In [7]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels.api as sm
from statsmodels.tsa.seasonal import seasonal_decompose
from statsmodels.tsa.holtwinters import SimpleExpSmoothing # SES
from statsmodels.tsa.holtwinters import Holt # Holts Exponential Smoothing
from statsmodels.tsa.holtwinters import ExponentialSmoothing #
import statsmodels.graphics.tsaplots as tsa_plots
import statsmodels.tsa.statespace as tm_models
from datetime import datetime,time
#from sm.tsa.statespace import sa
```

In [9]:
```
#Import the dataset
Cocacola=pd.read_excel("C:\\Users\\nishi\\Desktop\\Assignments\\Forecasting\\Coca
```

In [10]:
```
Cocacola
```

Out[10]:

	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
10	Q3_88	2128.320000
11	Q4_88	2026.828999
12	Q1_89	1910.603996
13	Q2_89	2331.164993
14	Q3_89	2206.549995
15	Q4_89	2173.967995
16	Q1_90	2148.278000
17	Q2_90	2739.307999
18	Q3_90	2792.753998
19	Q4_90	2556.009995
20	Q1_91	2480.973999
21	Q2_91	3039.522995
22	Q3_91	3172.115997
23	Q4_91	2879.000999
24	Q1_92	2772.000000
25	Q2_92	3550.000000
26	Q3_92	3508.000000
27	Q4_92	3243.859993
28	Q1_93	3056.000000
29	Q2_93	3899.000000
30	Q3_93	3629.000000
31	Q4_93	3373.000000
32	Q1_94	3352.000000
33	Q2_94	4342.000000

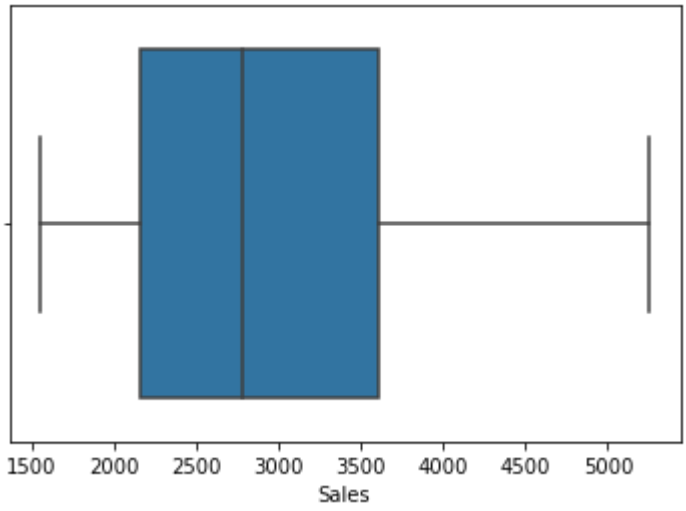
	Quarter	Sales
34	Q3_94	4461.000000
35	Q4_94	4017.000000
36	Q1_95	3854.000000
37	Q2_95	4936.000000
38	Q3_95	4895.000000
39	Q4_95	4333.000000
40	Q1_96	4194.000000
41	Q2_96	5253.000000

In [11]:
```
from statsmodels.tsa.seasonal import seasonal_decompose
```

In [12]:
```
# Boxplot for ever
sns.boxplot("Sales",data=Cocacola)
```

```
C:\Users\nishi\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWar
ning: Pass the following variable as a keyword arg: x. From version 0.12, the o
nly valid positional argument will be `data`, and passing other arguments witho
ut an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

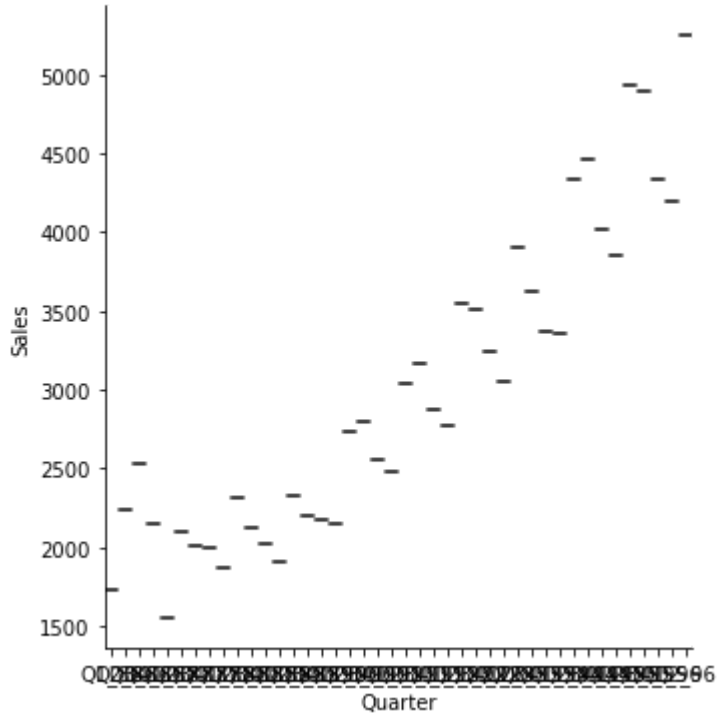
Out[12]: <AxesSubplot:xlabel='Sales'>



In [13]:
```
sns.factorplot("Quarter","Sales",data=Cocacola,kind="box")
```

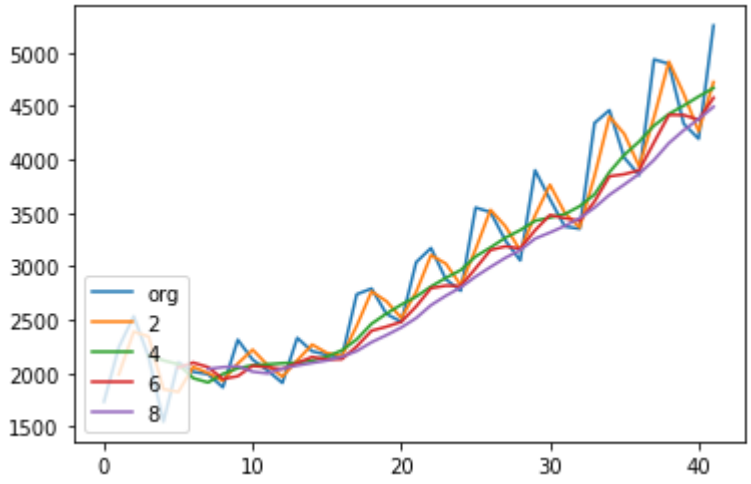
```
C:\Users\nishi\anaconda3\lib\site-packages\seaborn\categorical.py:3714: UserWar
ning: The `factorplot` function has been renamed to `catplot`. The original nam
e will be removed in a future release. Please update your code. Note that the d
efault `kind` in `factorplot` (``'point'``) has changed ``'strip'`` in `catplot`.
  warnings.warn(msg)
C:\Users\nishi\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWar
ning: Pass the following variables as keyword args: x, y. From version 0.12, th
e only valid positional argument will be `data`, and passing other arguments wi
thout an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[13]: <seaborn.axisgrid.FacetGrid at 0x1e72744a850>



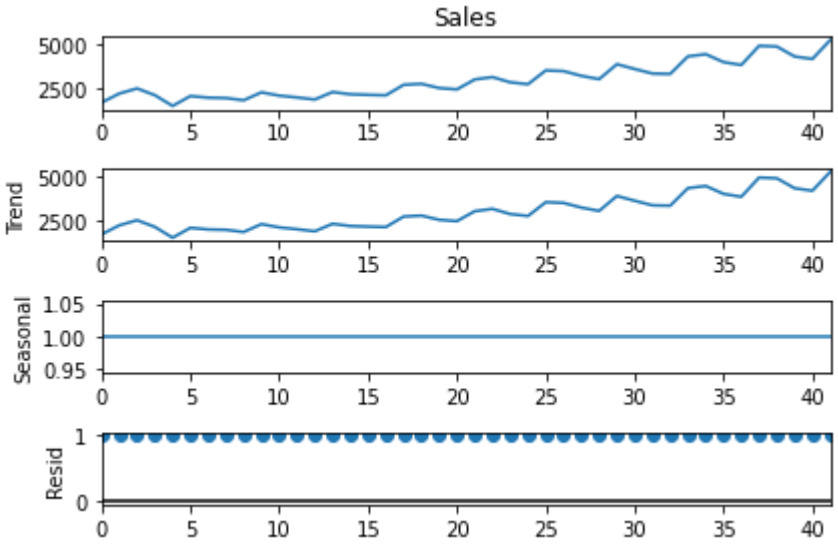
In [15]:
```
# moving average for the time series to understand better about the trend charact
Cocacola.Sales.plot(label="org")
for i in range(2,10,2):
    Cocacola["Sales"].rolling(i).mean().plot(label=str(i))
plt.legend(loc=3)
```

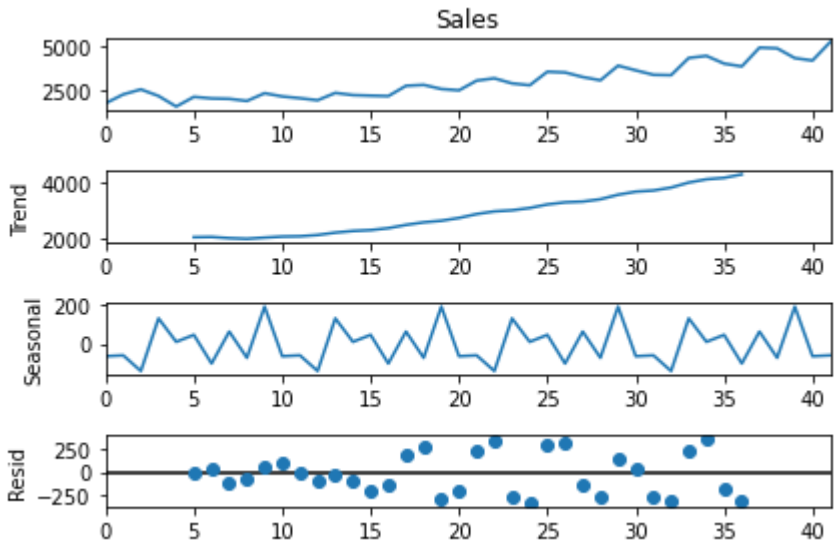
Out[15]: <matplotlib.legend.Legend at 0x1e727d488b0>

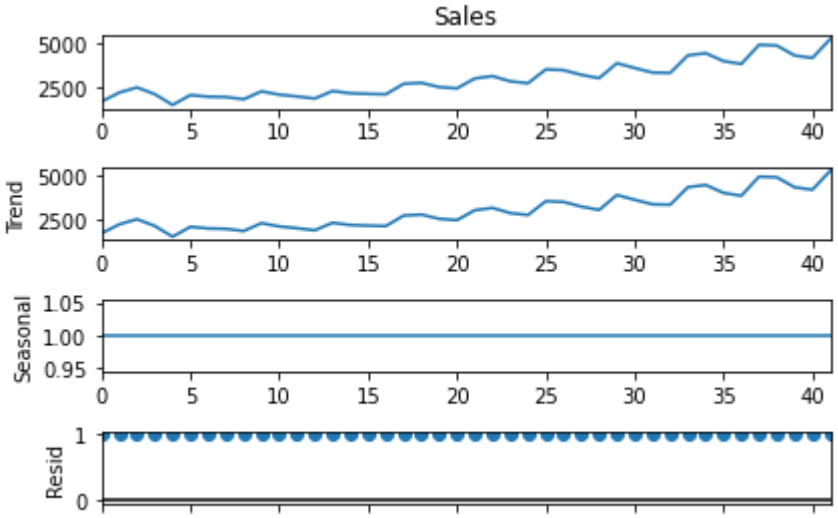


In [16]:
```
# Time series decomposition plot
decompose_ts_add = seasonal_decompose(Cocacola.Sales,model="additive",period=10)
decompose_ts_add.plot()
decompose_ts_mul = seasonal_decompose(Cocacola.Sales,model="multiplicative",perio
decompose_ts_mul.plot()
```

Out[16]:

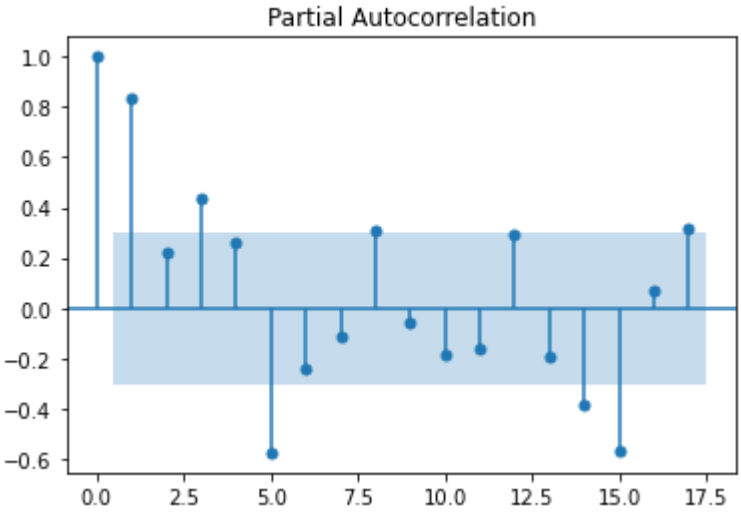


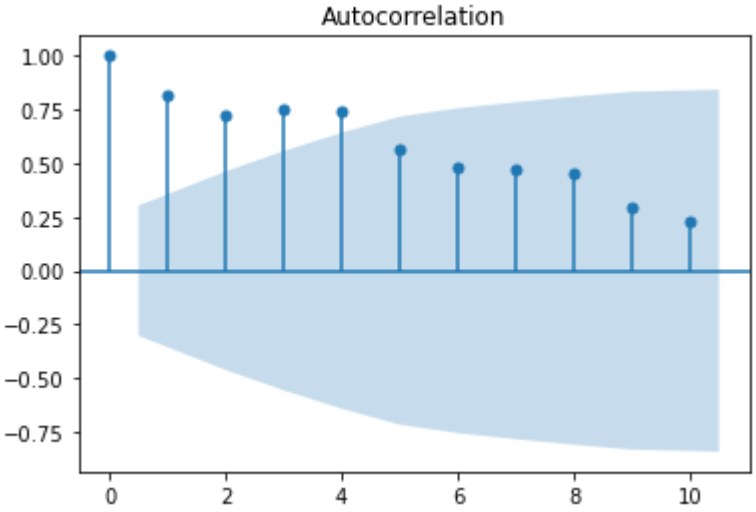


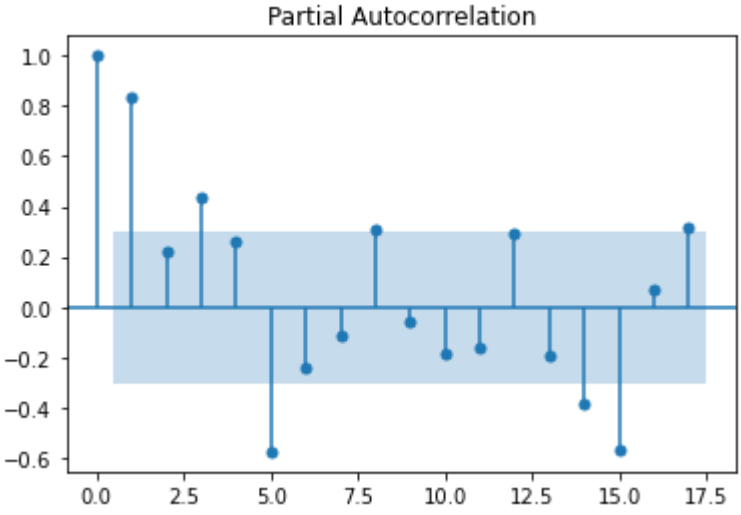

Sales
5000
2500
0
5
10
15
20
25
30
35
40
Trend
5000
2500
Seasonal
1.05
1.00
0.95
Resid
1
0

In [17]:
```
# ACF plots and PACF plots on Original data sets
tsa_plots.plot_acf(Cocacola.Sales,lags=10)
tsa_plots.plot_pacf(Cocacola.Sales)
```

Out[17]:







In [20]:
```
# Amtrak.index.freq = "MS"
# splitting the data into Train and Test data and considering the last 12 months
# Test data and left over data as train data

Train = Cocacola.head(48)
Test =Cocacola.tail(12)
```

In [21]:
```
# to change the index value in pandas data frame
# Test.set_index(np.arange(1,13),inplace=True)

# Creating a function to calculate the MAPE value for test data
def MAPE(pred,org):
    temp = np.abs((pred-org))*100/org
    return np.mean(temp)
```

In [22]:
```
# Simple Exponential Method
ses_model = SimpleExpSmoothing(Train["Sales"]).fit()
pred_ses = ses_model.predict(start = Test.index[0],end = Test.index[-1])
MAPE(pred_ses,Test.Sales) # 9.76
```

```
C:\Users\nishi\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.p
y:427: FutureWarning: After 0.13 initialization must be handled at model creati
on
  warnings.warn(
```

Out[22]:
```
9.68200492651463
```

In [23]:
```
# Holt method
hw_model = Holt(Train["Sales"]).fit()
pred_hw = hw_model.predict(start = Test.index[0],end = Test.index[-1])
MAPE(pred_hw,Test.Sales) # 9.82
```

Out[23]:
```
11.025182440957998
```

In [24]:
```
# Holts winter exponential smoothing with additive seasonality and additive trend
hwe_model_add_add = ExponentialSmoothing(Train["Sales"],seasonal="add",trend="add
pred_hwe_add_add = hwe_model_add_add.predict(start = Test.index[0],end = Test.ind
MAPE(pred_hwe_add_add,Test.Sales)# 3.10
```

