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

In [2]:
```
crimedf = pd.read_excel('crime_data.xlsx')
```

In [3]:
```
transactdf=pd.read_excel('transaction_data.xlsx')
crimedf.head()
```

Out[3]:

	Year	Population	Violent crime total	Murder and nonnegligent manslaughter	Forcible rape	Robbery	Aggravated assault	Property crime total	Burglary	Larceny the
0	1960	179323175	288460	9110	17190	107840	154320	3095700	912100	18554(
1	1961	182992000	289390	8740	17220	106670	156760	3198600	949600	19130(
2	1962	185771000	301510	8530	17550	110860	164570	3450700	994300	20896(
3	1963	188483000	316970	8640	17650	116470	174210	3792500	1086400	22978(
4	1964	191141000	364220	9360	21420	130390	203050	4200400	1213200	25144(

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

Out[4]:

	Transaction	Purchase Date	Customer ID	Gender	Marital Status	Homeowner	Children	Annual Income	City	State Provir
0	1	2014-12-18 00:00:00	7223	F	S	Y	2	$30K-$50K	Los Angeles	
1	2	2014-12-20 00:00:00	7841	M	M	Y	5	$70K-$90K	Los Angeles	
2	3	2014-12-21 00:00:00	8374	F	M	N	2	$50K-$70K	Bremerton	\
3	4	2014-12-21 00:00:00	9619	M	M	Y	3	$30K-$50K	Portland	
4	5	2014-12-22 00:00:00	1900	F	S	Y	3	$130K-$150K	Beverly Hills	

In [5]:
```
crimedf.dtypes
```

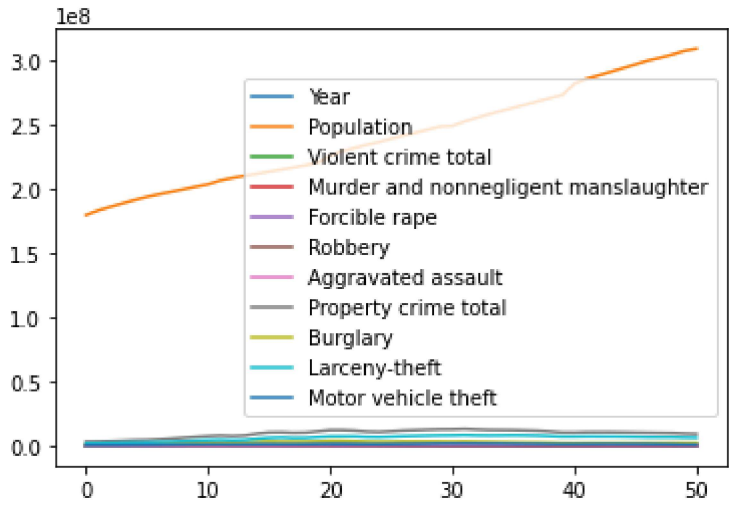
Out[5]:
```
Year                                  int64
Population                            int64
Violent crime total                   int64
Murder and nonnegligent manslaughter  int64
Forcible rape                         int64
Robbery                               int64
Aggravated assault                    int64
Property crime total                  int64
Burglary                              int64
Larceny-theft                         int64
Motor vehicle theft                   int64
dtype: object
```

In [6]:
```
transactdf.dtypes
```

Out[6]:
```
Transaction             int64
Purchase Date          object
Customer ID             int64
Gender                 object
Marital Status         object
Homeowner              object
Children                int64
Annual Income          object
City                   object
State or Province      object
Country                object
Product Family         object
Product Department     object
Product Category       object
Units Sold              int64
Revenue               float64
dtype: object
```

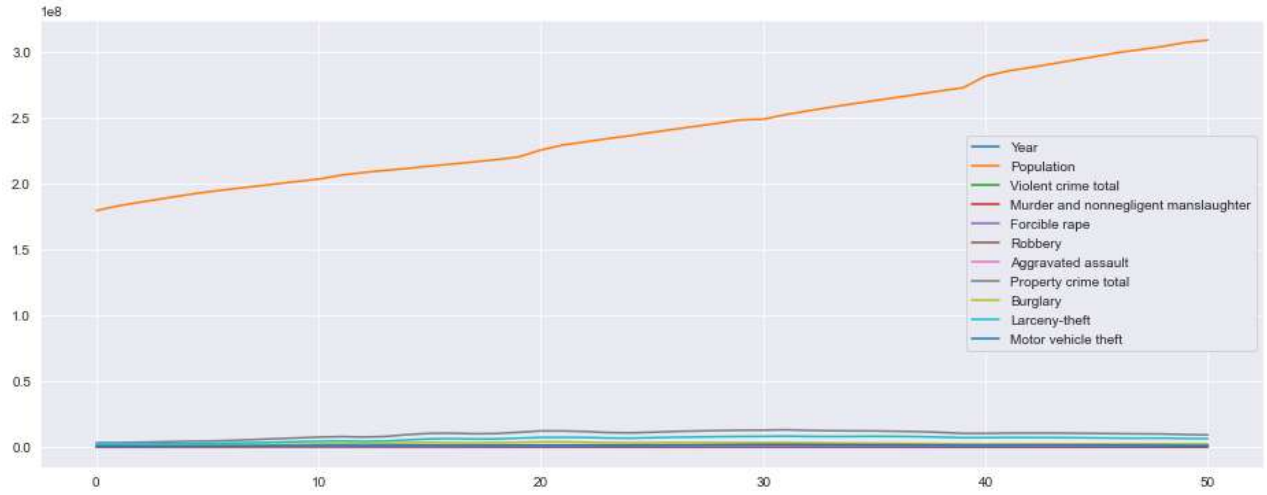
In [7]:
```
import matplotlib.pyplot as plt
import seaborn as sns
from statsmodels.tsa.ar_model import AutoReg, ar_select_order
from statsmodels.tsa.api import acf, pacf, graphics
```

In [8]:
```
fig,ax = plt.subplots()
ax = crimedf.plot(ax=ax)
```

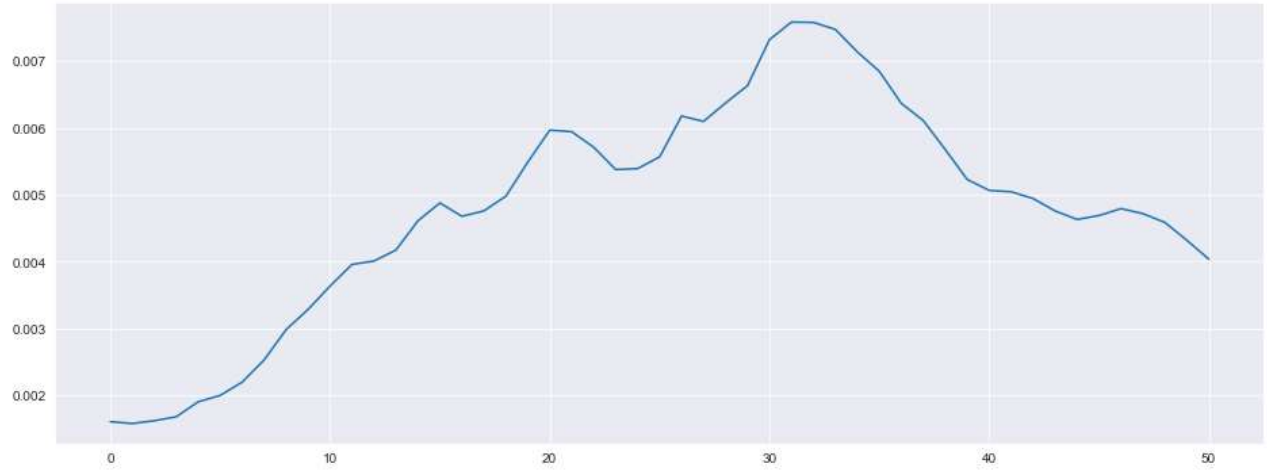


In [9]:
```
sns.set_style('darkgrid')
pd.plotting.register_matplotlib_converters()
sns.mpl.rc('figure', figsize=(16,6))
```

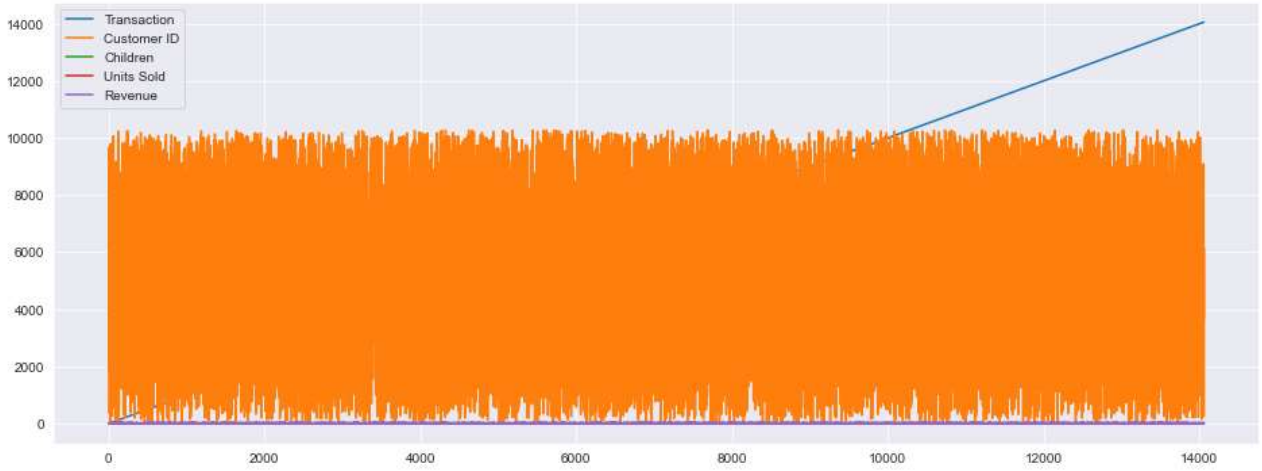
In [10]:
```
fig,ax = plt.subplots()
ax = crimedf.plot(ax=ax)
```



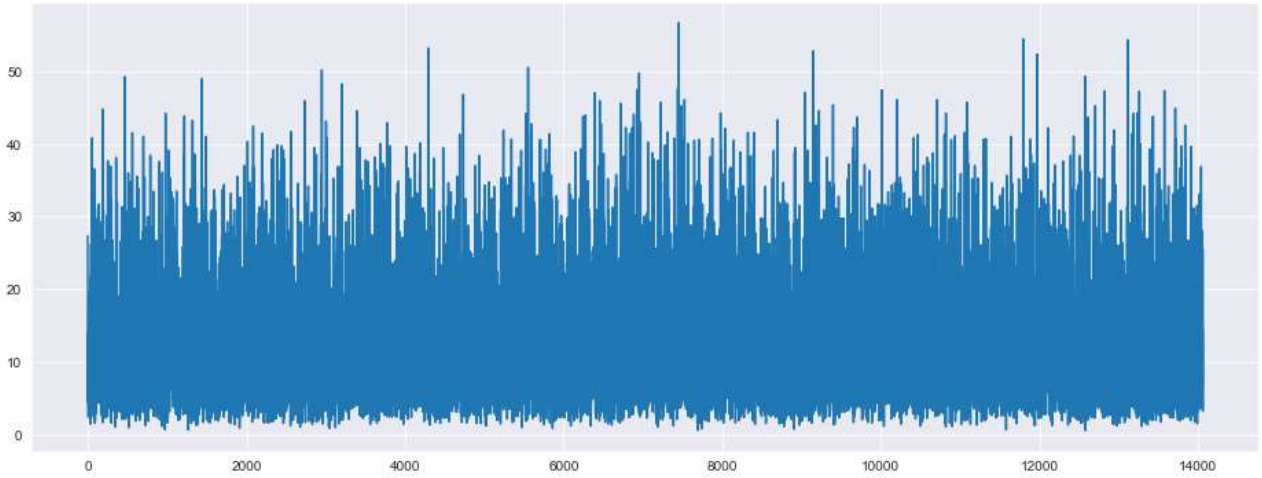
In [11]:
```
crime_rate=crimedf.copy()
crime_rate['Violent crime rate'] = crimedf['Violent crime total']/crimedf['Population']
crime_rate['Year']=crimedf['Year']
fig, ax = plt.subplots()
ax = crime_rate['Violent crime rate'].plot(ax=ax)
```



In [12]:
```
fig,ax = plt.subplots()
ax = transactdf.plot(ax=ax)
```



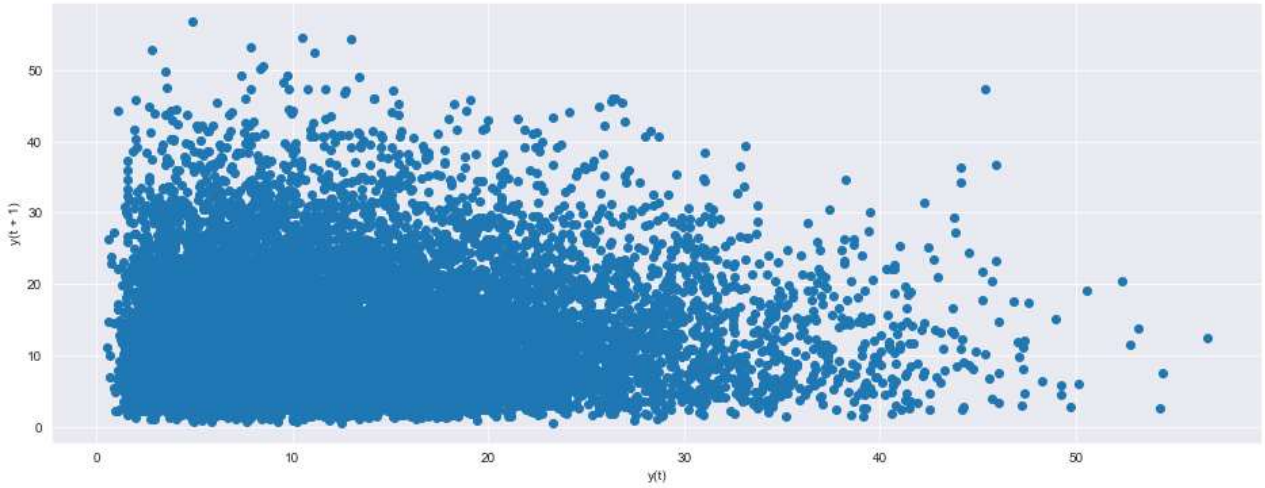
In [13]:
```
fig,ax = plt.subplots()
ax = transactdf['Revenue'].plot(ax=ax)
```



In [14]:
```
transact_small=transactdf.copy()
transact_small.drop(['Transaction'],axis=1, inplace=True)
transact_small.drop(['Customer ID'],axis=1, inplace=True)
```

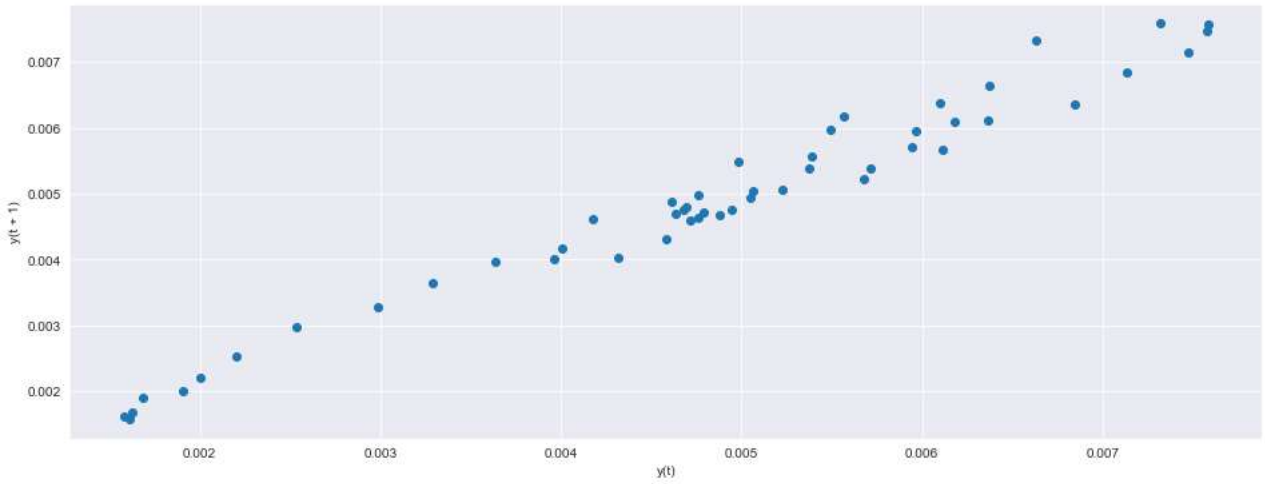
In [15]:
```
import matplotlib.pylab as plt
pd.plotting.lag_plot(transact_small['Revenue'])
```

Out[15]: <AxesSubplot:xlabel='y(t)', ylabel='y(t + 1)'>



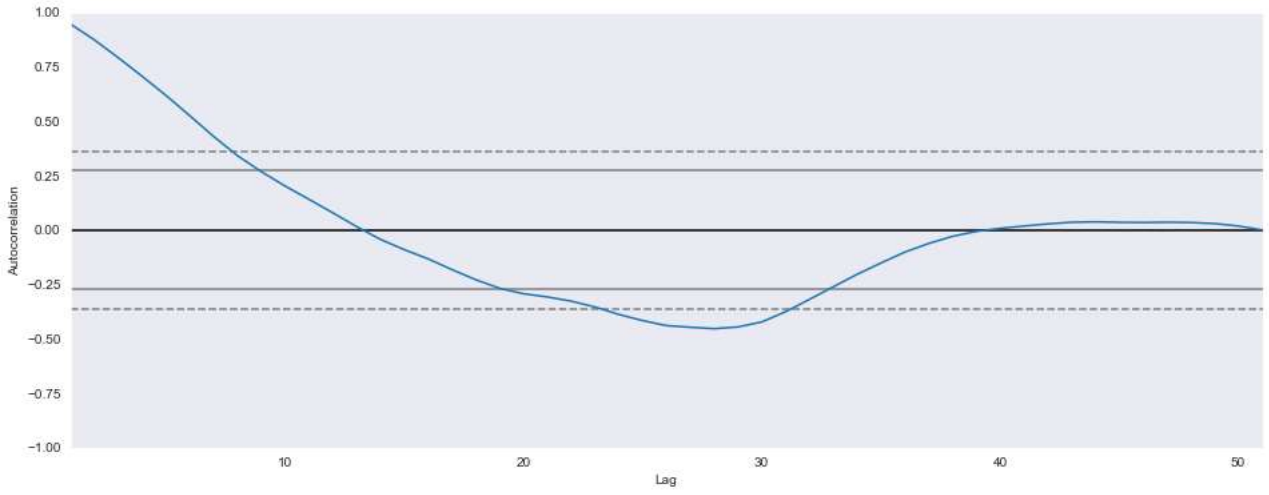
In [16]:
```
pd.plotting.lag_plot(crime_rate['Violent crime rate'])
```

Out[16]: <AxesSubplot:xlabel='y(t)', ylabel='y(t + 1)'>



In [17]:
```
pd.plotting.autocorrelation_plot(crime_rate['Violent crime rate'])
```

Out[17]: <AxesSubplot:xlabel='Lag', ylabel='Autocorrelation'>



In [18]:
```
transact_small['Revenue'].corr(transact_small['Revenue'].shift(50))
```

Out[18]:
```
-0.004421155977020592
```

In [19]:
```
crime_rate['Violent crime rate'].corr(crime_rate['Violent crime rate'].shift(30))
```

Out[19]:
```
-0.9654049596830483
```

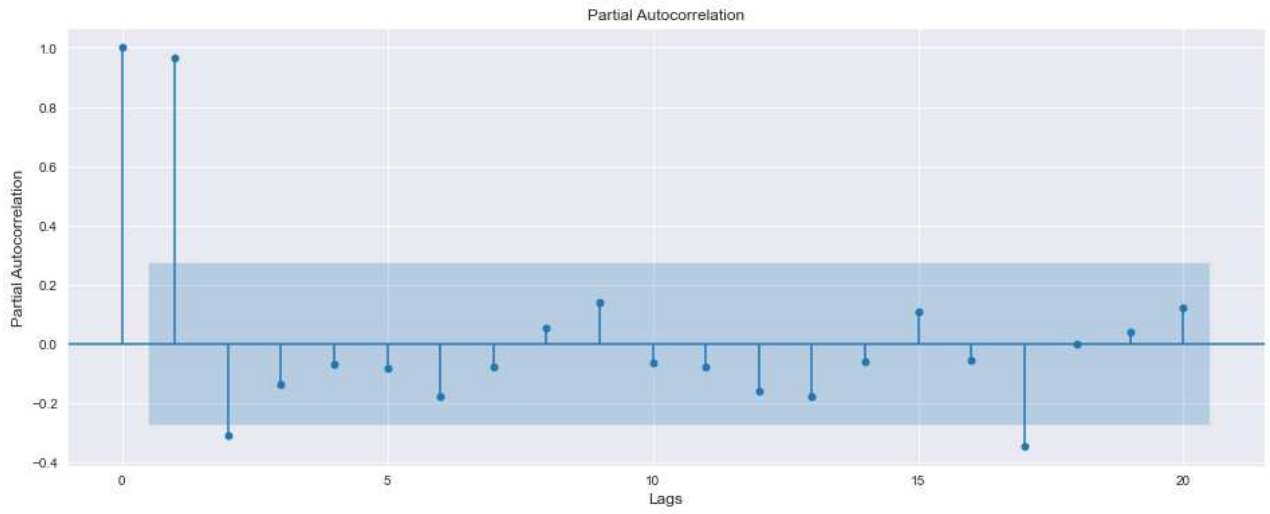
In [20]:
```
from statsmodels.tsa.ar_model import AutoReg
#to set up training set for time series data, use the first 80% of the data, and test i
model = AutoReg(crime_rate['Violent crime rate'],2, old_names=False)
model_fitted = model.fit()
```

In [21]:
```
model_fitted.params
```

Out[21]:
```
const                    0.000215
Violent crime rate.L1    1.573859
Violent crime rate.L2   -0.614711
dtype: float64
```

In [22]:
```
from statsmodels.graphics.tsaplots import plot_pacf
```

In [23]:
```
plot_pacf(crime_rate['Violent crime rate'], lags=20)
plt.xlabel('Lags', fontsize=12)
plt.ylabel('Partial Autocorrelation', fontsize=12)
plt.show()
#based on the graph below, use lags of 1 and 2 in the model at least
```



In [24]:
```
from statsmodels.tsa.stattools import adfuller

result = adfuller(crime_rate['Violent crime rate'])
print('p-value: %.2f' % result[1])
```

```
p-value: 0.24
```

In [25]:
```
crime_rate['Difference'] = crime_rate['Violent crime rate'].diff()
```

```
result = adfuller(crime_rate['Difference'].dropna())
print('p-value: %.2f' % result[1])
```

```
p-value: 0.03
```

In [26]:

```
model = AutoReg(crime_rate['Difference'].dropna(),2, old_names=False)
model_fitted = model.fit()
```

```
C:\Users\Top\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:578: ValueWar
ning: An unsupported index was provided and will be ignored when e.g. forecasting.
  warnings.warn('An unsupported index was provided and will be'
```

In [27]:

```
model_fitted = model.fit()
```

In [28]:

```
model_fitted.params
```

Out[28]:

```
const            0.000014
Difference.L1    0.669151
Difference.L2   -0.030646
dtype: float64
```

In [ ]: