

```
In [1]: import pandas as pd
```

```
In [2]: df = pd.read_csv("Crime_Data.csv")
df.head()
```

Out[2]:

	DATEEND	TIMESTART	TIMEEND	ADDRESS	CODE_DEFINED	Attempt	Arrest	LarcenyCode
<b>0</b>	2020/01/05 05:00:00+00	829	829	200 N STATE ST	LARCENY		NaN	
<b>1</b>	2020/02/14 05:00:00+00	1000	1000	1800 MIDLAND AV	LARCENY		NaN	A
<b>2</b>	2020/01/06 05:00:00+00	1731	1731	500 DELAWARE ST	RAPE		Yes	
<b>3</b>	2020/02/14 05:00:00+00	1933	1933	1 DESTINY USA DR	LARCENY		Yes	A
<b>4</b>	2020/02/14 05:00:00+00	0	1830	400 CHINOOK DR	LARCENY		NaN	A



```
In [3]: # Shape of the dataset
df.shape
```

```
# Column names
df.columns
```

```
# Basic info
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5266 entries, 0 to 5265
Data columns (total 9 columns):
 #   Column      Non-Null Count  Dtype  
 ---  --          --          --      
 0   DATEEND    5266 non-null   object 
 1   TIMESTART  5266 non-null   int64  
 2   TIMEEND    5266 non-null   int64  
 3   ADDRESS    5266 non-null   object 
 4   CODE_DEFINED 5266 non-null   object 
 5   Attempt    5266 non-null   object 
 6   Arrest     1262 non-null   object 
 7   LarcenyCode 3977 non-null   object 
 8   FID        5266 non-null   int64  
dtypes: int64(3), object(6)
memory usage: 370.4+ KB
```

```
In [4]: # Check missing values in each column
missing_values = df.isnull().sum()
```

## missing\_values

```
Out[4]: DATEEND      0
        TIMESTART     0
        TIMEEND       0
        ADDRESS       0
        CODE_DEFINED   0
        Attempt        0
        Arrest         4004
        LarcenyCode    1289
        FID            0
        dtype: int64
```

```
In [5]: # Percentage of missing values
(df.isnull().mean() * 100).round(2)
```

```
Out[5]: DATEEND      0.00
        TIMESTART     0.00
        TIMEEND       0.00
        ADDRESS       0.00
        CODE_DEFINED   0.00
        Attempt        0.00
        Arrest         76.03
        LarcenyCode    24.48
        FID            0.00
        dtype: float64
```

```
In [6]: df.columns
```

```
Out[6]: Index(['DATEEND', 'TIMESTART', 'TIMEEND', 'ADDRESS', 'CODE_DEFINED', 'Attempt',
               'Arrest', 'LarcenyCode', 'FID'],
              dtype='object')
```

```
In [13]: # Convert DATEEND to datetime
df['DATEEND'] = pd.to_datetime(df['DATEEND'], errors='coerce')
```

```
In [14]: df['DATEEND'].head()
```

```
Out[14]: 0    2020-01-05 05:00:00+00:00
1    2020-02-14 05:00:00+00:00
2    2020-01-06 05:00:00+00:00
3    2020-02-14 05:00:00+00:00
4    2020-02-14 05:00:00+00:00
Name: DATEEND, dtype: datetime64[ns, UTC]
```

```
In [15]: df['Year'] = df['DATEEND'].dt.year
df['Month'] = df['DATEEND'].dt.month
df['Day'] = df['DATEEND'].dt.day
df['Day_of_Week'] = df['DATEEND'].dt.day_name()
```

```
In [16]: df[['DATEEND', 'Year', 'Month', 'Day_of_Week']].head()
```

Out[16]:

	DATEEND	Year	Month	Day_of_Week
0	2020-01-05 05:00:00+00:00	2020	1	Sunday
1	2020-02-14 05:00:00+00:00	2020	2	Friday
2	2020-01-06 05:00:00+00:00	2020	1	Monday
3	2020-02-14 05:00:00+00:00	2020	2	Friday
4	2020-02-14 05:00:00+00:00	2020	2	Friday

In [17]:

```
offense_counts = df['CODE_DEFINED'].value_counts()
offense_counts.head()
```

Out[17]: CODE\_DEFINED

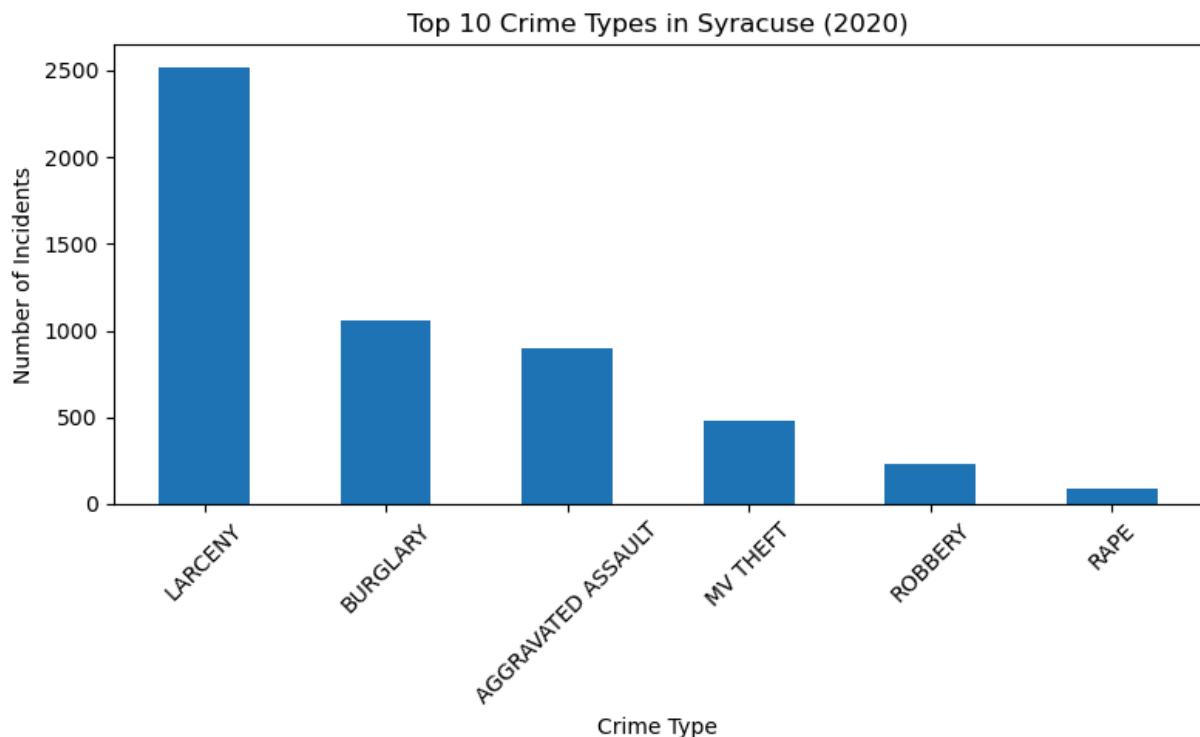
LARCENY	2522
BURGLARY	1054
AGGRAVATED ASSAULT	894
MV THEFT	480
ROBBERY	227

Name: count, dtype: int64

In [18]:

```
import matplotlib.pyplot as plt

offense_counts.head(10).plot(kind='bar', figsize=(8,5))
plt.title("Top 10 Crime Types in Syracuse (2020)")
plt.xlabel("Crime Type")
plt.ylabel("Number of Incidents")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



In [ ]: