In [1]:

```
# This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python Docker image: https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directory

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# You can write up to 5GB to the current directory (/kaggle/working/) that gets preserved as output when you create a version using "Save & Run All"
# You can also write temporary files to /kaggle/temp/, but they won't be saved outside of the current session
```

```
/kaggle/input/crimes-in-chicago/Chicago_Crimes_2012_to_2017.csv
/kaggle/input/crimes-in-chicago/Chicago_Crimes_2005_to_2007.csv
/kaggle/input/crimes-in-chicago/Chicago_Crimes_2001_to_2004.csv
/kaggle/input/crimes-in-chicago/Chicago_Crimes_2008_to_2011.csv
```

In [2]:

```
import pandas as pd
import  numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from fbprophet import Prophet
```

In [3]:

```
chicago_df_1=pd.read_csv('/kaggle/input/crimes-in-chicago/Chicago_Crimes_2005_to_2007.csv',error_bad_lines=False)
chicago_df_2=pd.read_csv('/kaggle/input/crimes-in-chicago/Chicago_Crimes_2008_to_2011.csv',error_bad_lines=False)
chicago_df_3=pd.read_csv('/kaggle/input/crimes-in-chicago/Chicago_Crimes_2012_to_2017.csv',error_bad_lines=False)
```

```
b'Skipping line 533719: expected 23 fields, saw 24\n'
b'Skipping line 1149094: expected 23 fields, saw 41\n'
```

In [4]:

```
print(chicago_df_1.shape)
print(chicago_df_2.shape)
print(chicago_df_3.shape)
```

```
(1872343, 23)
(2688710, 23)
(1456714, 23)
```

In [5]:

```
chicago_df=pd.concat([chicago_df_1,chicago_df_2,chicago_df_3],ignore_index=False, axis=0)
```

In [6]:

```
print(chicago_df.shape)
```

```
(6017767, 23)
```

In [7]:

```
chicago_df.head()
```

Out[7]:

	Unnamed: 0	ID	Case Number	Date	Block	IUCR	Primary Type	Description	Location Description	Arrest
0	0	4673626	HM274058	04/02/2006 01:00:00 PM	055XX N MANGO AVE	2825	OTHER OFFENSE	HARASSMENT BY TELEPHONE	RESIDENCE	False
1	1	4673627	HM202199	02/26/2006 01:40:48 PM	065XX S RHODES AVE	2017	NARCOTICS	MANU/DELIVER:CRACK	SIDEWALK	True
2	2	4673628	HM113861	01/08/2006 11:16:00 PM	013XX E 69TH ST	051A	ASSAULT	AGGRAVATED: HANDGUN	OTHER	False
3	4	4673629	HM274049	04/05/2006 06:45:00 PM	061XX W NEWPORT AVE	0460	BATTERY	SIMPLE	RESIDENCE	False
4	5	4673630	HM187120	02/17/2006 09:03:14 PM	037XX W 60TH ST	1811	NARCOTICS	POSS: CANNABIS 30GMS OR LESS	ALLEY	True

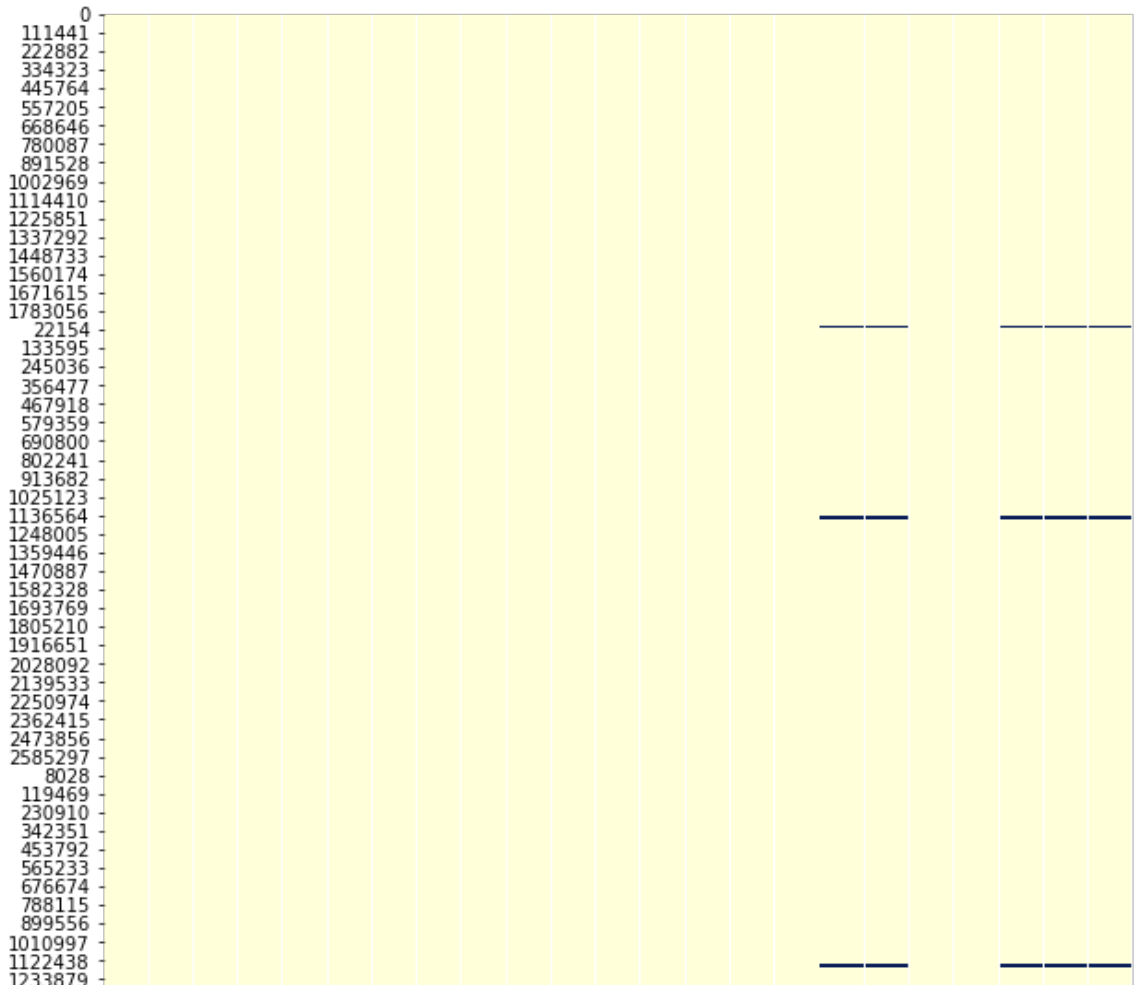
5 rows × 23 columns

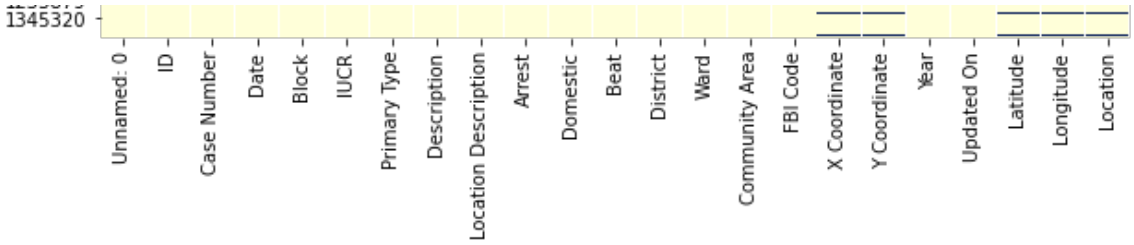
In [8]:

```
plt.figure(figsize=(10,10))
sns.heatmap(chicago_df.isnull(),cbar=False,cmap='YlGnBu')
```

Out[8]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f7110903410>
```





In [9]:

```
chicago_df.drop(['Unnamed: 0','Case Number','ID','IUCR','X Coordinate','Y Coordinate','Updated On','Year','FBI Code','Beat','Ward','Community Area','Location','District','Latitude','Longitude'],inplace=True,axis=1)
```

In [10]:

```
chicago_df.head(10)
```

Out[10]:

	Date	Block	Primary Type	Description	Location Description	Arrest	Domestic
0	04/02/2006 01:00:00 PM	055XX N MANGO AVE	OTHER OFFENSE	HARASSMENT BY TELEPHONE	RESIDENCE	False	False
1	02/26/2006 01:40:48 PM	065XX S RHODES AVE	NARCOTICS	MANU/DELIVER:CRACK	SIDEWALK	True	False
2	01/08/2006 11:16:00 PM	013XX E 69TH ST	ASSAULT	AGGRAVATED: HANDGUN	OTHER	False	False
3	04/05/2006 06:45:00 PM	061XX W NEWPORT AVE	BATTERY	SIMPLE	RESIDENCE	False	False
4	02/17/2006 09:03:14 PM	037XX W 60TH ST	NARCOTICS	POSS: CANNABIS 30GMS OR LESS	ALLEY	True	False
5	03/30/2006 10:30:00 PM	014XX W 73RD PL	ASSAULT	SIMPLE	APARTMENT	True	False
6	04/05/2006 12:10:00 PM	050XX N LARAMIE AVE	BATTERY	SIMPLE	SCHOOL, PUBLIC, BUILDING	True	False
7	04/05/2006 03:00:00 PM	067XX S ROCKWELL ST	THEFT	$500 AND UNDER	STREET	False	False
8	04/05/2006 09:30:00 PM	019XX W CHICAGO AVE	ASSAULT	SIMPLE	PARKING LOT/GARAGE(NON.RESID.)	False	False
9	04/03/2006 03:00:00 AM	063XX S EBERHART AVE	BATTERY	DOMESTIC BATTERY SIMPLE	SIDEWALK	False	True

In [11]:

```
chicago_df.Date = pd.to_datetime(chicago_df.Date, format='%m/%d/%Y %I:%M:%S %p')
```

In [12]:

```
chicago_df.Date.head(10)
```

Out[12]:

```
0   2006-04-02 13:00:00
1   2006-02-26 13:40:48
2   2006-01-08 23:16:00
3   2006-04-05 18:45:00
4   2006-02-17 21:03:14
5   2006-03-30 22:30:00
6   2006-04-05 12:10:00
7   2006-04-05 15:00:00
8   2006-04-05 21:30:00
9   2006-04-03 03:00:00
Name: Date, dtype: datetime64[ns]
```

In [13]:

```
chicago_df.head()
```

Out[13]:

	Date	Block	Primary Type	Description	Location Description	Arrest	Domestic
0	2006-04-02 13:00:00	055XX N MANGO AVE	OTHER OFFENSE	HARASSMENT BY TELEPHONE	RESIDENCE	False	False
1	2006-02-26 13:40:48	065XX S RHODES AVE	NARCOTICS	MANU/DELIVER:CRACK	SIDEWALK	True	False
2	2006-01-08 23:16:00	013XX E 69TH ST	ASSAULT	AGGRAVATED: HANDGUN	OTHER	False	False
3	2006-04-05 18:45:00	061XX W NEWPORT AVE	BATTERY	SIMPLE	RESIDENCE	False	False
4	2006-02-17 21:03:14	037XX W 60TH ST	NARCOTICS	POSS: CANNABIS 30GMS OR LESS	ALLEY	True	False

In [24]:

```
chicago_df.index=pd.DatetimeIndex(chicago_df.Date)
chicago_df.head()
```

Out[24]:

	Date	Block	Primary Type	Description	Location Description	Arrest	Domestic
Date							
2006-04-02 13:00:00	2006-04-02 13:00:00	055XX N MANGO AVE	OTHER OFFENSE	HARASSMENT BY TELEPHONE	RESIDENCE	False	False
2006-02-26 13:40:48	2006-02-26 13:40:48	065XX S RHODES AVE	NARCOTICS	MANU/DELIVER:CRACK	SIDEWALK	True	False
2006-01-08 23:16:00	2006-01-08 23:16:00	013XX E 69TH ST	ASSAULT	AGGRAVATED: HANDGUN	OTHER	False	False
2006-04-05 18:45:00	2006-04-05 18:45:00	061XX W NEWPORT AVE	BATTERY	SIMPLE	RESIDENCE	False	False
2006-02-17 21:03:14	2006-02-17 21:03:14	037XX W 60TH ST	NARCOTICS	POSS: CANNABIS 30GMS OR LESS	ALLEY	True	False

In [25]:

```
chicago_df['Primary Type'].value_counts()
```

Out[25]:

```
THEFT                                1245111
BATTERY                              1079178
CRIMINAL DAMAGE                       702702
NARCOTICS                             674831
BURGLARY                              369056
OTHER OFFENSE                         368169
ASSAULT                               360244
MOTOR VEHICLE THEFT                   271624
ROBBERY                               229467
DECEPTIVE PRACTICE                    225180
CRIMINAL TRESPASS                     171596
PROSTITUTION                           60735
WEAPONS VIOLATION                      60335
PUBLIC PEACE VIOLATION                 48403
OFFENSE INVOLVING CHILDREN             40260
CRIM SEXUAL ASSAULT                    22789
SEX OFFENSE                            20172
GAMBLING                               14755
INTERFERENCE WITH PUBLIC OFFICER       14009
```

```
LIQUOR LAW VIOLATION                 12129
ARSON                                 9269
HOMICIDE                              5879
KIDNAPPING                            4734
INTIMIDATION                          3324
STALKING                              2866
OBSCENITY                              422
PUBLIC INDECENCY                       134
OTHER NARCOTIC VIOLATION               122
NON-CRIMINAL                            96
CONCEALED CARRY LICENSE VIOLATION       90
NON - CRIMINAL                          38
HUMAN TRAFFICKING                       28
RITUALISM                               16
NON-CRIMINAL (SUBJECT SPECIFIED)         4
Name: Primary Type, dtype: int64
```

In [26]:

```
chicago_df['Primary Type'].value_counts().iloc[:15]
```

Out[26]:

```
THEFT                           1245111
BATTERY                         1079178
CRIMINAL DAMAGE                  702702
NARCOTICS                        674831
BURGLARY                         369056
OTHER OFFENSE                    368169
ASSAULT                          360244
MOTOR VEHICLE THEFT              271624
ROBBERY                          229467
DECEPTIVE PRACTICE               225180
CRIMINAL TRESPASS                171596
PROSTITUTION                      60735
WEAPONS VIOLATION                 60335
PUBLIC PEACE VIOLATION            48403
OFFENSE INVOLVING CHILDREN        40260
Name: Primary Type, dtype: int64
```

In [27]:

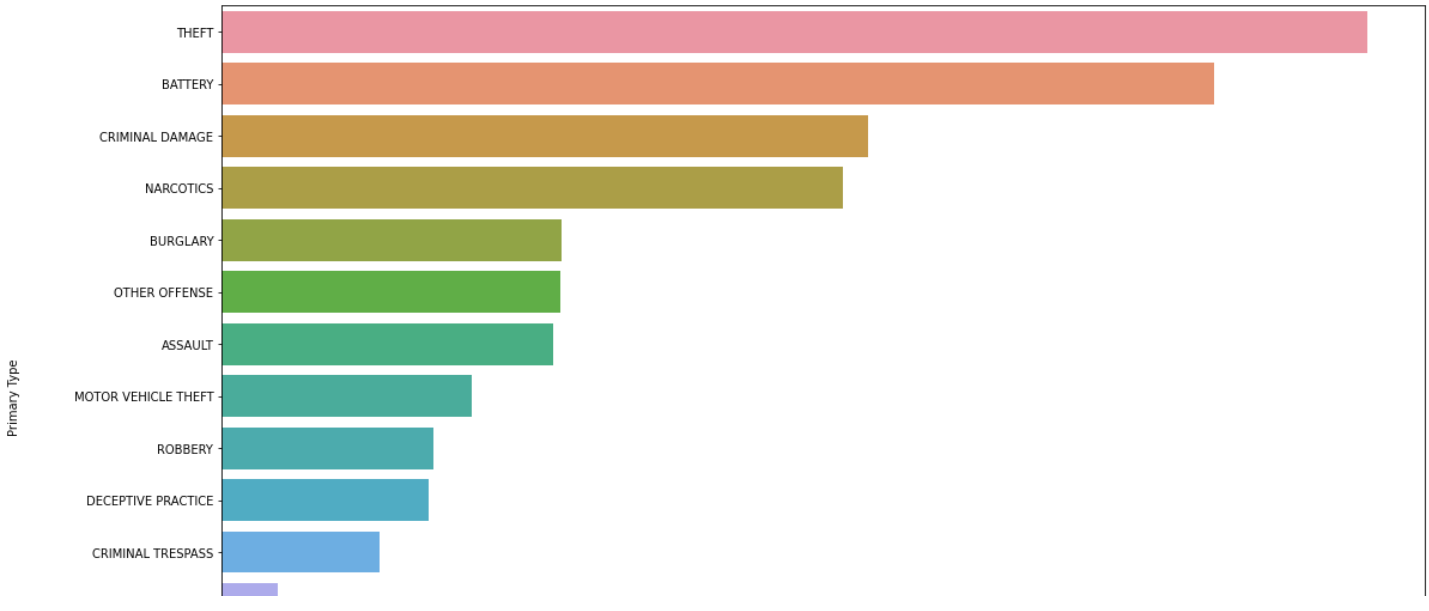
```
order_data=chicago_df['Primary Type'].value_counts().iloc[:15].index
```

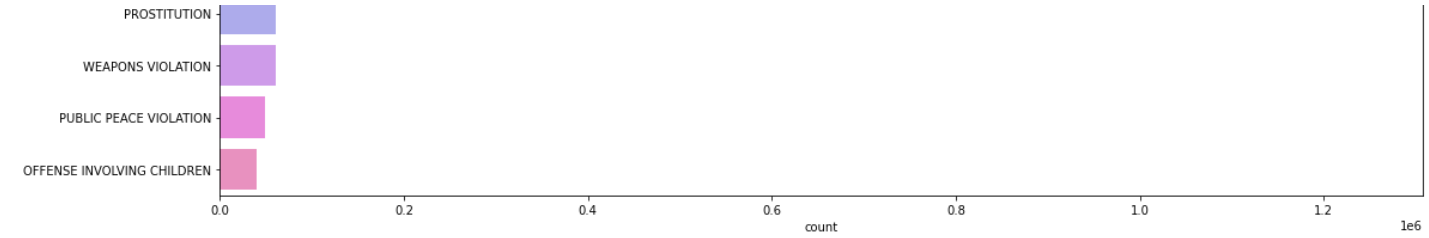
In [28]:

```
plt.figure(figsize=(18,12))
sns.countplot(y='Primary Type',data=chicago_df,order=order_data)
```

Out[28]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f70afdf6d10>
```



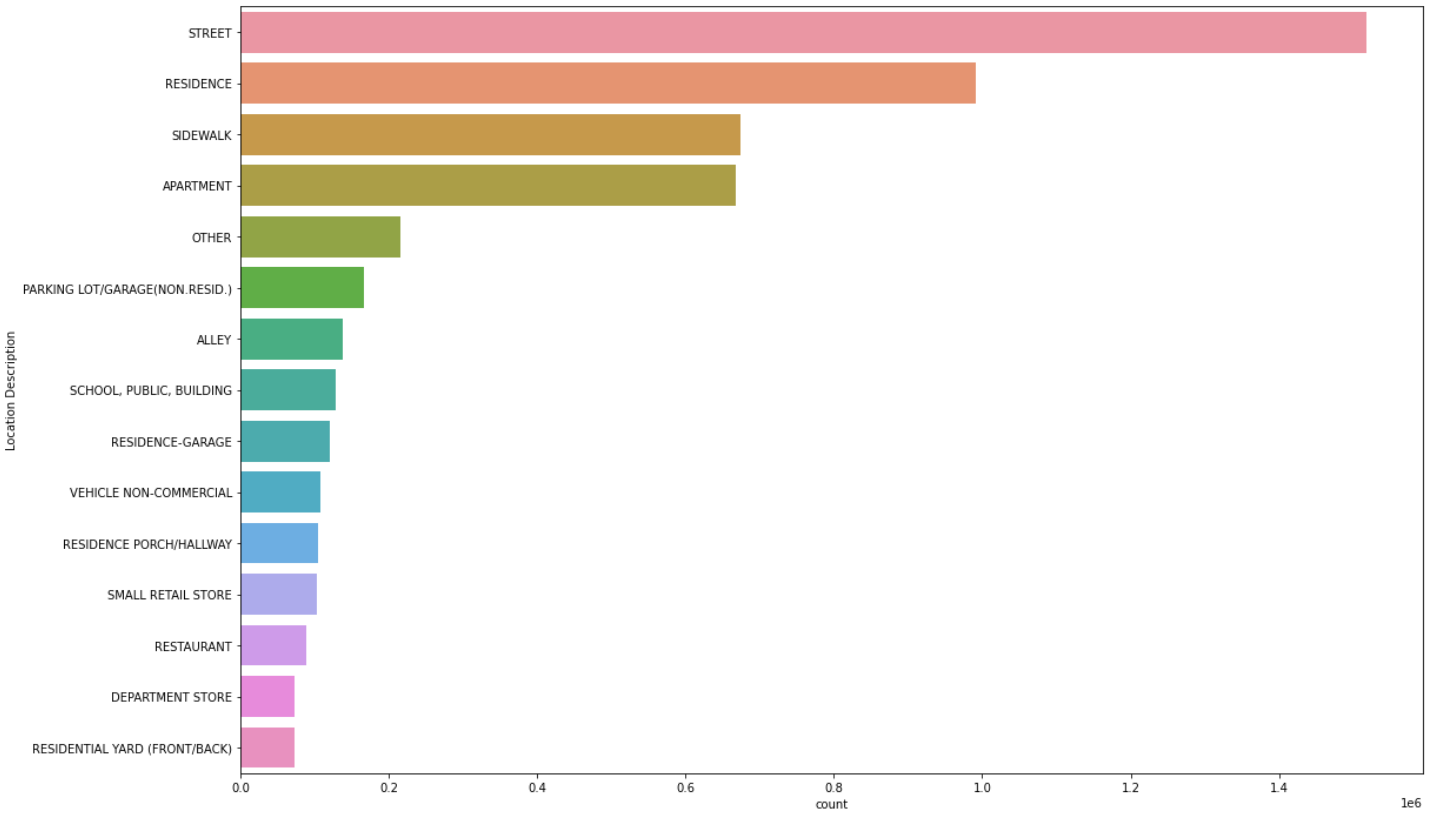


In [29]:

```
plt.figure(figsize=(18,12))
sns.countplot(y='Location Description',data=chicago_df,order=chicago_df['Location Description'].value_counts().iloc[:15].index)
```

Out[29]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f70afdae910>
```



In [30]:

```
chicago_df.resample('Y').size()
```

Out[30]:

```
Date
2005-12-31    455811
2006-12-31    794684
2007-12-31    621848
2008-12-31    852053
2009-12-31    783900
2010-12-31    700691
2011-12-31    352066
2012-12-31    335670
2013-12-31    306703
2014-12-31    274527
2015-12-31    262995
2016-12-31    265462
2017-12-31     11357
Freq: A-DEC, dtype: int64
```

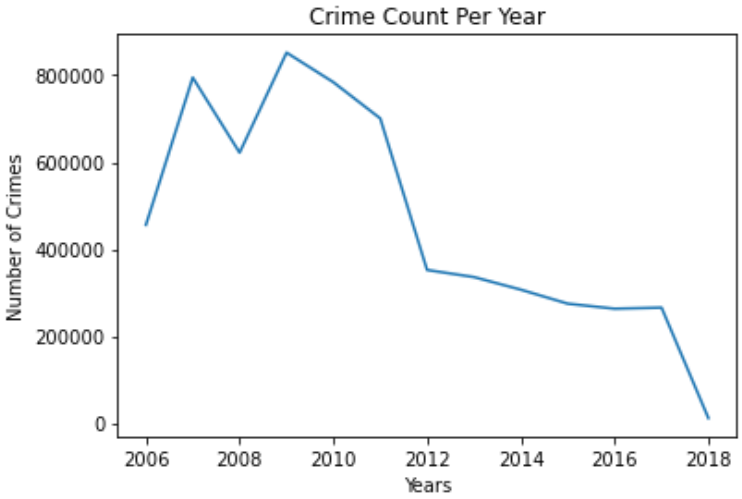
In [31]:

```
plt.plot(chicago_df.resample('Y').size())
plt.title('Crime Count Per Year')
```

```
plt.xlabel('Years')
plt.ylabel('Number of Crimes')
```

Out[31]:

```
Text(0, 0.5, 'Number of Crimes')
```

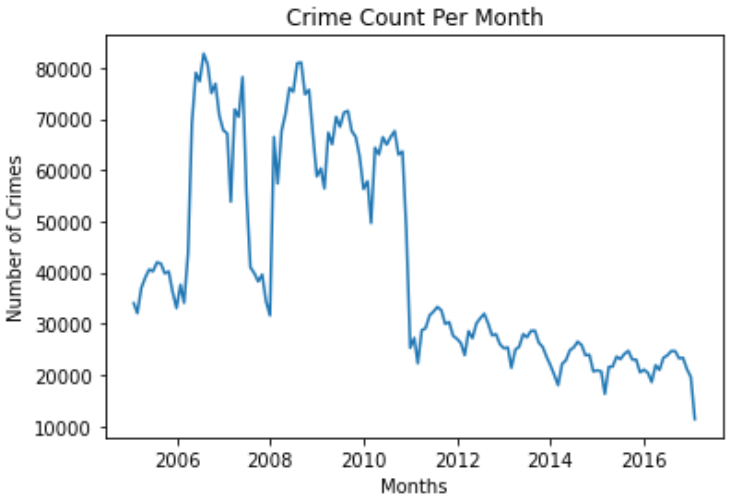


In [32]:

```
plt.plot(chicago_df.resample('M').size())
plt.title('Crime Count Per Month')
plt.xlabel('Months')
plt.ylabel('Number of Crimes')
```

Out[32]:

```
Text(0, 0.5, 'Number of Crimes')
```

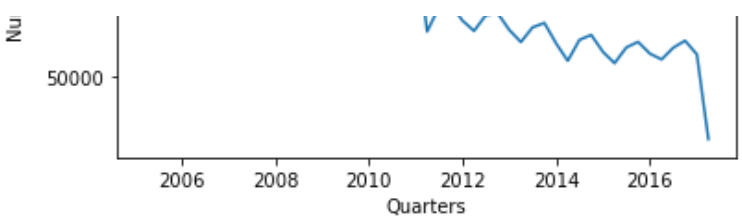


In [33]:

```
plt.plot(chicago_df.resample('Q').size())
plt.title('Crime Count Per Quarter')
plt.xlabel('Quarters')
plt.ylabel('Number of Crimes')
```

Out[33]:

```
Text(0, 0.5, 'Number of Crimes')
```



In [34]:

```
chicago_prophet=chicago_df.resample('M').size().reset_index()
```

In [35]:

```
chicago_prophet
```

Out[35]:

	Date	0
0	2005-01-31	33983
1	2005-02-28	32042
2	2005-03-31	36970
3	2005-04-30	38963
4	2005-05-31	40572
...	...	...
140	2016-09-30	23235
141	2016-10-31	23314
142	2016-11-30	21140
143	2016-12-31	19580
144	2017-01-31	11357

145 rows × 2 columns

In [38]:

```
chicago_prophet.columns=['Date','Crime Count']
```

In [39]:

```
chicago_prophet
```

Out[39]:

	Date	Crime Count
0	2005-01-31	33983
1	2005-02-28	32042
2	2005-03-31	36970
3	2005-04-30	38963
4	2005-05-31	40572
...	...	...
140	2016-09-30	23235
141	2016-10-31	23314
142	2016-11-30	21140
143	2016-12-31	19580
144	2017-01-31	11357

145 rows × 2 columns

In [49]:

```
chicago_proph=chicago_prophet.rename(columns={'Date':'ds','Crime Count':'y'})
```

In [50]:

```
chicago_proph
```

Out[50]:

	ds	y
0	2005-01-31	33983
1	2005-02-28	32042
2	2005-03-31	36970
3	2005-04-30	38963
4	2005-05-31	40572
...	...	...
140	2016-09-30	23235
141	2016-10-31	23314
142	2016-11-30	21140
143	2016-12-31	19580
144	2017-01-31	11357

145 rows × 2 columns

In [51]:

```
m=Prophet()
m.fit(chicago_proph)
```

Out[51]:

```
<fbprophet.forecaster.Prophet at 0x7f70a178fa90>
```

In [53]:

```
future=m.make_future_dataframe(periods=365)
forecast=m.predict(future)
```

In [54]:

```
forecast
```

Out[54]:

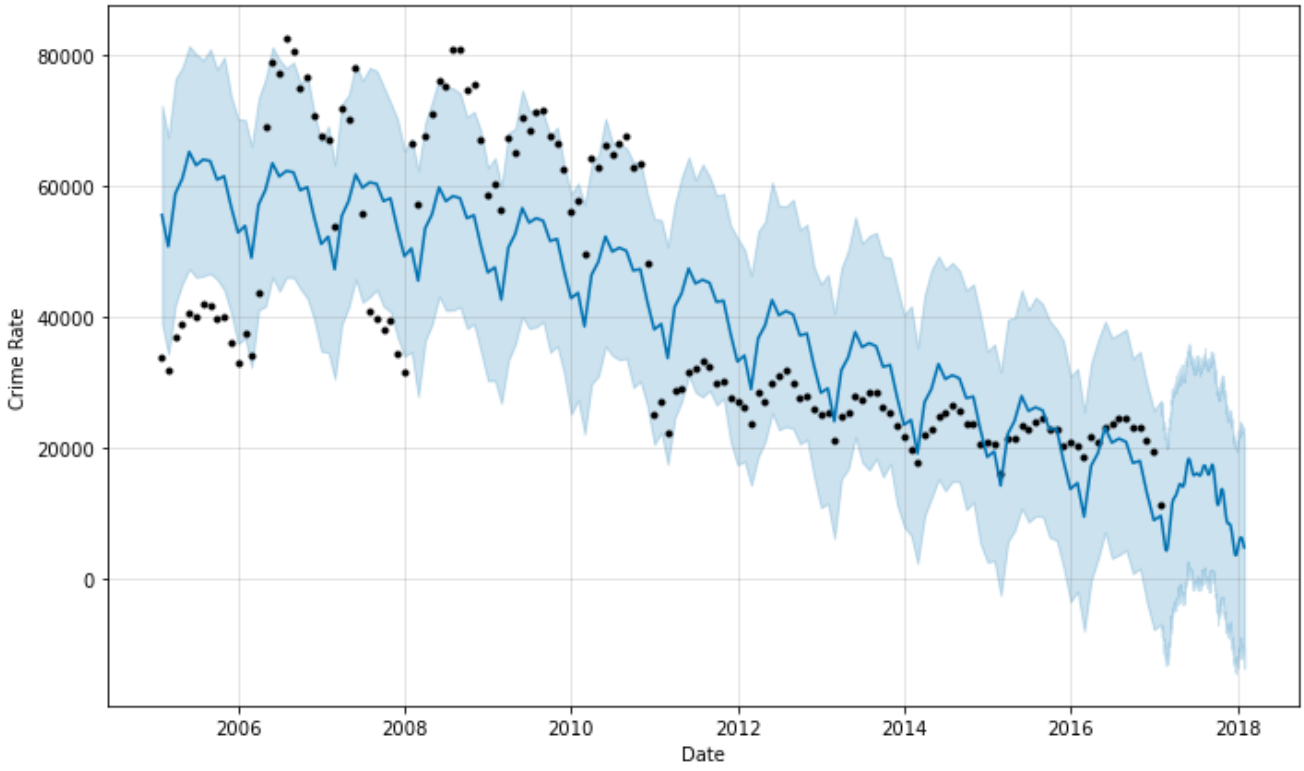
	ds	trend	yhat_lower	yhat_upper	trend_lower	trend_upper	additive_terms	additive_terms_lower	addi
0	2005-01-31	60463.432502	39079.648106	72364.447916	60463.432502	60463.432502	-4803.616721	-4803.616721	
1	2005-02-28	60332.766196	34414.224651	67558.089070	60332.766196	60332.766196	-9480.730557	-9480.730557	
2	2005-03-31	60188.099929	41970.122398	76459.571124	60188.099929	60188.099929	-1254.934724	-1254.934724	
3	2005-04-30	60048.100316	45187.083890	78238.832901	60048.100316	60048.100316	1114.162021	1114.162021	
4	2005-05-31	59903.434049	47438.465190	81474.424851	59903.434049	59903.434049	5396.906083	5396.906083	
...	...	...	...	...	...	...	...	...	
505	2018-01-27	9701.640380	-12026.678853	22819.008393	9545.141608	9869.622702	-4092.154604	-4092.154604	

	ds	trend	yhat_lower	yhat_upper	trend_lower	trend_upper	additive_terms	additive_terms_lower	addi
506	2018-01-28	9688.312243	-11602.594272	22158.585219	9531.241837	9856.895192	-4239.488484	-4239.488484	
507	2018-01-29	9674.984106	-11460.864587	22586.636803	9517.049682	9844.167681	-4399.420072	-4399.420072	
508	2018-01-30	9661.655969	-10527.817503	21580.512734	9502.857527	9831.787000	-4571.605324	-4571.605324	
509	2018-01-31	9648.327832	-13586.688912	20802.084120	9488.665371	9819.428663	-4755.746242	-4755.746242	

510 rows × 16 columns

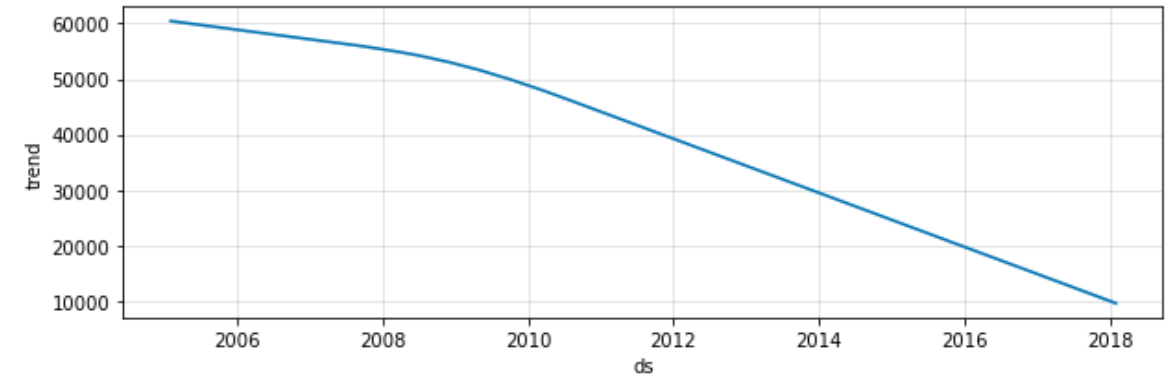
In [55]:

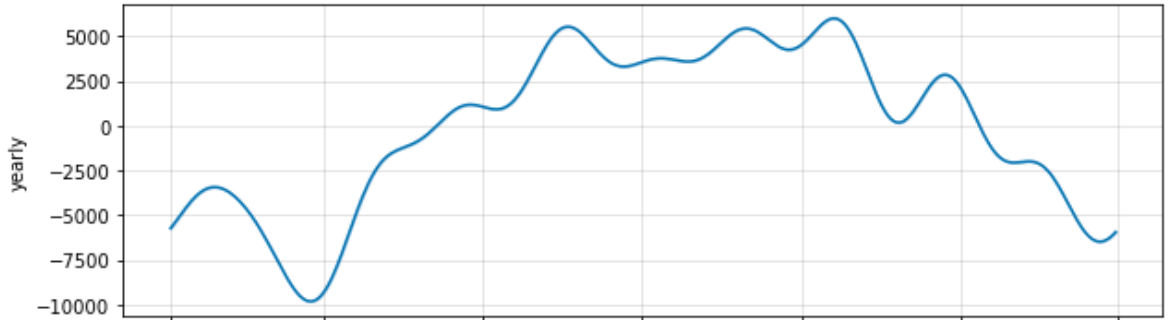
```
figure=m.plot(forecast,xlabel="Date",ylabel="Crime Rate")
```

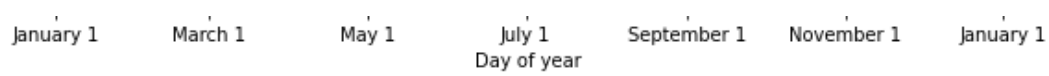


In [56]:

```
figure=m.plot_components(forecast)
```





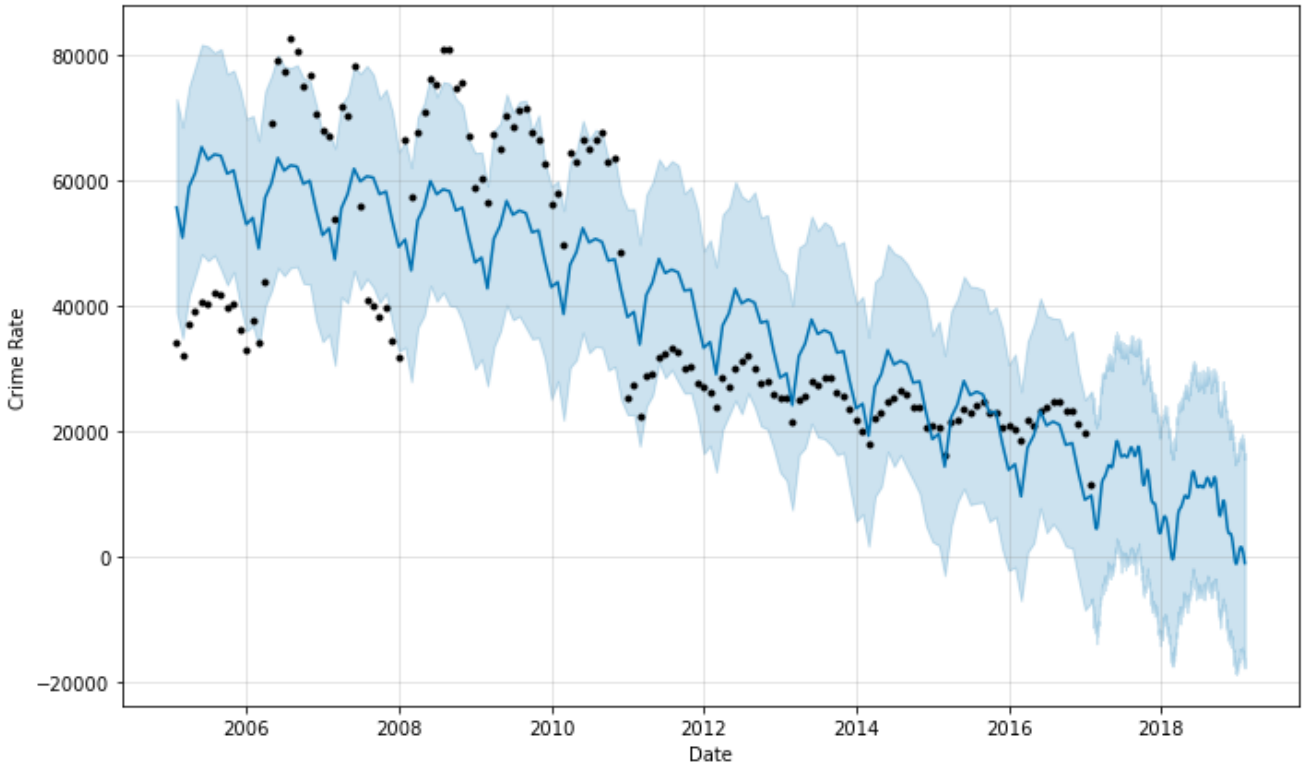


In [60]:

```
future=m.make_future_dataframe(periods=720)
forecast=m.predict(future)
```

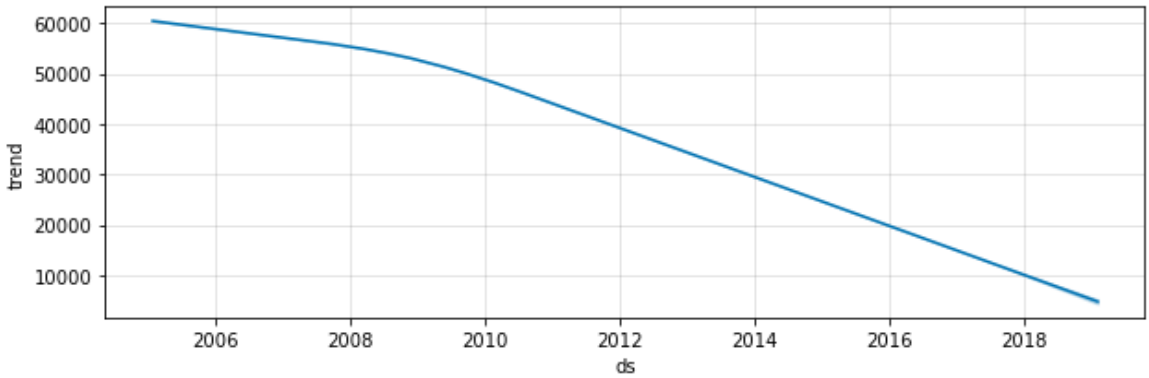
In [61]:

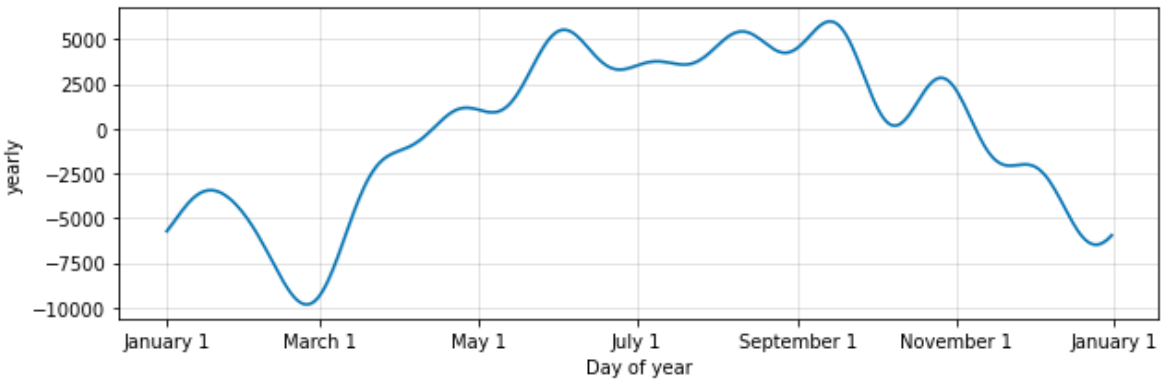
```
figure=m.plot(forecast,xlabel="Date",ylabel="Crime Rate")
```



In [62]:

```
figure=m.plot_components(forecast)
```





In [ ]: