                  8822    2P DIS M-SUN 0:00-23:59
975   01/10/2011 08:53:12 AM                  4984  1/2P MTR M-SAT 7:30-19:30
1000  01/10/2011 09:45:49 AM                  8141    1P MTR M-SAT 7:30-19:30
1060  01/10/2011 10:35:10 AM                 11102    1P MTR M-SAT 7:30-19:30

      Violation  Street ID  Device ID
74            1        907       1314
271           1        907       1302
975           1       1171        700
1000          1       1171        670
1060          1       1171        724
The exact number of infringements around Hardware:  77459

Streets in this area:  3
Finding the number of times a street has a violation:
                  Street
QUEEN STREET       49466
LONSDALE STREET    14717
Lt BOURKE STREET   13276

Counts of Side of Street: 5

Finding the number of times a street side has a violation:
    Side Of Street
1           50220
```

```
4           15019
5            7551
2            3665
3            1004

The Percentages of each Side of Street in a Violation:
 1    64.834299
4    19.389613
5     9.748383
2     4.731535
3     1.296170
Name: Side Of Street, dtype: float64

Counting Signs in Violations
                                Sign
1P MTR M-SAT 7:30-19:30        49051
1P SUN 7:30-18:30               7515
1/2P MTR M-SAT 7:30-19:30       6091
LZ 15M M-F 9:30-19:30           1435
2P DIS M-SUN 0:00-23:59          615
1P SAT 7:30-19:30                229
LZ 30M M-F 9:30-19:30            102
2P DIS AOT 9:30-23:59             89
2P DIS AOT 0:00-23:59             72
Temp Sign Plate SUN 8:30-5:30     34
2P DIS AOT 0:00-7:00               3
P5 THU 7:00-14:00                  3

Testing the Worst Ranked Signs

SIGN ONE: 1P MTR M-SAT 7:30-19:30
Value Counts for Parking Duration:         Parking Duration (s)
43192                   481
43193                   321
43191                    58
3901                     53
4036                     51
…                       …
20666                     1
8795                      1
8404                      1
16600                     1
25091                     1

[8589 rows x 1 columns]
The time (hours) of the max length of stay is:  12.0 Hours
```

### 2.4.11 Spencer

[14]:
```
spencer = parkingDB[parkingDB["Area"] == "Spencer"]
spencer = spencer[spencer["Violation"] == 1]
print(spencer.head())
print("The exact number of infringements around Spencer: ", len(spencer))
print("")

#STREETS
print("Streets in this area: ", spencer['Street'].unique().size)
spencer['Street'].unique()

print("Finding the number of times a street has a violation:\n",␣
↪spencer['Street'].value_counts().to_frame())
spencerHIGH = spencer[spencer["Street"].isin(['FLINDERS STREET', 'BOURKE␣
↪STREET', 'SPENCER STREET', 'FRANCIS STREET', 'Lt COLLINS STREET'])]
print("")

#STREET SIDES
print("Counts of Side of Street:", spencerHIGH['Side Of Street'].unique().size,␣
↪"\n")
spencerHIGH['Side Of Street'].unique()

print("Finding the number of times a street has a violation:\n",␣
↪spencerHIGH['Side Of Street'].value_counts().to_frame(), "\n")
print("The Percentages of each Side of Street in a Violation:", "\n",␣
↪spencerHIGH['Side Of Street'].value_counts(normalize=True)*100)
SPENCERHIGH34 = spencerHIGH[spencerHIGH["Side Of Street"].isin([4, 3])]

print("")

#SIGNS
print("Counting Signs in Violations \n", SPENCERHIGH34['Sign'].value_counts().
↪to_frame(), "\n")

print("Testing the Worst Ranked Signs\n")
#sign 1
print("SIGN ONE: 1P TKT A M-SAT 7:30-19:30")
SPENCERLOSS = SPENCERHIGH34[SPENCERHIGH34["Sign"] == "1P TKT A M-SAT 7:30-19:
↪30"]
print("Value Counts for Parking Duration: ", SPENCERLOSS['Parking Duration␣
↪(s)'].value_counts().to_frame())
print("The time (hours) of the max length of stay is: ",␣
↪(round(SPENCERLOSS['Parking Duration (s)'].max()/60)/60), "Hours")
print("")
```

```
        Area          Street  Street Limit 1     Street Limit 2  \
54    Spencer   BOURKE STREET  SPENCER STREET        KING STREET
```

```
113  Spencer   FLINDERS LANE  SPENCER STREET        KING STREET
161  Spencer   BOURKE STREET  SPENCER STREET        KING STREET
823  Spencer  SPENCER STREET   BOURKE STREET  Lt COLLINS STREET
824  Spencer  FRANCIS STREET  SPENCER STREET        KING STREET

     Side Of Street Street Marker            Arrival Time  \
54                4         2481S  01/10/2011 12:01:34 AM
113               3         1864N  01/10/2011 12:49:23 AM
161               3         2476N  01/10/2011 01:44:03 AM
823               2         1528E  01/10/2011 07:30:07 AM
824               3         2094N  01/10/2011 07:30:07 AM

             Departure Time  Parking Duration (s)                        Sign  \
54   01/10/2011 02:59:40 AM                 10686    2P DIS M-SUN 0:00-23:59
113  01/10/2011 06:40:23 AM                 21060    2P DIS M-SUN 0:00-23:59
161  01/10/2011 05:51:44 AM                 14861    2P DIS M-SUN 0:00-23:59
823  01/10/2011 09:42:13 AM                  7926      2P MTR SAT 7:30-20:30
824  01/10/2011 02:12:45 PM                 24158  1/2P MTR M-SAT 7:30-19:30

     Violation  Street ID  Device ID
54           1        123       1389
113          1        669       1096
161          1        123       1435
823          1       1285        781
824          1        679       1236
The exact number of infringements around Spencer:  76344

Streets in this area:  7
Finding the number of times a street has a violation:
                    Street
Lt COLLINS STREET   16222
FLINDERS LANE       13938
BOURKE STREET       12189
SPENCER STREET      11194
FRANCIS STREET      10477
COLLINS STREET       8092
FLINDERS STREET      4232

Counts of Side of Street: 4

Finding the number of times a street has a violation:
    Side Of Street
4           28921
3           14199
5            5825
2            5369

The Percentages of each Side of Street in a Violation:
```

```
 4    53.247781
3    26.142431
5    10.724675
2     9.885112
Name: Side Of Street, dtype: float64

Counting Signs in Violations
                                      Sign
1P TKT A M-SAT 7:30-19:30            11103
1P MTR M-SAT 7:30-19:30               7419
1/2P MTR M-SAT 7:30-19:30             5872
1/2P TKT A M-SAT 7:30-19:30           4692
P/ 10 M-F 16:00-19:30                 2277
2P TKT A M-SAT 7:30-20:30             1655
P/ 10 M-F 7:30-9:30                   1324
2P MTR M-F 9:30-16:00                 1283
2P DIS M-SUN 0:00-23:59               1220
LZ 15M M-F 7:30-19:30                 1169
S/ No Stop M-F 16:00-19:30            1035
S/ No Stop M-F 7:30-9:30               964
1P MTR M-F 9:30-16:00                  873
2P SUN 7:30-18:30                      688
1P MTR M-SAT 9:30-16:00                487
2P MTR SAT 7:30-20:30                  302
CW M-F 7:00-9:30                       260
2P SAT 7:30-19:30                      252
CW M-F 16:00-18:30                     242
Temp Sign Plate Sun-Sun 4:00-5:00        3

Testing the Worst Ranked Signs

SIGN ONE: 1P TKT A M-SAT 7:30-19:30
Value Counts for Parking Duration:        Parking Duration (s)
3944                     20
3942                     15
3986                     13
4016                     12
4116                     12
…                        …
7368                      1
9594                      1
7344                      1
9391                      1
6081                      1

[5120 rows x 1 columns]
The time (hours) of the max length of stay is:  12.0 Hours
```

### 2.4.12 Rialto

[15]:
```
rialto = parkingDB[parkingDB["Area"] == "Rialto"]
rialto = rialto[rialto["Violation"] == 1]
print(rialto.head())
print("The exact number of infringements around Rialto: ", len(rialto))
print("")

#STREETS
print("Streets in this area: ", rialto['Street'].unique().size)
rialto['Street'].unique()

print("Finding the number of times a street has a violation:\n",␣
 ↪rialto['Street'].value_counts().to_frame())
rialtoHIGH = rialto[rialto["Street"].isin(['FLINDERS STREET', 'COLLINS STREET',␣
 ↪'KINGS STREET'])]
print("")

#STREET SIDES
print("Counts of Side of Street:", rialtoHIGH['Side Of Street'].unique().size,␣
 ↪"\n")
rialtoHIGH['Side Of Street'].unique()

print("Finding the number of times a street has a violation:\n",␣
 ↪rialtoHIGH['Side Of Street'].value_counts().to_frame(), "\n")
print("The Percentages of each Side of Street in a Violation:", "\n",␣
 ↪rialtoHIGH['Side Of Street'].value_counts(normalize=True)*100)

print("")

#SIGNS
print("Counting Signs in Violations \n", rialtoHIGH['Sign'].value_counts().
 ↪to_frame(), "\n")

print("Testing the Worst Ranked Signs\n")
#sign 1
print("SIGN ONE: 1/2P M-SAT 7:30-19:30")
RIALTOloss = rialtoHIGH[rialtoHIGH["Sign"] == "1/2P M-SAT 7:30-19:30"]
print("Value Counts for Parking Duration: ", RIALTOloss['Parking Duration (s)'].
 ↪value_counts().to_frame())
print("The time (hours) of the max length of stay is: ",␣
 ↪(round(RIALTOloss['Parking Duration (s)'].max()/60)/60), "Hours")
print("")
```

```
      Area         Street Street Limit 1     Street Limit 2  Side Of Street  \
33   Rialto  FLINDERS LANE    KING STREET     WILLIAM STREET               4
36   Rialto    KING STREET  BOURKE STREET  Lt COLLINS STREET               2
42   Rialto  FLINDERS LANE    KING STREET     WILLIAM STREET               4
```

```
66   Rialto  FLINDERS LANE    KING STREET     WILLIAM STREET               4
108  Rialto    KING STREET  BOURKE STREET  Lt COLLINS STREET               2

    Street Marker            Arrival Time          Departure Time  \
33          1863S  01/10/2011 12:01:25 AM  01/10/2011 05:49:44 AM
36          1460E  01/10/2011 12:01:26 AM  01/10/2011 04:25:02 AM
42          1861S  01/10/2011 12:01:29 AM  01/10/2011 03:23:15 AM
66          1859S  01/10/2011 12:10:54 AM  01/10/2011 01:15:15 AM
108         1446E  01/10/2011 12:45:43 AM  01/10/2011 01:00:36 AM

     Parking Duration (s)                            Sign  Violation  \
33                  20899         S/ No Stop M-S 0:00-6:00          1
36                  15816  S/ No Stop AOT Buses 0:00-23:59          1
42                  12106         S/ No Stop M-S 0:00-6:00          1
66                   3861         S/ No Stop M-S 0:00-6:00          1
108                   893  S/ No Stop AOT Buses 0:00-23:59          1

     Street ID  Device ID
33         669       1046
36         839        892
42         669       1085
66         669       1004
108        839        900
The exact number of infringements around Rialto:  74588

Streets in this area:  6
Finding the number of times a street has a violation:
                    Street
KING STREET         27627
COLLINS STREET      14094
FLINDERS LANE       10506
BOURKE STREET        9273
Lt COLLINS STREET    7783
CHURCH STREET        5305

Counts of Side of Street: 2

Finding the number of times a street has a violation:
    Side Of Street
4            7700
3            6394

The Percentages of each Side of Street in a Violation:
 4    54.633177
3    45.366823
Name: Side Of Street, dtype: float64

Counting Signs in Violations
```

```
                                Sign
1/2P M-SAT 7:30-19:30          8074
1/2P MTR M-F 7:30-16:30        2548
LZ 15M M-F 7:30-16.30          1619
S/ No Stop M-F 16:30-18:00      960
1/2P MTR M-F 18:00-19:30        524
2P MTR SAT 7:30-20:30           188
2P SAT 7:30-19:30               108
1/2P MTR M-SAT 7:30-19:30        73

Testing the Worst Ranked Signs

SIGN ONE: 1/2P M-SAT 7:30-19:30
Value Counts for Parking Duration:         Parking Duration (s)
43193                       20
43192                       20
2242                        13
2108                        12
2170                        11
…                          …
8650                         1
6601                         1
4548                         1
6593                         1
34821                        1

[3774 rows x 1 columns]
The time (hours) of the max length of stay is:  12.0 Hours
```

### 2.4.13 RACV

[16]:
```
racv = parkingDB[parkingDB["Area"] == "RACV"]
racv = racv[racv["Violation"] == 1]
print(racv.head())
print("The exact number of infringements around RACV: ", len(racv))
print("")

#STREETS
print("Streets in this area: ", racv['Street'].unique().size)
racv['Street'].unique()

print("Finding the number of times a street has a violation:\n", racv['Street'].
 ↪value_counts().to_frame())
racvHIGH = racv[racv["Street"].isin(['QUEENS STREET', 'Lt COLLINS STREET',␣
 ↪'BOURKE STREET'])]
print("")
```

```
#STREET SIDES
print("Counts of Side of Street:", racvHIGH['Side Of Street'].unique().size,␣
 ↪"\n")
racvHIGH['Side Of Street'].unique()

print("Finding the number of times a street has a violation:\n", racvHIGH['Side␣
 ↪Of Street'].value_counts().to_frame(), "\n")
print("The Percentages of each Side of Street in a Violation:", "\n",␣
 ↪racvHIGH['Side Of Street'].value_counts(normalize=True)*100)

print("")

#SIGNS
print("Counting Signs in Violations \n", racvHIGH['Sign'].value_counts().
 ↪to_frame(), "\n")

print("Testing the Worst Ranked Signs\n")
#sign 1
print("SIGN ONE: 1P MTR M-SAT 7:30-19:30")
RACVLOSS = racvHIGH[racvHIGH["Sign"] == "1P MTR M-SAT 7:30-19:30"]
print("Value Counts for Parking Duration: ", RACVLOSS['Parking Duration (s)'].
 ↪value_counts().to_frame())
print("The time (hours) of the max length of stay is: ",␣
 ↪(round(RACVLOSS['Parking Duration (s)'].max()/60)/60), "Hours")
print("")
```

```
      Area              Street  Street Limit 1    Street Limit 2  \
12    RACV       BOURKE STREET  WILLIAM STREET      QUEEN STREET
689   RACV       BOURKE STREET    QUEEN STREET  ELIZABETH STREET
754   RACV  Lt COLLINS STREET  WILLIAM STREET      QUEEN STREET
1329  RACV       BOURKE STREET  WILLIAM STREET      QUEEN STREET
1415  RACV  Lt COLLINS STREET  WILLIAM STREET      QUEEN STREET

      Side Of Street Street Marker            Arrival Time  \
12                 4         2417S  01/10/2011 12:01:17 AM
689                4         2399S  01/10/2011 07:30:07 AM
754                3         2232N  01/10/2011 07:30:07 AM
1329               4         2417S  01/10/2011 07:31:09 AM
1415               3         2238N  01/10/2011 07:34:42 AM

              Departure Time  Parking Duration (s)                    Sign  \
12    01/10/2011 07:24:42 AM                 26605  2P DIS M-SUN 0:00-23:59
689   01/10/2011 09:33:43 AM                  7416  1P MTR M-SAT 7:30-19:30
754   01/10/2011 08:52:34 AM                  4947  1P MTR M-SAT 7:30-19:30
1329  01/10/2011 10:13:27 AM                  9738  2P DIS M-SUN 0:00-23:59
1415  01/10/2011 08:57:13 AM                  4951  1P MTR M-SAT 7:30-19:30
```

```
      Violation  Street ID  Device ID
12            1        123       1338
689           1        123       1349
754           1        911       1074
1329          1        123       1338
1415          1        911       1094
The exact number of infringements around RACV:  70159

Streets in this area:  4
Finding the number of times a street has a violation:
                    Street
QUEEN STREET        24561
BOURKE STREET       20782
Lt COLLINS STREET   19096
WILLIAM STREET       5720

Counts of Side of Street: 2

Finding the number of times a street has a violation:
    Side Of Street
3           24239
4           15639

The Percentages of each Side of Street in a Violation:
 3    60.782888
4    39.217112
Name: Side Of Street, dtype: float64

Counting Signs in Violations
                                    Sign
1P MTR M-SAT 7:30-19:30            21395
P10 M-F 7:30-19:30                  5054
1/2P MTR M-SAT 7:30-19:30           3349
1P SUN 7:30-18:30                   1961
LZ 15M M-SAT 7:30-19:30             1533
1/4P M-SAT 16:00-19:30              1487
1P DIS M-SUN 0:00-23:59             1407
2P DIS M-SUN 0:00-23:59             1318
1P MTR M-SAT 9:30-16:00              967
1/4P M-SAT 7:30-9:30                 774
1P SAT 7:30-19:30                    380
2P SUN 7:30-18:30                    243
2P DIS M-F 7:30-18:30                  4
Temp Sign Plate Sun-Sun 4:00-5:00      3
2P DIS SAT 7:30-20:30                  3

Testing the Worst Ranked Signs
```

```
SIGN ONE: 1P MTR M-SAT 7:30-19:30
Value Counts for Parking Duration:         Parking Duration (s)
43192                      97
43193                      39
3951                       27
3954                       25
3932                       23
…                         …
8786                        1
35399                       1
6725                        1
8754                        1
12350                       1

[6549 rows x 1 columns]
The time (hours) of the max length of stay is:  12.0 Hours
```

### 2.4.14 County

[17]:
```
county = parkingDB[parkingDB["Area"] == "County"]
county = county[county["Violation"] == 1]
print(county.head())
print("The exact number of infringements around County: ", len(county))
print("")

#STREETS
print("Streets in this area: ", county['Street'].unique().size)
county['Street'].unique()

print("Finding the number of times a street has a violation:\n",␣
 ↪county['Street'].value_counts().to_frame())
countyHIGH = county[county["Street"].isin(['LONSDALE STREET', 'Lt BOURKE␣
 ↪STREET', 'SPENCER STREET'])]
print("")

#STREET SIDES
print("Counts of Side of Street:", countyHIGH['Side Of Street'].unique().size,␣
 ↪"\n")
countyHIGH['Side Of Street'].unique()

print("Finding the number of times a street has a violation:\n",␣
 ↪countyHIGH['Side Of Street'].value_counts().to_frame(), "\n")
print("The Percentages of each Side of Street in a Violation:", "\n",␣
 ↪countyHIGH['Side Of Street'].value_counts(normalize=True)*100)
countyHIGH134 = countyHIGH[countyHIGH["Side Of Street"].isin([1, 4, 3])]
```

```
print("")

#SIGNS
print("Counting Signs in Violations \n", countyHIGH134['Sign'].value_counts().
 ↪to_frame(), "\n")

print("Testing the Worst Ranked Signs\n")
#sign 1
print("SIGN ONE: 1P MTR M-SAT 7:30-19:30")
countyLOSS = countyHIGH134[countyHIGH134["Sign"] == "1P MTR M-SAT 7:30-19:30"]
print("Value Counts for Parking Duration: ", countyLOSS['Parking Duration (s)'].
 ↪value_counts().to_frame())
print("The time (hours) of the max length of stay is: ",␣
 ↪(round(countyLOSS['Parking Duration (s)'].max()/60)/60), "Hours")
print("")
```

```
       Area           Street Street Limit 1  Street Limit 2  Side Of Street  \
29    County  LONSDALE STREET    KING STREET  WILLIAM STREET               1
121   County  LONSDALE STREET    KING STREET  WILLIAM STREET               1
156   County  LONSDALE STREET    KING STREET  WILLIAM STREET               1
157   County  LONSDALE STREET    KING STREET  WILLIAM STREET               1
158   County  LONSDALE STREET    KING STREET  WILLIAM STREET               1

    Street Marker            Arrival Time          Departure Time  \
29          C3150  01/10/2011 12:01:24 AM  01/10/2011 12:02:37 AM
121         C3144  01/10/2011 01:00:49 AM  01/10/2011 01:03:07 AM
156         C3146  01/10/2011 01:41:33 AM  01/10/2011 01:46:41 AM
157         C3142  01/10/2011 01:41:48 AM  01/10/2011 01:49:43 AM
158         C3148  01/10/2011 01:42:02 AM  01/10/2011 01:47:09 AM

     Parking Duration (s)                                  Sign  Violation  \
29                     73  S/ No Stop Auth Veh M-Sun 00:00-23:59          1
121                   138  S/ No Stop Auth Veh M-Sun 00:00-23:59          1
156                   308  S/ No Stop Auth Veh M-Sun 00:00-23:59          1
157                   475  S/ No Stop Auth Veh M-Sun 00:00-23:59          1
158                   307  S/ No Stop Auth Veh M-Sun 00:00-23:59          1

     Street ID  Device ID
29         894       2038
121        894       2086
156        894       2120
157        894       2115
158        894       2060
The exact number of infringements around County:  62866

Streets in this area:  4
Finding the number of times a street has a violation:
```

```
                   Street
LONSDALE STREET    33846
Lt BOURKE STREET   11717
SPENCER STREET     11447
KING STREET         5856

Counts of Side of Street: 5

Finding the number of times a street has a violation:
    Side Of Street
1           18928
3           14034
4           12601
5            8466
2            2981

The Percentages of each Side of Street in a Violation:
 1    33.201193
3    24.616734
4    22.103140
5    14.850026
2     5.228907
Name: Side Of Street, dtype: float64

Counting Signs in Violations
                                          Sign
1P MTR M-SAT 7:30-19:30                  16499
CW TOW M-F 16:00-18:30                    7033
1P TKT A M-SAT 7:30-19:30                 5654
CW TOW M-F 7:00-9:30                      3840
S/ No Stop Auth Veh M-Sun 00:00-23:59     2873
2P MTR M-SAT 7:30-20:30                   1681
1P MTR M-F 7:30-16:00                     1232
S/ No Stop Auth Veh M-Sun 0:00-23-59      1080
1/2P MTR M-SAT 7:30-19:30                  995
1/2P MTR M-F 9:30-19:30                    962
1P MTR M-F 9:30-19:30                      793
2P SUN 7:30-18:30                          774
LZ 30M M-F 7:30-19:30                      584
1/4P M-F 7:30-16:00                        337
2P DIS M-SUN 0:00-23:59                    331
1P MTR SAT 7:30-19:30                      308
1/4P M-F 7:30-19:30                        138
2P DIS AOT 0:00-23:59                      132
2P SAT 7:30-19:30                          123
2P DIS AOT 9:30-23:59                       97
1/4P M-F 18:30-19:30                        53
2P DIS AOT 0:00-7:00                        34
```

```
Temp Sign Plate Sun-Sun 4:00-5:00         10

Testing the Worst Ranked Signs

SIGN ONE: 1P MTR M-SAT 7:30-19:30
Value Counts for Parking Duration:         Parking Duration (s)
43192                    605
43193                    465
4015                      17
43191                     16
4092                      16
…                        …
22653                      1
15852                      1
7672                       1
10355                      1
14049                      1

[6507 rows x 1 columns]
The time (hours) of the max length of stay is:  12.0 Hours
```

### 2.4.15 REGENCY

[18]:
```
regency = parkingDB[parkingDB["Area"] == "Regency"]
regency = regency[regency["Violation"] == 1]
print(regency.head())
print("The exact number of infringements around Regency: ", len(regency))
print("")

#STREETS
print("Streets in this area: ", regency['Street'].unique().size)
regency['Street'].unique()

print("Finding the number of times a street has a violation:\n",␣
 ↪regency['Street'].value_counts().to_frame())
regencyHIGH = regency[regency["Street"].isin(['EXHIBITION STREET', 'SPRING␣
 ↪STREET', 'RUSSELL STREET'])]
print("")

#STREET SIDES
print("Counts of Side of Street:", regencyHIGH['Side Of Street'].unique().size,␣
 ↪"\n")
regencyHIGH['Side Of Street'].unique()

print("Finding the number of times a street has a violation:\n",␣
 ↪regencyHIGH['Side Of Street'].value_counts().to_frame(), "\n")
```

```
print("The Percentages of each Side of Street in a Violation:", "\n",␣
 ↪regencyHIGH['Side Of Street'].value_counts(normalize=True)*100)
# countyHIGH134 = regencyHIGH[regencyHIGH["Side Of Street"].isin([1, 4, 3])]

print("")

#SIGNS
print("Counting Signs in Violations \n", regencyHIGH['Sign'].value_counts().
 ↪to_frame(), "\n")

print("Testing the Worst Ranked Signs\n")
#sign 1
print("SIGN ONE: 2P MTR M-SAT 7:30-20:30")
REGLOSS = regencyHIGH[regencyHIGH["Sign"] == "2P MTR M-SAT 7:30-20:30"]
print("Value Counts for Parking Duration: ", REGLOSS['Parking Duration (s)'].
 ↪value_counts().to_frame())
print("The time (hours) of the max length of stay is: ",␣
 ↪(round(REGLOSS['Parking Duration (s)'].max()/60)/60), "Hours")
print("")
```

```
        Area              Street      Street Limit 1      Street Limit 2  \
1089  Regency      SPRING STREET     VICTORIA PARADE  Lt LONSDALE STREET
1096  Regency      SPRING STREET     VICTORIA PARADE     LONSDALE STREET
1097  Regency  EXHIBITION STREET  Lt LONSDALE STREET     LONSDALE STREET
1098  Regency  EXHIBITION STREET  Lt LONSDALE STREET     LONSDALE STREET
1196  Regency     RUSSELL STREET     VICTORIA STREET     LA TROBE STREET

      Side Of Street Street Marker            Arrival Time  \
1089               1          C188  01/10/2011 07:30:08 AM
1096               2           82E  01/10/2011 07:30:08 AM
1097               1          C544  01/10/2011 07:30:08 AM
1098               1          C548  01/10/2011 07:30:08 AM
1196               2         4574E  01/10/2011 07:30:08 AM

              Departure Time  Parking Duration (s)                     Sign  \
1089  01/10/2011 10:41:17 AM                 11469  2P MTR M-SAT 7:30-20:30
1096  01/10/2011 09:36:10 AM                  7562    2P MTR SAT 7:30-20:30
1097  01/10/2011 10:17:09 AM                 10021  2P MTR M-SAT 7:30-20:30
1098  01/10/2011 08:30:00 PM                 46792  2P MTR M-SAT 7:30-20:30
1196  01/10/2011 01:12:47 PM                 20559  2P MTR M-SAT 7:30-20:30

      Violation  Street ID  Device ID
1089          1       1288        125
1096          1       1288         60
1097          1        647        378
1098          1        647        343
1196          1       1221       3557
```

```
The exact number of infringements around Regency:  58816

Streets in this area:  6
Finding the number of times a street has a violation:
                     Street
EXHIBITION STREET    17403
SPRING STREET        13573
RUSSELL STREET       10192
Lt LONSDALE STREET    9533
LA TROBE STREET       5806
MACKENZIE STREET      2309

Counts of Side of Street: 3

Finding the number of times a street has a violation:
    Side Of Street
1           23439
2            9929
5            7800

The Percentages of each Side of Street in a Violation:
 1    56.934998
2    24.118247
5    18.946755
Name: Side Of Street, dtype: float64

Counting Signs in Violations
                                Sign
2P MTR M-SAT 7:30-20:30        22703
2P SUN 7:30-18:30               5114
2P TKT A M-SAT 7:30-20:30       2836
1P M-SAT 7:30-19:30             2474
2P MTR M-F 9:30-20:30           1317
2P MTR M-F 7:30-16:00           1244
1P MTR M-SAT 7:30-19:30          999
3P MTR M-SAT 7:30-20:30          936
2P M-SAT 7:30-20:30              892
2P DIS M-SUN 0:00-23:59          851
1P SUN 7:30-18:30                745
2P MTR SAT 7:30-20:30            408
S/ No Stop M-F  16:00-18:30      360
2P MTR M-SAT 7:00-20:30          171
CW M-F 7:30-9:30                  81
1/4P M-F 7:30-19:30               34
2P SAT 7:30-19:30                  3

Testing the Worst Ranked Signs
```

```
SIGN ONE: 2P MTR M-SAT 7:30-20:30
Value Counts for Parking Duration:         Parking Duration (s)
46792                     274
46793                      84
46791                      40
7690                       23
46789                      21
…                         …
14110                       1
16159                       1
20257                       1
30502                       1
32768                       1

[8301 rows x 1 columns]
The time (hours) of the max length of stay is:  13.0 Hours
```

### 1.2.8 2.4.16 Jolimont

[19]:
```
jolimont = parkingDB[parkingDB["Area"] == "Jolimont"]
jolimont = jolimont[jolimont["Violation"] == 1]
print(jolimont.head())
print("The exact number of infringements around Jolimont: ", len(jolimont))
print("")

#STREETS
print("Streets in this area: ", jolimont['Street'].unique().size)
jolimont['Street'].unique()

print("Finding the number of times a street has a violation:\n",␣
 ↪jolimont['Street'].value_counts().to_frame())
jolimontHIGH = jolimont[jolimont["Street"].isin(['ALBERT STREET', 'ST ANDREWS␣
 ↪PLACE'])]
print("")

#STREET SIDES
print("Counts of Side of Street:", jolimontHIGH['Side Of Street'].unique().
 ↪size, "\n")
jolimontHIGH['Side Of Street'].unique()

print("Finding the number of times a street has a violation:\n",␣
 ↪jolimontHIGH['Side Of Street'].value_counts().to_frame(), "\n")
print("The Percentages of each Side of Street in a Violation:", "\n",␣
 ↪jolimontHIGH['Side Of Street'].value_counts(normalize=True)*100)
# countyHIGH134 = regencyHIGH[regencyHIGH["Side Of Street"].isin([1, 4, 3])]
```

```
print("")

#SIGNS
print("Counting Signs in Violations \n", jolimontHIGH['Sign'].value_counts().
 ↪to_frame(), "\n")

print("Testing the Worst Ranked Signs\n")
#sign 1
print("SIGN ONE: 1P MTR M-SAT 7:30-18:30")
JOLLOSS = jolimontHIGH[jolimontHIGH["Sign"] == "1P MTR M-SAT 7:30-18:30"]
print("Value Counts for Parking Duration: ", JOLLOSS['Parking Duration (s)'].
 ↪value_counts().to_frame())
print("The time (hours) of the max length of stay is: ",␣
 ↪(round(JOLLOSS['Parking Duration (s)'].max()/60)/60), "Hours")
print("")
```

```
            Area            Street Street Limit 1   Street Limit 2  \
7333474   Jolimont  LANSDOWNE STREET  ALBERT STREET  CATHEDRAL PLACE
7333584   Jolimont  LANSDOWNE STREET  ALBERT STREET  CATHEDRAL PLACE
7333876   Jolimont  LANSDOWNE STREET  ALBERT STREET  CATHEDRAL PLACE
7334921   Jolimont  LANSDOWNE STREET  ALBERT STREET  CATHEDRAL PLACE
7335039   Jolimont  LANSDOWNE STREET  ALBERT STREET  CATHEDRAL PLACE

         Side Of Street Street Marker            Arrival Time  \
7333474               5        11895W  07/05/2012 07:30:08 AM
7333584               2        11866E  07/05/2012 07:30:08 AM
7333876               2        11898E  07/05/2012 07:33:55 AM
7334921               2        11874E  07/05/2012 08:04:45 AM
7335039               2        11878E  07/05/2012 08:08:06 AM

                Departure Time  Parking Duration (s)                  Sign  \
7333474  07/05/2012 09:03:03 AM                  5575  1P MTR M-F 7:30-18:30
7333584  07/05/2012 03:58:38 PM                 30510  2P MTR M-F 7:30-18:30
7333876  07/05/2012 12:01:08 PM                 16033  2P MTR M-F 7:30-18:30
7334921  07/05/2012 05:44:27 PM                 34782  2P MTR M-F 7:30-18:30
7335039  07/05/2012 03:46:27 PM                 27501  2P MTR M-F 7:30-18:30

         Violation  Street ID  Device ID
7333474          1        869       6321
7333584          1        869       6294
7333876          1        869       6324
7334921          1        869       6300
7335039          1        869       6304
The exact number of infringements around Jolimont:  54712

Streets in this area:  9
Finding the number of times a street has a violation:
```

```
                    Street
ALBERT STREET       14491
ST ANDREWS PLACE    11510
LANSDOWNE STREET     8303
CLARENDON STREET     7412
CATHEDRAL PLACE      5548
GISBORNE STREET      2242
NICHOLSON STREET     2072
PARLIAMENT PLACE     2063
WELLINGTON PARADE    1071

Counts of Side of Street: 2

Finding the number of times a street has a violation:
    Side Of Street
4           16198
3            9803

The Percentages of each Side of Street in a Violation:
 4    62.297604
3    37.702396
Name: Side Of Street, dtype: float64

Counting Signs in Violations
                          Sign
1P MTR M-SAT 7:30-18:30  8296
1P MTR M-F 9:30-18:30    4049
1P MTR M-F 7:30-16:00    4014
1/4P M-F 7:30-16:00      3138
2P MTR M-F 9:30-18:30    2108
2P MTR M-SAT 7:30-18:30  1502
1P SUN 7:30-18:30        1402
2P MTR M-F 7:30-16:00     510
2P SUN 7:30-18:30         310
2P MTR SAT 7:30-12:30     308
1P MTR SAT 7:30-12:30     226
1/4P SAT 7:30-12:30        89
1P MTR SAT 7:30-12:00      45
1/4P SAT 7:30-12:00         4

Testing the Worst Ranked Signs

SIGN ONE: 1P MTR M-SAT 7:30-18:30
Value Counts for Parking Duration:         Parking Duration (s)
39593                    159
39592                     23
4066                      11
3941                      10
```

```
4073                      10
…                         …
6228                       1
6105                       1
10199                      1
8144                       1
14732                      1

[4345 rows x 1 columns]
The time (hours) of the max length of stay is:  11.0 Hours
```

### 2.4.18 Magistrates

[20]:
```
magistrates = parkingDB[parkingDB["Area"] == "Magistrates"]
magistrates = magistrates[magistrates["Violation"] == 1]
print(magistrates.head())
print("The exact number of infringements around the Magistrates: ",␣
 ↪len(magistrates))
print("")

#STREETS
print("Streets in this area: ", magistrates['Street'].unique().size)
magistrates['Street'].unique()

print("Finding the number of times a street has a violation:\n",␣
 ↪magistrates['Street'].value_counts().to_frame())
magistratesHIGH = magistrates[magistrates["Street"].isin(['LA TROBE STREET',␣
 ↪'Lt LONSDALE STREET'])]
print("")

#STREET SIDES
print("Counts of Side of Street:", magistratesHIGH['Side Of Street'].unique().
 ↪size, "\n")
magistratesHIGH['Side Of Street'].unique()

print("Finding the number of times a street has a violation:\n",␣
 ↪magistratesHIGH['Side Of Street'].value_counts().to_frame(), "\n")
print("The Percentages of each Side of Street in a Violation:", "\n",␣
 ↪magistratesHIGH['Side Of Street'].value_counts(normalize=True)*100)
# countyHIGH134 = regencyHIGH[regencyHIGH["Side Of Street"].isin([1, 4, 3])]

print("")

#SIGNS
print("Counting Signs in Violations \n", magistratesHIGH['Sign'].value_counts().
 ↪to_frame(), "\n")
```

```
print("Testing the Worst Ranked Signs\n")
#sign 1
print("SIGN ONE: 2P MTR M-SAT 7:30-20:30")
magLoss = magistratesHIGH[magistratesHIGH["Sign"] == "2P MTR M-SAT 7:30-20:30"]
print("Value Counts for Parking Duration: ", magLoss['Parking Duration (s)'].
 ↪value_counts().to_frame())
print("The time (hours) of the max length of stay is: ",␣
 ↪(round(magLoss['Parking Duration (s)'].max()/60)/60), "Hours")
print("")
```

```
             Area              Street      Street Limit 1      Street Limit 2  \
24    Magistrates  Lt LONSDALE STREET      SPENCER STREET         KING STREET
954   Magistrates         KING STREET     LA TROBE STREET  Lt LONSDALE STREET
1210  Magistrates         KING STREET  Lt LONSDALE STREET     LONSDALE STREET
1236  Magistrates         KING STREET  Lt LONSDALE STREET     LONSDALE STREET
1243  Magistrates         KING STREET  Lt LONSDALE STREET     LONSDALE STREET

      Side Of Street Street Marker            Arrival Time  \
24                 3         3578N  01/10/2011 12:01:22 AM
954                2         4470E  01/10/2011 07:30:07 AM
1210               2         4458E  01/10/2011 07:30:08 AM
1236               5         4465W  01/10/2011 07:30:08 AM
1243               2         4464E  01/10/2011 07:30:08 AM

              Departure Time  Parking Duration (s)                   Sign  \
24    01/10/2011 11:59:00 PM                 86258      4P DIS ONLY M-SUN
954   01/10/2011 09:36:53 AM                  7606  2P MTR SAT 7:30-20:30
1210  01/10/2011 09:39:14 AM                  7746  2P MTR SAT 7:30-20:30
1236  01/10/2011 07:58:54 AM                  1726     P10 SAT 7:30-19:30
1243  01/10/2011 09:37:55 AM                  7667  2P MTR SAT 7:30-20:30

      Violation  Street ID  Device ID
24            1        926       1559
954           1        839        895
1210          1        839        777
1236          1        839        960
1243          1        839        901
The exact number of infringements around the Magistrates:  37862

Streets in this area:  5
Finding the number of times a street has a violation:
                     Street
Lt LONSDALE STREET   14314
LA TROBE STREET      14091
KING STREET           6990
SPENCER STREET        1317
```

```
LONSDALE STREET       1150

Counts of Side of Street: 2

Finding the number of times a street has a violation:
    Side Of Street
4           15777
3           12628

The Percentages of each Side of Street in a Violation:
 4    55.543038
3    44.456962
Name: Side Of Street, dtype: float64

Counting Signs in Violations
                                Sign
2P MTR M-SAT 7:30-20:30        10425
2P TKT A M-SAT 7:30-20:30       8945
1/2P TKT A M-SAT 7:30-19:30     2671
1/2P MTR M-SAT 7:30-19:30       2451
P/ 5 M-SAT 7:30-19:30           2099
2P SUN 7:30-18:30                890
LZ 30M M-SAT 7:30-19:30          501
4P DIS ONLY M-SUN                189
2P SUN 7:30-23:00                100
2P M-SAT 20:30-23:00              65
3P DIS M-SAT 7:30-19:30           39
LZ 30M SUN 7:30-18:30             30

Testing the Worst Ranked Signs

SIGN ONE: 2P MTR M-SAT 7:30-20:30
Value Counts for Parking Duration:         Parking Duration (s)
46792                   1439
46793                    353
46791                    148
46790                     84
46789                     54
…                        …
13962                      1
11913                      1
9864                       1
7815                       1
12246                      1

[5351 rows x 1 columns]
The time (hours) of the max length of stay is:  13.0 Hours
```

### 2.3.19 Supreme

[21]:
```
supreme = parkingDB[parkingDB["Area"] == "Supreme"]
supreme = supreme[supreme["Violation"] == 1]
print(supreme.head())
print("The exact number of infringements around Supreme: ", len(supreme))
print("")

#STREETS
print("Streets in this area: ", supreme['Street'].unique().size)
supreme['Street'].unique()

print("Finding the number of times a street has a violation:\n",␣
 ↪supreme['Street'].value_counts().to_frame())
supremeHIGH = supreme[supreme["Street"].isin(['LONSDALE STREET'])]
print("")

#STREET SIDES
print("Counts of Side of Street:", supremeHIGH['Side Of Street'].unique().size,␣
 ↪"\n")
supremeHIGH['Side Of Street'].unique()

print("Finding the number of times a street has a violation:\n",␣
 ↪supremeHIGH['Side Of Street'].value_counts().to_frame(), "\n")
print("The Percentages of each Side of Street in a Violation:", "\n",␣
 ↪supremeHIGH['Side Of Street'].value_counts(normalize=True)*100)
supremeHIGH1 = supremeHIGH[supremeHIGH["Side Of Street"].isin([1])]

print("")

#SIGNS
print("Counting Signs in Violations \n", supremeHIGH1['Sign'].value_counts().
 ↪to_frame(), "\n")

print("Testing the Worst Ranked Signs\n")
#sign 1
print("SIGN ONE: 1P MTR M-SAT 7:30-19:30")
supLoss = supremeHIGH1[supremeHIGH1["Sign"] == "1P MTR M-SAT 7:30-19:30"]
print("Value Counts for Parking Duration: ", supLoss['Parking Duration (s)'].
 ↪value_counts().to_frame())
print("The time (hours) of the max length of stay is: ",␣
 ↪(round(supLoss['Parking Duration (s)'].max()/60)/60), "Hours")
print("")
```

```
          Area           Street  Street Limit 1 Street Limit 2  \
1116   Supreme  LONSDALE STREET  WILLIAM STREET   QUEEN STREET
8435   Supreme  LONSDALE STREET  WILLIAM STREET   QUEEN STREET
8901   Supreme  LONSDALE STREET  WILLIAM STREET   QUEEN STREET
```

```
11067  Supreme  LONSDALE STREET  WILLIAM STREET   QUEEN STREET
15909  Supreme  LONSDALE STREET  WILLIAM STREET   QUEEN STREET

       Side Of Street Street Marker            Arrival Time  \
1116                1         C3076  01/10/2011 07:30:08 AM
8435                1         C3014  01/10/2011 11:14:27 AM
8901                1         C3010  01/10/2011 11:26:30 AM
11067               1         C3026  01/10/2011 12:23:34 PM
15909               1         C3126  01/10/2011 03:19:02 PM

               Departure Time  Parking Duration (s)                     Sign  \
1116   01/10/2011 07:30:00 PM                 43192  1P MTR M-SAT 7:30-19:30
8435   01/10/2011 01:01:52 PM                  6445  1P MTR M-SAT 7:30-19:30
8901   01/10/2011 12:37:02 PM                  4232  1P MTR M-SAT 7:30-19:30
11067  01/10/2011 03:08:47 PM                  9913  1P MTR M-SAT 7:30-19:30
15909  01/10/2011 05:39:29 PM                  8427  1P MTR M-SAT 7:30-19:30

       Violation  Street ID  Device ID
1116           1        894       2124
8435           1        894       1947
8901           1        894       1956
11067          1        894       2084
15909          1        894       2054
The exact number of infringements around Supreme:  25824

Streets in this area:  1
Finding the number of times a street has a violation:
                  Street
LONSDALE STREET   25824

Counts of Side of Street: 3

Finding the number of times a street has a violation:
    Side Of Street
1           18547
3            3859
4            3418

The Percentages of each Side of Street in a Violation:
 1    71.820787
3    14.943463
4    13.235750
Name: Side Of Street, dtype: float64

Counting Signs in Violations
                            Sign
1P MTR M-SAT 7:30-19:30  18547
```

```
Testing the Worst Ranked Signs

SIGN ONE: 1P MTR M-SAT 7:30-19:30
Value Counts for Parking Duration:         Parking Duration (s)
43192                    525
43193                    394
3946                      20
4118                      20
4005                      17
…                        …
22002                     1
7675                      1
30214                     1
17936                     1
18335                     1

[7203 rows x 1 columns]
The time (hours) of the max length of stay is:  12.0 Hours
```

### 2.3.20 Tavistock

[14]:
```
tavistock = parkingDB[parkingDB["Area"] == "Tavistock"]
tavistock = tavistock[tavistock["Violation"] == 1]
print(tavistock.head())
print("The exact number of infringements around Tavistock: ", len(tavistock))
print("")

#STREETS
print("Streets in this area: ", tavistock['Street'].unique().size)
tavistock['Street'].unique()

print("Finding the number of times a street has a violation:\n",␣
 ↪tavistock['Street'].value_counts().to_frame())
tavistockHIGH = tavistock[tavistock["Street"].isin(['COLLINS STREET'])]
print("")

#STREET SIDES
print("Counts of Side of Street:", tavistockHIGH['Side Of Street'].unique().
 ↪size, "\n")
tavistockHIGH['Side Of Street'].unique()

print("Finding the number of times a street has a violation:\n",␣
 ↪tavistockHIGH['Side Of Street'].value_counts().to_frame(), "\n")
print("The Percentages of each Side of Street in a Violation:", "\n",␣
 ↪tavistockHIGH['Side Of Street'].value_counts(normalize=True)*100)

print("")
```

```
#SIGNS
print("Counting Signs in Violations \n", tavistockHIGH['Sign'].value_counts().
 ↪to_frame(), "\n")

print("Testing the Worst Ranked Signs\n")
#sign 1
print("SIGN ONE: 1P MTR M-SAT 7:30-19:30")
tavistock = tavistockHIGH[tavistockHIGH["Sign"] == "1P MTR M-SAT 7:30-19:30"]
print("Value Counts for Parking Duration: ", tavistock['Parking Duration (s)'].
 ↪value_counts().to_frame())
print("The time (hours) of the max length of stay is: ",␣
 ↪(round(tavistock['Parking Duration (s)'].max()/60)/60), "Hours")
print("")
```

```
          Area          Street  Street Limit 1 Street Limit 2  \
1004  Tavistock  COLLINS STREET  WILLIAM STREET   QUEEN STREET
1048  Tavistock  COLLINS STREET  WILLIAM STREET   QUEEN STREET
4446  Tavistock  COLLINS STREET  WILLIAM STREET   QUEEN STREET
4954  Tavistock  COLLINS STREET  WILLIAM STREET   QUEEN STREET
4965  Tavistock  COLLINS STREET  WILLIAM STREET   QUEEN STREET

      Side Of Street Street Marker            Arrival Time  \
1004               3         2018N  01/10/2011 07:30:08 AM
1048               3         2026N  01/10/2011 07:30:08 AM
4446               3         2012N  01/10/2011 09:30:53 AM
4954               3         2014N  01/10/2011 09:45:46 AM
4965               3         2028N  01/10/2011 09:46:05 AM

              Departure Time  Parking Duration (s)                     Sign  \
1004  01/10/2011 10:46:38 AM                 11790  1P MTR M-SAT 7:30-19:30
1048  01/10/2011 09:59:17 AM                  8949  1P MTR M-SAT 7:30-19:30
4446  01/10/2011 11:22:00 AM                  6667  1P MTR M-SAT 7:30-19:30
4954  01/10/2011 11:02:48 AM                  4622  1P MTR M-SAT 7:30-19:30
4965  01/10/2011 11:23:33 AM                  5848  1P MTR M-SAT 7:30-19:30

      Violation  Street ID  Device ID
1004          1        528       1254
1048          1        528       1245
4446          1        528       1141
4954          1        528       1041
4965          1        528       1252
The exact number of infringements around Tavistock:  9398

Streets in this area:  1
Finding the number of times a street has a violation:
                Street
```

```
COLLINS STREET    9398

Counts of Side of Street: 2

Finding the number of times a street has a violation:
    Side Of Street
3            5248
4            4150

The Percentages of each Side of Street in a Violation:
 3    55.841668
4    44.158332
Name: Side Of Street, dtype: float64

Counting Signs in Violations
                           Sign
1P MTR M-SAT 7:30-19:30  9398

Testing the Worst Ranked Signs

SIGN ONE: 1P MTR M-SAT 7:30-19:30
Value Counts for Parking Duration:         Parking Duration (s)
43192                    145
43193                     51
3939                      14
3962                      14
3911                      13
…                        …
8328                       1
8320                       1
6271                       1
12410                      1
24496                      1

[4241 rows x 1 columns]
The time (hours) of the max length of stay is:  12.0 Hours
```

### 1.2.9 1.3.21 Docklands

[23]:
```
docklands = parkingDB[parkingDB["Area"] == "Docklands"]
docklands = docklands[docklands["Violation"] == 1]
print(docklands.head())
print("The exact number of infringements around Docklands: ", len(docklands))
print("")

#STREETS
```

```
print("Streets in this area: ", docklands['Street'].unique().size)
docklands['Street'].unique()

print("Finding the number of times a street has a violation:\n",␣
 ↪docklands['Street'].value_counts().to_frame())
print("All streets will be included given that the number of violations are␣
 ↪minimal.")

#STREET SIDES
print("Counts of Side of Street:", docklands['Side Of Street'].unique().size,␣
 ↪"\n")
docklands['Side Of Street'].unique()

print("Finding the number of times a street has a violation:\n",␣
 ↪docklands['Side Of Street'].value_counts().to_frame(), "\n")
print("The Percentages of each Side of Street in a Violation:", "\n",␣
 ↪docklands['Side Of Street'].value_counts(normalize=True)*100)
docklands431 = docklands[docklands["Side Of Street"].isin([1, 3, 4])]

print("")

#SIGNS
print("Counting Signs in Violations \n", docklands431['Sign'].value_counts().
 ↪to_frame(), "\n")

print("Testing the Worst Ranked Signs\n")
#sign 1
print("SIGN ONE: 1P TKT RPE M-SAT 7:30-23:00")
docklandsLoss = docklands431[docklands431["Sign"] == "1P TKT RPE M-SAT 7:30-23:
 ↪00 "]
print("Value Counts for Parking Duration: ", docklandsLoss['Parking Duration␣
 ↪(s)'].value_counts().to_frame())
print("The time (hours) of the max length of stay is: ",␣
 ↪(round(docklandsLoss['Parking Duration (s)'].max()/60)/60), "Hours")
print("")
#sign 2
print("SIGN TWO: 1P RPA M-F 7:30-18:30")
docklandsLoss = docklands431[docklands431["Sign"] == "1P RPA M-F 7:30-18:30"]
print("Value Counts for Parking Duration: ", docklandsLoss['Parking Duration␣
 ↪(s)'].value_counts().to_frame())
print("The time (hours) of the max length of stay is: ",␣
 ↪(round(docklandsLoss['Parking Duration (s)'].max()/60)/60), "Hours")
print("")
```

```
          Area           Street   Street Limit 1 Street Limit 2  \
676  Docklands  ANDERSON STREET  VICTORIA STREET  MILLER STREET
677  Docklands  ANDERSON STREET  VICTORIA STREET  MILLER STREET
```

```
678  Docklands  ANDERSON STREET  VICTORIA STREET  MILLER STREET
679  Docklands  ANDERSON STREET  VICTORIA STREET  MILLER STREET
944  Docklands  ANDERSON STREET  VICTORIA STREET  MILLER STREET

     Side Of Street Street Marker            Arrival Time  \
676               1        C10218  01/10/2011 07:30:07 AM
677               1        C10224  01/10/2011 07:30:07 AM
678               1        C10226  01/10/2011 07:30:07 AM
679               1        C10242  01/10/2011 07:30:07 AM
944               1        C10216  01/10/2011 07:30:07 AM

             Departure Time  Parking Duration (s)                    Sign  \
676  01/10/2011 12:30:00 PM                 17993  1P RPA SAT 7:30-12:30
677  01/10/2011 08:46:10 AM                  4563  1P RPA SAT 7:30-12:30
678  01/10/2011 12:30:00 PM                 17993  1P RPA SAT 7:30-12:30
679  01/10/2011 09:59:37 AM                  8970  1P RPA SAT 7:30-12:30
944  01/10/2011 09:52:00 AM                  8513  1P RPA SAT 7:30-12:30

     Violation  Street ID  Device ID
676          1         28       2388
677          1         28       2545
678          1         28       2586
679          1         28       2531
944          1         28       2483
The exact number of infringements around Docklands:  8770

Streets in this area:  5
Finding the number of times a street has a violation:
                  Street
JEFFCOTT STREET     5523
ANDERSON STREET     1509
SPENCER STREET      1358
KING STREET          321
BATMAN STREET         59
All streets will be included given that the number of violations are minimal.
Counts of Side of Street: 5

Finding the number of times a street has a violation:
    Side Of Street
4            3525
3            1998
1            1568
5             990
2             689

The Percentages of each Side of Street in a Violation:
 4    40.193843
3    22.782212
```

```
1    17.879133
5    11.288483
2     7.856328
Name: Side Of Street, dtype: float64

Counting Signs in Violations
                                     Sign
1P TKT RPE M-SAT 7:30-23:00        3074
1P RPA M-F 7:30-18:30              1329
1/4P M-F 7:30-18:30                 910
2P TKT A M-F 7:30-18:30             837
1P RPE SUN 7:30-23:00               545
1P RPA SAT 7:30-12:30               176
2P TKT A SAT 7:30-12:30             113
4P MTR M-F 7:30-18:30                58
2P SAT 7:30-12:30                    44
Temp Sign Plate Sun-Sun 4:00-5:00     4
4P MTR SAT 7:30-12:30                 1

Testing the Worst Ranked Signs

SIGN ONE: 1P TKT RPE M-SAT 7:30-23:00
Value Counts for Parking Duration:         Parking Duration (s)
55792                    248
55793                     83
55791                     25
55789                     22
55790                     14
…                        …
6752                       1
4703                       1
6750                       1
4701                       1
14764                      1

[2503 rows x 1 columns]
The time (hours) of the max length of stay is:  15.5 Hours

SIGN TWO: 1P RPA M-F 7:30-18:30
Value Counts for Parking Duration:         Parking Duration (s)
39593                    156
39592                     87
39591                     11
39039                      2
11084                      2
…                        …
32067                      1
5440                       1
```

```
30264                    1
9534                     1
30720                    1

[1060 rows x 1 columns]
The time (hours) of the max length of stay is:  11.0 Hours
```

### 1.3.22 WEST MELBOURNE

[25]:
```
west = parkingDB[parkingDB["Area"] == "West Melbourne"]
west = west[west["Violation"] == 1]
print(west.head())
print("The exact number of infringements around Docklands: ", len(west))
print("")

#STREETS
print("Streets in this area: ", west['Street'].unique().size)
west['Street'].unique()

print("Finding the number of times a street has a violation:\n", west['Street'].
 ↪value_counts().to_frame())
print("All streets will be included given that the number of violations are␣
 ↪minimal.")

#STREET SIDES
print("Counts of Side of Street:", west['Side Of Street'].unique().size, "\n")
west['Side Of Street'].unique()

print("Finding the number of times a street has a violation:\n", west['Side Of␣
 ↪Street'].value_counts().to_frame(), "\n")
print("The Percentages of each Side of Street in a Violation:", "\n",␣
 ↪west['Side Of Street'].value_counts(normalize=True)*100)
west431 = west[west["Side Of Street"].isin([1, 3, 4])]

print("")

#SIGNS
print("Counting Signs in Violations \n", west431['Sign'].value_counts().
 ↪to_frame(), "\n")

print("Testing the Worst Ranked Signs\n")
#sign 1
print("SIGN ONE: 4P MTR M-F 7:30-18:30")
westLoss = west431[west431["Sign"] == "4P MTR M-F 7:30-18:30"]
print("Value Counts for Parking Duration: ", westLoss['Parking Duration (s)'].
 ↪value_counts().to_frame())
```

```
print("The time (hours) of the max length of stay is: ",␣
 ↪(round(westLoss['Parking Duration (s)'].max()/60)/60), "Hours")
print("")
#sign 2
print("SIGN TWO: 2P M-SAT 7:30-18:30")
westLoss = west431[west431["Sign"] == "2P M-SAT 7:30-18:30"]
print("Value Counts for Parking Duration: ", westLoss['Parking Duration (s)'].
 ↪value_counts().to_frame())
print("The time (hours) of the max length of stay is: ",␣
 ↪(round(westLoss['Parking Duration (s)'].max()/60)/60), "Hours")
print("")
```

```
                  Area         Street Street Limit 1  Street Limit 2  \
7253025  West Melbourne  BATMAN STREET    KING STREET  SPENCER STREET
7253026  West Melbourne  BATMAN STREET    KING STREET  SPENCER STREET
7253635  West Melbourne  BATMAN STREET    KING STREET  SPENCER STREET
7270778  West Melbourne  BATMAN STREET    KING STREET  SPENCER STREET
7279430  West Melbourne  BATMAN STREET    KING STREET  SPENCER STREET

         Side Of Street Street Marker            Arrival Time  \
7253025              1         C8066  04/05/2012 07:30:07 AM
7253026              1         C8068  04/05/2012 07:30:07 AM
7253635              1         C8070  04/05/2012 07:30:08 AM
7270778              3        11600N  04/05/2012 12:55:50 PM
7279430              3        11600N  04/05/2012 03:34:40 PM

                Departure Time  Parking Duration (s)                   Sign  \
7253025  04/05/2012 02:46:08 PM                 26161  4P MTR M-F 7:30-18:30
7253026  04/05/2012 03:19:16 PM                 28149  4P MTR M-F 7:30-18:30
7253635  04/05/2012 03:32:12 PM                 28924  4P MTR M-F 7:30-18:30
7270778  04/05/2012 03:13:36 PM                  8266    2P M-SAT 7:30-18:30
7279430  04/05/2012 06:30:00 PM                 10520    2P M-SAT 7:30-18:30

         Violation  Street ID  Device ID
7253025          1         78       6118
7253026          1         78       6119
7253635          1         78       6120
7270778          1         78       6078
7279430          1         78       6078
The exact number of infringements around Docklands:  668

Streets in this area:  1
Finding the number of times a street has a violation:
                Street
BATMAN STREET     668
All streets will be included given that the number of violations are minimal.
Counts of Side of Street: 3
```

```
Finding the number of times a street has a violation:
    Side Of Street
1             316
3             255
4              97

The Percentages of each Side of Street in a Violation:
 1    47.305389
3    38.173653
4    14.520958
Name: Side Of Street, dtype: float64

Counting Signs in Violations
                         Sign
4P MTR M-F 7:30-18:30   393
2P M-SAT 7:30-18:30     255
4P MTR SAT 7:30-12:30    20

Testing the Worst Ranked Signs

SIGN ONE: 4P MTR M-F 7:30-18:30
Value Counts for Parking Duration:         Parking Duration (s)
39593                       3
39592                       3
19844                       2
15018                       1
32933                       1
…                         …
17013                       1
22365                       1
28065                       1
14771                       1
15907                       1

[388 rows x 1 columns]
The time (hours) of the max length of stay is:  11.0 Hours

SIGN TWO: 2P M-SAT 7:30-18:30
Value Counts for Parking Duration:         Parking Duration (s)
8447                        2
8156                        2
7998                        2
9700                        2
8660                        2
…                         …
8012                        1
15690                       1
```

```
19273                    1
15176                    1
8192                     1

[250 rows x 1 columns]
The time (hours) of the max length of stay is:  11.0 Hours
```

### 1.3 3.0 Machine Learning Algorithm

So because this is a categorical output, we will therefore be using logistic regression modelling. This will be a binary classifier.

[47]:
```
# create algorithm
df = parkingDB

#import
from sklearn.datasets import make_classification
from matplotlib import pyplot as plt
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
import pandas as pd
```

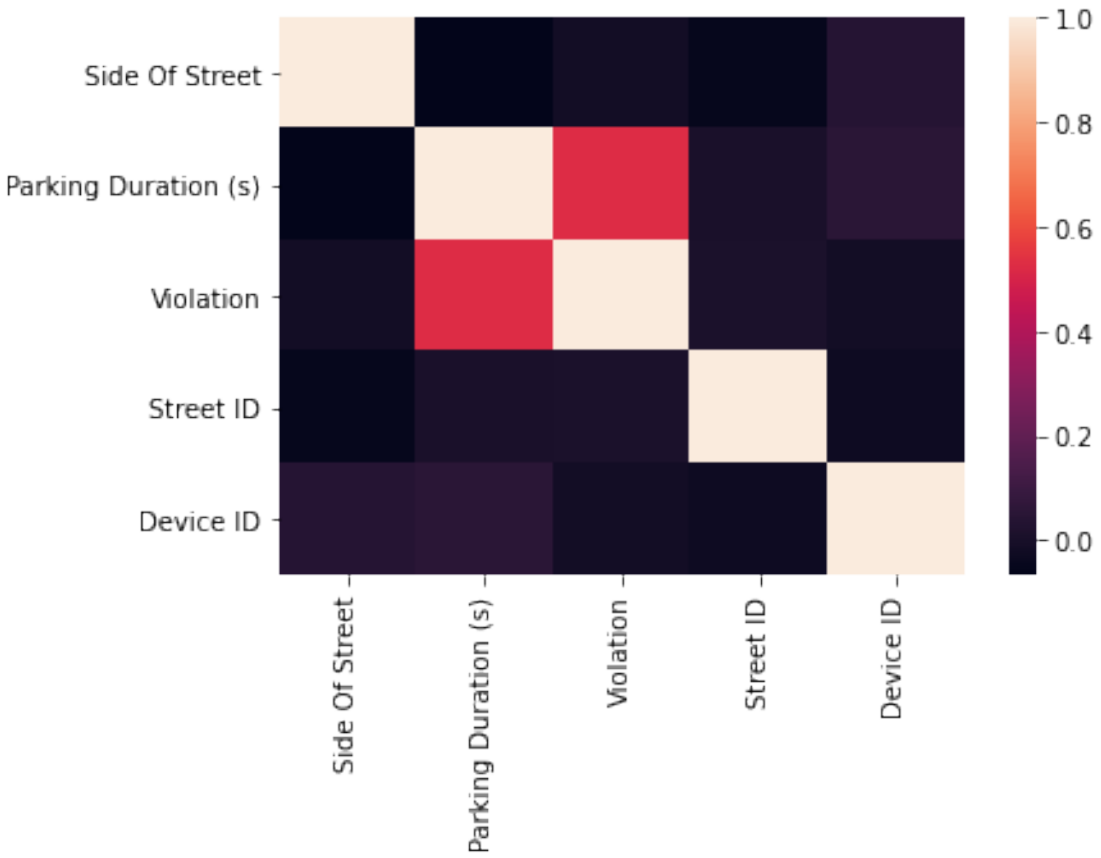
[48]:
```
df.drop(columns = ['Arrival Time', 'Departure Time', 'Street'], inplace=True)
df.head()
```

[48]:
```
              Area    Street Limit 1      Street Limit 2  Side Of Street  \
0            Banks     FLINDERS LANE     FLINDERS STREET               5
1        Chinatown     BOURKE STREET   Lt COLLINS STREET               2
2         Courtney     PELHAM STREET  QUEENSBERRY STREET               2
3  Princes Theatre  Lt BOURKE STREET       BOURKE STREET               2
4  Princes Theatre     BOURKE STREET   Lt COLLINS STREET               5

  Street Marker  Parking Duration (s)                        Sign  Violation  \
0         1263W                    94     2P DIS M-SUN 0:00-23:59          0
1          742E                 14229       4P DIS AOT 12:00-7:30          0
2         4744E                 39548     2P DIS M-SUN 0:00-23:59          1
3          452E                  2189     2P DIS M-SUN 0:00-23:59          0
4          429W                 19616  TKT AREA M-SUN 0:00 - 7:30          0

   Street ID  Device ID
0        957       1667
1       1221        525
2        627       3202
3        647        201
4        647        176
```

Arrival Time and Depature time were removed because we already have the duration of the parking event, and the signage to give us an indication of the reasoning behind the violation. This also simplifies the feature selection of the supervised machine learning model.

[49]:
```
import seaborn as sb

sb.heatmap(df.corr())
```

[49]: <AxesSubplot:>




This shows that parking duration has a really high correlation

The following columns will be label-encoded:

- Street limit 1
- Street limit 2
- Area
- Sign
- Street Marker

[50]:
```
from sklearn.preprocessing import LabelEncoder
cols = ['Area', 'Street Limit 1', 'Street Limit 2', 'Sign', 'Street Marker']
df[cols] = df[cols].apply(LabelEncoder().fit_transform)
df.head()
```

[50]:

| | Area | Street Limit 1 | Street Limit 2 | Side Of Street | Street Marker | \ |
|---|---|---|---|---|---|---|
| 0 | 0 | 21 | 23 | 5 | 648 | |
| 1 | 1 | 5 | 42 | 2 | 2859 | |
| 2 | 4 | 53 | 54 | 2 | 2085 | |
| 3 | 10 | 42 | 2 | 2 | 1970 | |
| 4 | 10 | 5 | 42 | 5 | 1890 | |

| | Parking Duration (s) | Sign | Violation | Street ID | Device ID |
|---|---|---|---|---|---|
| 0 | 94 | 126 | 0 | 957 | 1667 |
| 1 | 14229 | 199 | 0 | 1221 | 525 |
| 2 | 39548 | 126 | 1 | 627 | 3202 |
| 3 | 2189 | 126 | 0 | 647 | 201 |
| 4 | 19616 | 324 | 0 | 647 | 176 |

Now to check that my dafaset is sufficient

[51]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 12208178 entries, 0 to 12208177
Data columns (total 10 columns):
 #   Column                Dtype
---  ------                -----
 0   Area                  int64
 1   Street Limit 1        int64
 2   Street Limit 2        int64
 3   Side Of Street        int64
 4   Street Marker         int64
 5   Parking Duration (s)  int64
 6   Sign                  int64
 7   Violation             int64
 8   Street ID             int64
 9   Device ID             int64
dtypes: int64(10)
memory usage: 1.0 GB
```

[52]:
```
#DROP ALL COLUMNS WITH NULL VALUES
df.dropna(axis=0, how="any", thresh=None, subset=None, inplace=False)
#SEPARATE INTO ATTRIBUTES AND TARGET VALUE
x = df.drop('Violation',axis=1)
x = df.drop('Street Marker',axis=1)
y = df['Violation']
```

[53]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 12208178 entries, 0 to 12208177
Data columns (total 10 columns):
 #   Column                Dtype
---  ------                -----
 0   Area                  int64
 1   Street Limit 1        int64
 2   Street Limit 2        int64
 3   Side Of Street        int64
 4   Street Marker         int64
 5   Parking Duration (s)  int64
 6   Sign                  int64
 7   Violation             int64
 8   Street ID             int64
 9   Device ID             int64
dtypes: int64(10)
memory usage: 1.0 GB
```

[54]:
```
#SEPARATE INTO TEST AND TRAINING SETS
x_train, x_test, y_train, y_test = train_test_split(x,y, test_size=0.25,␣
↪random_state=42)
```

[60]:
```
#BUILD MODEL
log_reg = LogisticRegression(max_iter=1000)
log_reg.fit(x_train, y_train)
```

```
/Users/alanatobgui/opt/anaconda3/lib/python3.8/site-
packages/sklearn/linear_model/_logistic.py:763: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  n_iter_i = _check_optimize_result(
```

[60]:
```
LogisticRegression(max_iter=1000)
```

[61]:
```
#MAKE PREDICTIONS
y_pred = log_reg.predict(x_test)
```

[62]:
```
#CONFUSION MATRIX
```

```
confusion_matrix(y_test, y_pred)
```

[62]:
```
array([[2634498,       0],
       [      0,  417547]])
```

[63]:
```
from sklearn.metrics import classification_report

print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00   2634498
           1       1.00      1.00      1.00    417547

    accuracy                           1.00   3052045
   macro avg       1.00      1.00      1.00   3052045
weighted avg       1.00      1.00      1.00   3052045
```

[ ]:
```
Current List of Hyper-Parameter Tuning:

Features:
    -removed street marker
    -
LogReg Parameters:
    -max iteration increased to
```