In [109]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels.graphics.tsaplots as tsa_plot
from statsmodels.tsa.seasonal import seasonal_decompose
import statsmodels.formula.api as smf
from statsmodels.tsa.holtwinters import SimpleExpSmoothing
from statsmodels.tsa.holtwinters import Holt
from statsmodels.tsa.holtwinters import ExponentialSmoothing
from statsmodels.tsa.arima_model import ARMA
from sklearn.metrics import mean_absolute_error
from datetime import datetime as dlt

import warnings
warnings.filterwarnings('ignore')
```

In [77]:
```
sales = pd.read_excel('CocaCola_Sales_Rawdata.xlsx')
sales.head(10)
```

Out[77]:

	Quarter	Sales
0	Q1_86	1734.827000
1	Q2_86	2244.960999
2	Q3_86	2533.804993
3	Q4_86	2154.962997
4	Q1_87	1547.818996
5	Q2_87	2104.411995
6	Q3_87	2014.362999
7	Q4_87	1991.746998
8	Q1_88	1869.049999
9	Q2_88	2313.631996

In [78]:
```
sales.shape
```

Out[78]:
```
(42, 2)
```

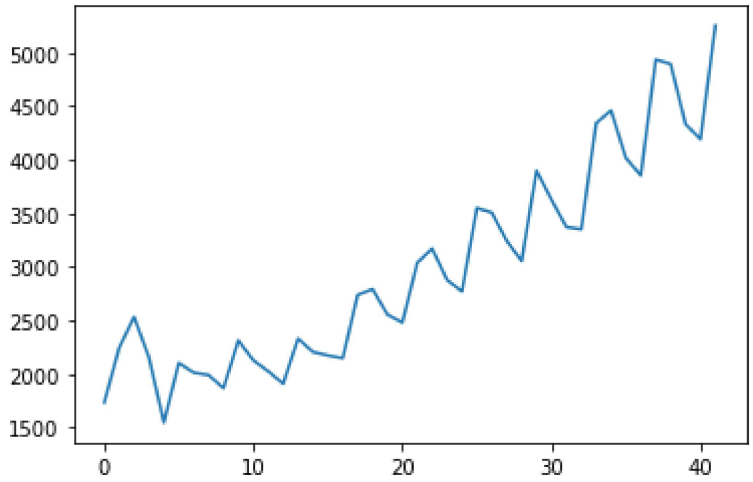
In [79]:
```
sales.dtypes
```

Out[79]:
```
Quarter     object
Sales      float64
dtype: object
```

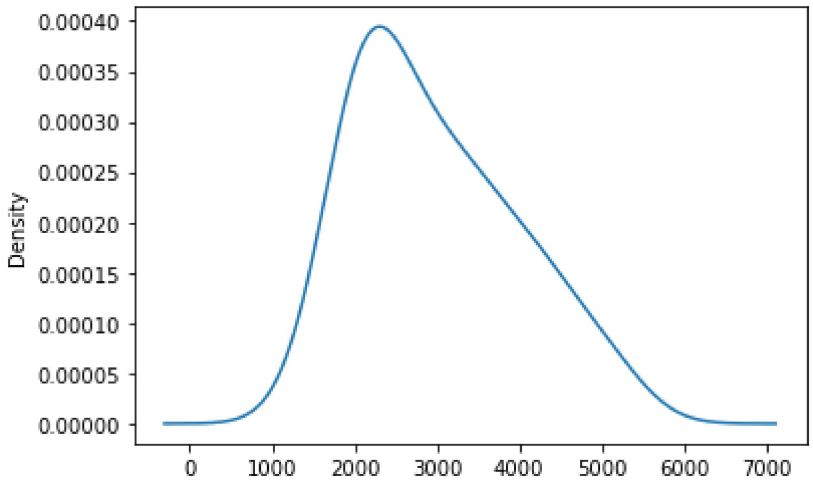
In [80]:
```
sales.isna().sum()
```

Out[80]:
```
Quarter    0
Sales      0
dtype: int64
```

In [81]:
```
sales['Sales'].plot()
plt.show()
```



In [82]:
```
sales['Sales'].plot(kind = 'kde')
plt.show()
```



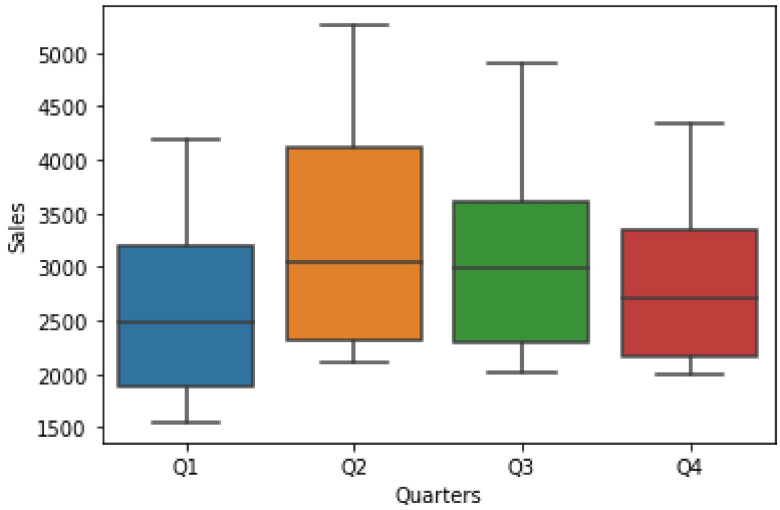
In [83]:
```
sales['Quarters'] = 0
sales['Year'] = 0
for i in range(sales.shape[0]):
    p = sales['Quarter'][i]
    sales['Quarters'][i] = p[0:2]
    sales['Year'][i] = p[3:5]
```

In [84]:
```
sales.head()
```

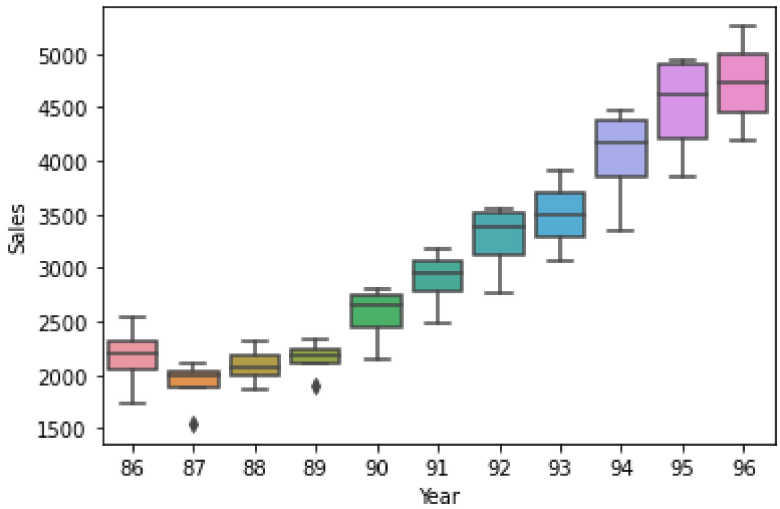
Out[84]:

	Quarter	Sales	Quarters	Year
0	Q1_86	1734.827000	Q1	86
1	Q2_86	2244.960999	Q2	86
2	Q3_86	2533.804993	Q3	86
3	Q4_86	2154.962997	Q4	86
4	Q1_87	1547.818996	Q1	87

In [85]:
```
sns.boxplot(x='Quarters',y='Sales',data = sales)
plt.show()
```

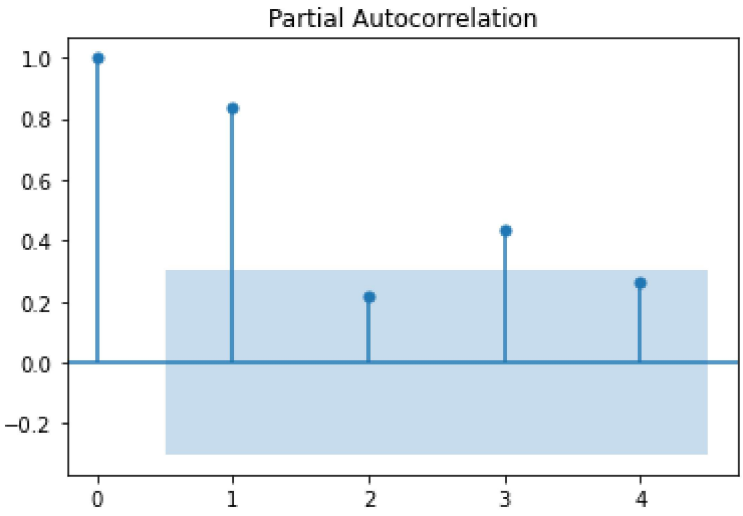


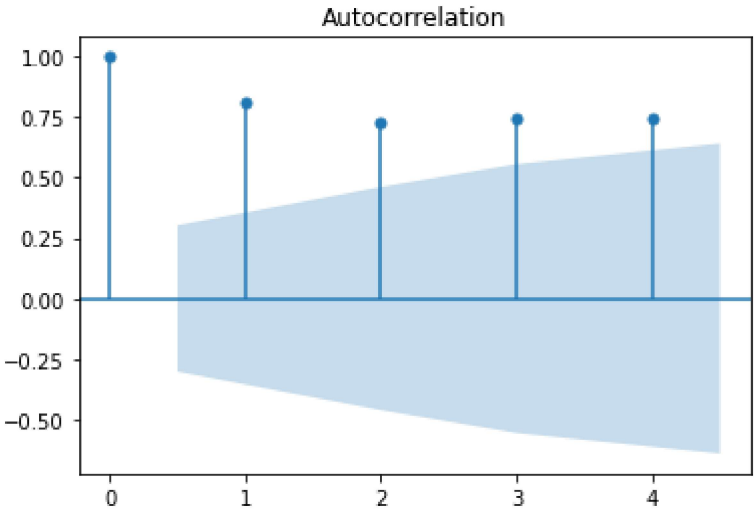
In [86]:
```
sns.boxplot(x='Year',y='Sales',data=sales)
plt.show()
```

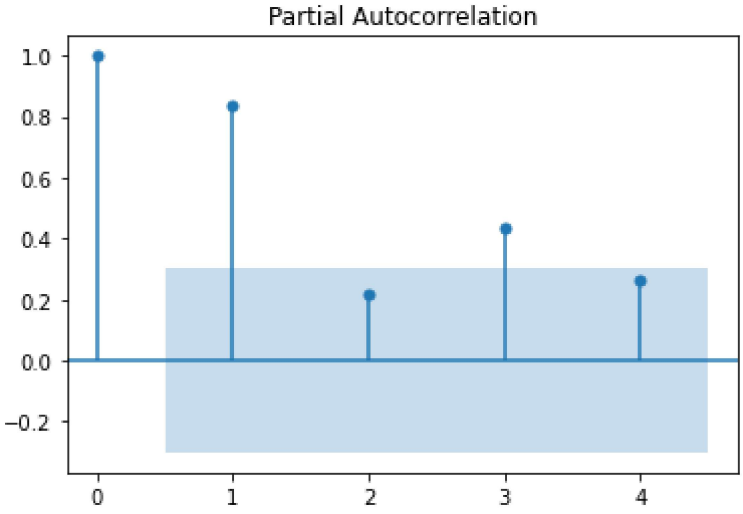


In [87]:
```
tsa_plot.plot_acf(sales.Sales,lags=4)
tsa_plot.plot_pacf(sales.Sales,lags=4)
```

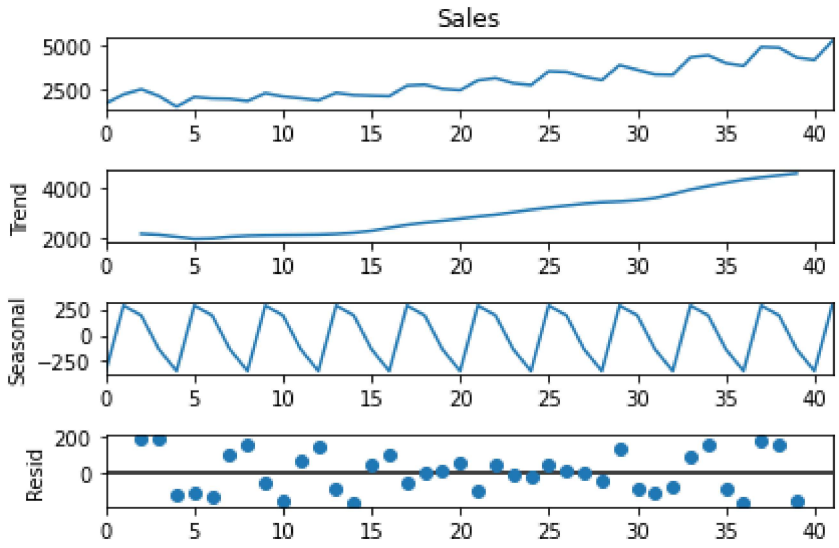
Out[87]:







In [88]:
```
decompose_ts_add = seasonal_decompose(sales['Sales'],period=4)
decompose_ts_add.plot()
plt.show()
```



In [89]:
```
sales['t'] = pd.RangeIndex(start=0,stop=42,step=1)
sales['sq_t'] = np.square(sales['t'])
sales['log_sale'] = np.log(sales['Sales'])
sales = pd.get_dummies(data=sales,columns=['Quarters'])
sales
```

Out[89]:

	Quarter	Sales	Year	t	sq_t	log_sale	Quarters_Q1	Quarters_Q2	Quarters_Q3	Quar
0	Q1_86	1734.827000	86	0	0	7.458663	1	0	0	
1	Q2_86	2244.960999	86	1	1	7.716443	0	1	0	
2	Q3_86	2533.804993	86	2	4	7.837477	0	0	1	
3	Q4_86	2154.962997	86	3	9	7.675529	0	0	0	
4	Q1_87	1547.818996	87	4	16	7.344602	1	0	0	
5	Q2_87	2104.411995	87	5	25	7.651791	0	1	0	
6	Q3_87	2014.362999	87	6	36	7.608058	0	0	1	
7	Q4_87	1991.746998	87	7	49	7.596767	0	0	0	
8	Q1_88	1869.049999	88	8	64	7.533186	1	0	0	
9	Q2_88	2313.631996	88	9	81	7.746574	0	1	0	
10	Q3_88	2128.320000	88	10	100	7.663088	0	0	1	
11	Q4_88	2026.828999	88	11	121	7.614228	0	0	0	
12	Q1_89	1910.603996	89	12	144	7.555175	1	0	0	
13	Q2_89	2331.164993	89	13	169	7.754123	0	1	0	
14	Q3_89	2206.549995	89	14	196	7.699185	0	0	1	
15	Q4_89	2173.967995	89	15	225	7.684309	0	0	0	
16	Q1_90	2148.278000	90	16	256	7.672422	1	0	0	
17	Q2_90	2739.307999	90	17	289	7.915461	0	1	0	
18	Q3_90	2792.753998	90	18	324	7.934783	0	0	1	
19	Q4_90	2556.009995	90	19	361	7.846203	0	0	0	
20	Q1_91	2480.973999	91	20	400	7.816407	1	0	0	
21	Q2_91	3039.522995	91	21	441	8.019456	0	1	0	
22	Q3_91	3172.115997	91	22	484	8.062154	0	0	1	
23	Q4_91	2879.000999	91	23	529	7.965199	0	0	0	
24	Q1_92	2772.000000	92	24	576	7.927324	1	0	0	
25	Q2_92	3550.000000	92	25	625	8.174703	0	1	0	
26	Q3_92	3508.000000	92	26	676	8.162801	0	0	1	
27	Q4_92	3243.859993	92	27	729	8.084519	0	0	0	
28	Q1_93	3056.000000	93	28	784	8.024862	1	0	0	
29	Q2_93	3899.000000	93	29	841	8.268475	0	1	0	
30	Q3_93	3629.000000	93	30	900	8.196712	0	0	1	

	Quarter	Sales	Year	t	sq_t	log_sale	Quarters_Q1	Quarters_Q2	Quarters_Q3	Quar
31	Q4_93	3373.000000	93	31	961	8.123558	0	0	0	
32	Q1_94	3352.000000	94	32	1024	8.117312	1	0	0	
33	Q2_94	4342.000000	94	33	1089	8.376090	0	1	0	
34	Q3_94	4461.000000	94	34	1156	8.403128	0	0	1	
35	Q4_94	4017.000000	94	35	1225	8.298291	0	0	0	
36	Q1_95	3854.000000	95	36	1296	8.256867	1	0	0	
37	Q2_95	4936.000000	95	37	1369	8.504311	0	1	0	
38	Q3_95	4895.000000	95	38	1444	8.495970	0	0	1	
39	Q4_95	4333.000000	95	39	1521	8.374015	0	0	0	
40	Q1_96	4194.000000	96	40	1600	8.341410	1	0	0	
41	Q2_96	5253.000000	96	41	1681	8.566555	0	1	0	

In [90]:
```
train = sales.head(30)
test = sales.head(12)
```

In [91]:
```
linear_model = smf.ols('Sales~t',data=train).fit()
linear_pred=linear_model.predict(train['t'])
linear_rms=mean_absolute_error(train['Sales'],linear_pred)
linear_rms
```

Out[91]: 257.4609816851609

In [92]:
```
Exp = smf.ols('log_sale~t',data=train).fit()
pred_Exp = pd.Series(Exp.predict(pd.DataFrame(train['t'])))
rmse_Exp = mean_absolute_error(train['Sales'],pred_Exp)
rmse_Exp
```

Out[92]: 2496.3274984983696

In [93]:
```
quad = smf.ols('Sales~t+sq_t',data=train).fit()
pred_quad = pd.Series(quad.predict(train[["t","sq_t"]]))
rmse_quad = mean_absolute_error(train['Sales'],pred_quad)
rmse_quad
```

Out[93]: 213.16749202305735

In [94]:
```
~Quarters_Q1+Quarters_Q2+Quarters_Q3+Quarters_Q4',data=train).fit()
(add_sea.predict(train[['Quarters_Q1','Quarters_Q2','Quarters_Q3','Quarters_Q4']]
olute_error(train['Sales'],pred_add_sea)
```

Out[94]: 458.8212516375951

In [96]:
```
add_sea_quad = smf.ols('Sales~t+sq_t+Quarters_Q1+Quarters_Q2+Quarters_Q3+Quarters
pred_add_sea_quad = pd.Series(add_sea_quad.predict(train[['Quarters_Q1','Quarters
rmse_add_sea_quad = mean_absolute_error(train['Sales'],pred_add_sea_quad)
rmse_add_sea_quad
```

Out[96]: 99.69068092542595

In [97]:
```
Mul_sea = smf.ols('log_sale~Quarters_Q1+Quarters_Q2+Quarters_Q3+Quarters_Q4',data
pred_Mult_sea = pd.Series(Mul_sea.predict(train[['Quarters_Q1','Quarters_Q2','Qua
rmse_Mult_sea = mean_absolute_error(train['Sales'],pred_Mult_sea)
rmse_Mult_sea
```

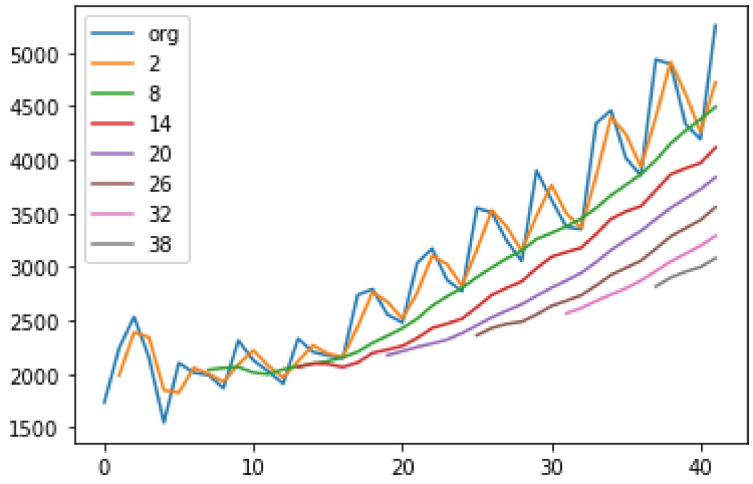
Out[97]: 2496.3274984983705

In [98]:
```
Mul_Add_sea = smf.ols('log_sale~t+Quarters_Q1+Quarters_Q2+Quarters_Q3+Quarters_Q4
pred_Mult_add_sea = pd.Series(Mul_Add_sea.predict(train[['t','Quarters_Q1','Quart
rmse_Mult_add_sea = mean_absolute_error(train['Sales'],pred_Mult_add_sea)
rmse_Mult_add_sea
```

Out[98]: 2496.32749849837

In [100]:
```
sales['Sales'].plot(label="org")
for i in range(2,43,6):
    sales['Sales'].rolling(i).mean().plot(label=str(i))
plt.legend(loc='best')
```

Out[100]: <matplotlib.legend.Legend at 0x15b328f0490>



In [103]:
```
ses_model = SimpleExpSmoothing(train["Sales"]).fit(smoothing_level=0.2)
pred_ses = ses_model.predict(start = train.index[0],end = train.index[-1])
ses_rms=mean_absolute_error(pred_ses,train['Sales'])
ses_rms
```

Out[103]: 284.2907677932735

In [104]:
```
hw_model = Holt(train["Sales"]).fit(smoothing_level=0.8, smoothing_slope=0.2) # t
pred_hw = hw_model.predict(start = train.index[0],end = train.index[-1])
hw_rms=mean_absolute_error(pred_hw,train['Sales'])
hw_rms
```

Out[104]: 299.17346422213893

In [106]:
```
hwe_model_mul_add = ExponentialSmoothing(train["Sales"],seasonal="mul",trend="add
pred_hwe_mul_add = hwe_model_mul_add.predict(start = train.index[0],end = train.i
hw_ma_rms=mean_absolute_error(pred_hwe_mul_add,train['Sales'])
hw_ma_rms
```

Out[106]: 135.94949307336117

In [107]:
```
hwe_model_mul_add = ExponentialSmoothing(train["Sales"],seasonal="add",trend="mul
pred_hwe_mul_add = hwe_model_mul_add.predict(start = train.index[0],end = train.i
hw_ma_rms=mean_absolute_error(pred_hwe_mul_add,train['Sales'])
hw_ma_rms
```

Out[107]: 133.8422739605856

In [108]:
```
hwe_model_add_add = ExponentialSmoothing(train["Sales"],seasonal="add",trend="add
pred_hwe_add_add = hwe_model_add_add.predict(start = train.index[0],end = train.i
hw_aa_rms=mean_absolute_error(pred_hwe_add_add,train['Sales'])
hw_aa_rms
```

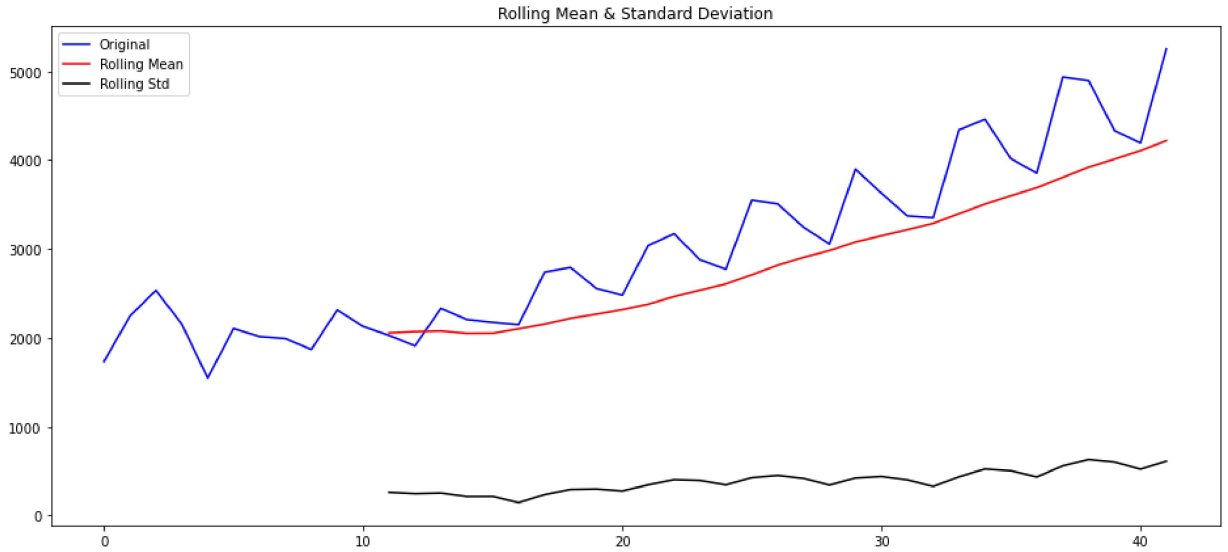
Out[108]: 129.7301462404645

In [110]:
```
ARMAmodel = ARMA(train['Sales'], order=(3, 0)) #model with AR=1 and MA=0
ARMAmodel_fit = ARMAmodel.fit()
ARMA_pred = ARMAmodel_fit.predict()
ARMA_pred
arma_rms=mean_absolute_error(ARMA_pred,train['Sales'])
arma_rms
```

Out[110]: 277.18726942124533

In [111]:
```
rolLmean = sales['Sales'].rolling(12).mean() # 12 entries
rolLstd = sales['Sales'].rolling(12).std()
plt.figure(figsize=(16,7))
fig = plt.figure(1)

orig = plt.plot(sales['Sales'], color='blue',label='Original')
mean = plt.plot(rolLmean, color='red', label='Rolling Mean')
std = plt.plot(rolLstd, color='black', label = 'Rolling Std')
plt.legend(loc='best')
plt.title('Rolling Mean & Standard Deviation')
plt.show(block=False)
```



In [112]:
```
add_sea_quad = smf.ols('Sales~t+sq_t+Quarters_Q1+Quarters_Q2+Quarters_Q3+Quarters
pred_add_sea_quad = add_sea_quad.predict(test[['Quarters_Q1','Quarters_Q2','Quart
rmse_add_sea_quad = mean_absolute_error(test['Sales'],pred_add_sea_quad)
rmse_add_sea_quad
```

Out[112]: 119.56046711071104