

Boston Crime Data Analysis and Visualization

```
In [2]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import plotly.graph_objects
```

```
In [3]: data = pd.read_csv('C:/Users/zizhe/Desktop/Leo/Crimes in Boston/crime.csv',encoding='latin')
```

```
In [4]: data.head()
```

```
Out[4]:
```

	INCIDENT_NUMBER	OFFENSE_CODE	OFFENSE_CODE_GROUP	OFFENSE_DESCRIPTION	DISTRICT	REPORTING_AREA	SHOOTING	OCCURRED_ON_DATE	YEAR	MC
0	I182070945	619	Larceny	LARCENY ALL OTHERS	D14	808	NaN	2018-09-02 13:00:00	2018	
1	I182070943	1402	Vandalism	VANDALISM	C11	347	NaN	2018-08-21 00:00:00	2018	
2	I182070941	3410	Towed	TOWED MOTOR VEHICLE	D4	151	NaN	2018-09-03 19:27:00	2018	
3	I182070940	3114	Investigate Property	INVESTIGATE PROPERTY	D4	272	NaN	2018-09-03 21:16:00	2018	
4	I182070938	3114	Investigate Property	INVESTIGATE PROPERTY	B3	421	NaN	2018-09-03 21:05:00	2018	

Data Visualisation

```
In [5]: def treemap(categories,title,path,values):
fig = px.treemap(categories, path=path, values=values, height=700,
title=title, color_discrete_sequence = px.colors.sequential.RdBu)
fig.data[0].textinfo = 'label+text+value'
fig.show()
```

```
In [6]: def histogram(data,path,color,title,xaxis,yaxis):
fig = px.histogram(data, x=path,color=color)
fig.update_layout(
title_text=title,
xaxis_title_text=xaxis,
yaxis_title_text=yaxis,
)
fig.show()
```

```
In [7]: def bar(categories,x,y,color,title,xlab,ylab):
fig = px.bar(categories, x=x, y=y,
color=color,
height=400)
fig.update_layout(
title_text=title,
xaxis_title_text=xlab,
yaxis_title_text=ylab,
)
fig.show()
```

General Analysis

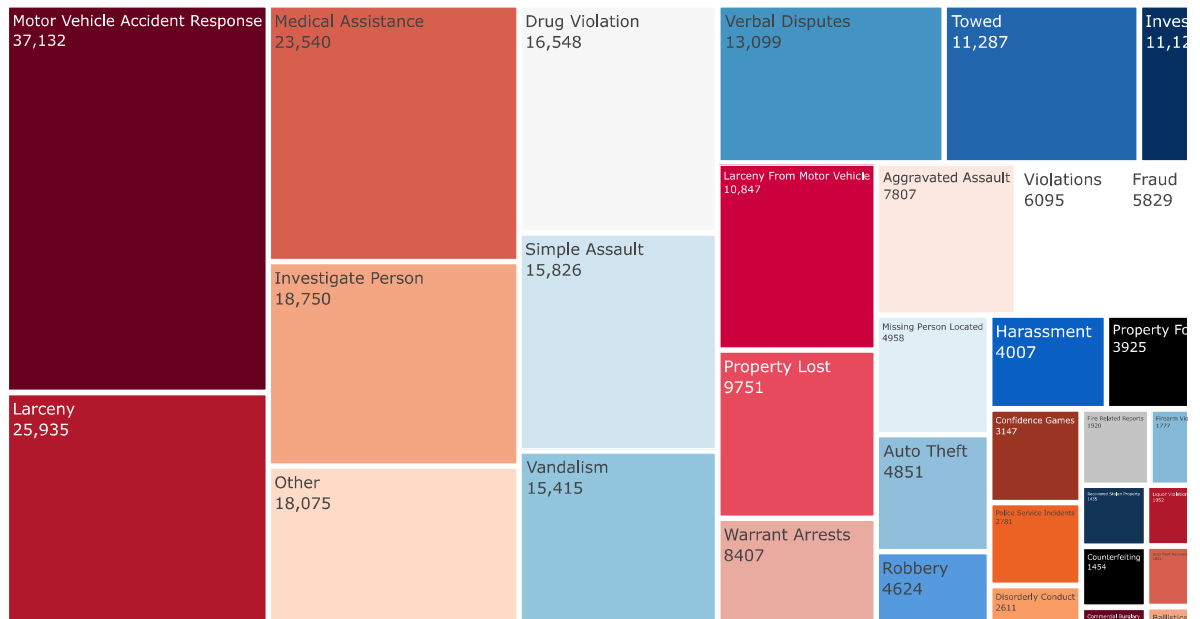
Number of crimes per category

```
In [8]: Number_crimes = data['OFFENSE_CODE_GROUP'].value_counts()
values = Number_crimes.values
categories = pd.DataFrame(data=Number_crimes.index, columns=["OFFENSE_CODE_GROUP"])
categories['values'] = values
```

```
In [9]: treemap(categories,'Major Crimes in Boston',['OFFENSE_CODE_GROUP'],categories['values'])
```

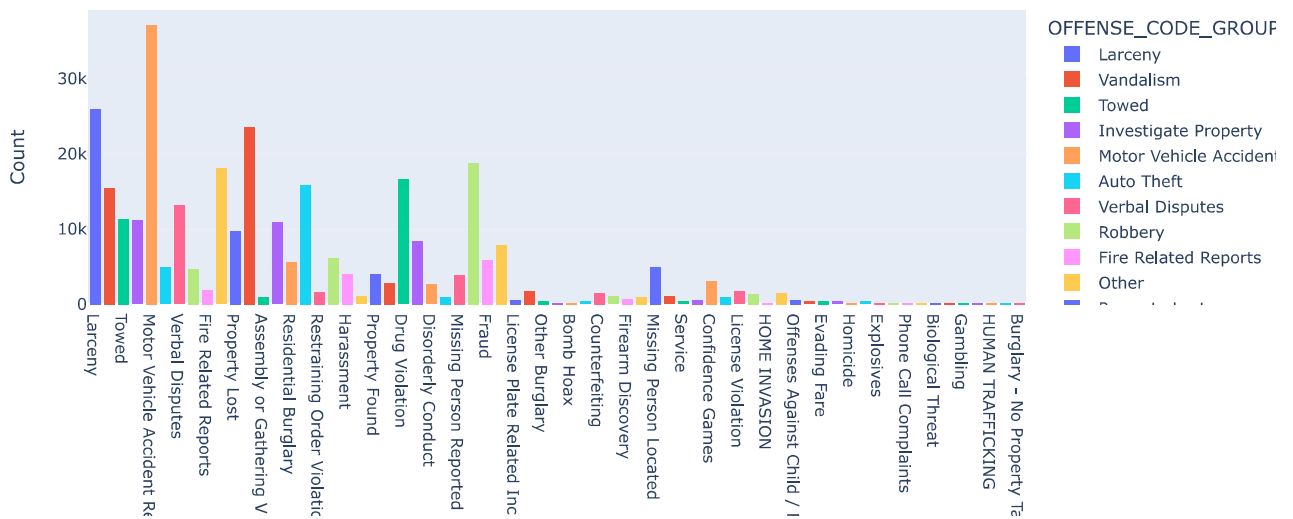
```
C:\Users\zizhe\New folder\lib\site-packages\plotly\express\_core.py:1637: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a future version. Use pandas.concat instead.
df_all_trees = df_all_trees.append(df_tree, ignore_index=True)
```

Major Crimes in Boston



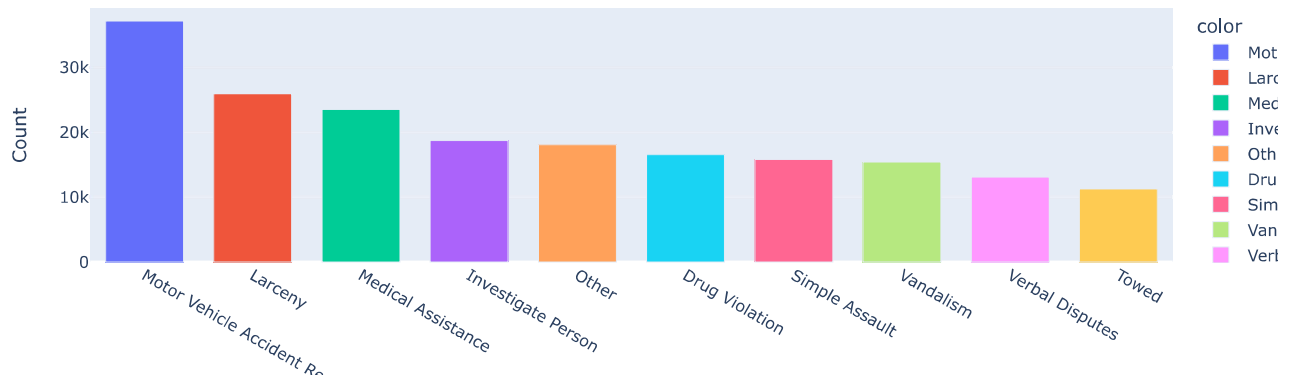
```
In [10]: histogram(data,"OFFENSE_CODE_GROUP","OFFENSE_CODE_GROUP",Major Crimes in Boston','Crime','Count')
```

Major Crimes in Boston



```
In [11]: bar(categories, categories['OFFENSE_CODE_GROUP'][0:10], categories['values'][0:10], categories['OFFENSE_CODE_GROUP'][0:10], 'Top 10 Major Crimes in Boston', 'Crime', 'Count')
```

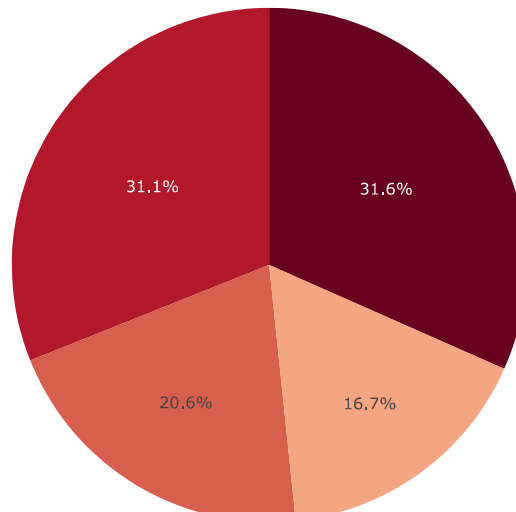
Top 10 Major Crimes in Boston



Number of crimes per year

```
In [12]: Number_crimes_year = data['YEAR'].value_counts()
years = pd.DataFrame(data=Number_crimes_year.index, columns=["YEAR"])
years['values'] = Number_crimes_year.values

In [13]: fig = px.pie(years, values='values', names='YEAR', color_discrete_sequence=px.colors.sequential.RdBu)
fig.show()
```



```
In [14]: Number_crimes_month = data['MONTH'].value_counts()
months = pd.DataFrame(data=Number_crimes_month.index, columns=["MONTH"])
months['values'] = Number_crimes_month.values
```

```
In [15]: fig = go.Figure(go.Bar(
    x=months['values'],
    y=months['MONTH'],
    marker=dict(
        color='rgb(13,143,129)',
    ),
    orientation='h'))
fig.update_layout(
    title_text='Major Crimes in Boston per month',
    xaxis_title_text='Count',
    yaxis_title_text='Month',
)
fig.show()
```

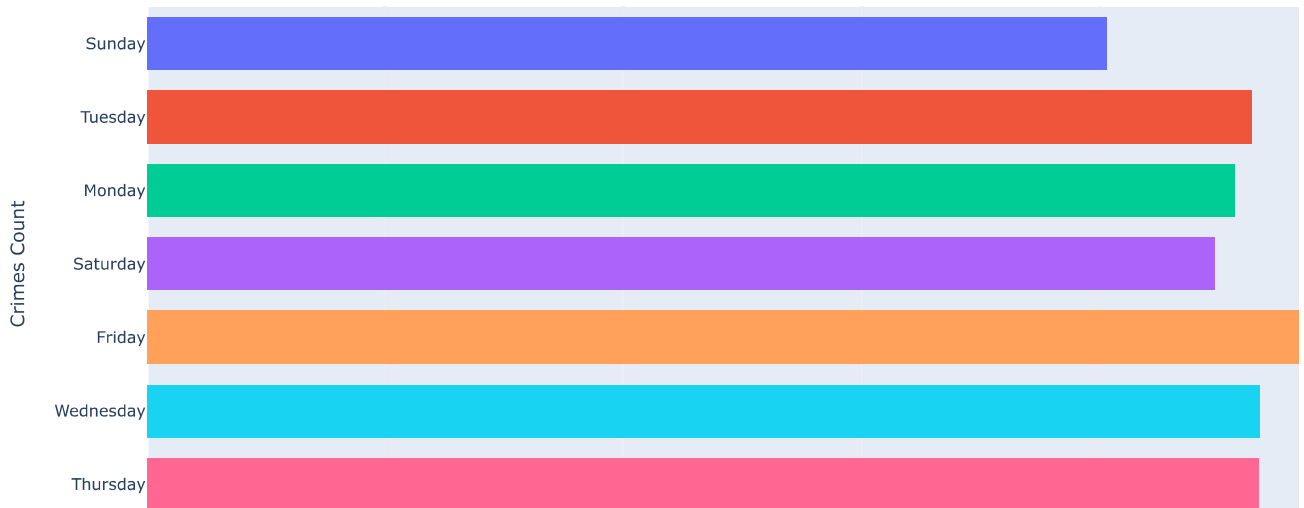
Major Crimes in Boston per month



```
In [16]: Number_crimes_days = data['DAY_OF_WEEK'].value_counts()
days = pd.DataFrame(data=Number_crimes_days.index, columns=["DAY_OF_WEEK"])
days['values'] = Number_crimes_days.values
```

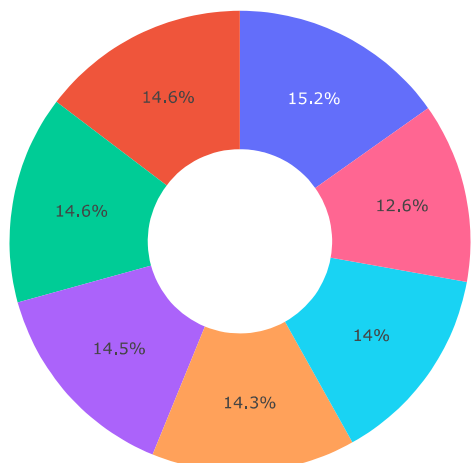
```
In [17]: fig = px.histogram(data, y="DAY_OF_WEEK", color="DAY_OF_WEEK")
fig.update_layout(
    title_text='Crime count on each day',
    xaxis_title_text='Day',
    yaxis_title_text='Crimes Count',
)
fig.show()
```

Crime count on each day



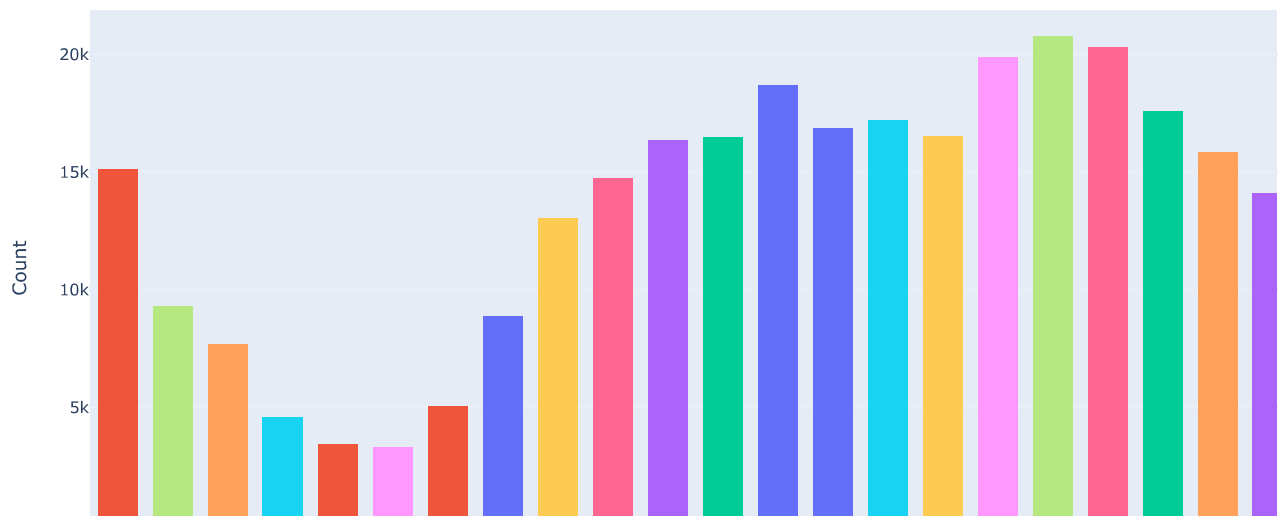
```
In [18]: fig = go.Figure(data=[go.Pie(labels=days['DAY_OF_WEEK'], values=days['values'], hole=.4)])
fig.update_layout(
    title_text='Crime count on each day',
)
fig.show()
```

Crime count on each day



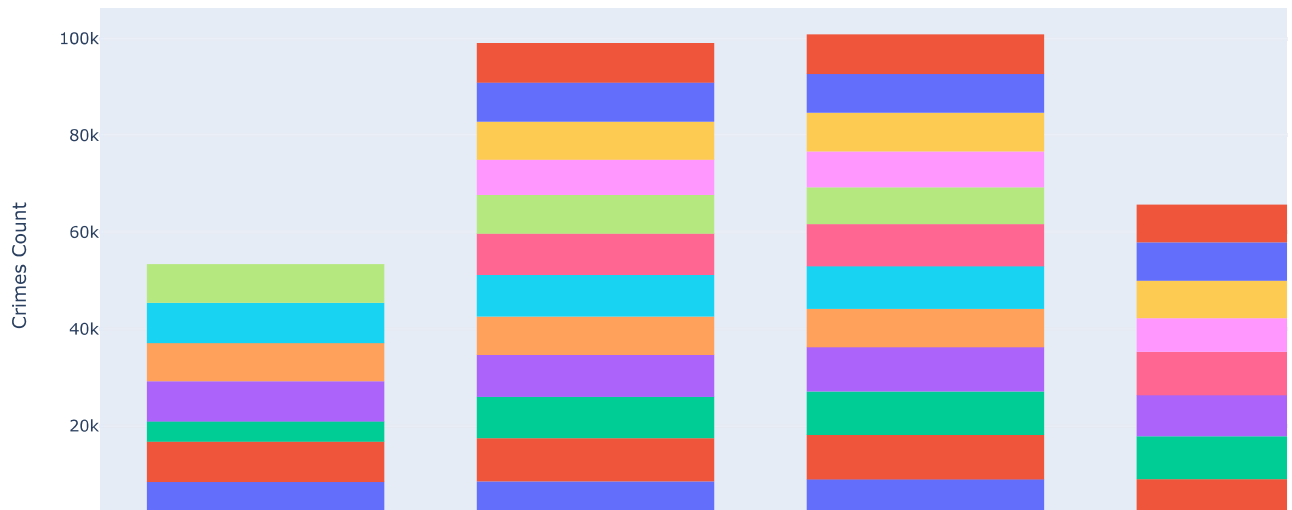
In [19]: `histogram(data,"HOUR","HOUR",'Crime count on each Hour','Hour','Count')`

Crime count on each Hour



In [20]: `histogram(data,"YEAR","MONTH",'Crime count on each year per month','Year','Crimes Count')`

Crime count on each year per month



```
In [21]: Number_crimes_street = data['STREET'].value_counts()
street = pd.DataFrame(data=Number_crimes_street.index, columns=["STREET"])
street['values'] = Number_crimes_street.values
```

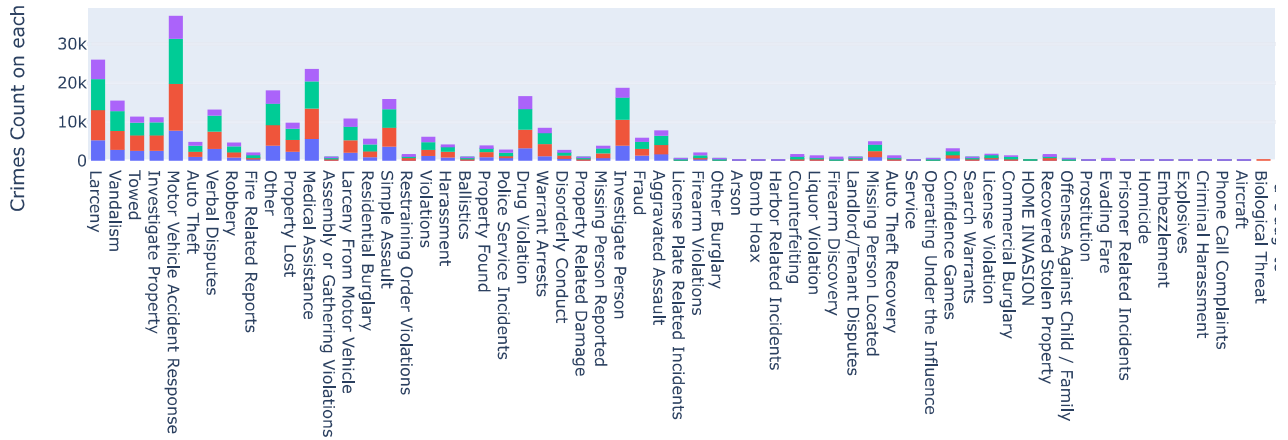
```
In [22]: bar(street,street['STREET'][0:10],street['values'][0:10]
,street['STREET'][0:10], 'Top 10 Crime count on each Street', 'Street', ' Crime Count')
```

Top 10 Crime count on each Street



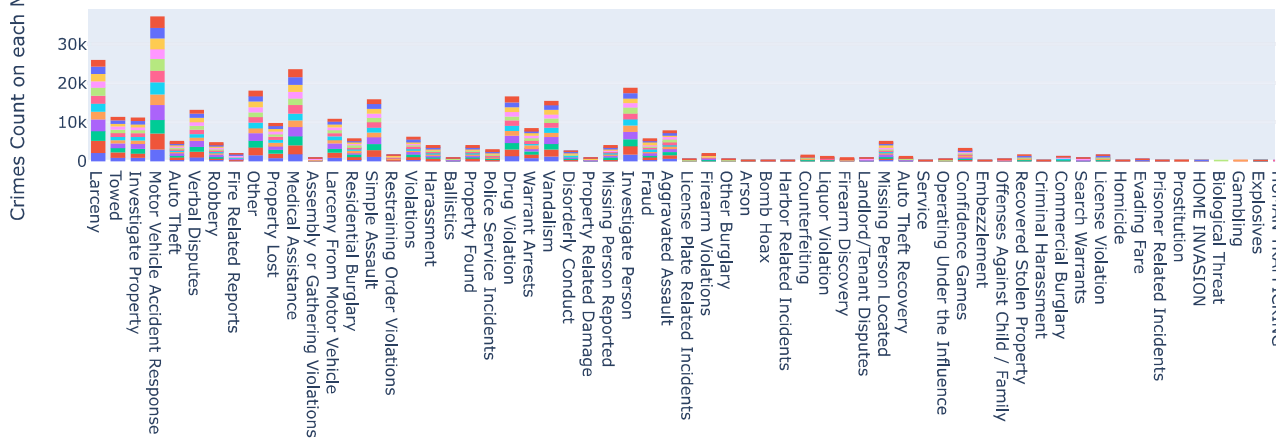
```
In [23]: histogram(data,"OFFENSE_CODE_GROUP","YEAR",'Crime count per Category on each Year','Category','Crimes Count on each Year')
```

Crime count per Category on each Year



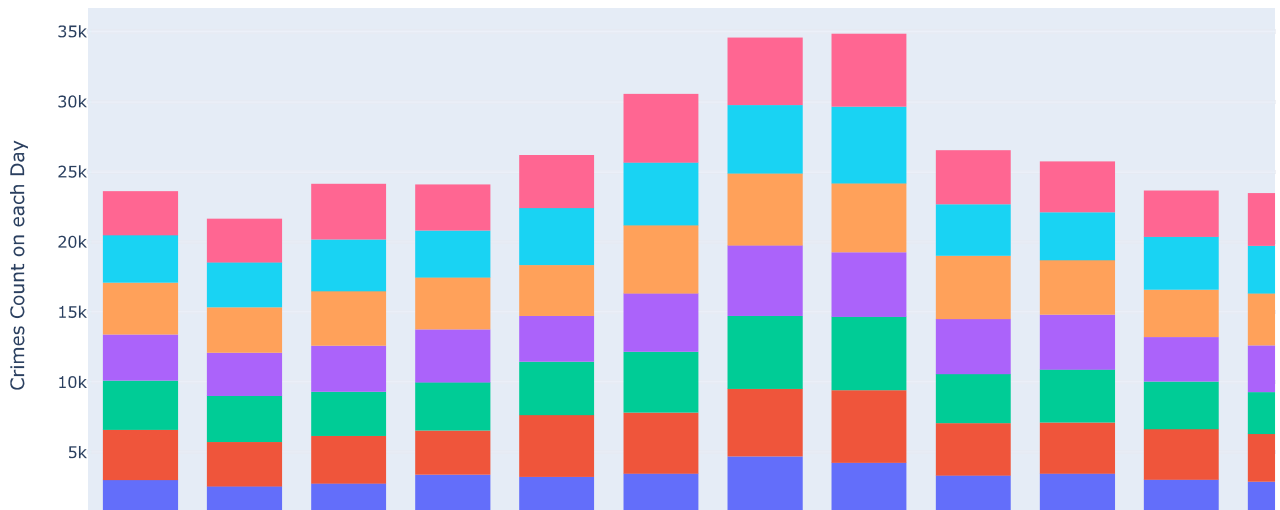
```
In [24]: histogram(data, "OFFENSE_CODE_GROUP", "MONTH", 'Crime count per Category on each Month', 'Category', 'Crimes Count on each Month')
```

Crime count per Category on each Month



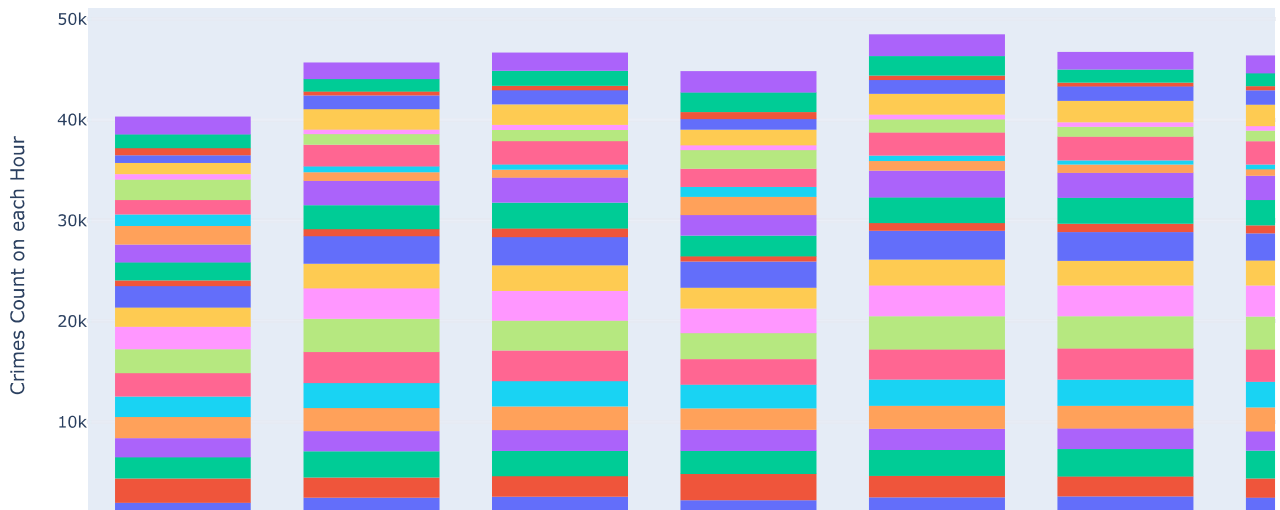
```
In [25]: histogram(data, "MONTH", "DAY_OF_WEEK", 'Crime count per Month on each Day', 'Month', 'Crimes Count on each Day')
```

Crime count per Month on each Day



```
In [26]: histogram(data,"DAY_OF_WEEK","HOURL", 'Crime count per Day on each Hour', 'Day', 'Crimes Count on each Hour')
```

Crime count per Day on each Hour



2018 Crimes Analysis

```
In [27]: Data_2018 = data [(data['YEAR'] == 2018) ].reset_index(drop=True)
```

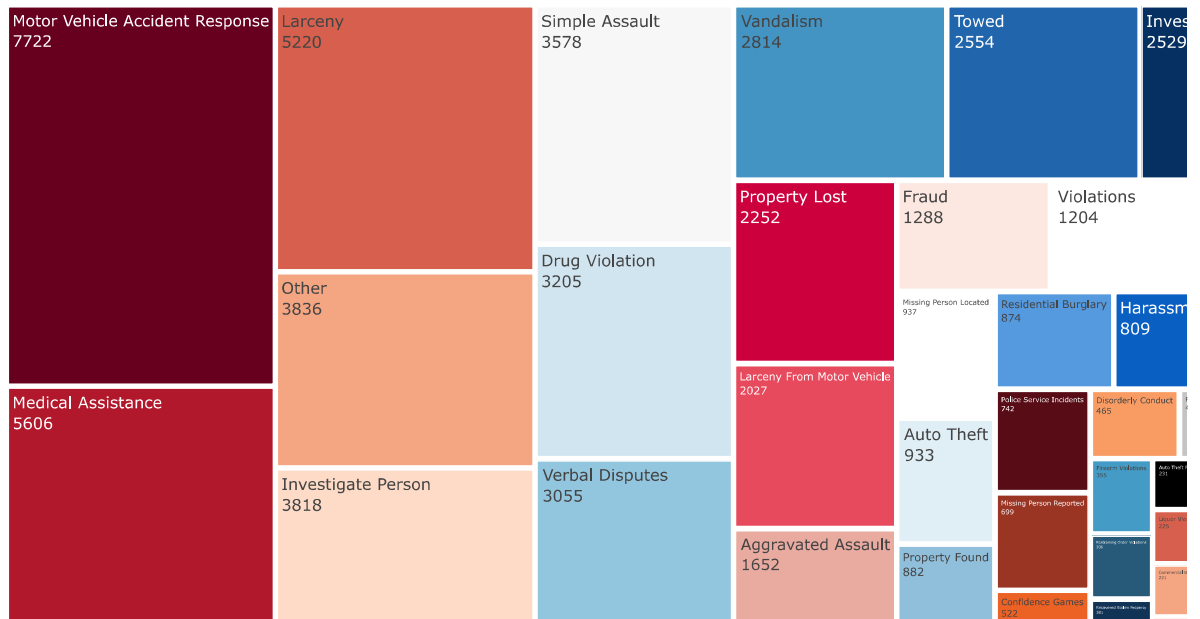
```
In [28]: Number_crimes_2018 = Data_2018['OFFENSE_CODE_GROUP'].value_counts()
categories_2018 = pd.DataFrame(data=Number_crimes_2018.index, columns=["OFFENSE_CODE_GROUP"])
categories_2018['values'] = Number_crimes_2018.values
```

```
In [29]: treemap(categories_2018,'Major Crimes in Boston in 2018',['OFFENSE_CODE_GROUP'],categories_2018['values'])
```

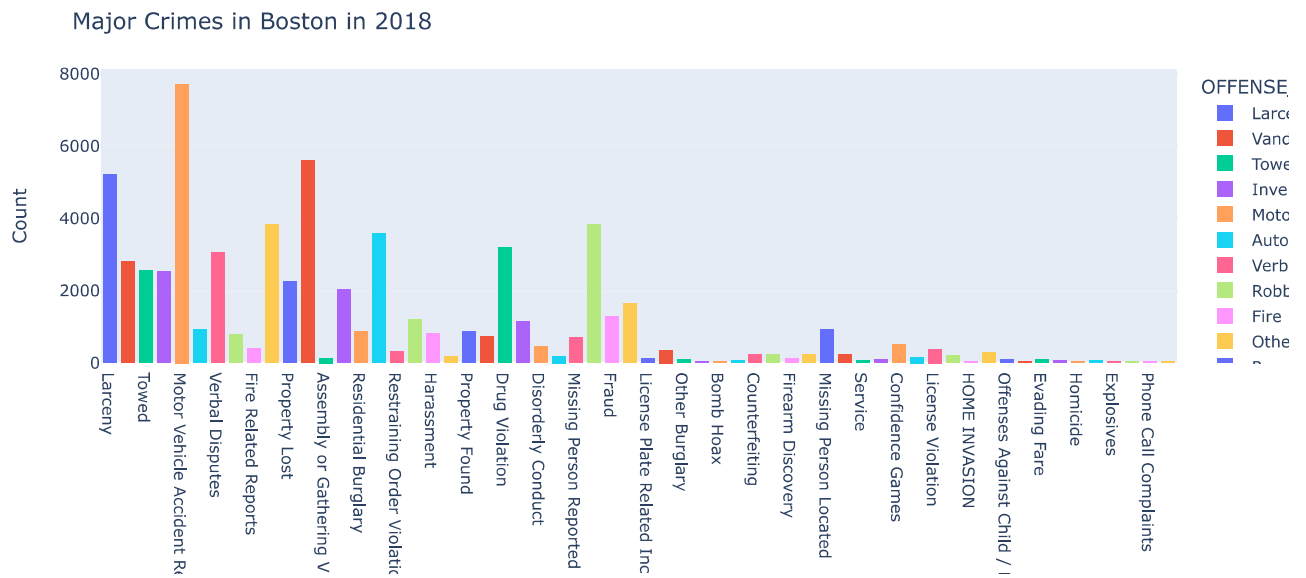
C:\Users\zizhe\New folder\lib\site-packages\plotly\express\core.py:1637: FutureWarning:

The frame.append method is deprecated and will be removed from pandas in a future version. Use pandas.concat instead.

Major Crimes in Boston in 2018

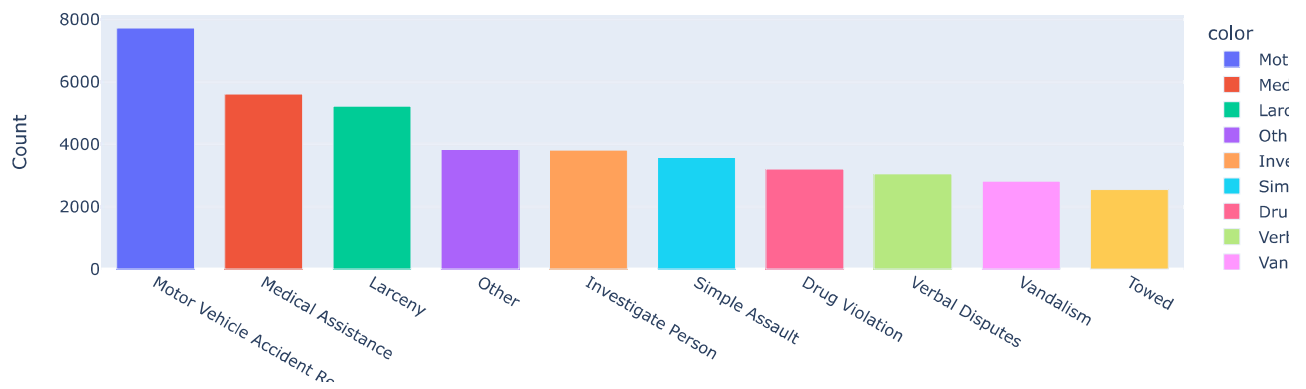


```
In [30]: histogram(Data_2018,"OFFENSE_CODE_GROUP","OFFENSE_CODE_GROUP",'Major Crimes in Boston in 2018','Crime','Count')
```



```
In [31]: fig = px.bar(categories_2018, x=categories_2018['OFFENSE_CODE_GROUP'][0:10], y=categories_2018['values'][0:10],
                color=categories_2018['OFFENSE_CODE_GROUP'][0:10],
                labels={'pop':'population of Canada'}, height=400)
fig.update_layout(
    title_text='Top 10 Major Crimes in Boston in 2018', # title of plot
    xaxis_title_text='Crime', # xaxis label
    yaxis_title_text='Count', # yaxis label
)
fig.show()
```

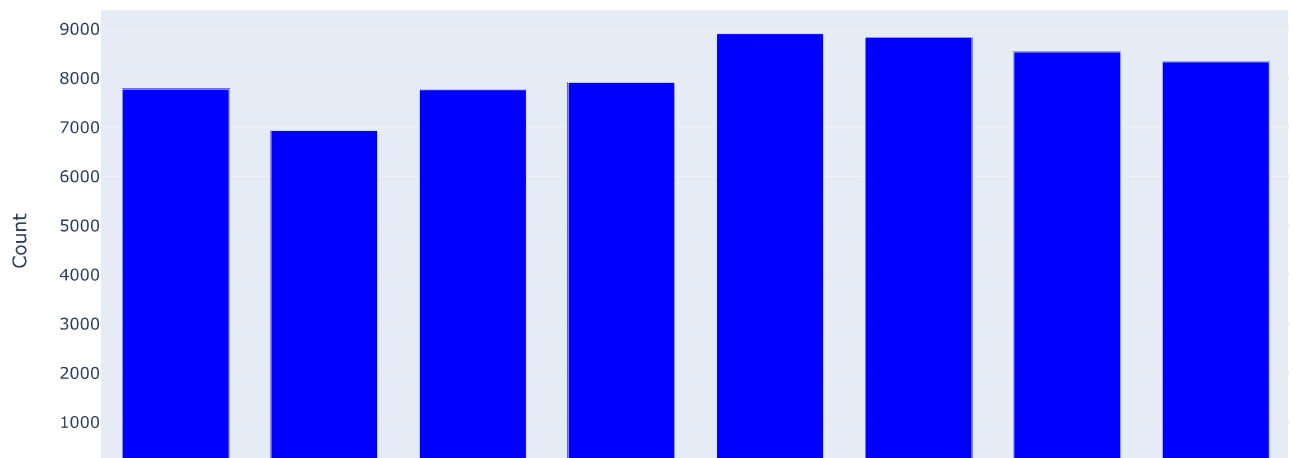
Top 10 Major Crimes in Boston in 2018



```
In [32]: Number_crimes_month_2018 = Data_2018['MONTH'].value_counts()
months_2018 = pd.DataFrame(data=Number_crimes_month_2018.index, columns=["MONTH"])
months_2018['values'] = Number_crimes_month_2018.values
```

```
In [33]: fig = go.Figure(go.Bar(
    y=months_2018['values'],
    x=months_2018['MONTH'],
    marker=dict(
        color='blue',
    ),
    orientation='v'))
fig.update_layout(
    title_text='Major Crimes in Boston per month in 2018', # title of plot
    xaxis_title_text='Month', # xaxis Label
    yaxis_title_text='Count ', # yaxis Label
)
fig.show()
```

Major Crimes in Boston per month in 2018



```
In [34]: Number_crimes_days_2018 = Data_2018['DAY_OF_WEEK'].value_counts()
days_2018 = pd.DataFrame(data=Number_crimes_days_2018.index, columns=["DAY_OF_WEEK"])
days_2018['values'] = Number_crimes_days_2018.values
```

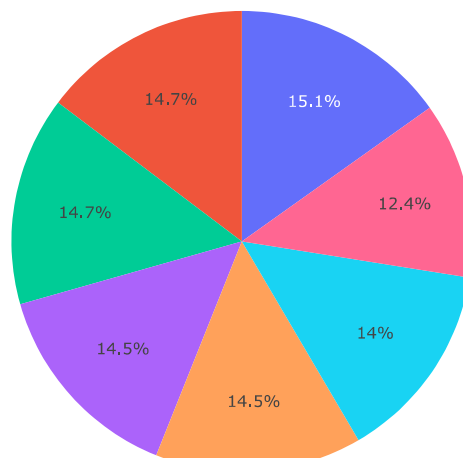
```
In [35]: histogram(Data_2018, "DAY_OF_WEEK", "DAY_OF_WEEK", 'Crime count on each day in 2018', 'Day', 'Crimes Count')
```

Crime count on each day in 2018



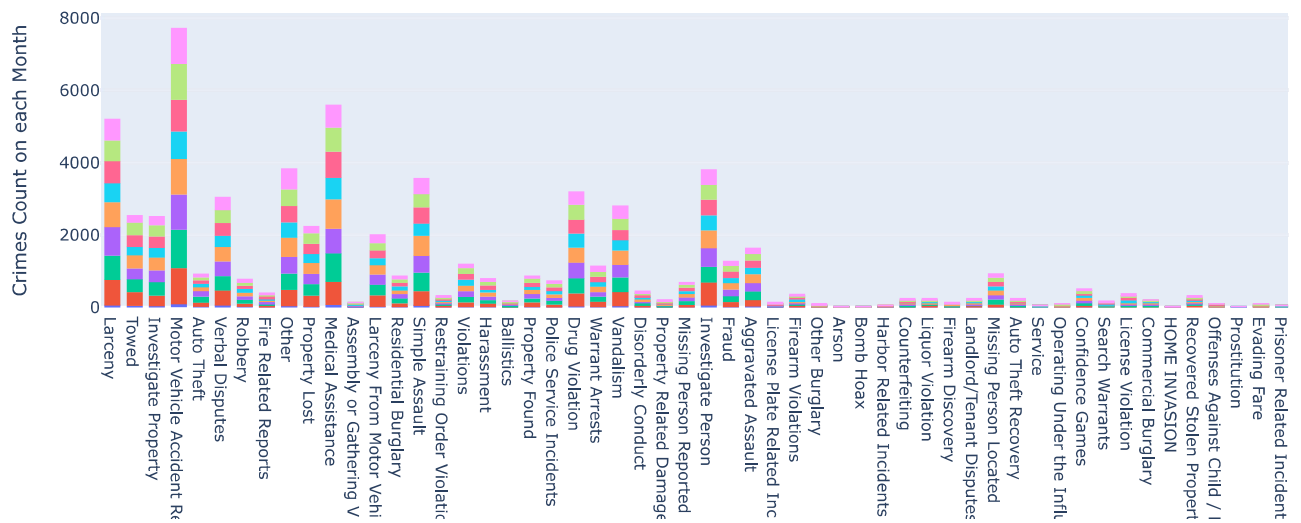
```
In [36]: fig = go.Figure(data=[go.Pie(labels=days_2018['DAY_OF_WEEK'], values=days_2018['values'])])
fig.update_layout(
    title_text='Crime count on each day in 2018', # title of plot
)
fig.show()
```

Crime count on each day in 2018



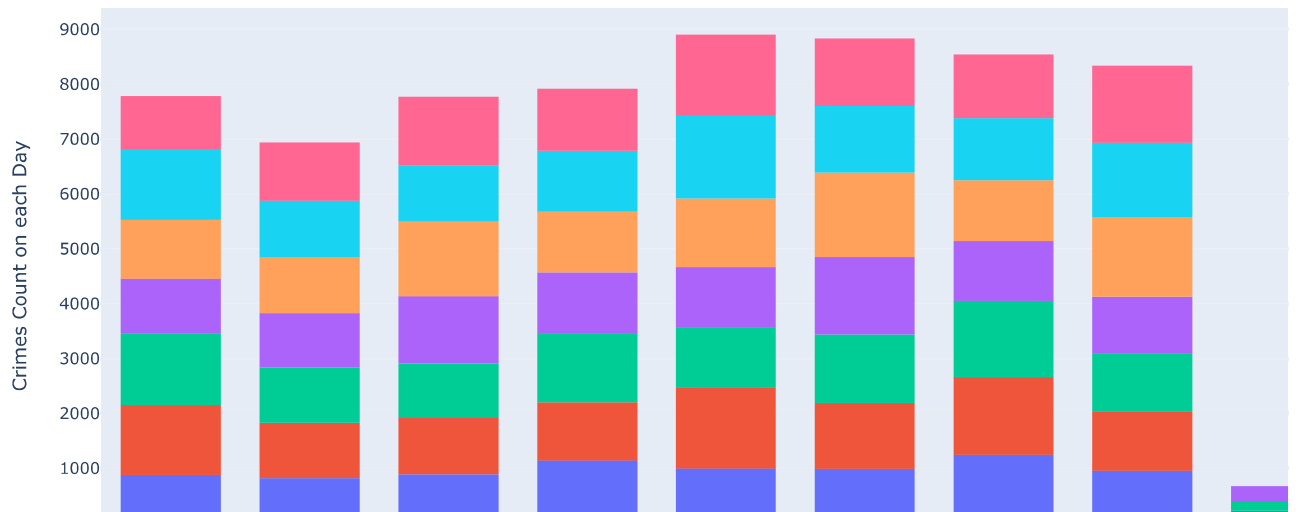
```
In [37]: histogram(Data_2018,"OFFENSE_CODE_GROUP","MONTH",'Crime count per Category on each Month in 2018','Category','Crimes Count on each Month')
```

Crime count per Category on each Month in 2018



```
In [38]: histogram(Data_2018,"MONTH","DAY_OF_WEEK",'Crime count per Month on each Day in 2018','Month','Crimes Count on each Day')
```

Crime count per Month on each Day in 2018



```
In [39]: histogram(Data_2018,"DAY_OF_WEEK","HOUR",'Crime count per Day on each Hour in 2018','Day','Crimes Count on each Hour')
```

Crime count per Day on each Hour in 2018

