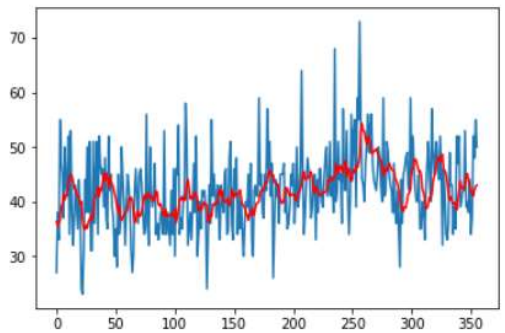
Assignment 5

Compare the accuracies of Moving Average, Single Exponential Smoothing, Double Exponential Smoothing and Triple Exponential Smoothing

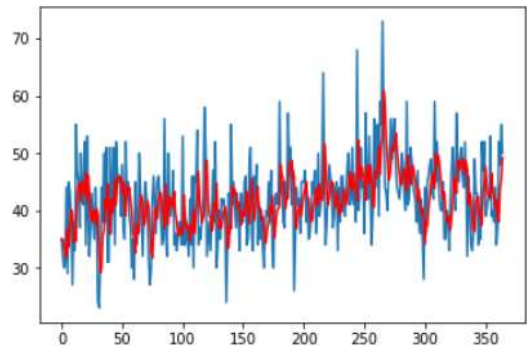
```
def MAM(X,window):
    history = [X[i] for i in range(window)] #Initially observed only 3 values
    test = [X[i] for i in range(window, len(X))] #WE can't predict X[2] as there is not enough data
    predictions = []
    for t in range(len(test)):
        length = len(history)
        yhat = mean([history[i] for i in range(length-window,length)])
        obs = test[t]
        predictions.append(yhat)
        history.append(obs)
        #print(predictions[t],history[t])
    print("Moving Averge Method")
    forecast_accuracy(predictions,test)
    pyplot.plot(test)
    pyplot.plot(predictions, color='red')
    pyplot.show()
```

```
Moving Averge Method
MAPE : 0.14114561084079616 ME : -0.12453183520599262 MAE : 5.796816479400749 RMSE : 7.242330118152365
```



```
def ES(series, alpha):
    result = [series[0]] # first value is same as series
    for n in range(1, len(series)):
        result.append(alpha * series[n-1] + (1 - alpha) * result[n-1])
    print("Single Exponential Method")
    forecast_accuracy(result,series)
    pyplot.plot(series)
    pyplot.plot(result, color='red')
    pyplot.show()
```

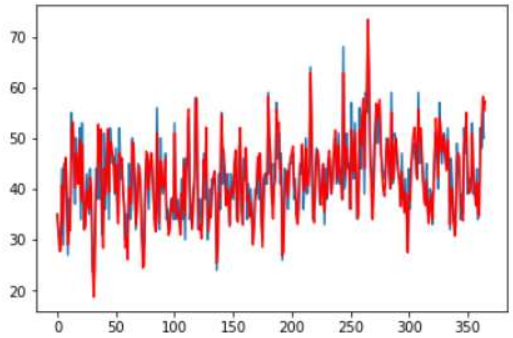
```
Single Exponential Method
MAPE : 0.145967014705052 ME : -0.09954965504183189 MAE : 5.999246665741553 RMSE : 7.48058983618224
```




```
def DES(series, alpha, beta):
    result = [series[0]]
    for n in range(1, len(series)+1):
        if n == 1:
            level, trend = series[0], series[1] - series[0]
        if n >= len(series): # we are forecasting
          val = result[-1]
        else:
          val = series[n]
        last_level, level = level, alpha*val + (1-alpha)*result[n-1]
        trend = beta*(level-last_level) + (1-beta)*trend
```

```
        result.append(level+trend)
    print("Double Exponential Method")
    forecast_accuracy(result,series)
    pyplot.plot(series)
    pyplot.plot(result, color='red')
    pyplot.show()
```

```
Double Exponential Method
MAPE : 0.0613773320400511 ME : 0.023649581070345357 MAE : 2.506384460038338 RMSE : 3.081471038959605
```




```
def TES(series, slen, alpha, beta, gamma, n_preds):
    result = []
    seasonals = initial_seasonal_components(series, slen)
    for i in range(len(series)+n_preds):
        if i == 0: # initial values
            smooth = series[0]
            trend = initial_trend(series, slen)
            result.append(series[0])
            continue
        if i >= len(series): # we are forecasting
            m = i - len(series) + 1
            result.append((smooth + m*trend) + seasonals[i%slen])
        else:
            val = series[i]
            last_smooth, smooth = smooth, alpha*(val-seasonals[i%slen]) + (1-alpha)*(smooth+trend)
```

```
        trend = beta * (smooth-last_smooth) + (1-beta)*trend
        seasonals[i%slen] = gamma*(val-smooth) + (1-gamma)*seasonals[i%slen]
        result.append(smooth+trend+seasonals[i%slen])
print("Triple Exponential Method")
forecast_accuracy(result,series)
pyplot.plot(series)
pyplot.plot(result, color='red')
pyplot.show()
```

```
Triple Exponential Method
MAPE : 0.04585691789560022 ME : 0.04027563906218409 MAE : 1.872279931930349 RMSE : 2.313749710292178
```