In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pylab as plt
%matplotlib inline
```

In [2]:
```
data = pd.read_csv('f:/machine learning/AirPassengers.csv')
print (data.head())
print( '\n Data Types:')
print(data.dtypes)
```

```
     Month  #Passengers
0  1949-01          112
1  1949-02          118
2  1949-03          132
3  1949-04          129
4  1949-05          121

 Data Types:
Month          object
#Passengers     int64
dtype: object
```

In [3]:
```
from datetime import datetime
con=data['Month']
data['Month']=pd.to_datetime(data['Month'])
data.set_index('Month', inplace=True)
#check datatype of index
data.index
```

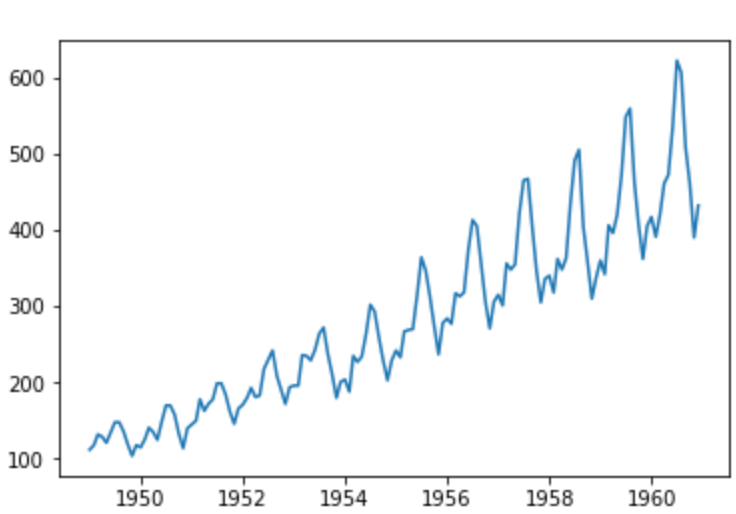
Out[3]:
```
DatetimeIndex(['1949-01-01', '1949-02-01', '1949-03-01', '1949-04-01',
               '1949-05-01', '1949-06-01', '1949-07-01', '1949-08-01',
               '1949-09-01', '1949-10-01',
               ...
               '1960-03-01', '1960-04-01', '1960-05-01', '1960-06-01',
               '1960-07-01', '1960-08-01', '1960-09-01', '1960-10-01',
               '1960-11-01', '1960-12-01'],
              dtype='datetime64[ns]', name='Month', length=144, freq=None)
```

In [4]:
```
#convert to time series:
ts = data['#Passengers']
ts.head(10)
```

Out[4]:
```
Month
1949-01-01    112
1949-02-01    118
1949-03-01    132
1949-04-01    129
1949-05-01    121
1949-06-01    135
1949-07-01    148
1949-08-01    148
1949-09-01    136
1949-10-01    119
Name: #Passengers, dtype: int64
```

In [5]:
```
plt.plot(ts)
```

Out[5]:
```
[<matplotlib.lines.Line2D at 0x2ec4268bbe0>]
```

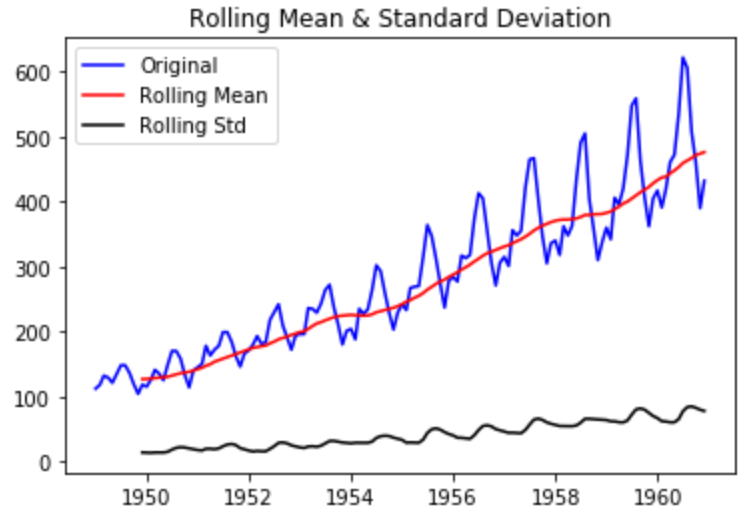


In [15]:
```
from statsmodels.tsa.stattools import adfuller
def test_stationarity(timeseries):

    #Determing rolling statistics
    rolmean = pd.Series.rolling(timeseries, window=12).mean()
    rolstd = pd.Series.rolling(timeseries, window=12).std()
#Plot rolling statistics:
    plt.plot(timeseries, color='blue',label='Original')
    plt.plot(rolmean, color='red', label='Rolling Mean')
    plt.plot(rolstd, color='black', label = 'Rolling Std')
    plt.legend(loc='best')
    plt.title('Rolling Mean & Standard Deviation')
    plt.show()
#Perform Dickey-Fuller test:
    print('Results of Dickey-Fuller Test:')
    dftest = adfuller(timeseries, autolag='AIC')
    dfoutput = pd.Series(dftest[0:4], index=['Test Statistic','p-value','#Lags Used','Number of Observations Use
    for key,value in dftest[4].items():
        dfoutput['Critical Value (%s)'%key] = value
    print(dfoutput)
```

In [ ]:
```
```

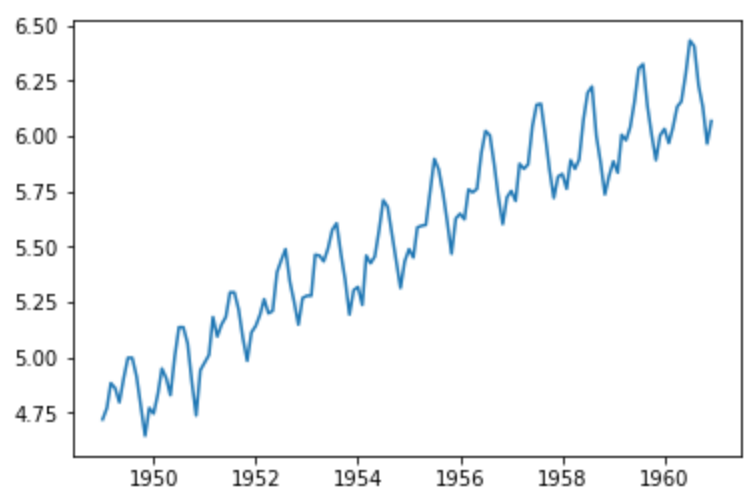
In [16]:
```
test_stationarity(ts)
```




```
Results of Dickey-Fuller Test:
Test Statistic                   0.815369
p-value                          0.991880
#Lags Used                      13.000000
Number of Observations Used    130.000000
Critical Value (1%)             -3.481682
Critical Value (5%)             -2.884042
Critical Value (10%)            -2.578770
dtype: float64
```

In [17]:
```
ts_log=np.log(ts)
plt.plot(ts_log)
```

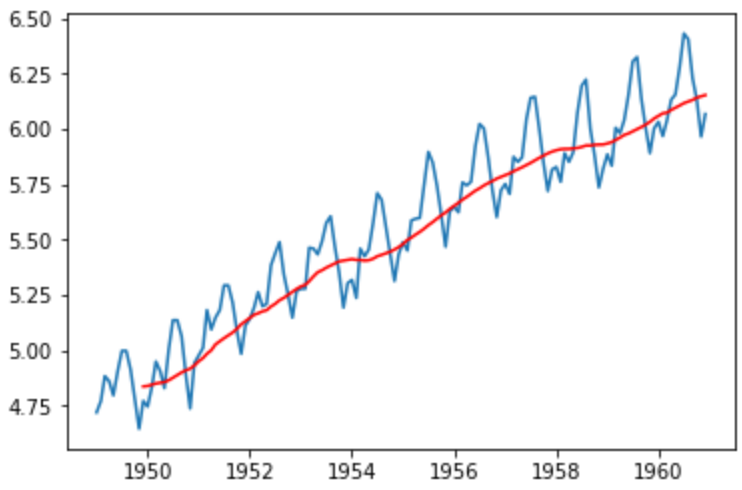
Out[17]:
```
[<matplotlib.lines.Line2D at 0x2ec4ae39e10>]
```



In [18]:
```
moving_avg=pd.Series.rolling(ts_log, 12).mean()
```

In [20]:
```
plt.plot(ts_log)
plt.plot(moving_avg, color='red')
```

Out[20]: [<matplotlib.lines.Line2D at 0x2ec4bf66ef0>]



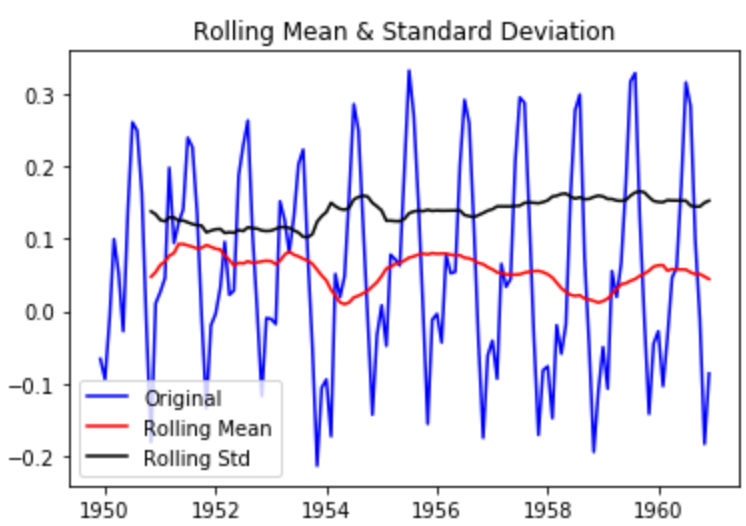
In [21]:
```
# Subtract the rolling mean from the original series
ts_log_moving_avg_diff=ts_log - moving_avg
ts_log_moving_avg_diff.head(12)
```

Out[21]:
```
Month
1949-01-01         NaN
1949-02-01         NaN
1949-03-01         NaN
1949-04-01         NaN
1949-05-01         NaN
1949-06-01         NaN
1949-07-01         NaN
1949-08-01         NaN
1949-09-01         NaN
1949-10-01         NaN
1949-11-01         NaN
1949-12-01   -0.065494
Name: #Passengers, dtype: float64
```

In [22]:
```
ts_log_moving_avg_diff.dropna(inplace=True)
ts_log_moving_avg_diff.head(12)
```

Out[22]:
```
Month
1949-12-01   -0.065494
1950-01-01   -0.093449
1950-02-01   -0.007566
1950-03-01    0.099416
1950-04-01    0.052142
1950-05-01   -0.027529
1950-06-01    0.139881
1950-07-01    0.260184
1950-08-01    0.248635
1950-09-01    0.162937
1950-10-01   -0.018578
1950-11-01   -0.180379
Name: #Passengers, dtype: float64
```

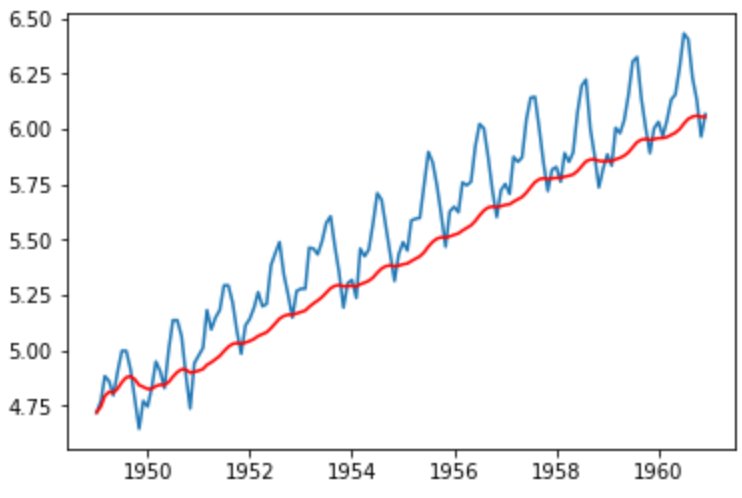
In [23]:
```
test_stationarity(ts_log_moving_avg_diff)
```



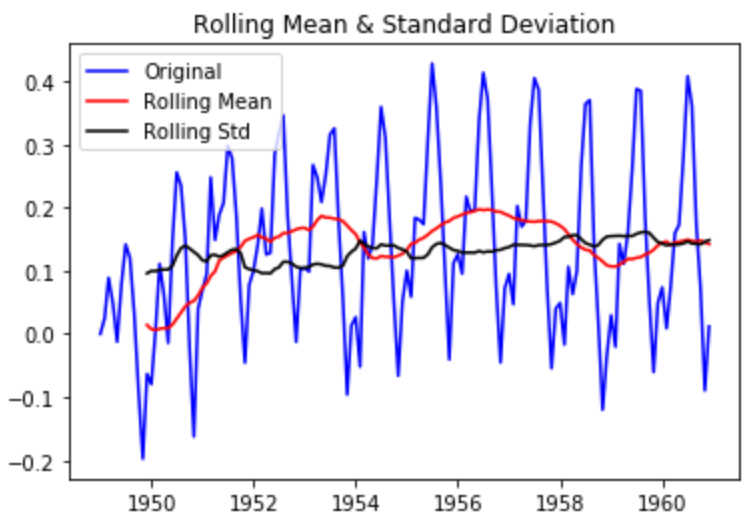

```
Results of Dickey-Fuller Test:
Test Statistic                  -3.162908
p-value                          0.022235
#Lags Used                      13.000000
Number of Observations Used    119.000000
Critical Value (1%)             -3.486535
Critical Value (5%)             -2.886151
Critical Value (10%)            -2.579896
dtype: float64
```

In [25]:
```
# Exponential weighted average
expweighted_avg=pd.Series.ewm(ts_log, halflife=12).mean()
plt.plot(ts_log)
plt.plot(expweighted_avg, color='red')
```

Out[25]: [<matplotlib.lines.Line2D at 0x2ec40edaa20>]



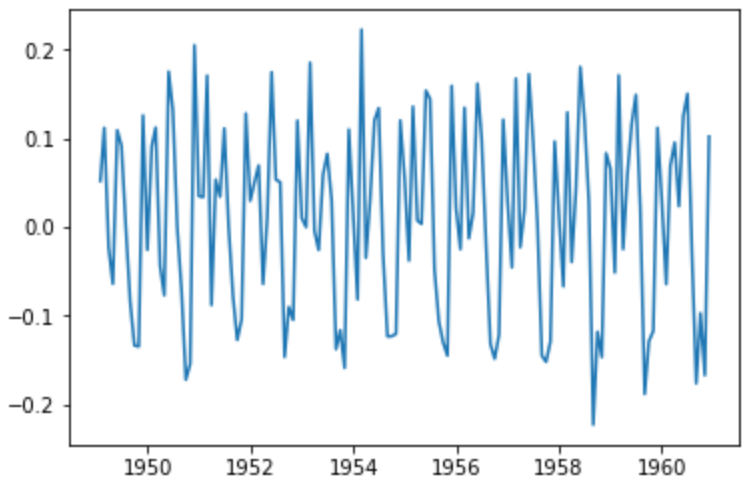
In [26]:
```
ts_log_ewma_diff=ts_log - expweighted_avg
test_stationarity(ts_log_ewma_diff)
```



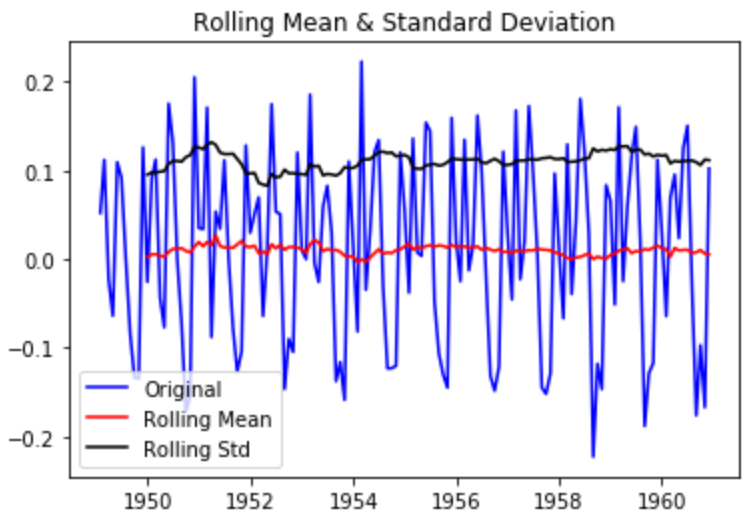

```
Results of Dickey-Fuller Test:
Test Statistic                  -3.601262
p-value                          0.005737
#Lags Used                      13.000000
Number of Observations Used    130.000000
Critical Value (1%)             -3.481682
Critical Value (5%)             -2.884042
Critical Value (10%)            -2.578770
dtype: float64
```

In [27]:
```
# Seasonality with trends
#Take the first difference:
ts_log_diff = ts_log - ts_log.shift()
plt.plot(ts_log_diff)
```

Out[27]: [<matplotlib.lines.Line2D at 0x2ec4c265d68>]



In [28]:
```
ts_log_diff.dropna(inplace=True)
test_stationarity(ts_log_diff)
```




```
Results of Dickey-Fuller Test:
Test Statistic                  -2.717131
p-value                          0.071121
#Lags Used                      14.000000
Number of Observations Used    128.000000
Critical Value (1%)             -3.482501
Critical Value (5%)             -2.884398
Critical Value (10%)            -2.578960
dtype: float64
```

In [ ]: