In [1]:

```
import pandas as pd
import numpy as np
import matplotlib as mpl
from matplotlib import pyplot as plt
import plotly.express as px
pd.set_option('display.max_columns', 50)
pd.set_option('display.max_rows', 550)
total = pd.read_csv('C:/Users/Laptop/Desktop/Grypa/Dane/total.csv')
total['Rok_str'] = total['Rok_str'].astype(str)
total.drop(total.columns[total.columns.str.contains('unnamed', case = False)], axis=1,
#total['Unique_ID'] = total['Rok_str'] + '_' + total['Tydzien']
#total
```

In [2]:

```
total['Sezon_str'] = total['Sezon'].astype(str)
#total
```

In [5]:

```
#zmienic zakres na dataframe!!
for x in zakres:
    print(x)
    sezon = total.loc[total['Sezon_str'] == x]
sezon
```

```
2011
2019
2018
2013
2012
2014
2017
2015
2010
2016
```

In [6]:

```
1 total['Total'].describe()
```

Out[6]:

```
count       476.000000
mean      72532.334034
std       59087.554448
min        3221.000000
25%       27348.250000
50%       56827.000000
75%      102172.250000
max      363583.000000
Name: Total, dtype: float64
```

In [7]:

```
1 total2 = total.loc[96:]
2 total2
```

Out[7]:

	URL	Plik	Rok	Miesia
96	http://wwwold.pzh.gov.pl/oldpage/epimeld/grypa...	C:/Users/Laptop/Desktop/Grypa/Dane/2012/09A.pdf	2012	9
97	http://wwwold.pzh.gov.pl/oldpage/epimeld/grypa...	C:/Users/Laptop/Desktop/Grypa/Dane/2012/09B.pdf	2012	9
98	http://wwwold.pzh.gov.pl/oldpage/epimeld/grypa...	C:/Users/Laptop/Desktop/Grypa/Dane/2012/09C.pdf	2012	9
99	http://wwwold.pzh.gov.pl/oldpage/epimeld/grypa...	C:/Users/Laptop/Desktop/Grypa/Dane/2012/09D.pdf	2012	9
100	http://wwwold.pzh.gov.pl/oldpage/epimeld/grypa...	C:/Users/Laptop/Desktop/Grypa/Dane/2012/10A.pdf	2012	10
101	http://wwwold.pzh.gov.pl/oldpage/epimeld/grypa...	C:/Users/Laptop/Desktop/Grypa/Dane/2012/10B.pdf	2012	10

In [8]:

```
1 seria2 = pd.DataFrame
2 seria2 = total2[['data', 'Total']]
3 seria2.to_csv('C:/Users/Laptop/Desktop/Grypa/Dane/seria2.csv')
```

In [11]:

```
1 #seria2
```

In [9]:

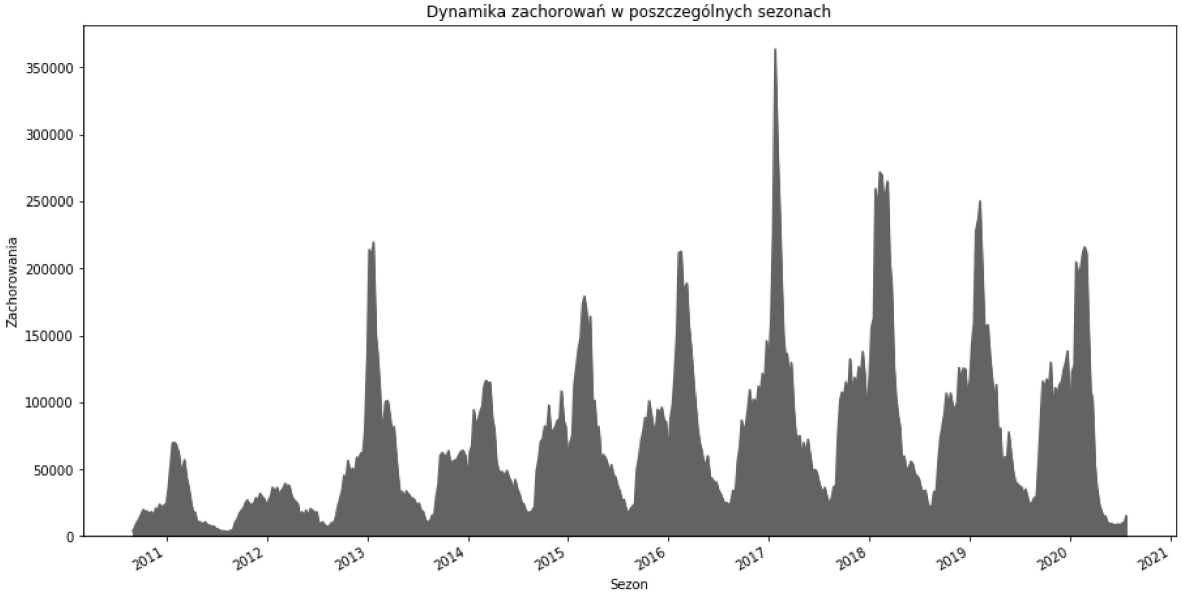
```
1 seria = pd.DataFrame
2 seria = total[['data', 'Total']]
3 seria.to_csv('C:/Users/Laptop/Desktop/Grypa/Dane/seria.csv')
```

In [10]:

```
series = pd.read_csv('C:/Users/Laptop/Desktop/Grypa/Dane/seria.csv', parse_dates=True,
series.drop(series.columns[series.columns.str.contains('unnamed', case = False)], axis=
#series
```

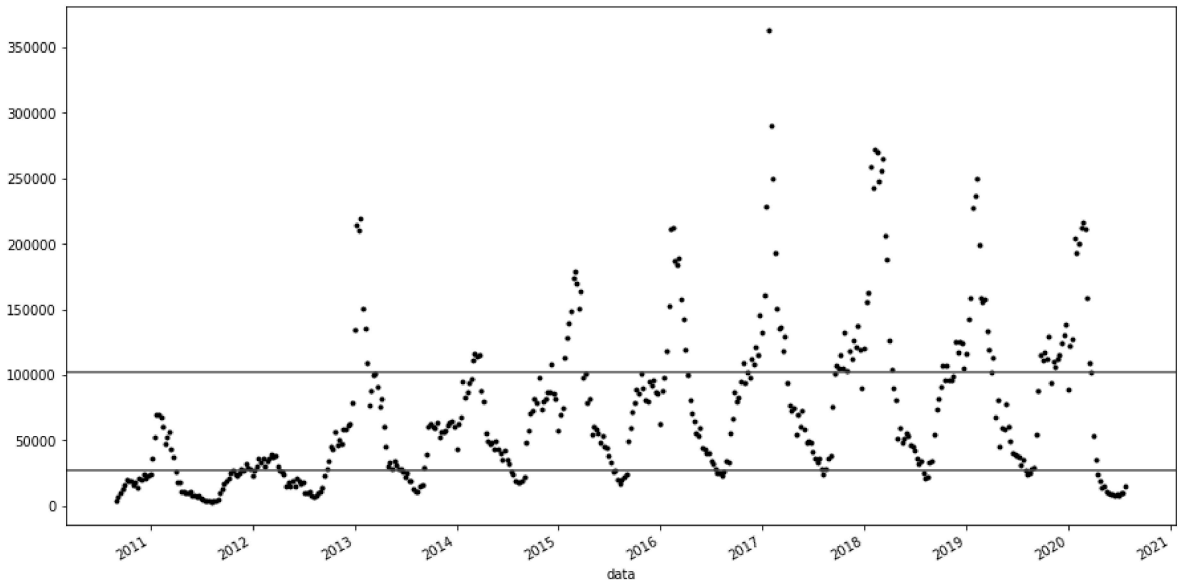
In [11]:

```
series.plot.area(figsize=(15,8), legend=None)
plt.title('Dynamika zachorowań w poszczególnych sezonach')
plt.ylabel('Zachorowania')
plt.xlabel('Sezon')
plt.show()
```



In [12]:

```
series.plot(style='k.', figsize=(15,8), legend=None)
plt.axhline(y=27348, color='red')
plt.axhline(y=102172, color='red')
plt.show()
```

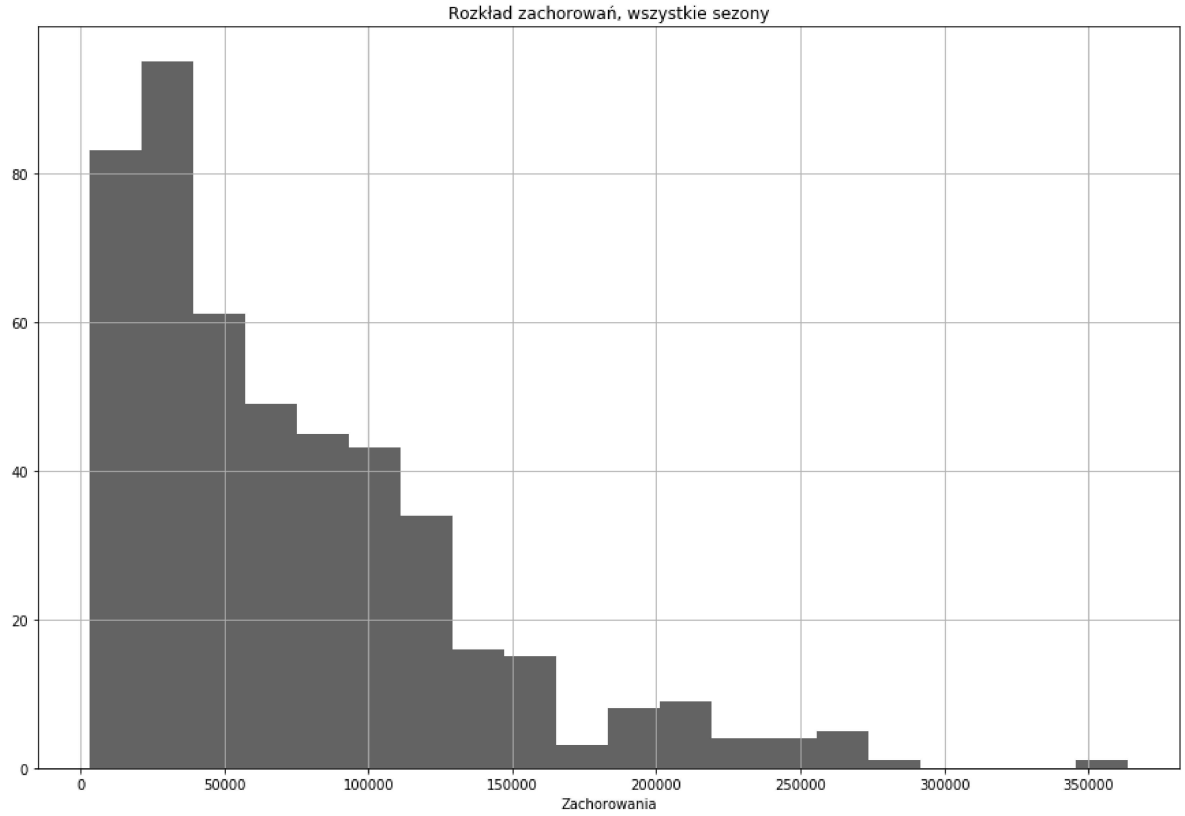


In [13]:

```
from pandas import Grouper
from pandas import DataFrame
from matplotlib import pyplot
from pandas import read_csv
```

In [15]:

```
series3 = pd.read_csv('C:/Users/Laptop/Desktop/Grypa/Dane/seria.csv', parse_dates=True
series3.drop(series3.columns[series3.columns.str.contains('unnamed', case = False)], a
series.hist(figsize = (15, 10), bins=20)
plt.xlabel('Zachorowania')
plt.title('Rozkład zachorowań, wszystkie sezony')
plt.show()
```



In [82]:

```
#series3
```

In [38]:

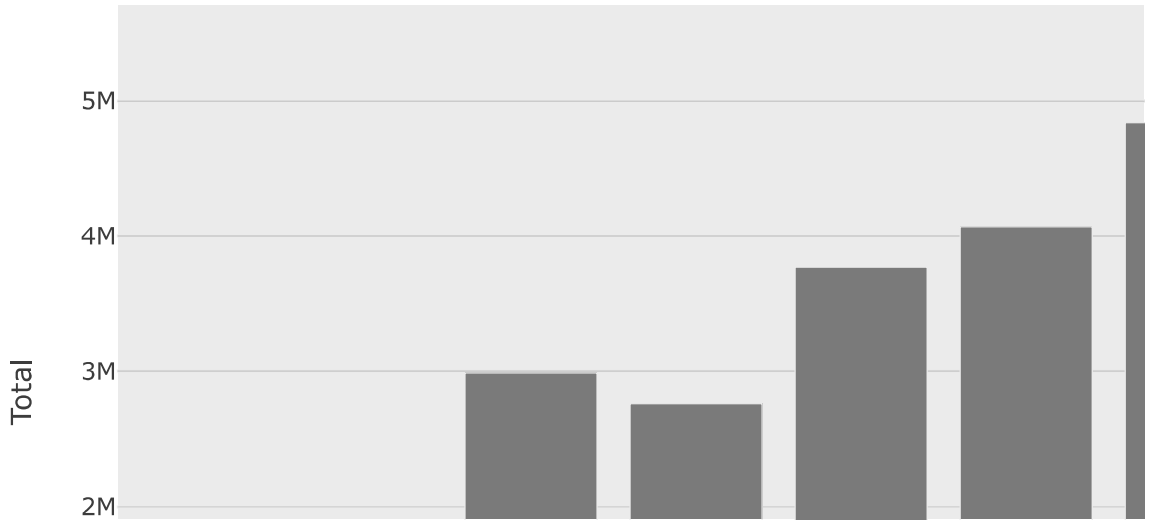
```
#series
```

In [16]:

```
suma_sezonow = total.groupby('Sezon', as_index=False)['Total'].sum()
#suma_sezonow
#suma_sezonow.to_csv('C:/Users/Laptop/Desktop/Grypa/Dane/suma_sezonow.csv')
```

In [18]:

```
fig = px.bar(suma_sezonow, x='Sezon', y='Total')
fig.show()
```



In [17]:

```
series3 = pd.read_csv('C:/Users/Laptop/Desktop/Grypa/Dane/seria.csv', parse_dates=True
series3.drop(series3.columns[series3.columns.str.contains('unnamed', case = False)], a
#series3
```

In [18]:

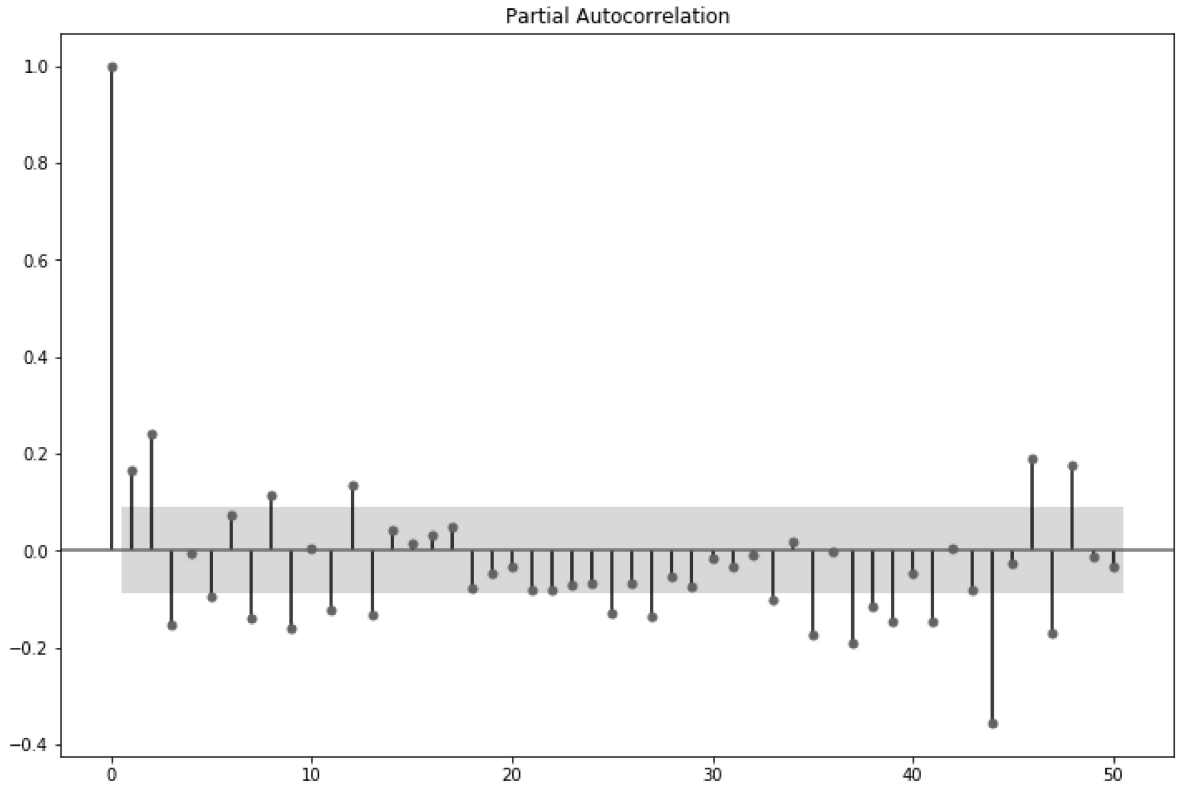
```
from pandas.plotting import autocorrelation_plot
from statsmodels.graphics.tsaplots import plot_pacf
```

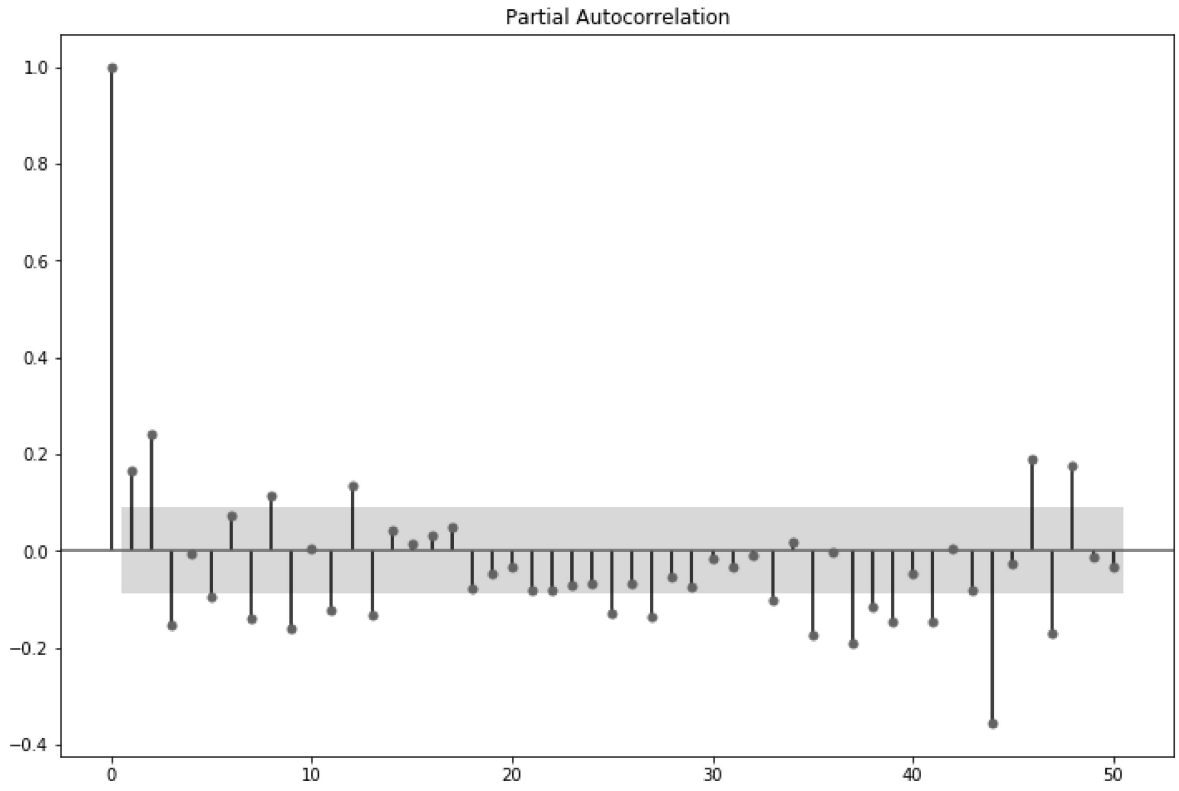
```
C:\Users\Laptop\Anaconda3\lib\site-packages\statsmodels\tools\_testing.py:1
9: FutureWarning:

pandas.util.testing is deprecated. Use the functions in the public API at pa
ndas.testing instead.
```

In [60]:

```
1 plot_pacf(series5['Differenced'], lags=50)
```

Out[60]:





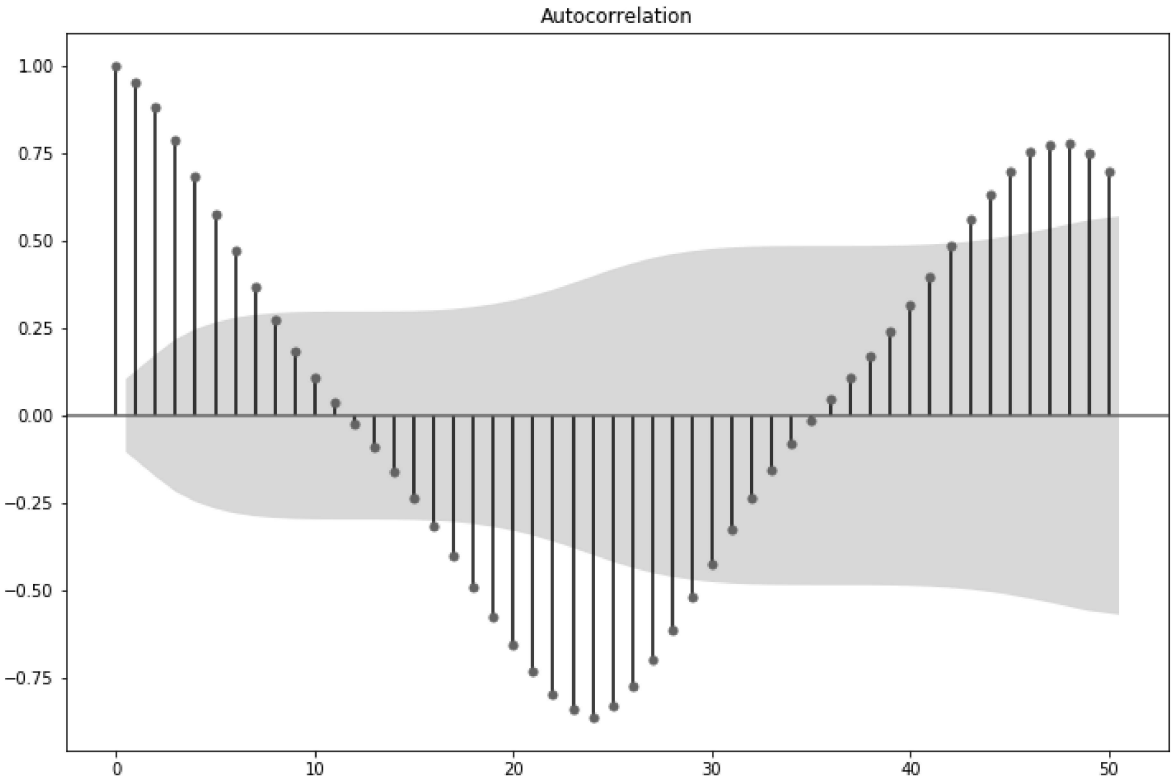
In [68]:

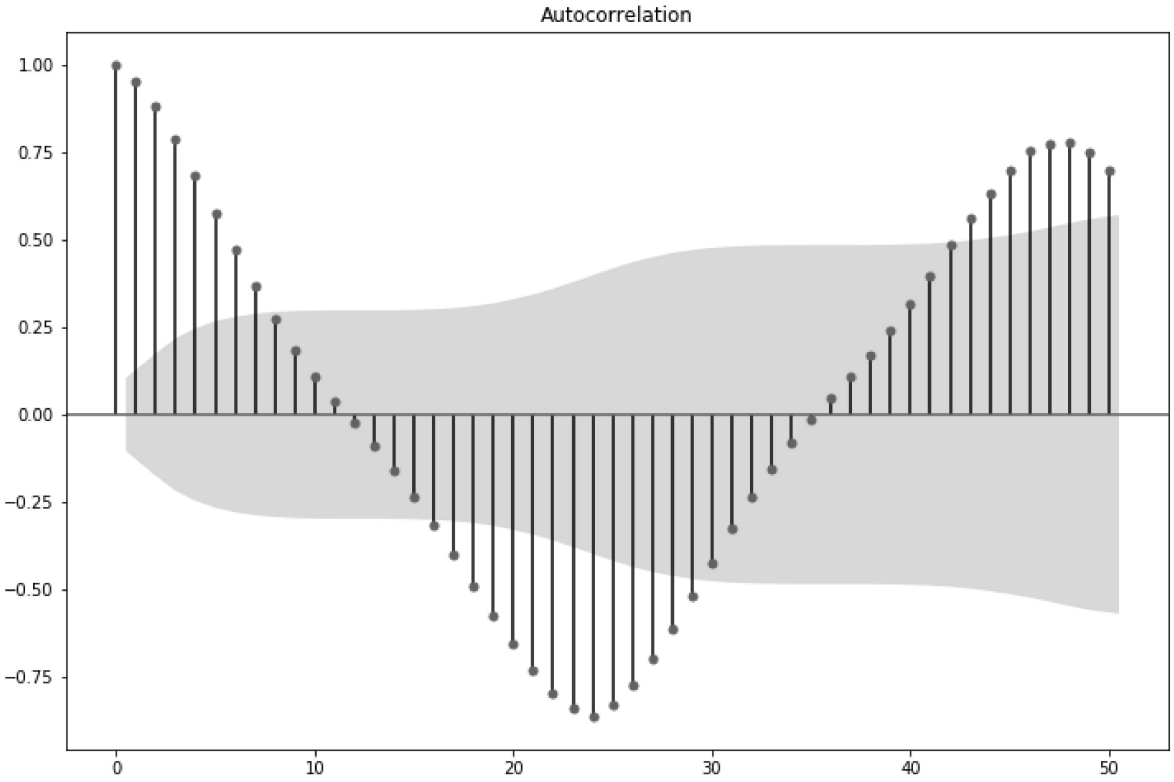
```
1 from statsmodels.graphics.tsaplots import plot_acf
```

In [69]:

```
1 plot_acf(series5['Differenced'], lags=50)
```

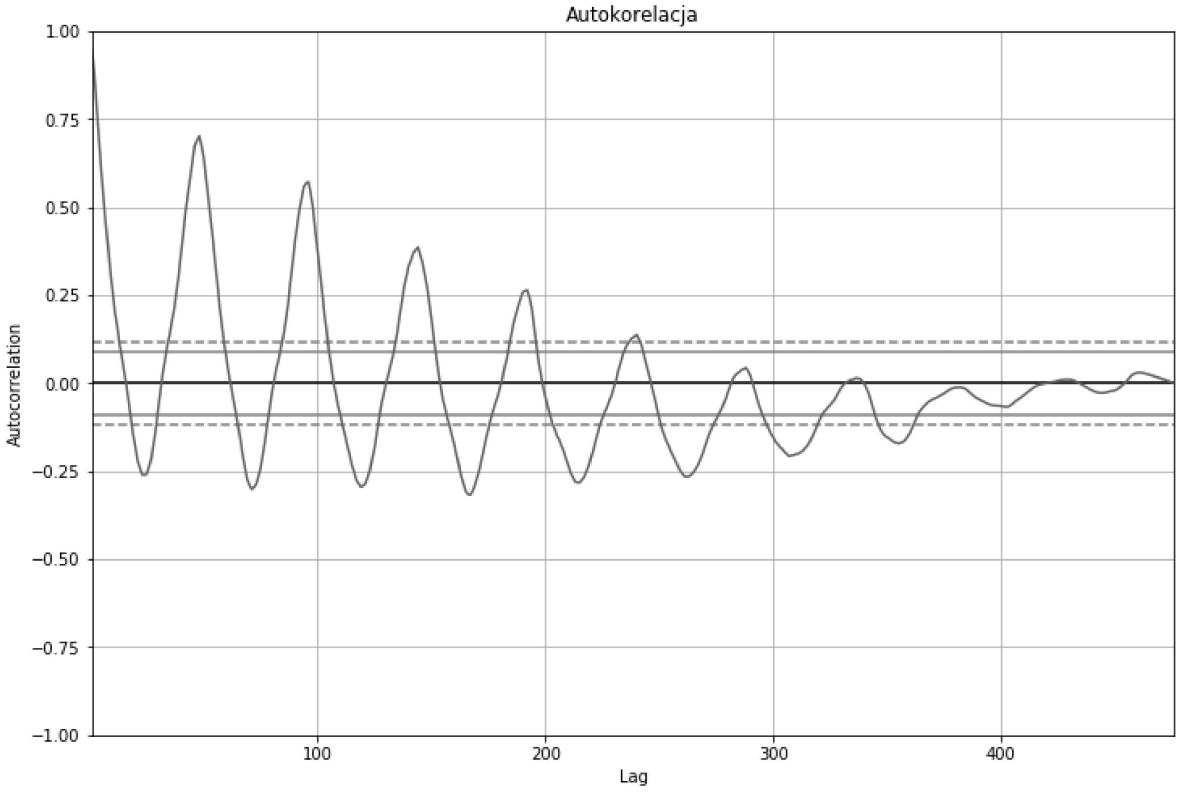
Out[69]:





In [36]:

```
autocorrelation_plot(series3)
plt.title('Autokorelacja')
#plt.xlabel=None
plt.show()
```



In [29]:

```
from statsmodels.tsa.seasonal import seasonal_decompose
```

In [30]:

```
from matplotlib import pylab
```

In [34]:

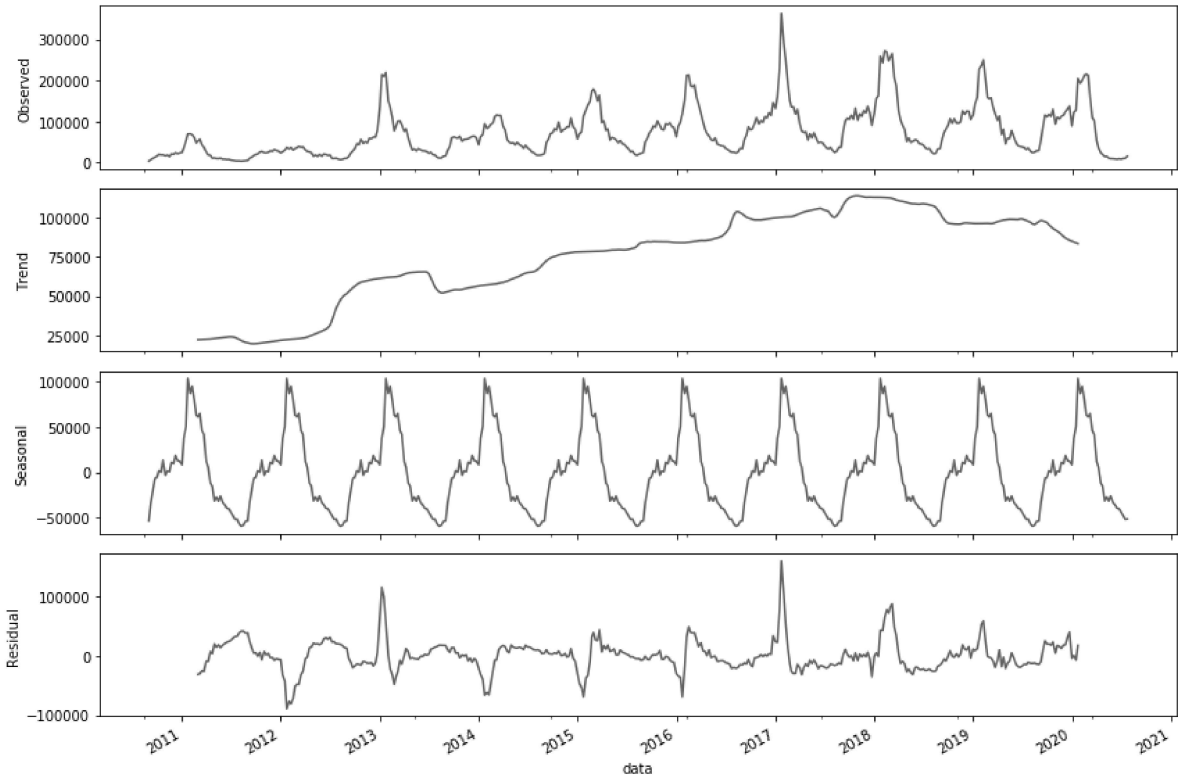
```
series3
#len(series3)
```

Out[34]:

data	Total
2010-09-01	3604
2010-09-08	6892
2010-09-16	10398
2010-09-23	12631
2010-10-01	15750
2010-10-08	19872
2010-10-16	18634
2010-10-23	18753
2010-11-01	16372

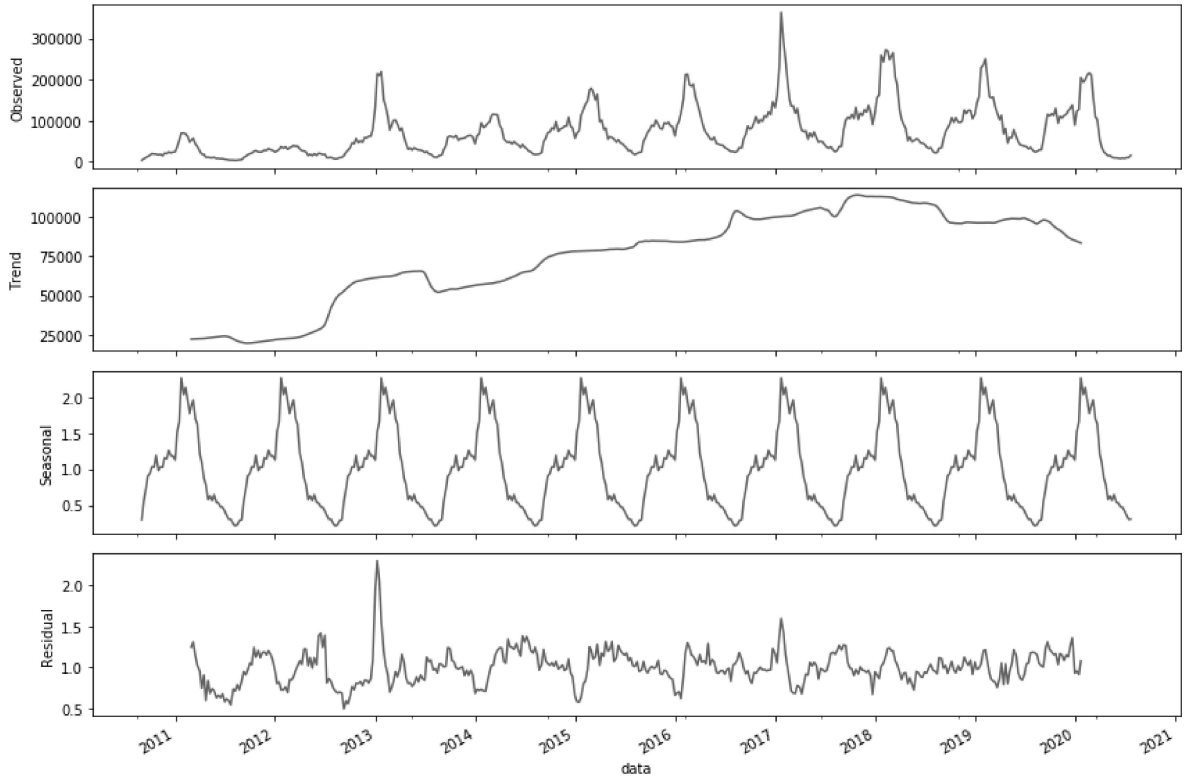
In [31]:

```
result = seasonal_decompose(series3, model='additive', freq=48)
pylab.rcParams['figure.figsize'] = 12,8
#plt.title('Dekompozycja szeregu czasowego')
result.plot()
plt.show()
```



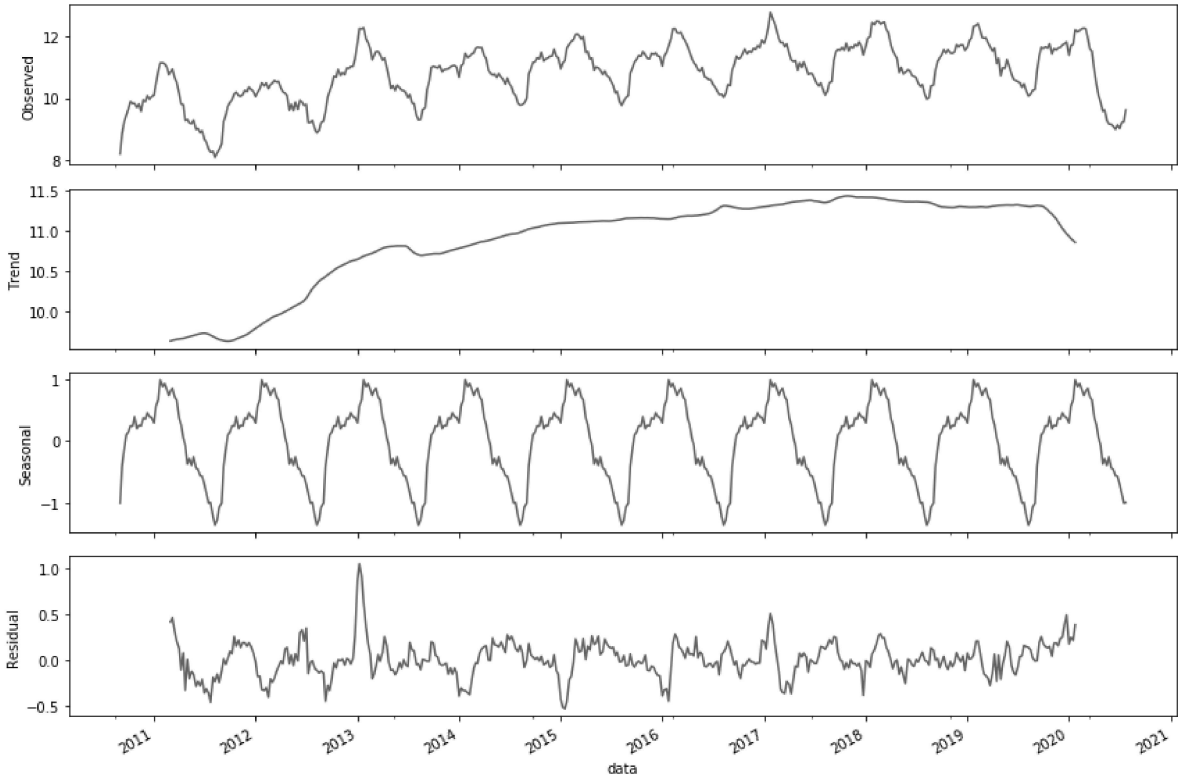
In [32]:

```
result = seasonal_decompose(series3, model='multiplicative', freq=48)
pylab.rcParams['figure.figsize'] = 12,8
#plt.title('Dekompozycja szeregu czasowego')
result.plot()
plt.show()
```



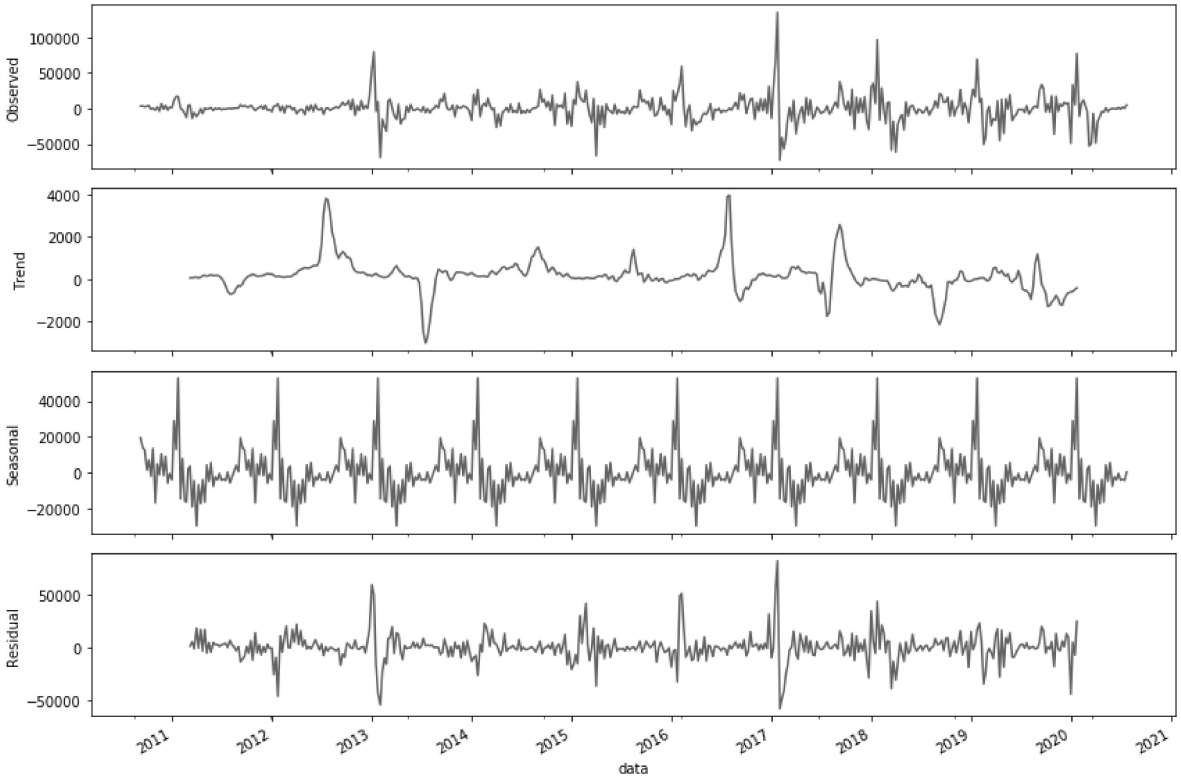
In [33]:

```
#dekompozycja po transformacji log
result = seasonal_decompose(series4, model='additive', freq=48)
pylab.rcParams['figure.figsize'] = 12,8
#plt.title('Dekompozycja szeregu czasowego')
result.plot()
plt.show()
```



In [54]:

```
result = seasonal_decompose(series5['Differenced'], model='additive', freq=48)
pylab.rcParams['figure.figsize'] = 12,8
#plt.title('Dekompozycja szeregu czasowego')
result.plot()
plt.show()
```



In [25]:

```
series4 = pd.read_csv('C:/Users/Laptop/Desktop/Grypa/Dane/seria.csv', parse_dates=True
series4.drop(series4.columns[series4.columns.str.contains('unnamed', case = False)], a
```

In [26]:

```
#transformacja logarytmiczna
series4['Total'] = np.log1p(series4['Total'])
series4
```

Out[26]:

data	Total
2010-09-01	8.190077
2010-09-08	8.838262
2010-09-16	9.249465
2010-09-23	9.443989
2010-10-01	9.664659
2010-10-08	9.897117
2010-10-16	9.832797
2010-10-23	9.839162
2010-11-01	9.703389

In [39]:

```
series5 = series3
series5['t-1'] = ""
```

In [41]:

```
series5['t-1'] = series5['Total'].shift(periods=1)
```

In [47]:

```
#series
series5['Differenced'] = series5['Total'] - series5['t-1']
series5.dropna()
```

Out[47]:

data	Total	t-1	Differenced
2010-09-08	6892	3604.0	3288.0
2010-09-16	10398	6892.0	3506.0
2010-09-23	12631	10398.0	2233.0
2010-10-01	15750	12631.0	3119.0
2010-10-08	19872	15750.0	4122.0
2010-10-16	18634	19872.0	-1238.0
2010-10-23	18753	18634.0	119.0
2010-11-01	16372	18753.0	-2381.0
2010-11-08	18387	16372.0	2015.0

In [52]:

```
series5.dropna(inplace=True)
```

In [53]:

```
series5
```

Out[53]:

data	Total	t-1	Differenced
2010-09-08	6892	3604.0	3288.0
2010-09-16	10398	6892.0	3506.0
2010-09-23	12631	10398.0	2233.0
2010-10-01	15750	12631.0	3119.0
2010-10-08	19872	15750.0	4122.0
2010-10-16	18634	19872.0	-1238.0
2010-10-23	18753	18634.0	119.0
2010-11-01	16372	18753.0	-2381.0
2010-11-08	18387	16372.0	2015.0

In [65]:

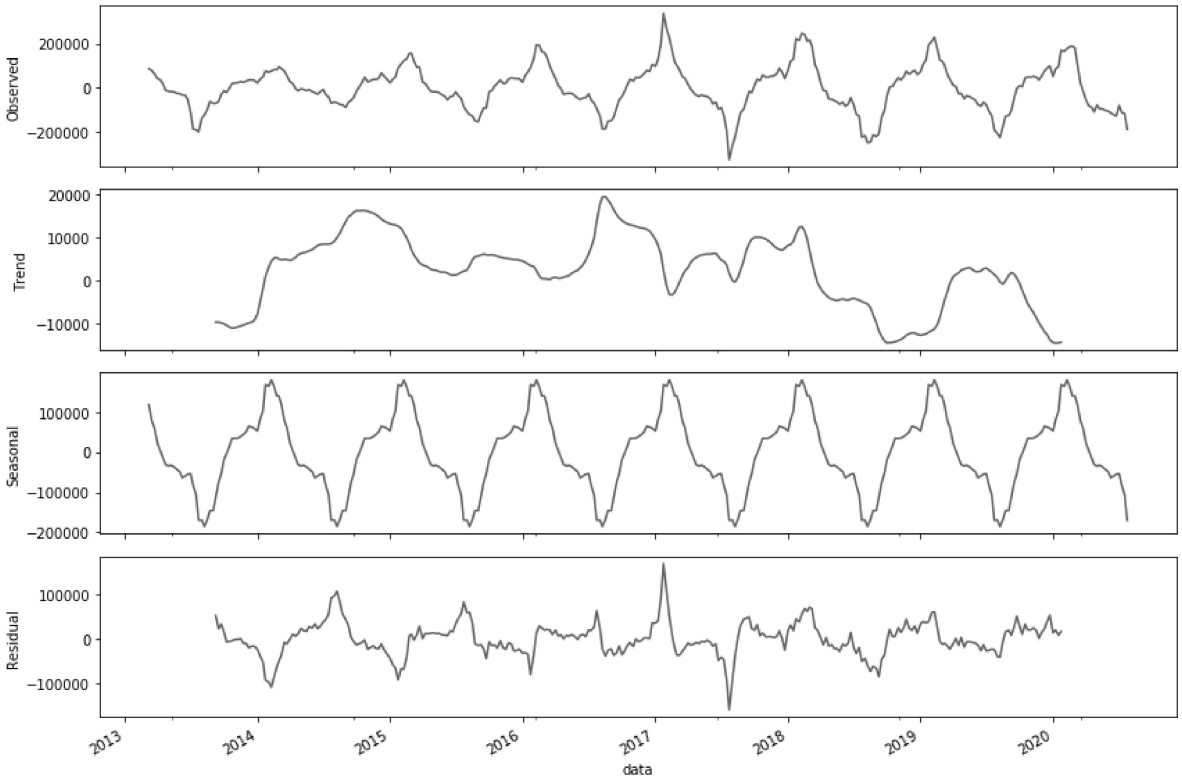
```
series7 = series3
series7['t-24'] = ""
series7['t-24'] = series6['Total'].shift(periods=24)
series7['Differenced'] = series7['Total'] - series7['t-24']
series7.dropna(inplace=True)
series7
```

Out[65]:

data	Total	t-1	Differenced	t-48	t-24
2013-03-08	100518	39499.0	86143.0	39499.0	14375.0
2013-03-16	101205	37546.0	78055.0	37546.0	23150.0
2013-03-23	91358	38258.0	63494.0	38258.0	27864.0
2013-04-01	75947	30644.0	41715.0	30644.0	34232.0
2013-04-08	81863	27130.0	36506.0	27130.0	45357.0
2013-04-16	60122	26019.0	16454.0	26019.0	43668.0
2013-04-23	44866	23871.0	-11567.0	23871.0	56433.0
2013-05-01	30305	14958.0	-16407.0	14958.0	46712.0
2013-05-08	33372	18206.0	-17482.0	18206.0	50854.0

In [66]:

```
result = seasonal_decompose(series7['Differenced'], model='additive', freq=48)
pylab.rcParams['figure.figsize'] = 12,8
#plt.title('Dekompozycja szeregu czasowego')
result.plot()
plt.show()
```




```
1
```