CRIME DATA

February 8, 2024

[66]:
```
pip install pymysql
```

```
Requirement already satisfied: pymysql in c:\users\admin\new folder\lib\site-
packages (1.1.0)
Note: you may need to restart the kernel to use updated packages.
```

[67]:
```
import pymysql
import seaborn as sns
from matplotlib import pyplot as plt
import pandas as pd
```

[68]:
```
connection=pymysql.
 ↪connect(host="localhost",user="root",password="NavyaReddy56789",database="minicap")
```

[69]:
```
query="select * from minicap.crimedata_csv"
```

[70]:
```
df=pd.read_sql(query,connection)
```

```
C:\Users\Admin\AppData\Local\Temp\ipykernel_11044\3823902772.py:1: UserWarning:
pandas only supports SQLAlchemy connectable (engine/connection) or database
string URI or sqlite3 DBAPI2 connection. Other DBAPI2 objects are not tested.
Please consider using SQLAlchemy.
  df=pd.read_sql(query,connection)
```

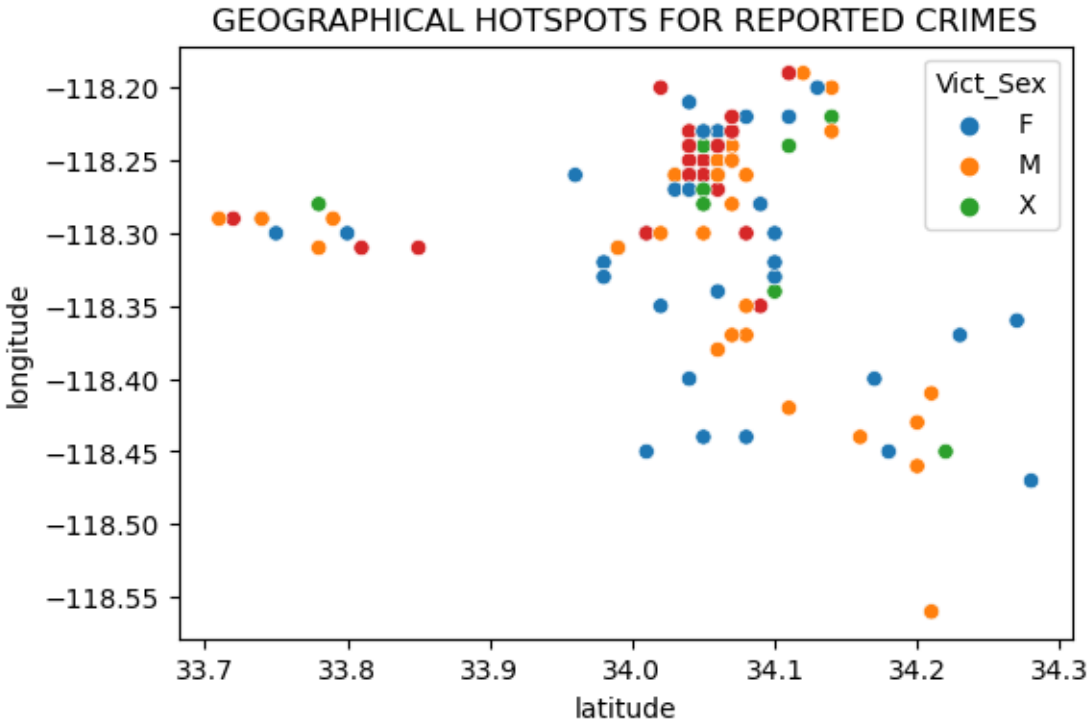
WHERE ARE THE GEOGRAPHICAL HOTSPOTS FOR REPORTED CRIMES?

[71]:
```
query1 = """
SELECT Vict_Sex,LAT,LON,COUNT(*) location_crime
FROM minicap.crimedata_csv
GROUP BY Vict_Sex,LAT,LON;
"""
```

[72]:
```
df1=pd.read_sql(query1,connection)
```

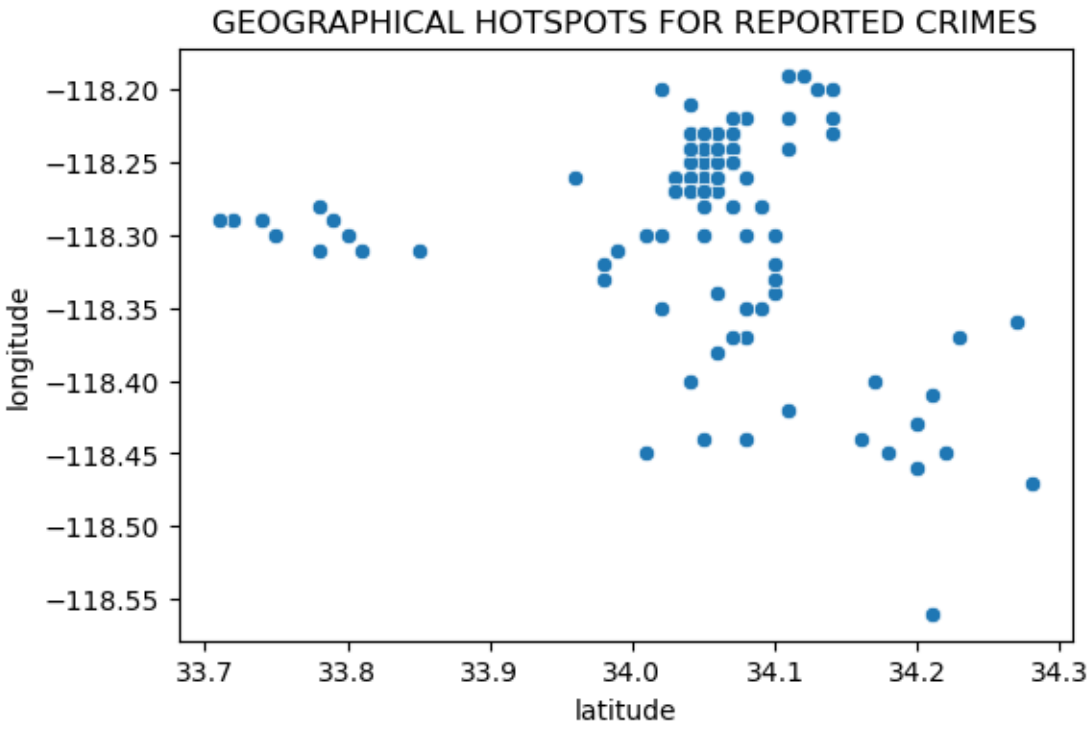
```
C:\Users\Admin\AppData\Local\Temp\ipykernel_11044\2803407773.py:1: UserWarning:
pandas only supports SQLAlchemy connectable (engine/connection) or database
string URI or sqlite3 DBAPI2 connection. Other DBAPI2 objects are not tested.
Please consider using SQLAlchemy.
  df1=pd.read_sql(query1,connection)
```

[73]:
```
plt.figure(figsize=(6,4))
plt.xlabel('latitude')
plt.ylabel('longitude')
plt.title('GEOGRAPHICAL HOTSPOTS FOR REPORTED CRIMES')
sns.scatterplot(x=df1['LAT'], y=df1['LON'], hue=df1['Vict_Sex'])
plt.show()
```



——BY THE IMAGE WE GET TO KNOW THAT X-AXIS REPRESENTS THE LATITUDE AND Y-AXIS REPRESENTS THE LONGITUDE MORE DENSER THE POINTS IN THE PLOT MORE REPORTED CRIMES OCCURED AT THAT POINTS AND HUE DETERMINES THE COLOR OF THE VALUES IN THE COLUMN WE USED THAT IS VICT_SEX.

[74]:
```
plt.figure(figsize=(6,4))
plt.xlabel('latitude')
plt.ylabel('longitude')
plt.title('GEOGRAPHICAL HOTSPOTS FOR REPORTED CRIMES')
sns.scatterplot(x=df1['LAT'], y=df1['LON'])
plt.show()
```



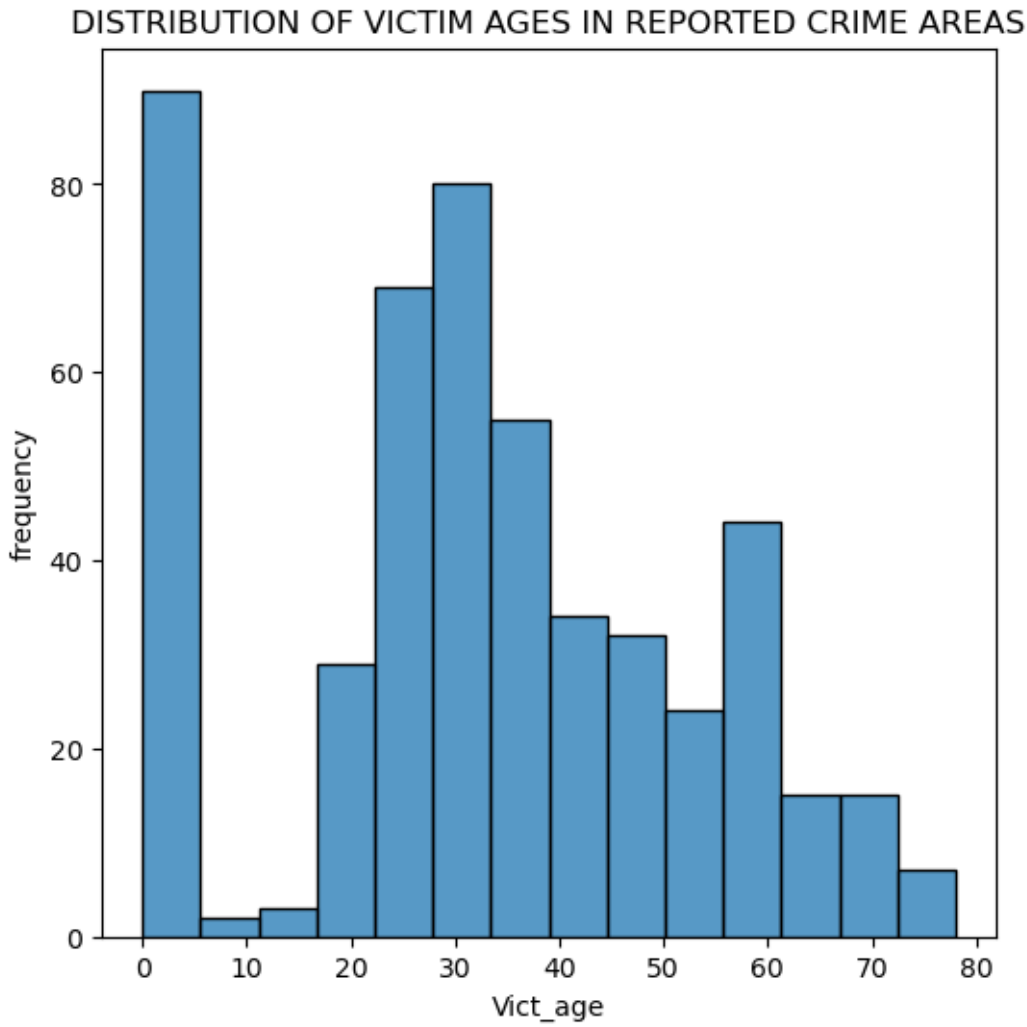
WHAT IS THE DISTRIBUTION OF VICTIM AGES IN REPORTED CRIMES?

[75]:
```
query2_age="""
select Vict_age from minicap.crimedata_csv;
"""
```

[76]:
```
df2_age=pd.read_sql(query2_age,connection)
```

```
C:\Users\Admin\AppData\Local\Temp\ipykernel_11044\2032936899.py:1: UserWarning:
pandas only supports SQLAlchemy connectable (engine/connection) or database
string URI or sqlite3 DBAPI2 connection. Other DBAPI2 objects are not tested.
Please consider using SQLAlchemy.
  df2_age=pd.read_sql(query2_age,connection)
```

[77]:
```
plt.figure(figsize=(6,6))
sns.histplot(df2_age['Vict_age'])
plt.xlabel('Vict_age')
plt.ylabel('frequency')
plt.title('DISTRIBUTION OF VICTIM AGES IN REPORTED CRIME AREAS')
plt.show()
```



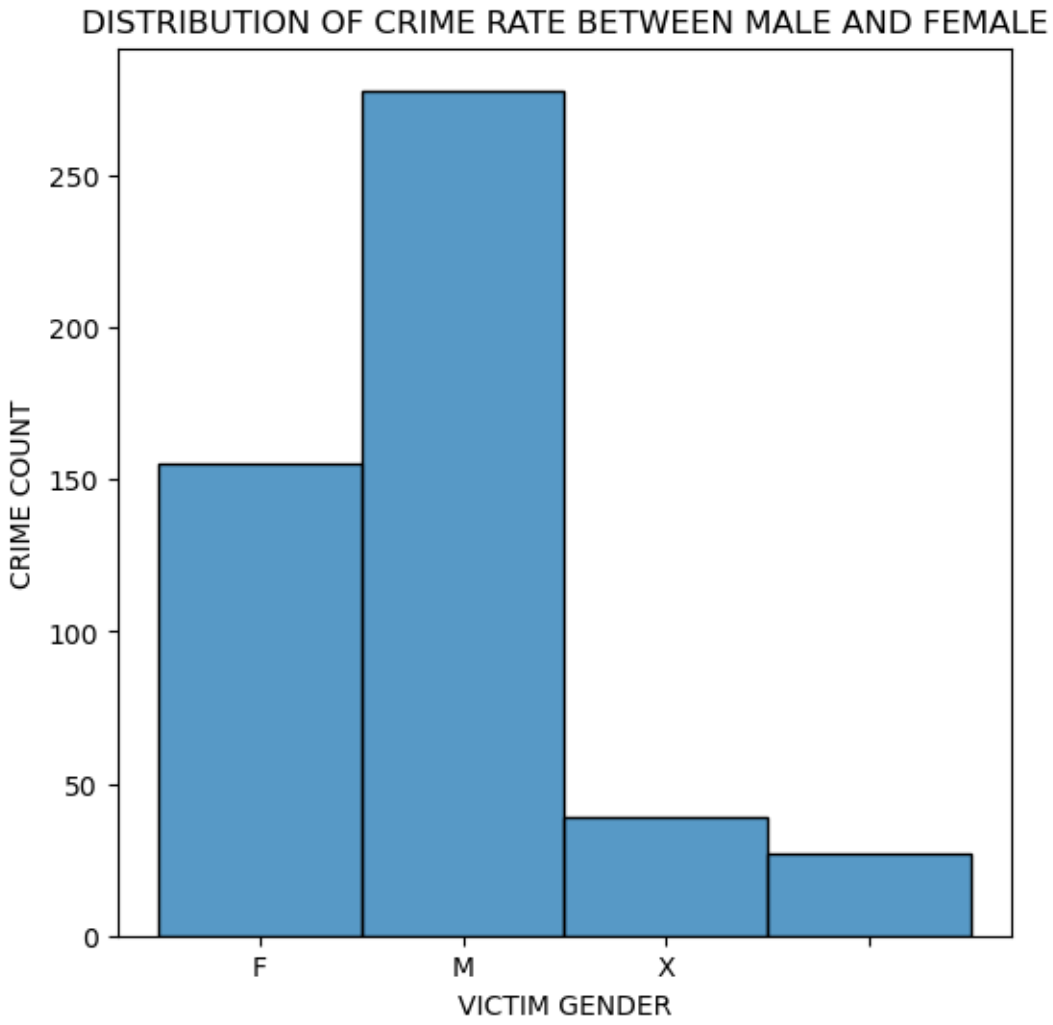
[78]:
```
query2_sex="""
select Vict_sex from minicap.crimedata_csv;
"""
```

[79]:
```
df2_sex=pd.read_sql(query2_sex,connection)
```

```
C:\Users\Admin\AppData\Local\Temp\ipykernel_11044\2004729100.py:1: UserWarning:
pandas only supports SQLAlchemy connectable (engine/connection) or database
string URI or sqlite3 DBAPI2 connection. Other DBAPI2 objects are not tested.
Please consider using SQLAlchemy.
  df2_sex=pd.read_sql(query2_sex,connection)
```

[80]:
```
plt.figure(figsize=(6,6))
sns.histplot(df2_sex['Vict_sex'])
plt.xlabel('VICTIM GENDER')
```

```
plt.ylabel('CRIME COUNT')
plt.title('DISTRIBUTION OF CRIME RATE BETWEEN MALE AND FEMALE')
plt.show()
```



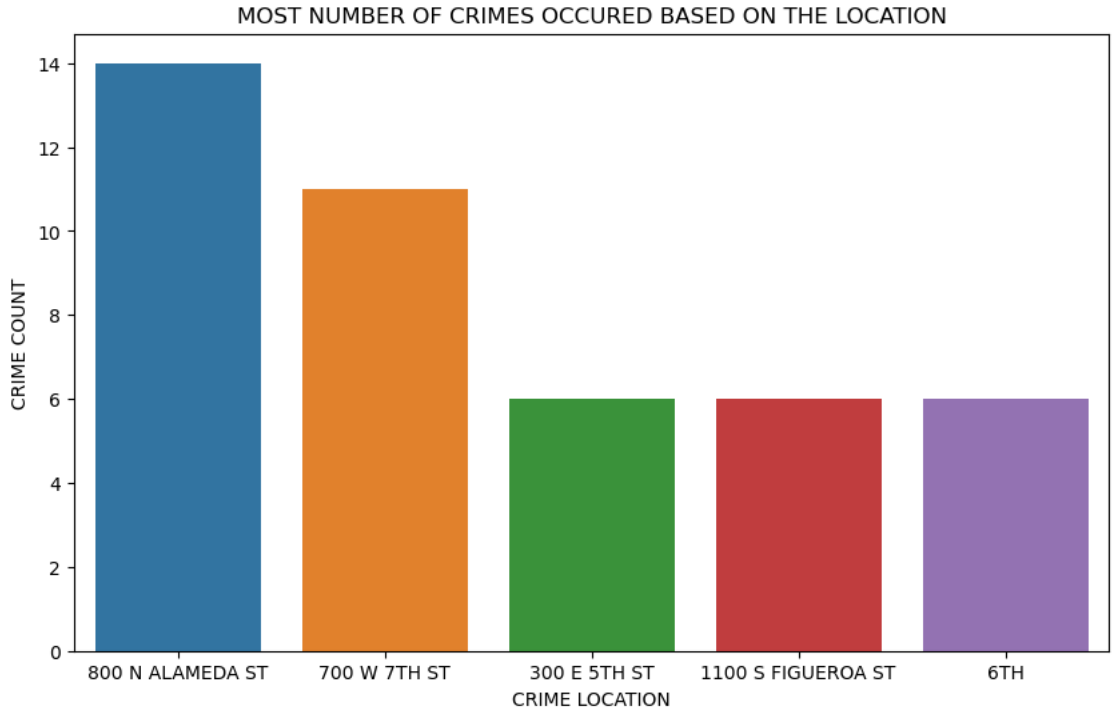
WHERE DO MOST NUMBER OF CRIMES OCCUR BASED ON THE LOCATION?

[123]:
```
Location="""
select Location,count(*) as total_no_crimes
from minicap.crimedata_csv
group by(Location)
order by total_no_crimes desc
limit 5;
"""
```

[124]:
```
a=pd.read_sql(Location,connection)
```

```
C:\Users\Admin\AppData\Local\Temp\ipykernel_11044\3670261003.py:1: UserWarning:
pandas only supports SQLAlchemy connectable (engine/connection) or database
string URI or sqlite3 DBAPI2 connection. Other DBAPI2 objects are not tested.
Please consider using SQLAlchemy.
  a=pd.read_sql(Location,connection)
```

[125]:
```
plt.figure(figsize=(10,6))
sns.barplot(x='Location', y='total_no_crimes', data=a)
plt.xlabel('CRIME LOCATION')
plt.ylabel('CRIME COUNT')
plt.title('MOST NUMBER OF CRIMES OCCURED BASED ON THE LOCATION')
plt.show()
```



WHAT IS THE DISTRIBUTION OF REPORTED CRIMES BASED ON CRIME CODE?

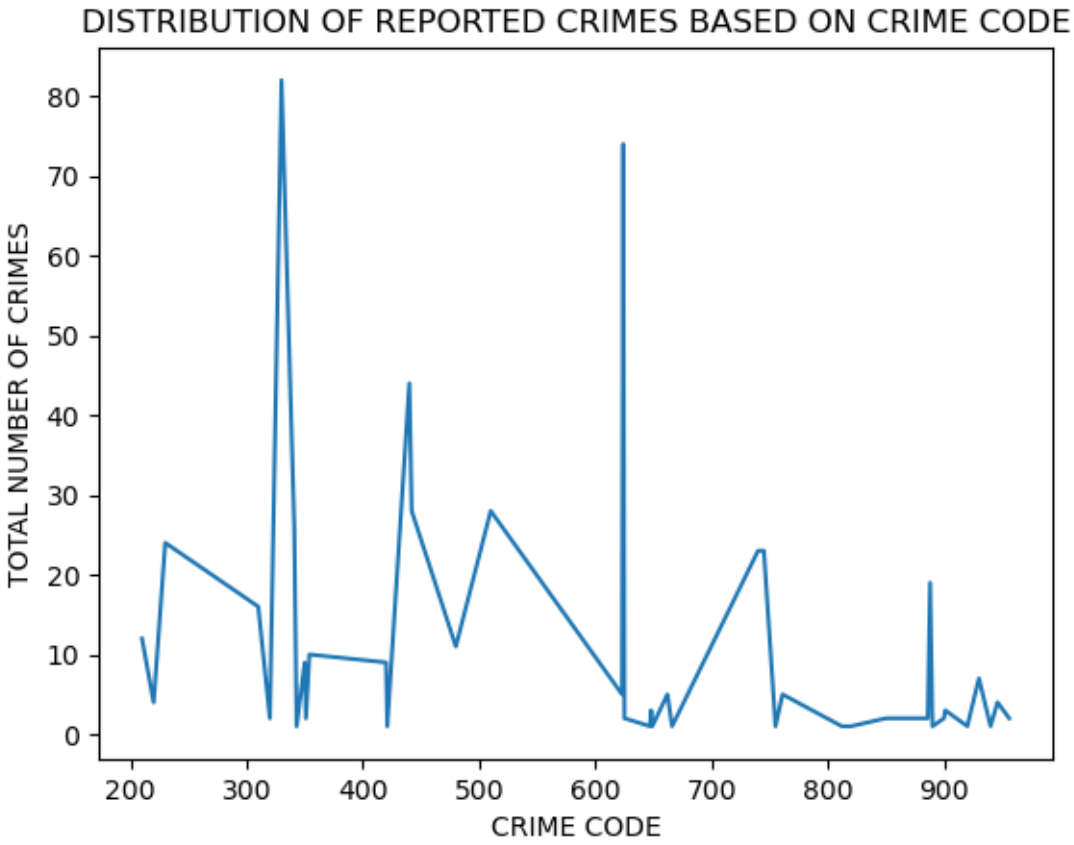
[102]:
```
Crime_cd="""
select Crm_Cd,count(*) as total_crimes
from minicap.crimedata_csv
group by Crm_Cd ;
"""
```

[103]:
```
df4=pd.read_sql(Crime_cd,connection)
```

```
C:\Users\Admin\AppData\Local\Temp\ipykernel_11044\669298768.py:1: UserWarning:
pandas only supports SQLAlchemy connectable (engine/connection) or database
```

```
string URI or sqlite3 DBAPI2 connection. Other DBAPI2 objects are not tested.
Please consider using SQLAlchemy.
  df4=pd.read_sql(Crime_cd,connection)
```

[106]:
```
sns.lineplot(x='Crm_Cd',y='total_crimes',data=df4)
plt.title('DISTRIBUTION OF REPORTED CRIMES BASED ON CRIME CODE')
plt.xlabel('CRIME CODE')
plt.ylabel('TOTAL NUMBER OF CRIMES')
plt.show()
```



[ ]: