In this assignment students have to make ARIMA model over shampoo sales data and check the MSE between predicted and actual value.

Student can download data in .csv format from the following link:

https://datamarket.com/data/set/22r0/sales-of-shampoo-over-a-three-year-period#!ds=22r0&display=line (https://datamarket.com/data/set/22r0/sales-of-shampoo-over-a-three-year-period#!ds=22r0&display=line)

In [1]:
```
import pandas as pd
from pandas import datetime
import matplotlib.pyplot as plt
%matplotlib inline
```

In [2]:
```
def parser(x):
    return datetime.strptime('190'+x,'%Y-%m')

sales = pd.read_csv("shampoo-sales.csv",index_col =0, parse_dates=[0],date_parser = parser)
#sales = pd.read_csv("shampoo-sales.csv",index_col =0)
sales
```

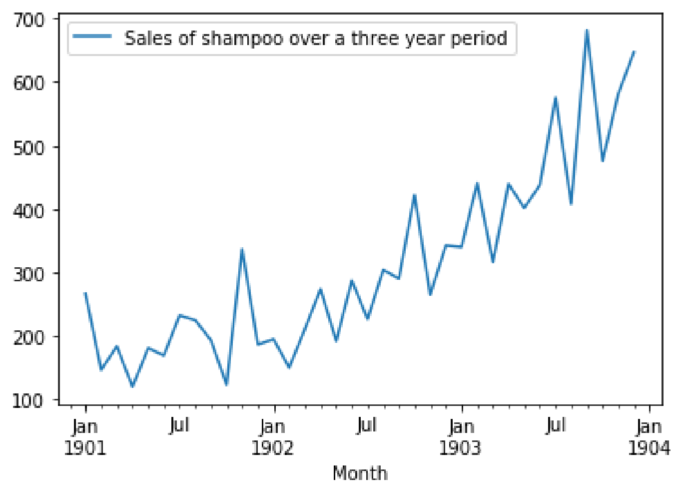
Out[2]:

Month	Sales of shampoo over a three year period
1901-01-01	266.0
1901-02-01	145.9
1901-03-01	183.1
1901-04-01	119.3
1901-05-01	180.3
1901-06-01	168.5
1901-07-01	231.8
1901-08-01	224.5
1901-09-01	192.8
1901-10-01	122.9
1901-11-01	336.5
1901-12-01	185.9
1902-01-01	194.3
1902-02-01	149.5
1902-03-01	210.1
1902-04-01	273.3
1902-05-01	191.4
1902-06-01	287.0
1902-07-01	226.0
1902-08-01	303.6
1902-09-01	289.9
1902-10-01	421.6
1902-11-01	264.5
1902-12-01	342.3
1903-01-01	339.7
1903-02-01	440.4
1903-03-01	315.9
1903-04-01	439.3
1903-05-01	401.3
1903-06-01	437.4
1903-07-01	575.5
1903-08-01	407.6
1903-09-01	682.0

	Sales of shampoo over a three year period
Month	
1903-10-01	475.3
1903-11-01	581.3
1903-12-01	646.9

In [3]:
```
sales.plot()
```

Out[3]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1febf1945f8>
```

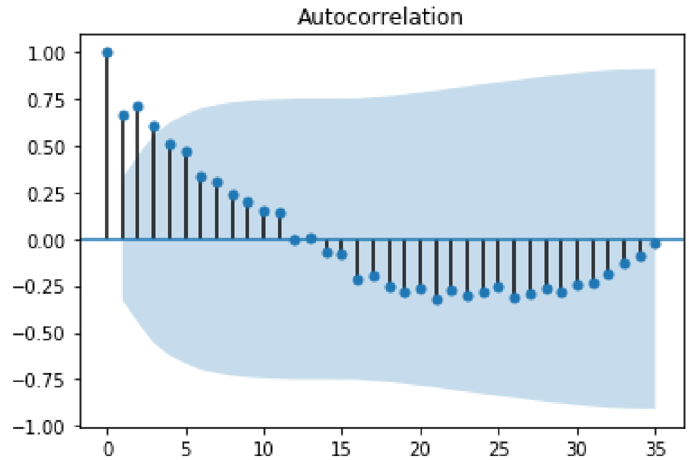


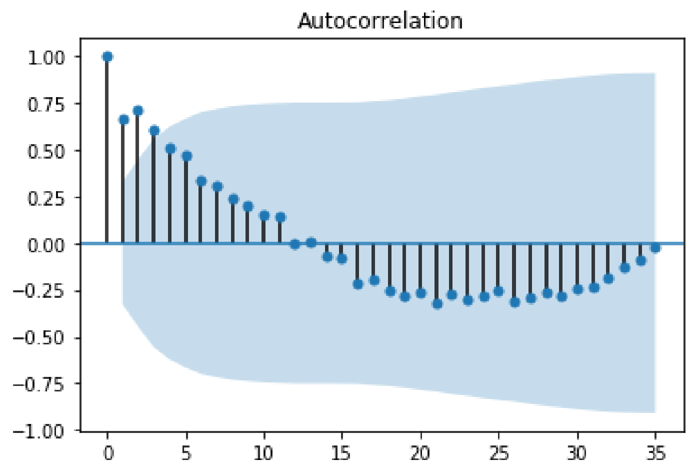
Stationary means mean,variance and covariance is constant over periods.

In [4]:
```
from statsmodels.graphics.tsaplots import plot_acf
plot_acf(sales)
```

```
C:\Users\prashant_gupta1\AppData\Local\Continuum\anaconda3\lib\site-packages\statsmodels\c
ompat\pandas.py:56: FutureWarning: The pandas.core.datetools module is deprecated and will
be removed in a future version. Please use the pandas.tseries module instead.
  from pandas.core import datetools
```

Out[4]:





Converting series to stationary

In [5]:
```
sales.head()
```

Out[5]:

Month	Sales of shampoo over a three year period
1901-01-01	266.0
1901-02-01	145.9
1901-03-01	183.1
1901-04-01	119.3
1901-05-01	180.3

In [6]:
```
sales.shift(1)
```

Out[6]:

Month	Sales of shampoo over a three year period
1901-01-01	NaN
1901-02-01	266.0
1901-03-01	145.9
1901-04-01	183.1
1901-05-01	119.3
1901-06-01	180.3
1901-07-01	168.5
1901-08-01	231.8
1901-09-01	224.5
1901-10-01	192.8
1901-11-01	122.9
1901-12-01	336.5
1902-01-01	185.9
1902-02-01	194.3
1902-03-01	149.5
1902-04-01	210.1
1902-05-01	273.3
1902-06-01	191.4
1902-07-01	287.0
1902-08-01	226.0
1902-09-01	303.6
1902-10-01	289.9
1902-11-01	421.6
1902-12-01	264.5
1903-01-01	342.3
1903-02-01	339.7
1903-03-01	440.4
1903-04-01	315.9
1903-05-01	439.3
1903-06-01	401.3
1903-07-01	437.4
1903-08-01	575.5
1903-09-01	407.6
1903-10-01	682.0
1903-11-01	475.3
1903-12-01	581.3

In [7]:
```
sales_diff = sales.diff(periods=1)
#integrated of order 1, denoted by d(for diff), one of the parameter of ARIMA model
```

In [8]:
```
sales_diff = sales_diff[1:]
sales_diff.head()
```

Out[8]:

	Sales of shampoo over a three year period
Month	
1901-02-01	-120.1
1901-03-01	37.2
1901-04-01	-63.8
1901-05-01	61.0
1901-06-01	-11.8

In [9]:
```
sales.shift(1)
```

Out[9]:

	Sales of shampoo over a three year period
Month	
1901-01-01	NaN
1901-02-01	266.0
1901-03-01	145.9
1901-04-01	183.1
1901-05-01	119.3
1901-06-01	180.3
1901-07-01	168.5
1901-08-01	231.8
1901-09-01	224.5
1901-10-01	192.8
1901-11-01	122.9
1901-12-01	336.5
1902-01-01	185.9
1902-02-01	194.3
1902-03-01	149.5
1902-04-01	210.1
1902-05-01	273.3
1902-06-01	191.4
1902-07-01	287.0
1902-08-01	226.0
1902-09-01	303.6
1902-10-01	289.9
1902-11-01	421.6
1902-12-01	264.5
1903-01-01	342.3
1903-02-01	339.7
1903-03-01	440.4
1903-04-01	315.9
1903-05-01	439.3
1903-06-01	401.3
1903-07-01	437.4
1903-08-01	575.5
1903-09-01	407.6
1903-10-01	682.0
1903-11-01	475.3
1903-12-01	581.3

In [10]:
```
sales_diff = sales.diff(periods=1)
#integrated of order 1, denoted by d(for diff), one of the parameter of ARIMA model
```

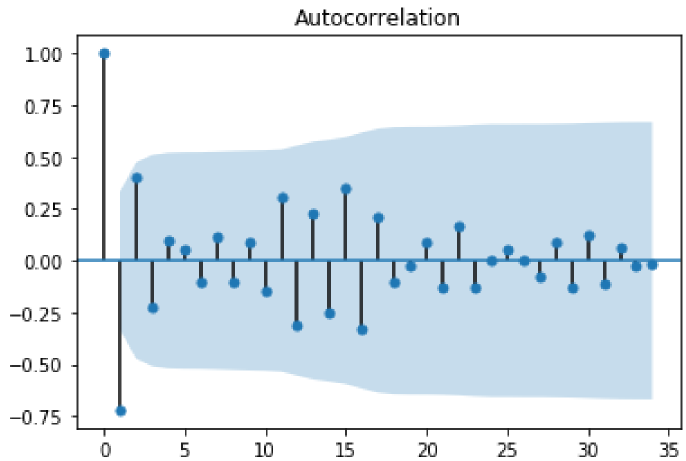
In [11]:
```
sales_diff = sales_diff[1:]
sales_diff.head()
```

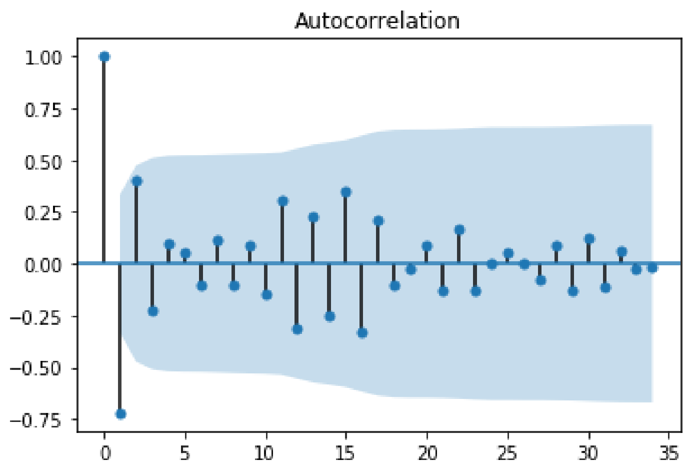
Out[11]:

Month	Sales of shampoo over a three year period
1901-02-01	-120.1
1901-03-01	37.2
1901-04-01	-63.8
1901-05-01	61.0
1901-06-01	-11.8

In [12]:
```
plot_acf(sales_diff)
```

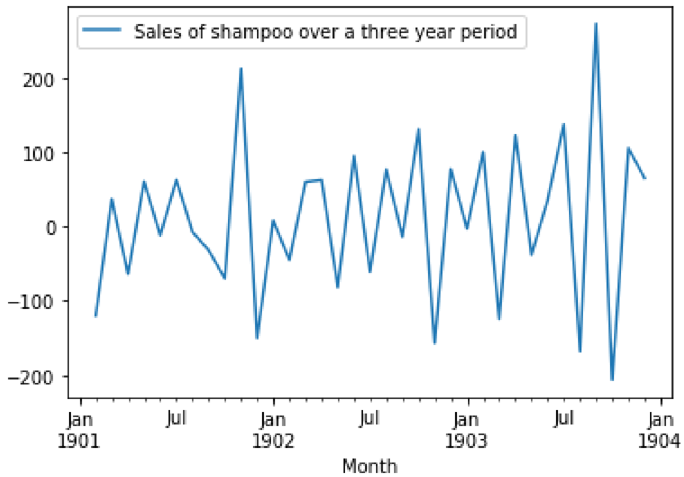
Out[12]:





In [13]:
```
sales_diff.plot()
```

Out[13]: <matplotlib.axes._subplots.AxesSubplot at 0x1fec30dbc18>



In [14]:
```
X = sales.values
train = X[0:28] # 27 data as train data
test = X[28:] # 9 data as test data
print(train.size)
print(test.size)
predictions = []
```

```
28
8
```

ARIMA model

In [15]:
```
from statsmodels.tsa.arima_model import ARIMA
```

In [16]:

```python
import itertools
p=d=q=range(0,6)
pdq=list(itertools.product(p,d,q))
pdq
```

```
 (0, 2, 5),
 (0, 3, 0),
 (0, 3, 1),
 (0, 3, 2),
 (0, 3, 3),
 (0, 3, 4),
 (0, 3, 5),
 (0, 4, 0),
 (0, 4, 1),
 (0, 4, 2),
 (0, 4, 3),
 (0, 4, 4),
 (0, 4, 5),
 (0, 5, 0),
 (0, 5, 1),
 (0, 5, 2),
 (0, 5, 3),
 (0, 5, 4),
 (0, 5, 5),
 (1, 0, 0),
```

In [17]:
```python
import warnings
warnings.filterwarnings('ignore')
for param in pdq:
    try:
        model_arima = ARIMA(train, order=param)
        model_arima_fit = model_arima.fit()
        print(param,model_arima_fit.aic)
    except:
        continue
```

```
(0, 0, 0) 335.09040511436183
(0, 0, 1) 334.3872829960065
(0, 0, 2) 329.3188116463245
(0, 0, 3) 330.92416191329755
(0, 0, 4) 326.20991814556083
(0, 0, 5) nan
(0, 1, 0) 324.4220452613395
(0, 1, 1) 308.00170527527325
(0, 1, 2) 306.76985736180313
(0, 2, 0) 343.4502397203673
(0, 2, 1) 318.2621879566042
(1, 0, 0) 330.891809426
(1, 0, 1) 325.6333949306371
(1, 0, 2) 322.4525346634769
(1, 1, 0) 309.11887677524066
(1, 1, 1) 306.79093608912103
(1, 1, 2) 306.91086651441833
(1, 1, 3) 311.87130308527367
(1, 1, 4) 309.11946068137985
(1, 1, 5) 314.62738257879744
(1, 2, 0) 317.80308367611093
(1, 2, 1) 304.28854627184654
(1, 2, 4) 301.84243592770645
(2, 0, 0) 322.0934945916866
(2, 0, 1) 323.6906512129914
(2, 1, 0) 310.4797159942252
(2, 1, 1) 308.67508697382726
(2, 1, 4) 310.7392463440925
(2, 1, 5) 309.26130294792756
(2, 2, 0) 317.2438967005482
(2, 2, 1) 305.70536447196343
(2, 2, 3) 301.9734131614505
(3, 0, 0) 324.0564894664129
(3, 0, 1) 345.56189420353957
(3, 1, 0) 305.2133148241061
(3, 1, 1) 306.4702994791127
(3, 1, 2) 303.36848271124654
(3, 1, 3) 305.18376856738263
(3, 1, 4) 306.97823625865476
(3, 2, 0) 309.7952134616359
(3, 2, 1) 300.1087027284869
(3, 2, 2) 303.66039523146463
(3, 2, 3) 298.77068698523084
(3, 2, 4) 300.6672665956482
(4, 0, 0) 330.7646131531544
(4, 0, 1) 333.30147262133727
(4, 1, 0) 306.53005409134755
(4, 1, 1) 308.44169440230746
(4, 1, 2) 305.2617876918041
(4, 1, 3) 307.17530796213805
(4, 1, 4) 335.2735615876484
(4, 2, 0) 307.1934977443044
```

```
(4, 2, 1) 301.1607743489167
(4, 2, 2) 304.81455102589587
(5, 0, 0) 333.44426873756004
(5, 1, 0) 308.3655922901959
(5, 1, 1) 310.09896254081184
(5, 1, 2) 306.1810888934841
(5, 1, 3) 308.8768446366264
(5, 2, 0) 306.8900144573346
(5, 2, 1) 302.8531580359661
(5, 2, 2) 302.5657340585613
(5, 2, 3) 308.3799433205256
```

It seems that out of different combinations ranging from order (0,0,0) to (5,5,5) param with values as p=3, d=2 and q=3 is the best because of lowest AIC value

In [18]:

```
#p,d,q
#p -> Periods taken for auto regressive model
#d -> Integrated order, difference
#q -> Periods in moving average model
model_arima = ARIMA(train, order=(3,2,3))
model_arima_fit = model_arima.fit()
print(model_arima_fit.aic)
```

```
298.77068698523084
```

In [19]:

```
predictions = model_arima_fit.forecast(steps=8)[0]
predictions
```

Out[19]:

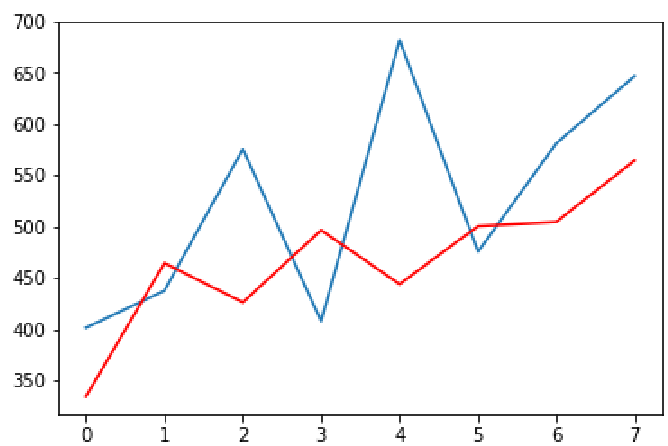
```
array([334.25497944, 464.33196912, 426.08758442, 496.34781322,
       443.78522094, 500.24819382, 504.44186036, 564.70722907])
```

In [20]:

```
plt.plot(test)
plt.plot(predictions, color='red')
```

Out[20]:

```
[<matplotlib.lines.Line2D at 0x1fec32697f0>]
```



In [21]:

```
from sklearn.metrics import mean_squared_error
mean_squared_error(test,predictions)
```

Out[21]:

```
13181.51607295262
```