```
<ipython-input-24-383dbf709136>:2: FutureWarning: the 'damped'' keyword is depr
ecated, use 'damped_trend' instead
  hwe_model_add_add = ExponentialSmoothing(Train["Sales"],seasonal="add",trend
="add",seasonal_periods=4,damped=True).fit()
```

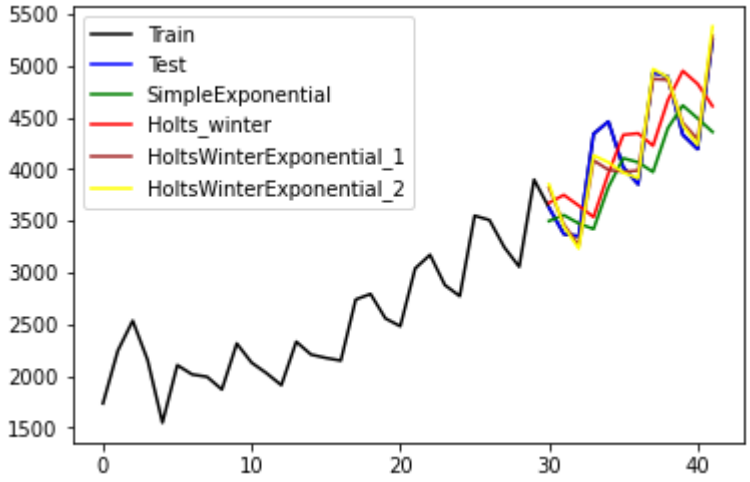
Out[24]:
```
3.245837019668915
```

In [25]:
```
# Holts winter exponential smoothing with multiplicative seasonality and additive
hwe_model_mul_add = ExponentialSmoothing(Train["Sales"],seasonal="mul",trend="add
pred_hwe_mul_add = hwe_model_mul_add.predict(start = Test.index[0],end = Test.ind
MAPE(pred_hwe_mul_add,Test.Sales) # 2.35
```

Out[25]:
```
2.8845556504947196
```

In [26]:
```
# Visualization of Forecasted values for Test data set using different methods
plt.plot(Train.index, Train["Sales"], label='Train',color="black")
plt.plot(Test.index, Test["Sales"], label='Test',color="blue")
plt.plot(pred_ses.index, pred_ses, label='SimpleExponential',color="green")
plt.plot(pred_hw.index, pred_hw, label='Holts_winter',color="red")
plt.plot(pred_hwe_add_add.index,pred_hwe_add_add,label="HoltsWinterExponential_1"
plt.plot(pred_hwe_mul_add.index,pred_hwe_mul_add,label="HoltsWinterExponential_2"
plt.legend(loc='best')
```

Out[26]:
```
<matplotlib.legend.Legend at 0x1e729866f40>
```



In [33]:
```
# AIRLINES DATA
Air = pd.read_excel("C:\\Users\\nishi\\Desktop\\Assignments\\Forecasting\\Airline

from statsmodels.tsa.seasonal import seasonal_decompose
#decomposition = seasonal_decompose(indexedDataset_logScale)
Air
```

Out[33]:

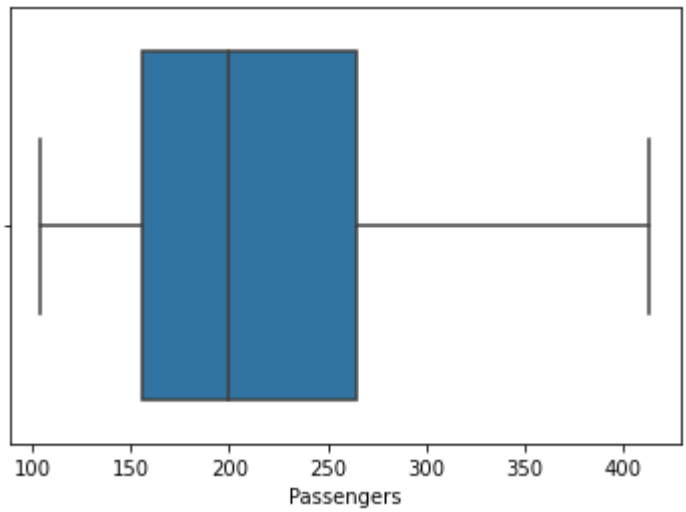
	Month	Passengers
0	1995-01-01	112
1	1995-02-01	118
2	1995-03-01	132
3	1995-04-01	129
4	1995-05-01	121
...	...	...
91	2002-08-01	405
92	2002-09-01	355
93	2002-10-01	306
94	2002-11-01	271
95	2002-12-01	306

96 rows × 2 columns

In [34]:
```
# Boxplot for the dataset
sns.boxplot("Passengers",data=Air)
```

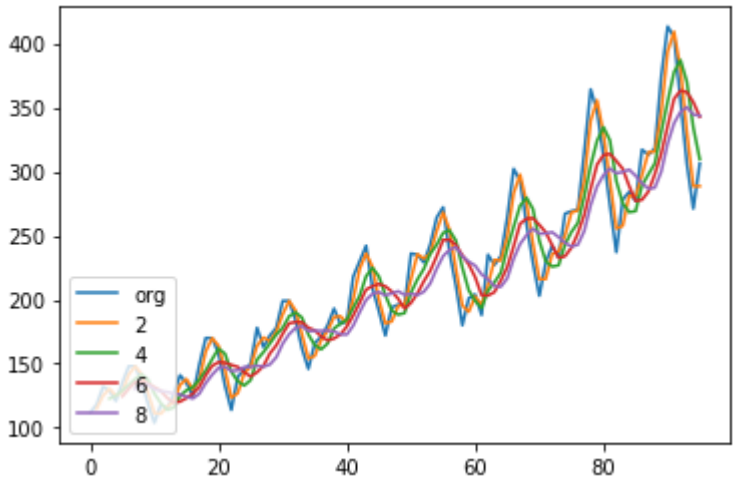
```
C:\Users\nishi\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWar
ning: Pass the following variable as a keyword arg: x. From version 0.12, the o
nly valid positional argument will be `data`, and passing other arguments witho
ut an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[34]: <AxesSubplot:xlabel='Passengers'>



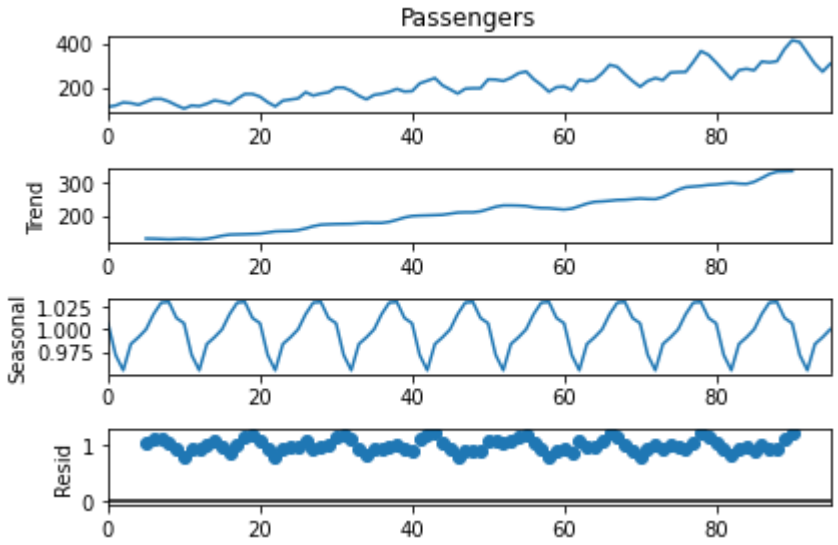
In [36]:
```
#sns.factorplot("Quarter","Sales",data=cocola,kind="box")

# moving average for the time series to understand better about the trend charact
Air.Passengers.plot(label="org")
for i in range(2,10,2):
    Air["Passengers"].rolling(i).mean().plot(label=str(i))
plt.legend(loc=3)
```

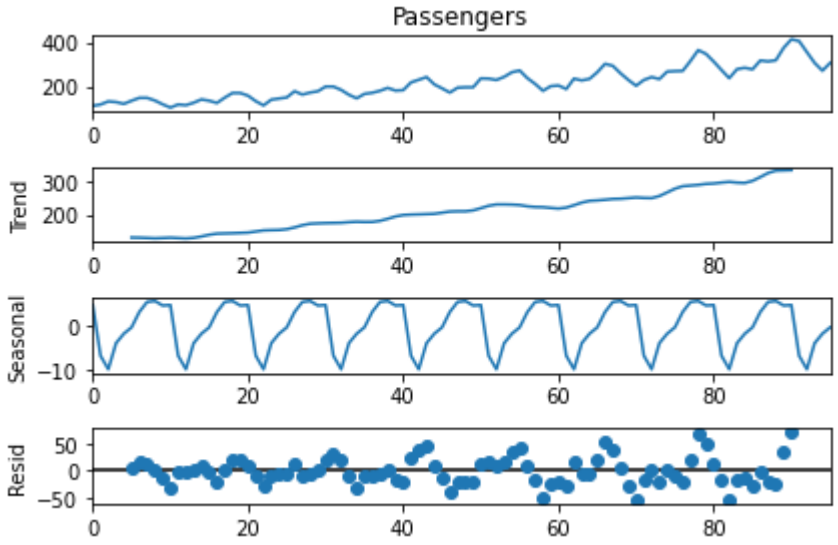
Out[36]:
```
<matplotlib.legend.Legend at 0x1e727e1da00>
```

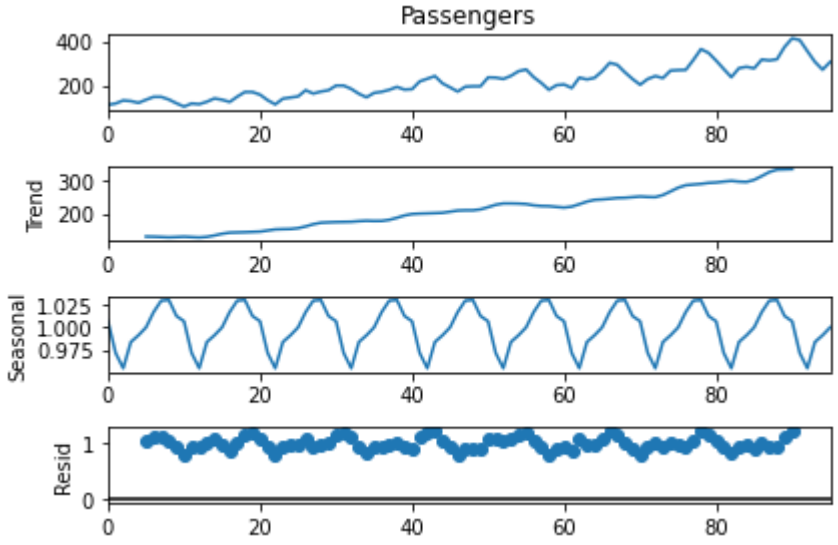


In [38]:
```
decompose_ts_add = seasonal_decompose(Air.Passengers,model="additive",period=10)
decompose_ts_add.plot()
decompose_ts_mul = seasonal_decompose(Air.Passengers,model="multiplicative",perio
decompose_ts_mul.plot()
```

Out[38]:

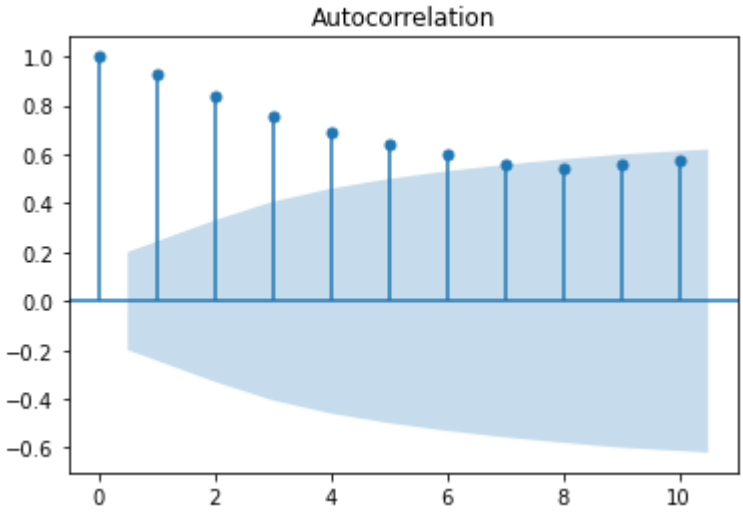


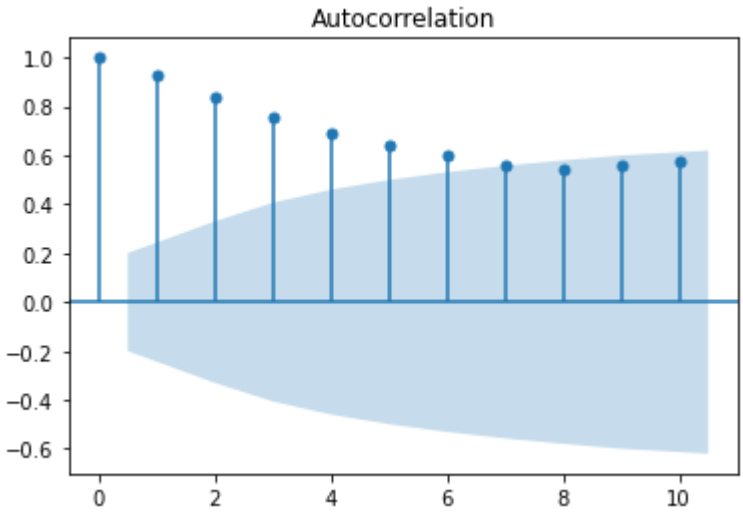




In [39]:
```
tsa_plots.plot_acf(Air.Passengers,lags=10)
```

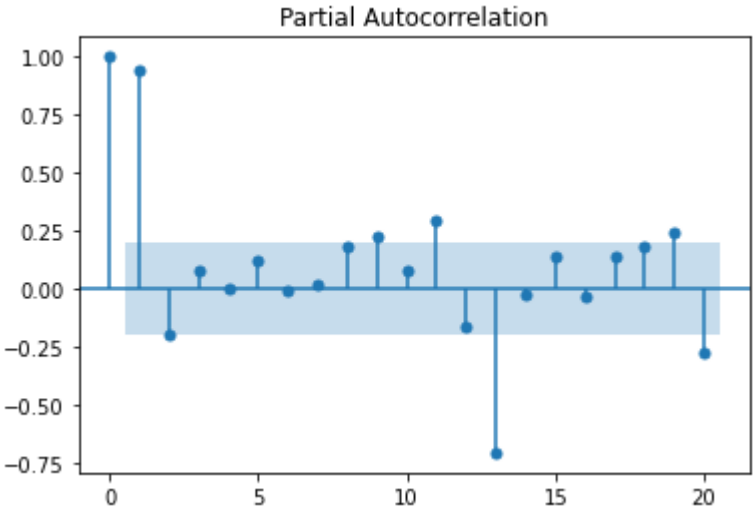
Out[39]:

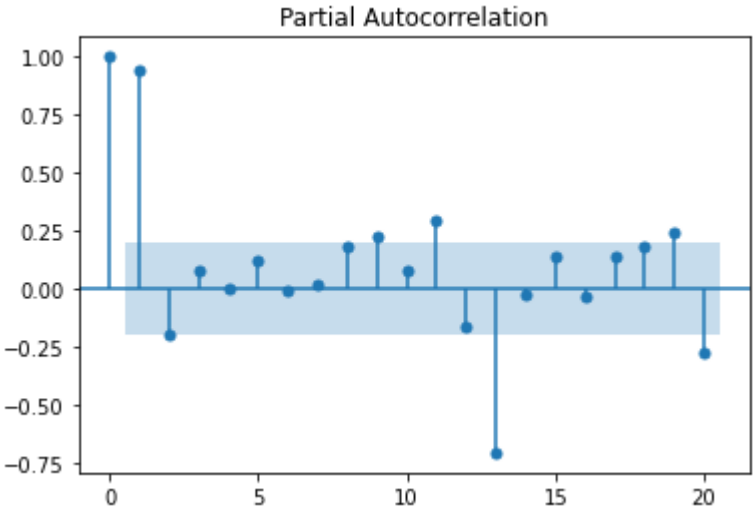




In [41]:
```
tsa_plots.plot_pacf(Air.Passengers)
```

Out[41]:





In [42]:
```
Train = Air.head(48)
Test =Air.tail(12)
```

In [43]:
```
def MAPE(pred,org):
    temp = np.abs((pred-org))*100/org
    return np.mean(temp)


# Simple Exponential Method
ses_model = SimpleExpSmoothing(Train["Passengers"]).fit()
pred_ses = ses_model.predict(start = Test.index[0],end = Test.index[-1])
MAPE(pred_ses,Test.Passengers)
```

```
C:\Users\nishi\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.p
y:427: FutureWarning: After 0.13 initialization must be handled at model creati
on
  warnings.warn(
```

Out[43]:
```
39.807494673483454
```

In [44]:
```
hw_model = Holt(Train["Passengers"]).fit()
pred_hw = hw_model.predict(start = Test.index[0],end = Test.index[-1])
MAPE(pred_hw,Test.Passengers)
```

Out[44]:
```
17.147769574198275
```

In [45]:
```
hwe_model_add_add = ExponentialSmoothing(Train["Passengers"],seasonal="add",trend
pred_hwe_add_add = hwe_model_add_add.predict(start = Test.index[0],end = Test.ind
MAPE(pred_hwe_add_add,Test.Passengers)
```

```
<ipython-input-45-caa01d2aede1>:1: FutureWarning: the 'damped'' keyword is depr
ecated, use 'damped_trend' instead
  hwe_model_add_add = ExponentialSmoothing(Train["Passengers"],seasonal="add",t
rend="add",seasonal_periods=4,damped=True).fit()
```

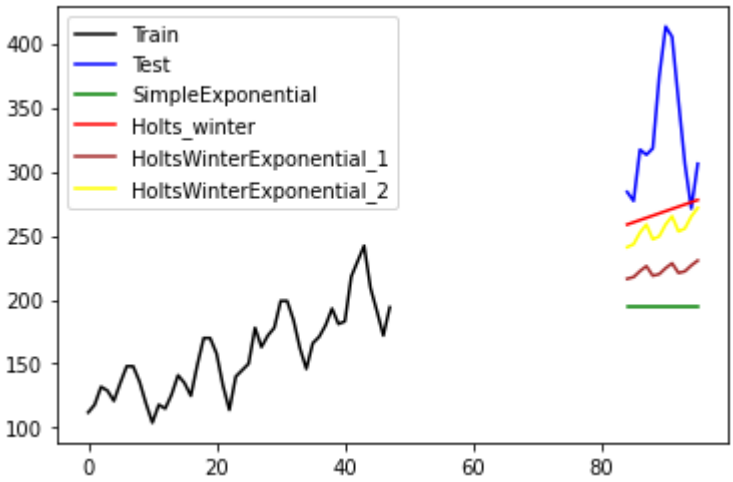
Out[45]:
```
30.910936406693008
```

In [46]:
```
# Holts winter exponential smoothing with multiplicative seasonality and additive
hwe_model_mul_add = ExponentialSmoothing(Train["Passengers"],seasonal="mul",trend
pred_hwe_mul_add = hwe_model_mul_add.predict(start = Test.index[0],end = Test.ind
MAPE(pred_hwe_mul_add,Test.Passengers)
```

Out[46]:
```
20.952935957124662
```

In [47]:
```
# Visualization of Forecasted values for Test data set using different methods
plt.plot(Train.index, Train["Passengers"], label='Train',color="black")
plt.plot(Test.index, Test["Passengers"], label='Test',color="blue")
plt.plot(pred_ses.index, pred_ses, label='SimpleExponential',color="green")
plt.plot(pred_hw.index, pred_hw, label='Holts_winter',color="red")
plt.plot(pred_hwe_add_add.index,pred_hwe_add_add,label="HoltsWinterExponential_1")
plt.plot(pred_hwe_mul_add.index,pred_hwe_mul_add,label="HoltsWinterExponential_2")
plt.legend(loc='best')
```

Out[47]: <matplotlib.legend.Legend at 0x1e729e47f70>



In [ ]: