MSIS 5223: Tutorial 13 – Time Series Analysis in R

1. Prediction Versus Forecasting

At the heart of analytics is the desire to predict the future and make fortunes. The majority of models covered in this class and the field as a whole focus on the concept of prediction. Another similar concept exists which is often misunderstood. This is the idea of forecasting. *Prediction* uses interpolation of data while *forecasting* relies on extrapolation. Interpolation is the process of creating new data points within a field of known data points. Extrapolation is the process of creating new data points outside a field of known data points.

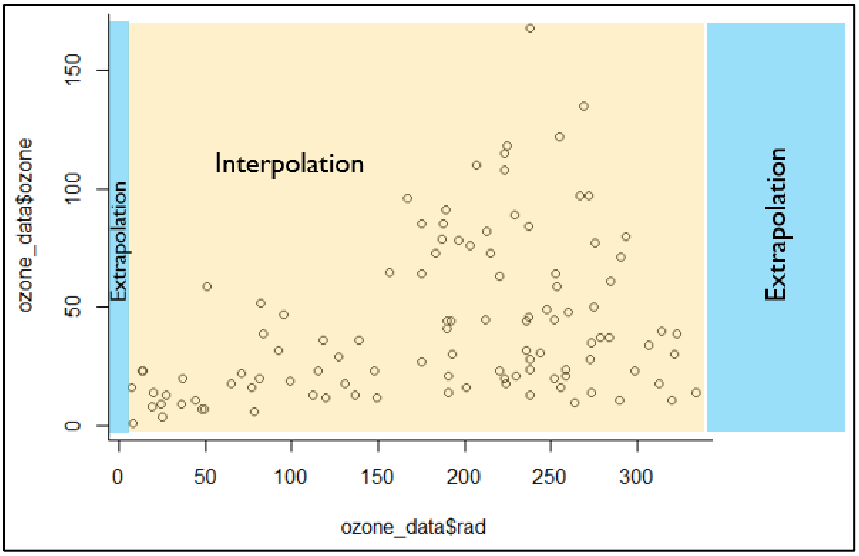


Figure 1 Extrapolation Versus Interpolation

Look at the colored areas in Figure 1. This data comes from the pollution data you have used in previous exercises. Notice the yellow area contains the entire field of data. The blue area, on the other hand, contains no data. When working with regression, classification, neural networks, and many other models the predicted values will only appear in the yellow area. This is interpolation. You will never receive a prediction for a value of *x* outside of this yellow boundary.

The blue area is beyond the known boundary. This is the realm of forecasting, which is a technique designed to extrapolate. Extrapolation is a dangerous endeavor, even for seasoned data miners. Prediction is safer because when predicting a *y* for a given *x*, the new *x* you are dealing with is similar to many other values of *x*. This means the predicted *y* you receive is also going to be similar to the *y* values you already have. With extrapolation, there are no similar values of *x* because the new *x* lies outside the known boundary. This also means that any *y* you "predict" won't be surrounded by other *y* values. An example should help illustrate this problem.

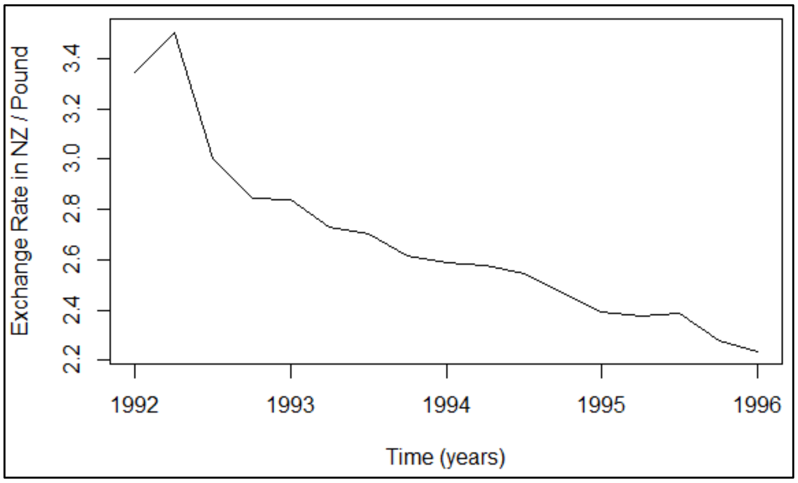


Figure 2 Exchange Rate of NZ $ to GB Pounds

Figure 2 shows the exchange rate between New Zealand's dollar and Britain's pound. Starting in 1992 the exchange rate increases, but decreases spectacularly until the data ends in 1996. The trend of this data is obviously a negative slope. If you took this data and placed it in a regression model, you would predict a continuation of this downward trend. The problem, though, is you are attempting to extrapolate beyond the known data. This is not prediction, because you are not trying to predict what will happen in August of 1994 or October of 1995; you are trying to determine what will happen in February of 1996, data that you do not have.

Assuming you make your prediction about a continuation of a downward trend, you would have lost on many investments. Look at Figure 3. In the middle of the timeline is the year 1996. This is the lowest value and where the negative trend stops. Starting in 1996 and continuing on passed January of 2000, the data has a positive trend. In short, the moral of the story is extrapolation is dangerous. Conveniently, time series offers many techniques and methods to mitigate some of the potential danger inherent in forecasting.

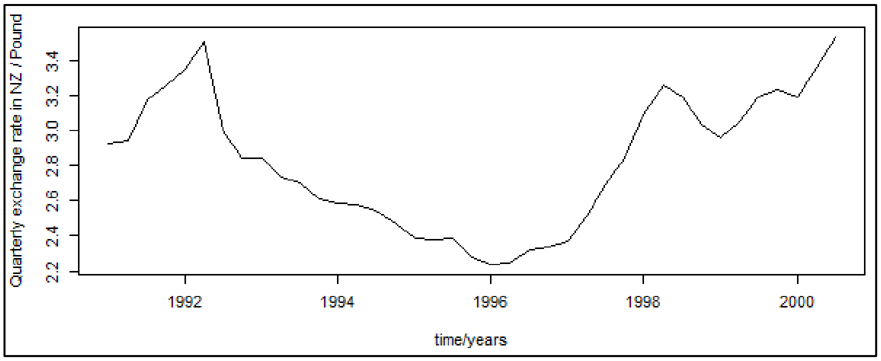


Figure 3 Quarterly Exchange Rate GB Pound to NZ $

Many forecasting techniques exist depending on the type of data used. If the data is horizontal in nature, then the following techniques can be used: Naïve Method, Simple Mean Method, Simple Moving Average, Weighted Moving Average, and Exponential Smoothing. When data has a trend, positive or negative, a Trend-Adjusted Exponential Smoothing function will do or simple linear regression. The following table illustrates these differences.

	Horizontal Data	**Trend Data**	**Seasonal Data**
Simple Moving Average	X		
Weighted Moving Average	X		
Exponential Smoothing	X		
Trend-Adjusted Exponential Smoothing		X	
Linear Regression		X	
Holt-Winters			X

Table 4 Time Series Analysis by Data Type

2. Basic Forecasting Techniques

Many basic forecasting methods exist when assessing horizontal data. Most of these basic techniques are rarely used now because the majority of real-world data possesses trends or seasonality. Why bother to learn these? These are foundational techniques and help teach important concepts related to time series analysis. The more complicated models originated from these.

The first method is the Naïve Method. The forecast for the next period equals the demand for the current period and is represented by the following equation:

$$F_{t+1} + 1 = A_t$$

where *F* is the forecast for the next period after *t* and *A* is the actual value. It should be noted that *t* represents the current period under consideration.

Look at the data in the table below. This data is simple and contains three periods of data. The purpose is to forecast the fourth period. The forecast for each period is the actual demand from the previous period. Since Period 1 does not have a prior period, its forecast is not given a value; that is, it is initialized as null. Moving to Period 2, the previous period's demand is 15, so the forecast will be 15.

Period	Demand	Forecast
1	15	-
2	28	15
3	25	28
4		25

Table 5 Naive Method

This forecasting method is very simple, but it doesn't take into consideration any data past the previous period. The Simple Mean Method is an improvement on the Naïve Method because it takes the average of all past periods. Look at the next table of data, which is an expansion of the previous table. Each successive period's forecast is the average value of all previous demand values. The equation is given as follows:

$$F_{t+1} = {\sum_{i=t}^{t} A_i}/_{t}$$

To illustrate how this method works, look at the forecast for Period 6. This is the average of the demand from Periods 1 through 5. Each period incorporates the values of demand from previous periods.

Period	Demand	Forecast	Calculations
1	15	-	No calculation
2	28	15	15/1
3	25	21.5	(15+28) / 2
4	34	22.7	(15+28+25) / 3
5	53	25.5	(15+28+25+34) / 4
6	42	31	(15+28+25+34+53) / 5
7		32.8	(15+28+25+34+53+42) / 6

Table 6 Simple Mean Method

The weakness of the Simple Mean Method is obvious. This method is not very responsive to recent changes in the data. Currently, the demand and forecast are following an upward trend. Looking at Period 6, the data may start a downward trend. By continuing to average all previous values together, future forecasts will not have the ability to make similar adjustments.

Simple Moving Average is a technique designed to place emphasis on recent data. This method places the same weight on all periods. This method works well when data is fairly stable over time. When a trend is present, this method does not work very well. The forecast lags the actual demand because of the lag effect. Also keep in mind, that decreasing the number of periods in the averaging effect makes the forecast more responsive. The equation for Simple Moving Average is as follows:

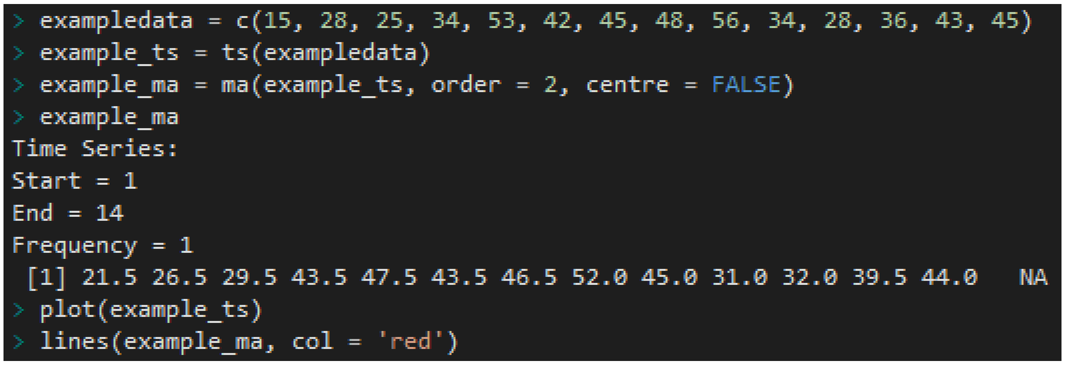
$$F_{t+1} = \frac{\sum_{i=t-n+1}^{t} A_i}{n}$$

where *n* is the number of periods used to calculate the forecast. Note, *n* cannot exceed the total number of previous periods. For example, if a forecast is being made for 2007 and the number of periods to average is 5, but the data only goes back to 2004, then the method will not work.

Period	Demand	Forecast	Calculations
1	15	-	No calculation
2	28	-	No calculation
3	25	21.5	(15+28) / 2
4		26.5	(25+25) / 2

Table 7 Simple Moving Average

The example shown in the table above uses 2 previous periods. This means Periods 1 and 2 will not receive a forecast because 2 periods are not available. Period 3 is the first period that can have a forecast, followed by Period 4. The following code in R provides the full example of this analysis followed by a plot of the actual demand with the forecasted demands.



```
> exampledata = c(15, 28, 25, 34, 53, 42, 45, 48, 56, 34, 28, 36, 43, 45)
> example_ts = ts(exampledata)
> example_ma = ma(example_ts, order = 2, centre = FALSE)
> example_ma
Time Series:
Start = 1
End = 14
Frequency = 1
 [1] 21.5 26.5 29.5 43.5 47.5 43.5 46.5 52.0 45.0 31.0 32.0 39.5 44.0   NA
> plot(example_ts)
> lines(example_ma, col = 'red')
```

Figure 8 Simple Moving Average in R

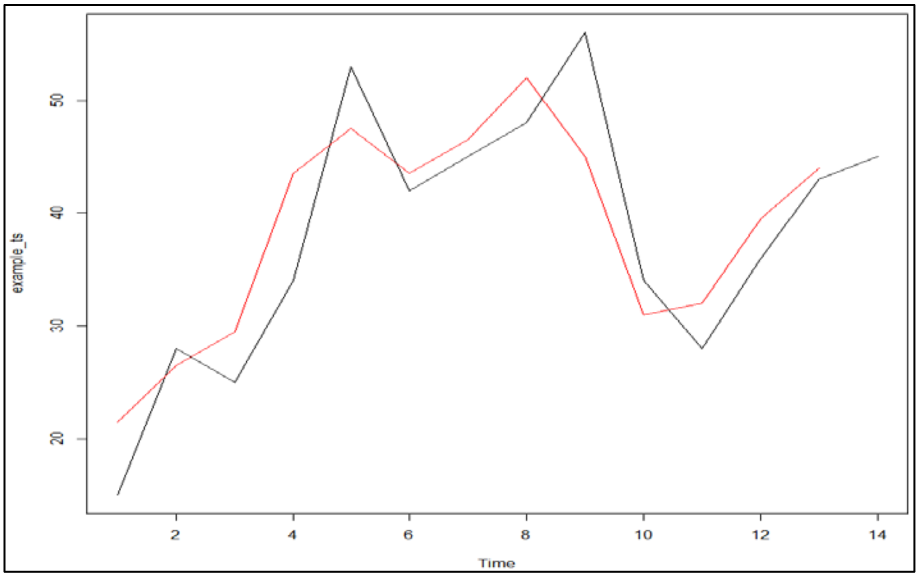


Figure 9 Actual Demand (black) vs. Forecast (red)

Notice the forecast lags the actual slightly. When the actual data has a change in direction, positive or negative, the forecast needs one additional time period to catch up. This lag is caused by the number of averaging periods. To illustrate this, look at the example again, but this time a new forecast is created using 4 averaging periods.

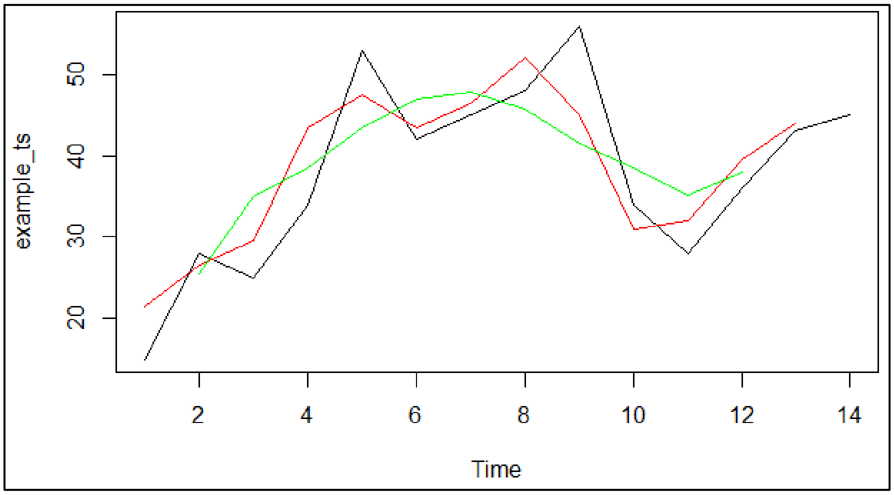


Figure 10 Actual Demand (black), Forecast 2-Periods (red), Forecast 4-Periods (green)

The green line in Figure 10 is the forecast with 4 averaging periods. This line is smoother than the red line, but not as responsive to changes. Importantly, the larger *n* is, the smoother the forecast and the less responsive it is to changes in demand; the smaller *n* is, the more responsive the forecast is to changes in demand.

Unfortunately, the Simple Moving Average has limitations. All time periods averaged must share the same weight. A more realistic approach would give more recent periods a greater emphasis and those further back in time less. This is where Weighted Moving Average has its advantage. The following should be kept in mind when using Weighted Moving Average:

- Weighted moving average allows different emphasis to be placed on different time periods.
- This method works well when the demand is fairly stable over time, but does not do a good job of forecasting when a trend is present.
- The forecast lags the actual demand because of the averaging effect.
- Weights tend to be based on the forecaster’s experience.
- Decreasing the number of periods in a forecast and/or increasing the size of the weights for more recent demand creates a more responsive forecast.

The following represents the equation for the Weighted Moving Average, which is very similar to that of the Simple Moving Average:

$$F_{t+1} = \frac{\sum_{i=t-n+1}^{t} w_i A_i}{\sum_{i=t-n+1}^{t} w_i}$$

where w_i is the weight assigned to period i. The following table illustrates a simple example using weights of 3 and 2 for time period t and t-1 respectively.

Period	Demand	Forecast	Calculations
1	15	-	No calculation
2	28	-	No calculation
3	25	22.8	(3*28+2*15) / (3+2)
4		26.2	(3*25+2*28) / (3+2)

Figure 11 Weighted Moving Average

The next evolution of time series functions is Exponential Smoothing. Exponential smoothing forecasting is a sophisticated weighted moving average forecasting technique that relies on a single smoothing parameter; unlike Weighted Moving Average which relies on several. Like the previous time series models, exponential smoothing is suitable for data without a trend and the forecast lags the actual demand because of the averaging effect. In addition, this model still finds use in the real-world for forecasting, unlike the previous models discussed.

Exponential Smoothing is a more robust method and is recommended over the previous examples. The next period's forecast equals the current period's forecast adjusted by a fraction of the difference between the current period's actual demand and its forecast. Two alternate versions are provided where the forecast is

$$F_{t+1} = F_t + \alpha(A_t - F_t)$$

or

$$F_{t+1} = \alpha A_t + (1 - \alpha)F_t$$

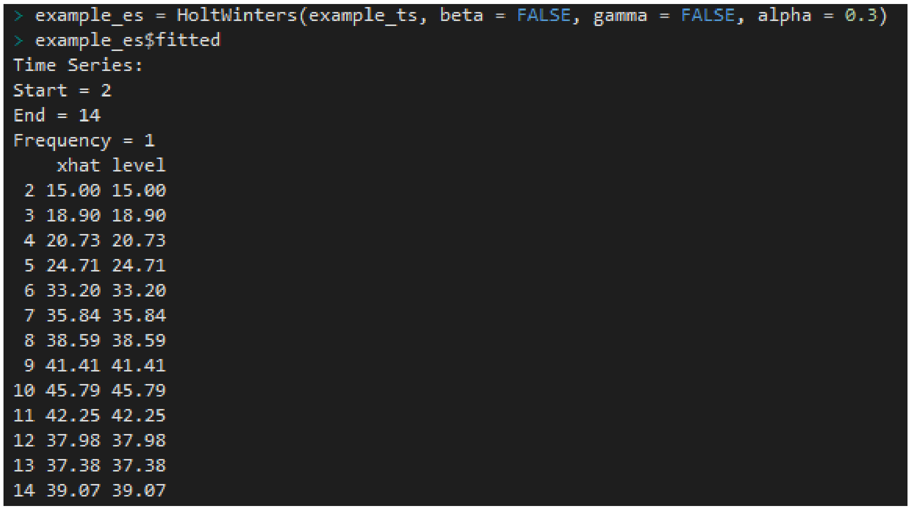
where α is a smoothing parameter such that $0 \leq \alpha \leq 1$.

Using the same demand data, the following table illustrates the process of using Exponential Smoothing. This example uses a smoothing parameter of 0.3. Unlike the previous methods, the first period must have an initial forecast created. This usually entails taking the demand and using it as the forecast. For Period 1, the forecast is initialized as 15 because the demand is 15.

Period	Demand	Forecast	Calculations
1	15	15	Must initialize first forecast!
2	28	15	15+0.3 (15-15)
3	25	18.9	15+0.3 (28-15)
4		20.7	18.9+0.3 (25-18.9)

Figure 12 Exponential Smoothing Example

To perform Exponential Smoothing within R, use the HoltWinters() function and set beta and gamma equal to FALSE. This is because Exponential Smoothing only uses the alpha parameter. See Figure 13 below for the code. Notice the predicted values correspond to those shown in the table above.



```
> example_es = HoltWinters(example_ts, beta = FALSE, gamma = FALSE, alpha = 0.3)
> example_es$fitted
Time Series:
Start = 2
End = 14
Frequency = 1
    xhat level
 2 15.00 15.00
 3 18.90 18.90
 4 20.73 20.73
 5 24.71 24.71
 6 33.20 33.20
 7 35.84 35.84
 8 38.59 38.59
 9 41.41 41.41
10 45.79 45.79
11 42.25 42.25
12 37.98 37.98
13 37.38 37.38
14 39.07 39.07
```

Figure 13 Exponential Smoothing Alpha=0.3

What happens when α is adjusted to a different value? The same model is run with an alpha value of 0.7. The new model and original are plotted and shown below in Figure 14. Notice the difference in both of the plots. The first plot has the smaller alpha; the second plot has the larger alpha. A larger smoothing parameter emphasizes recent demand and yields a forecast which is more responsive to changes in actual demand; a smaller alpha places more uniform emphasis on demand and yields a forecast which is more stable and less responsive to changes in actual demand.

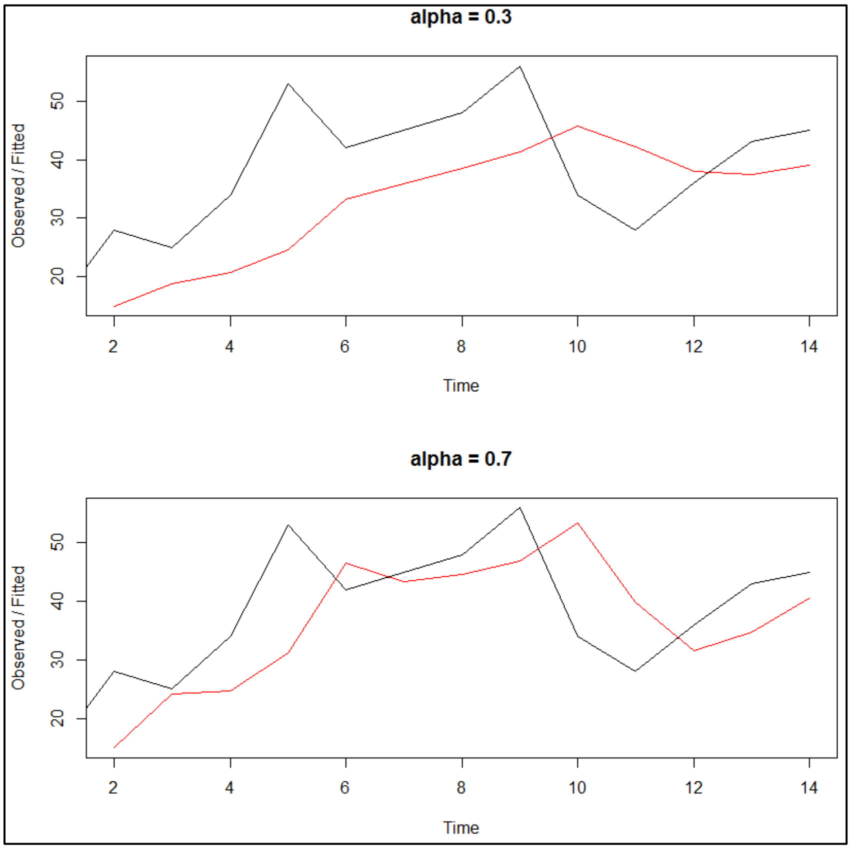
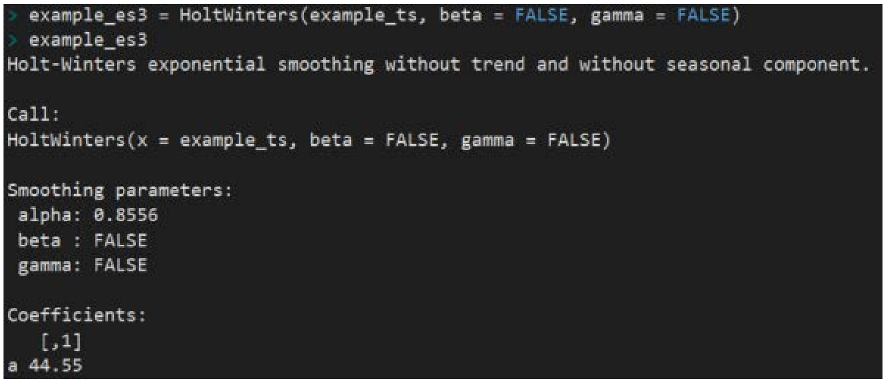


Figure 14 Comparison of Alpha Values

Which of the models is more appropriate? This depends on your preference for a more stable model versus something more flexible. Additionally, this decision should only be made once sufficient experience with the industry is had.

As an alternative, the modeling technique can provide an estimate of the smoothing parameter. The code in Figure 15 shows the code and resulting output. For the model, do not provide a value for the smoothing parameter, alpha. Looking at the output, the alpha value for the model is estimated at 0.86, indicating a much more aggressive model in terms of responsiveness than the 0.7 chosen previously. Once obtained, the fitted, or predicted, values can be obtained and used to forecast.



```
> example_es3 = HoltWinters(example_ts, beta = FALSE, gamma = FALSE)
> example_es3
Holt-Winters exponential smoothing without trend and without seasonal component.

Call:
HoltWinters(x = example_ts, beta = FALSE, gamma = FALSE)

Smoothing parameters:
 alpha: 0.8556
 beta : FALSE
 gamma: FALSE

Coefficients:
   [,1]
a 44.55
```

Figure 15 Exponential Smoothing - Estimated Smoothing Parameter

Looking at the model plotted, it is very apparent that the model is extremely responsive to the actual data. The forecast is limited to the range of data, which is not very different from the idea of interpolation. Since the purpose of forecasting is to extrapolate beyond the range of data, some adjustment to the code is required.

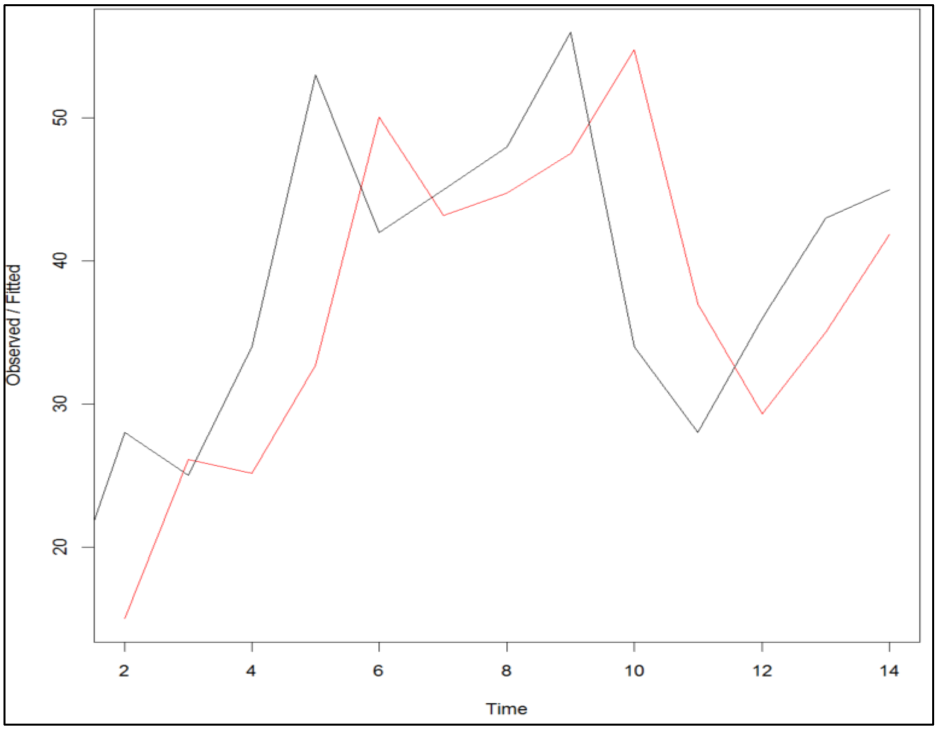
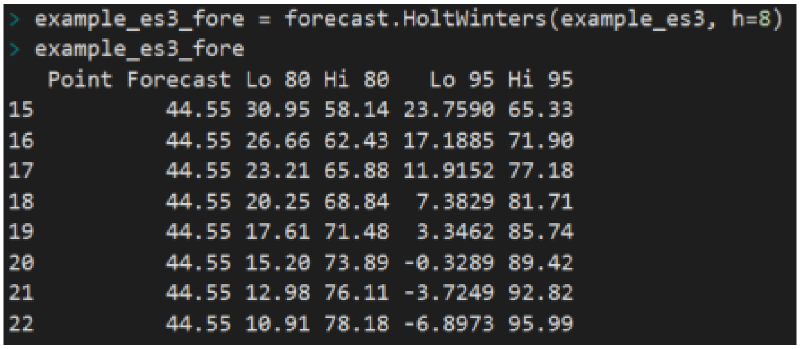


Figure 16 Exponential Smoothing Model with Alpha of 0.86

To make future forecasts beyond the current data set, the function forecast.HoltWinters() from the library forecast is used. Use the argument h to set the number of periods after the last data point you want to forecast. Figure 17 provides the output and code for creating the model.



```
> example_es3_fore = forecast.HoltWinters(example_es3, h=8)
> example_es3_fore
   Point Forecast Lo 80 Hi 80    Lo 95 Hi 95
15          44.55 30.95 58.14 23.7590 65.33
16          44.55 26.66 62.43 17.1885 71.90
17          44.55 23.21 65.88 11.9152 77.18
18          44.55 20.25 68.84  7.3829 81.71
19          44.55 17.61 71.48  3.3462 85.74
20          44.55 15.20 73.89 -0.3289 89.42
21          44.55 12.98 76.11 -3.7249 92.82
22          44.55 10.91 78.18 -6.8973 95.99
```

Figure 17 Forecasting 8 Periods

For each forecasted period an 80% prediction interval and 95% prediction interval are provided. For example, Period 21 has a forecasted value of 44.55 with a 95% interval of -3.73 to 92.82. The real beauty comes from plotting the model in R. The 80% interval is the dark, gray-blue shaded area while the 95% interval is the gray shaded area.

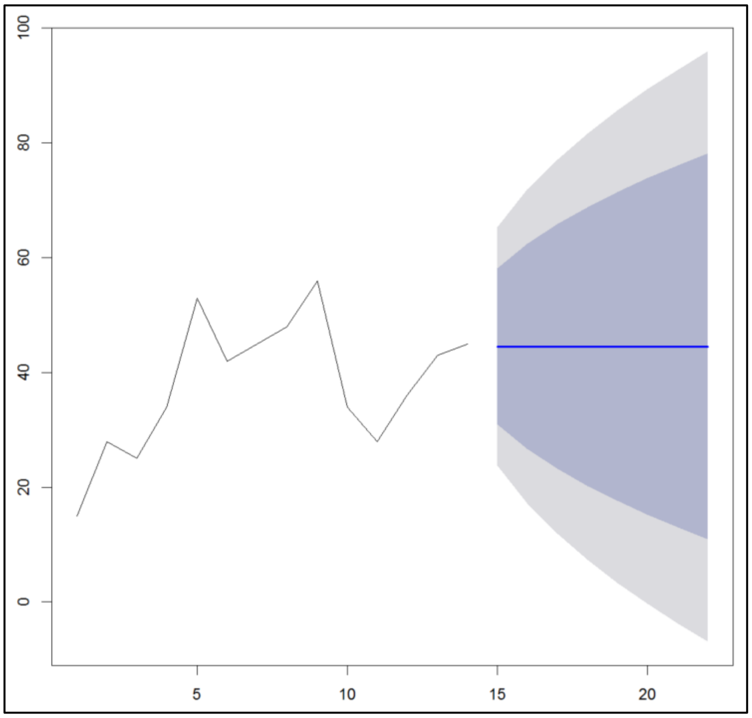


Figure 18 Estimated Forecast Beyond Known Data

Just like with regression, it is important to assess the residuals for the forecasted model and ensure they are normally distributed with a mean close to zero and homoscedasticity, or constant variance. To check for constant variance, simply plot the residuals, the errors, from the forecasted model. The data does not exhibit homoscedasticity, but reveals heteroscedasticity, or non-constant variance. The data is not spread consistently about the mean, the red line in Figure 19.

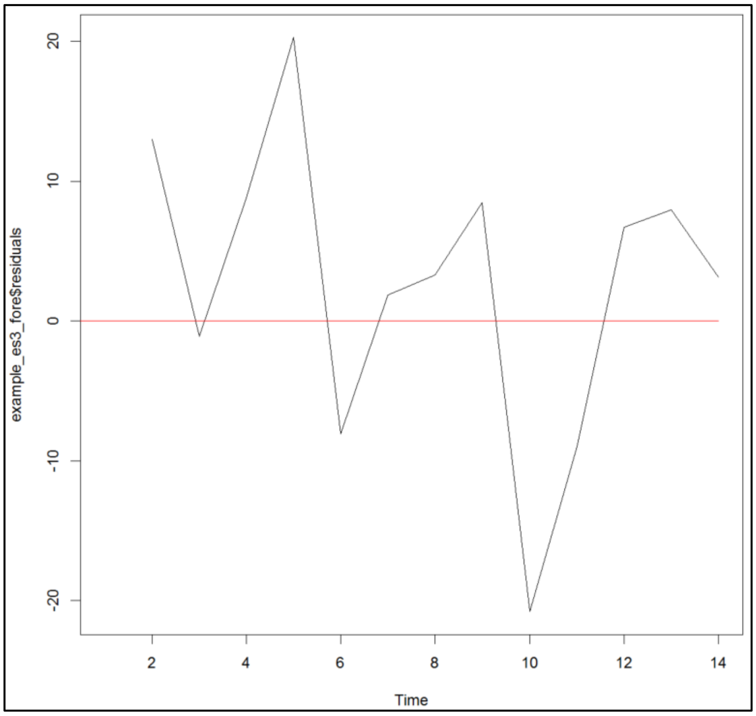


Figure 19 Plot of Residuals for Assessing Constant Variance

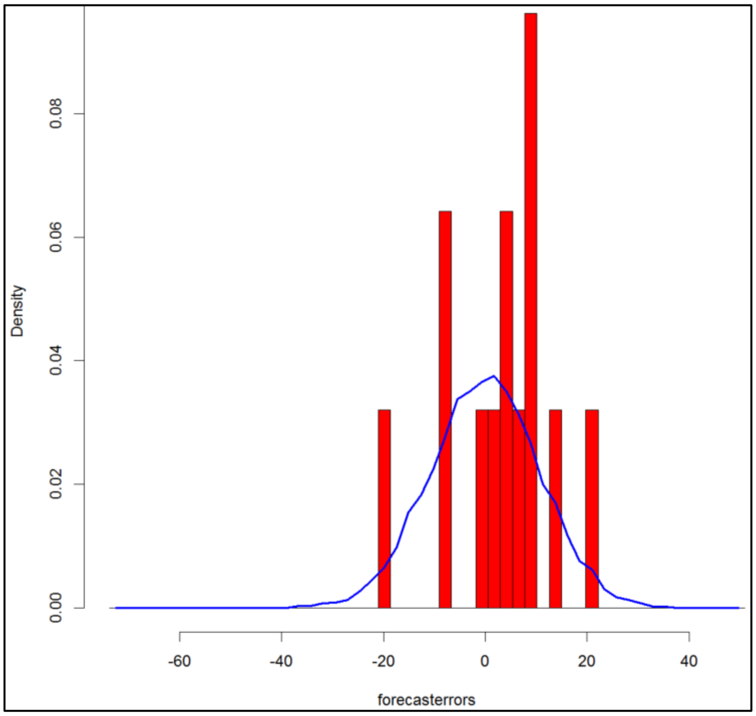


Figure 20 Assessing Normality of the Forecast

The next step is to assess the normality of the residuals, similar to what is done in regression. The R script file for this tutorial provides a function to create a histogram of the residual distribution. The output is shown above in Figure 20. The errors do not come close to forming a normal distribution. The conclusion, based on these two assessments, is a different model should be selected.

3. Components of Time Series: Trends

Time series data possess three main components: 1) trends, 2) seasonality, and 3) irregularity. The first component is the easiest to understand; the other two will be discussed later in this tutorial. This is merely the direction data moves over time and can be positive or negative. Data is not limited to just a single direction. This means data can begin in a positive direction and change direction by moving downward.

All the previous time series modeling techniques are not capable of estimating trends. Exponential Smoothing has the ability if an additional parameter is added. This parameter is β and is used to estimate the slope of the line. Once added, the beta parameter changes the model to a Trend-Adjusted Exponential Smoothing. The equation is given below:

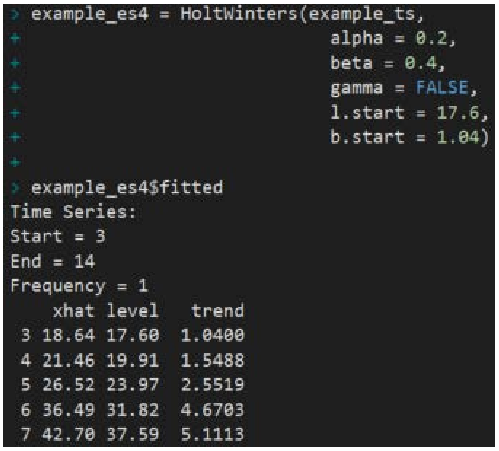
Step 1: $F_t = \alpha A_t + (1 - \alpha)(F_{t-1} + T_{t-1})$

Step 2: $T_t = \beta(F_t - F_{t-1}) + (1 - \beta)T_{t-1}$

Step 3: $TAF_{t+m} = F_t + mT_t$

where the smoothing parameter for β is $0 \leq \beta \leq 1$ and *m* is the number of future periods to forecast.

Figure 21 shows the R code with the example data. Alpha is given a value of 0.2, beta is 0.4, and gamma is still set to FALSE. A couple of additional arguments are used. The first one, l.start, is the initial value for F_t and b.start is the initial slope, or T_t. Since the first period shown in the output is Period 3, this means the initial values of 17.6 and 1.04 are for Period 2. Step 3



```
> example_es4 = HoltWinters(example_ts,
+                           alpha = 0.2,
+                           beta = 0.4,
+                           gamma = FALSE,
+                           l.start = 17.6,
+                           b.start = 1.04)
+
> example_es4$fitted
Time Series:
Start = 3
End = 14
Frequency = 1
    xhat level   trend
 3 18.64 17.60  1.0400
 4 21.46 19.91  1.5488
 5 26.52 23.97  2.5519
 6 36.49 31.82  4.6703
 7 42.70 37.59  5.1113
```

Figure 21 Trend-Adjusted Exponential Smoothing

in the formula above is found in the first column of the output, labeled `xhat`. These are the forecasted values for this time series, or TAF_{t+m}.

Just like Exponential Smoothing, the forecasted values of Trend-Adjusted Exponential Smoothing are influenced by changes in α and β. The figure below illustrates these differences.

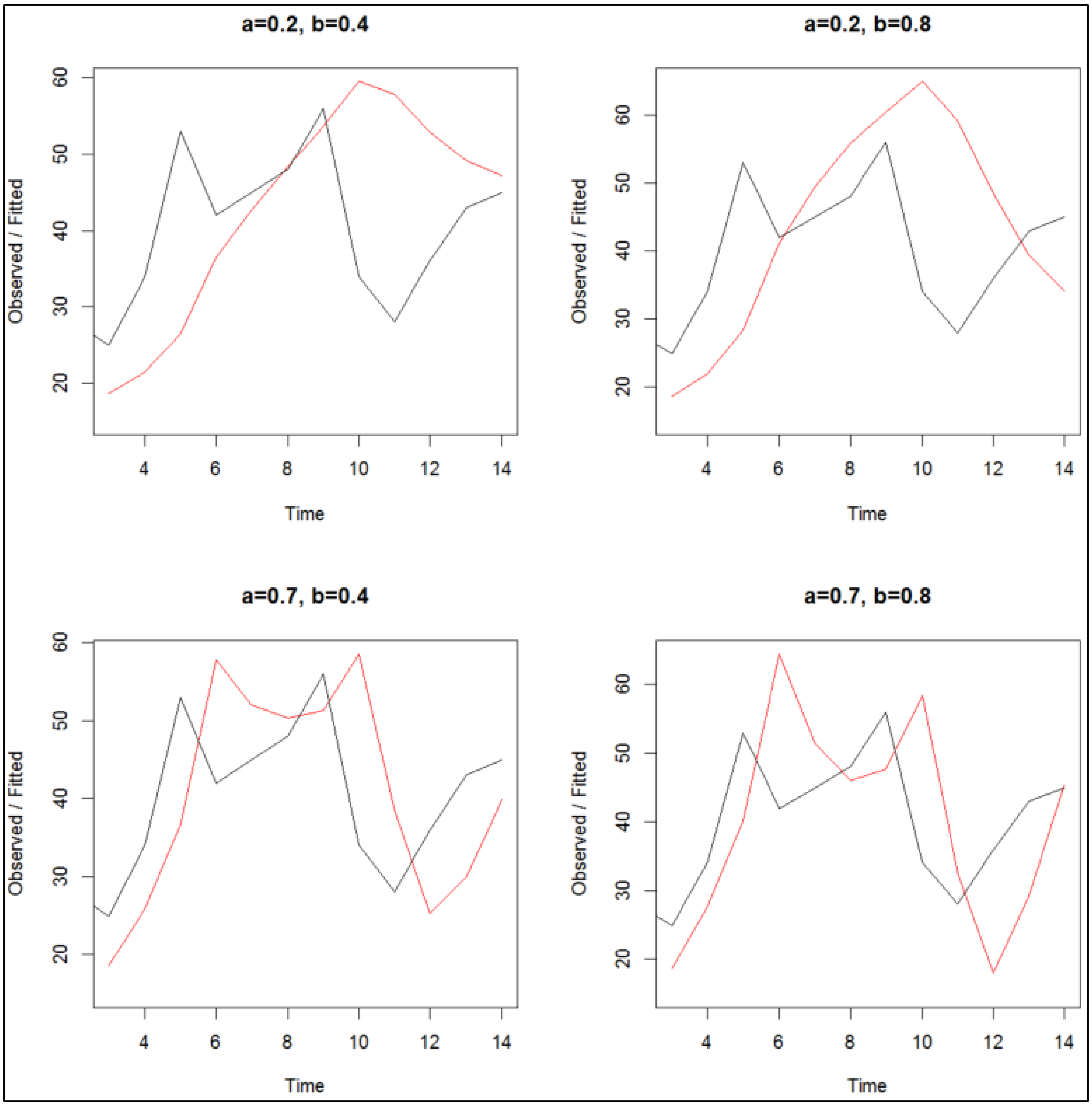


Figure 22 Comparison of Alpha and Beta

Notice that the larger the smoothing parameter, the more responsive the forecast is to changes. The smaller parameters force the model to be less responsive, but more consistent over time. Allowing R to estimate alpha and beta for us provides the values 0.96 and 0.0 respectively. The value of 0 for beta may at first appear puzzling, but looking at Figure 22 it would appear that the data contains several trends. The first trend is positive starting at Period 1 and ending at Period 9. The below figures present the forecasted data, assessment of homoscedasticity, and normally distributed residuals. The residuals look better using this model compared to the Exponential Smoothing model provided in the previous section.

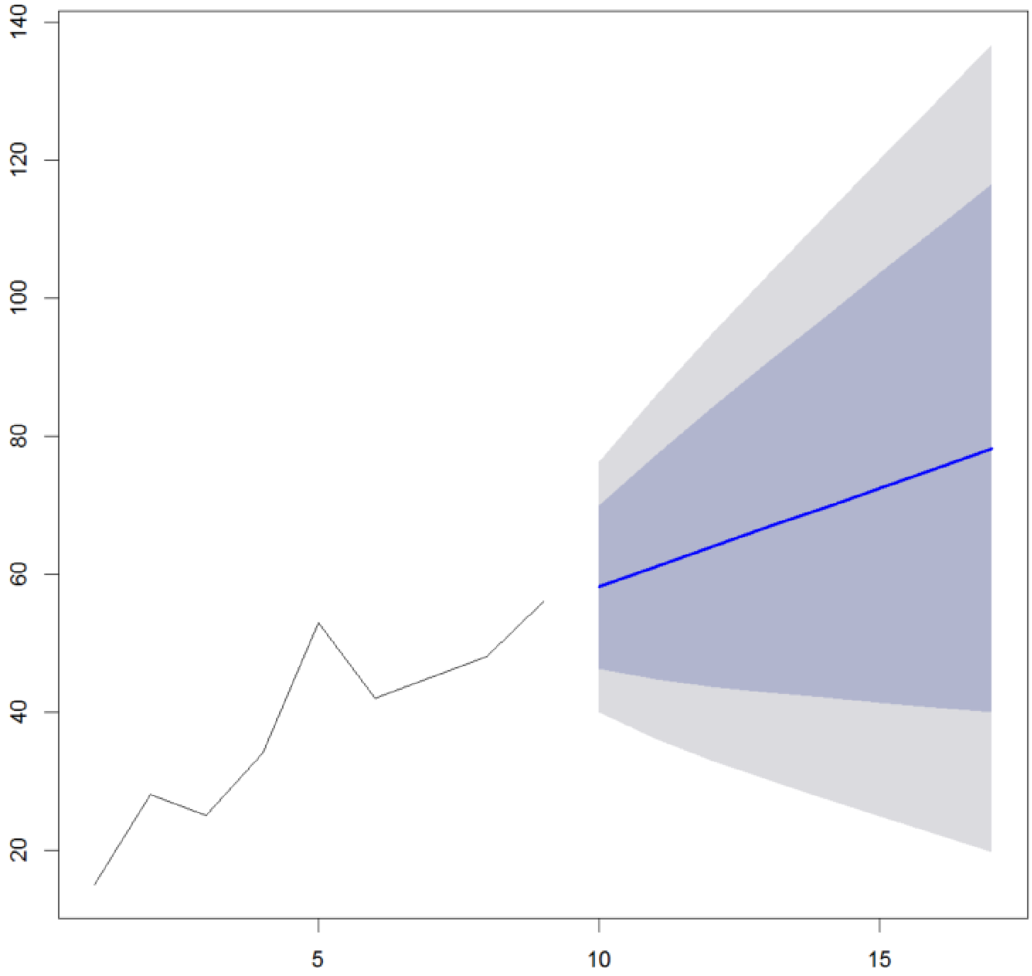


Figure 23 Forecast for Periods 1 - 9

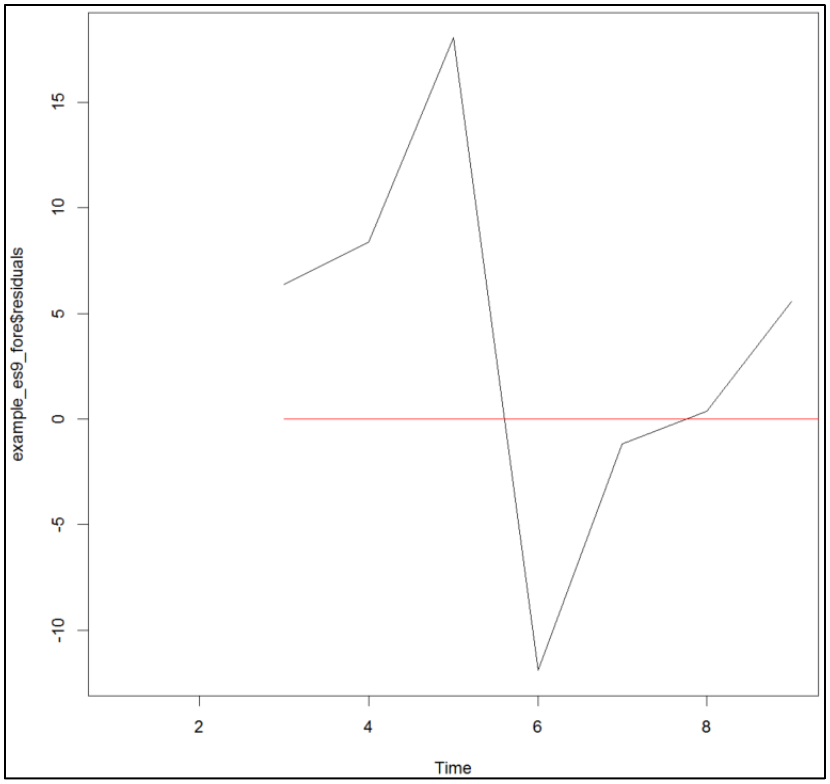


Figure 24 Assessing Constant Variance

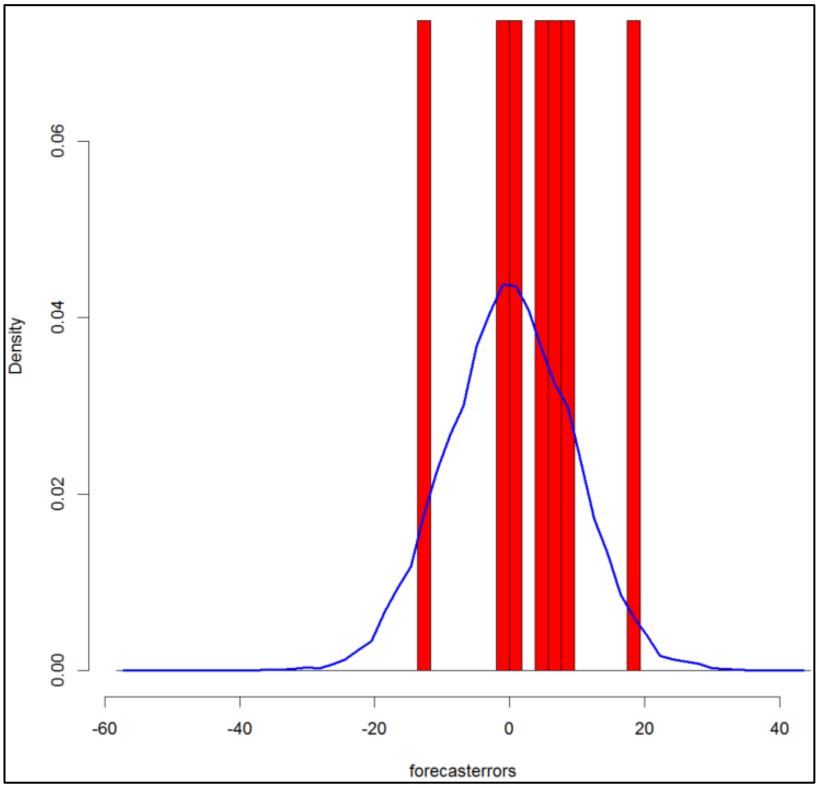
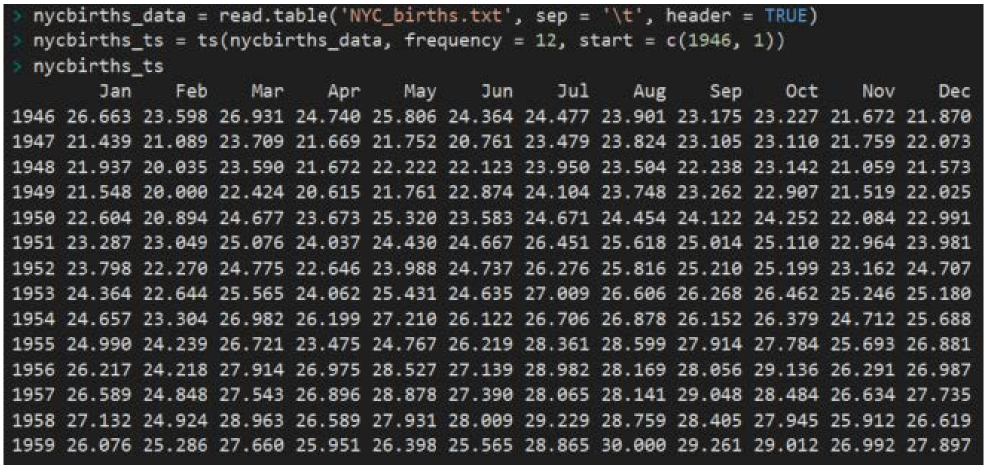


Figure 25 Distribution of Residuals

4. Components of Time Series: Seasonality

Seasonality is the second component of time series models and refers to the pattern of oscillations over time. These seasonal effects can be either additive or multiplicative. In this tutorial Holt-Winters Exponential Smoothing is used to assess time series data with a seasonal component that is additive. Holt-Winters uses three smoothing parameters to forecast: alpha for leveling, beta for slope, and gamma the seasonal component. Like alpha, a larger gamma indicates recent effects have a stronger influence on the model compared to smaller gamma values which indicate prolonged effects.



```
> nycbirths_data = read.table('NYC_births.txt', sep = '\t', header = TRUE)
> nycbirths_ts = ts(nycbirths_data, frequency = 12, start = c(1946, 1))
> nycbirths_ts
        Jan    Feb    Mar    Apr    May    Jun    Jul    Aug    Sep    Oct    Nov    Dec
1946 26.663 23.598 26.931 24.740 25.806 24.364 24.477 23.901 23.175 23.227 21.672 21.870
1947 21.439 21.089 23.709 21.669 21.752 20.761 23.479 23.824 23.105 23.110 21.759 22.073
1948 21.937 20.035 23.590 21.672 22.222 22.123 23.950 23.504 22.238 23.142 21.059 21.573
1949 21.548 20.000 22.424 20.615 21.761 22.874 24.104 23.748 23.262 22.907 21.519 22.025
1950 22.604 20.894 24.677 23.673 25.320 23.583 24.671 24.454 24.122 24.252 22.084 22.991
1951 23.287 23.049 25.076 24.037 24.430 24.667 26.451 25.618 25.014 25.110 22.964 23.981
1952 23.798 22.270 24.775 22.646 23.988 24.737 26.276 25.816 25.210 25.199 23.162 24.707
1953 24.364 22.644 25.565 24.062 25.431 24.635 27.009 26.606 26.268 26.462 25.246 25.180
1954 24.657 23.304 26.982 26.199 27.210 26.122 26.706 26.878 26.152 26.379 24.712 25.688
1955 24.990 24.239 26.721 23.475 24.767 26.219 28.361 28.599 27.914 27.784 25.693 26.881
1956 26.217 24.218 27.914 26.975 28.527 27.139 28.982 28.169 28.056 29.136 26.291 26.987
1957 26.589 24.848 27.543 26.896 28.878 27.390 28.065 28.141 29.048 28.484 26.634 27.735
1958 27.132 24.924 28.963 26.589 27.931 28.009 29.229 28.759 28.405 27.945 25.912 26.619
1959 26.076 25.286 27.660 25.951 26.398 25.565 28.865 30.000 29.261 29.012 26.992 27.897
```

Figure 26 New York City Births per Month

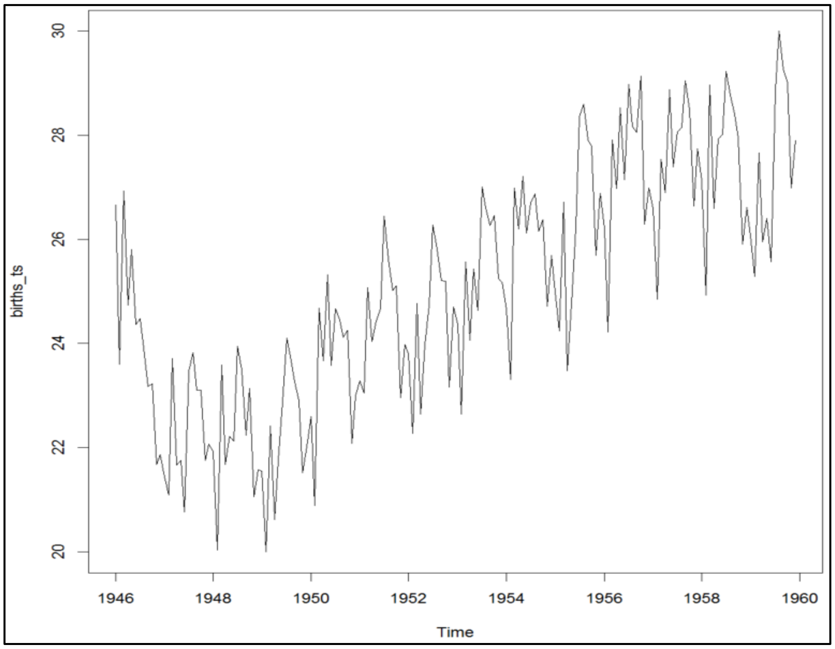


Figure 27 NYC Births per Month Plotted Over Time

The data comes from data collected from 1946 to 1959. This data shows the number of births per month in New York City collected for a research study. During the summer months, an increase in births occur. The data is indicative of seasonality because regular, predictable fluctuations appear every single year. Seasonal components do not have to be limited to annual fluctuations, but can occur every month, week, or even every second.

The plot illustrates the three main components of a time series: trend, season, and irregularities. To break the time series into separate components, use the function decompose(). Figure 28 presents the output using the plot() function. Notice how each component is separated out from the main time series, with the original at the top. Observe how each component matches up with the actual data.

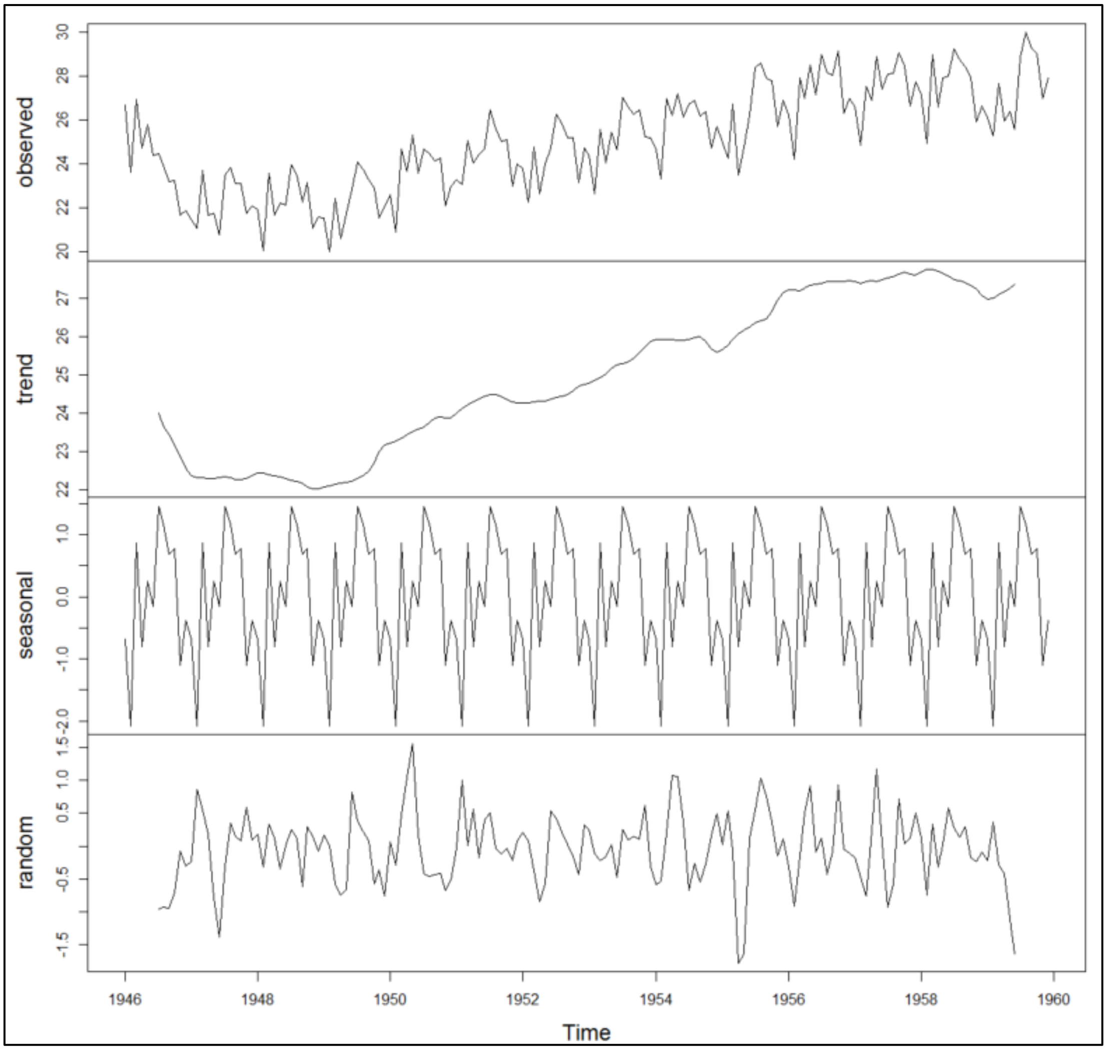
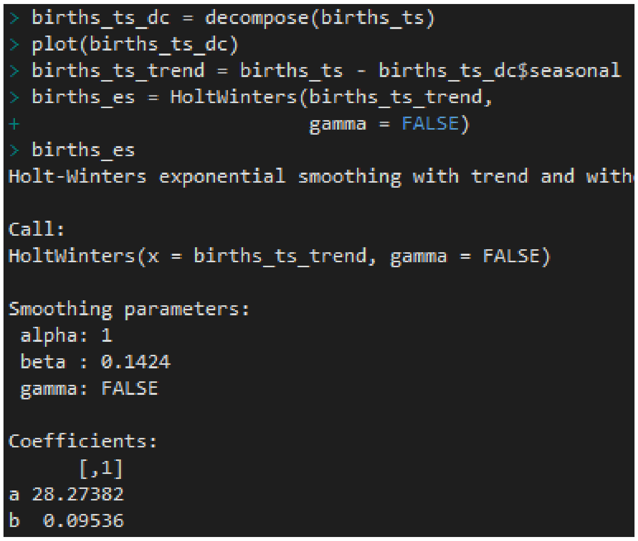


Figure 28 Decomposition of Additive Model: NYC Birth Data

The irregularities, or random component, is noise that cannot be explained by trends or seasonal effects. The advantage of decomposing the time series is the various components can be subtracted out of the time series model. For example, because this data contains a seasonal component, it cannot be modeled with Trend-Adjusted Exponential Smoothing. By removing the seasonal component, the data can then be run through Trend-Adjusted Exponential Smoothing. The figure below illustrates this. The alpha is estimated to be 1.00 with a beta of 0.14.



```
> births_ts_dc = decompose(births_ts)
> plot(births_ts_dc)
> births_ts_trend = births_ts - births_ts_dc$seasonal
> births_es = HoltWinters(births_ts_trend,
+                         gamma = FALSE)
> births_es
Holt-Winters exponential smoothing with trend and with

Call:
HoltWinters(x = births_ts_trend, gamma = FALSE)

Smoothing parameters:
 alpha: 1
 beta : 0.1424
 gamma: FALSE

Coefficients:
      [,1]
a 28.27382
b  0.09536
```

Figure 29 NYC Birth Without Season

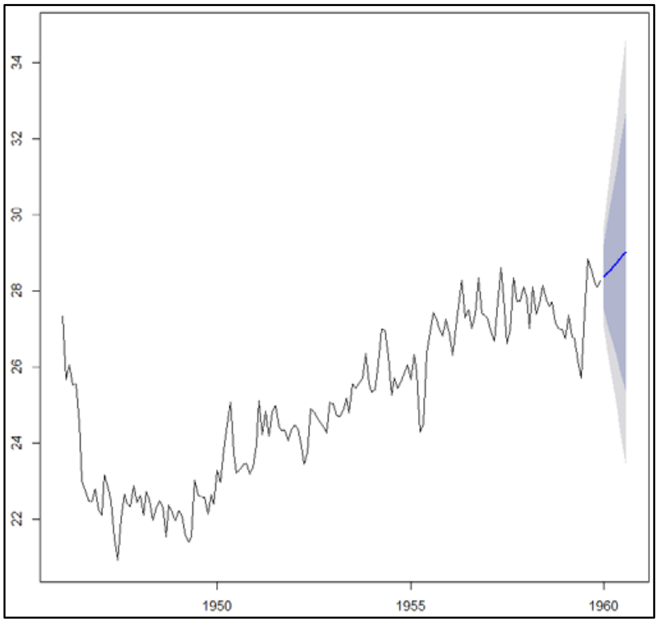


Figure 30Birth Forecast for 8 Periods

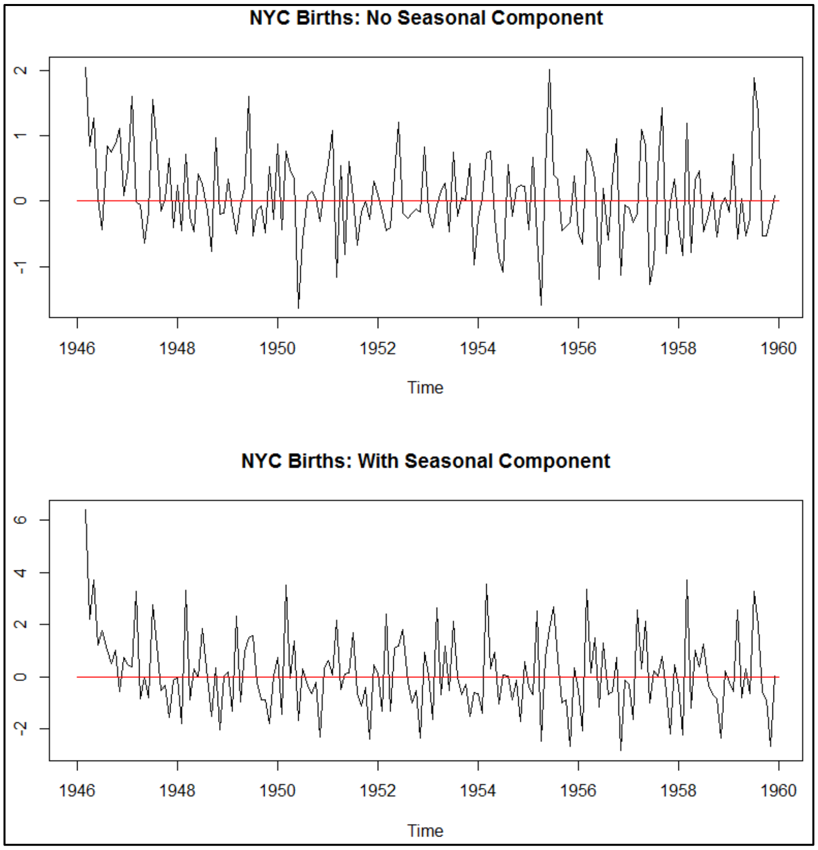


Figure 31 Homoscedasticity Assessment

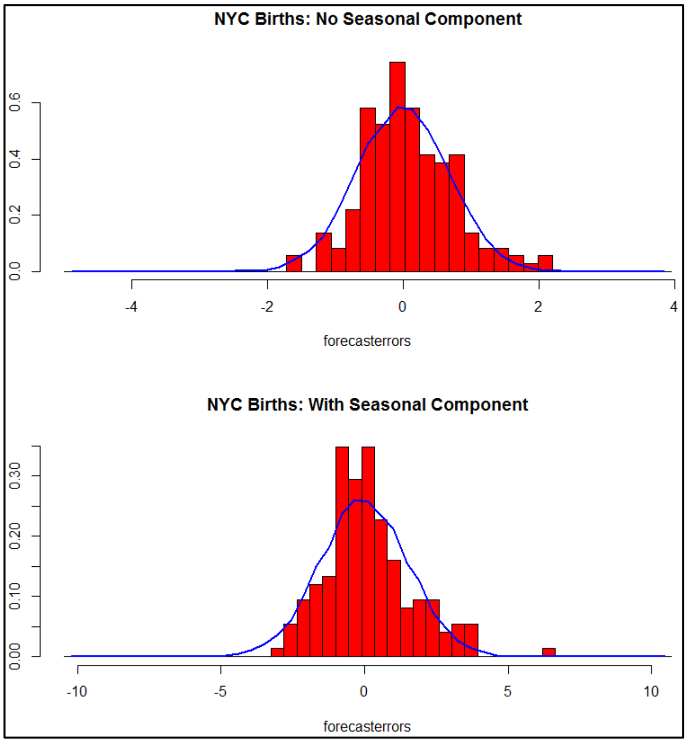
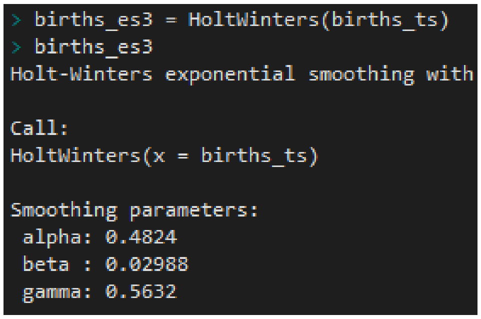


Figure 32 Assessment of Normal Distribution

Figures 31 and 32 present assessments of constant variance and normality. Each figure presents the data with the seasonal component removed and the data with the seasonal component intact. Removing the seasonal component did improve the model. Keep in mind, the Trend-Adjusted Exponential Smoothing does not assess seasons, so lots of information was lost.



```
> births_es3 = HoltWinters(births_ts)
> births_es3
Holt-Winters exponential smoothing with

Call:
HoltWinters(x = births_ts)

Smoothing parameters:
 alpha: 0.4824
 beta : 0.02988
 gamma: 0.5632
```

Figure 33 Birth Data in Holt-Winters

Looking at Figure 33, the first difference noticed is the change in alpha and beta. Alpha has decreased from 1.0 to 0.48; this means the model is less responsive to recent effects, but still considers them to some extent. Beta has also decreased down to 0.03, placing less emphasis on recent slopes, maintaining a slope equal to the initial slope. Gamma, like alpha, is in the middle and places emphasis on both recent and distant values. Figure 34 shows a forecast for the next 40 periods.

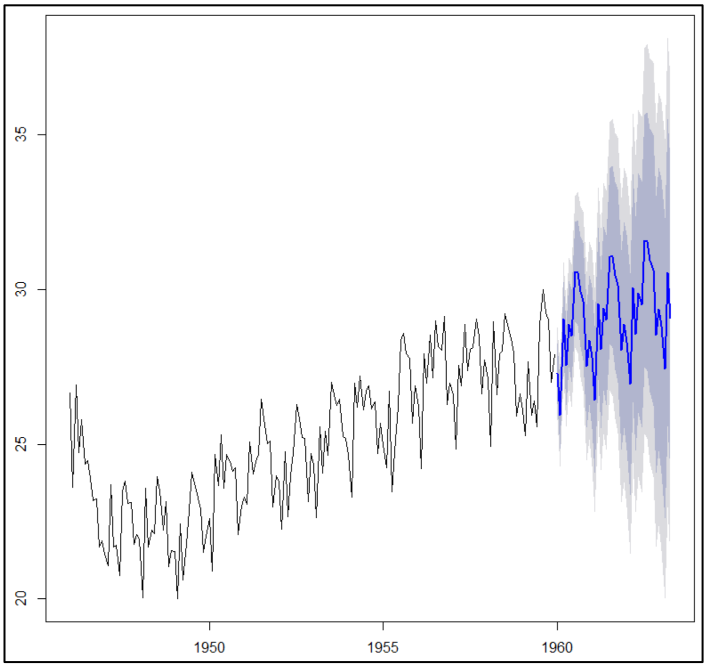


Figure 34 Forecast for 40 Periods

5. Additive vs. Multiplicative

All of the models discussed assume the data is additive and not multiplicative. While these two terms sound scary because of their “mathiness”, they are in fact simple concepts to understand. The characteristic that determines whether a time series is additive or multiplicative is the change of the amplitude over time.

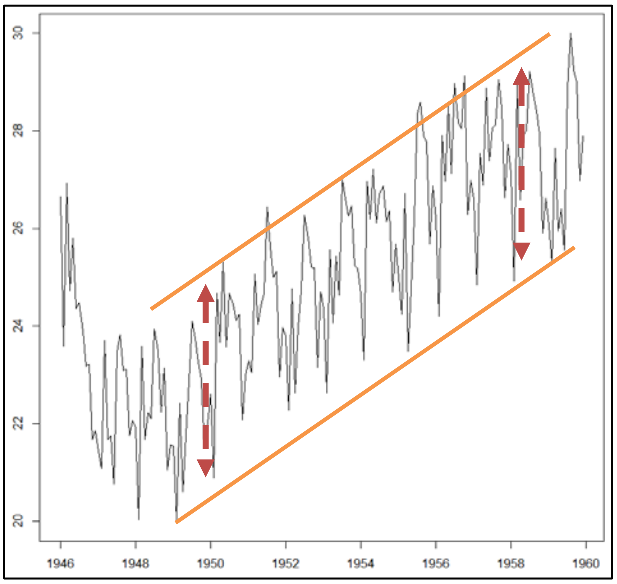


Figure 35 NYC Birth Data: Additive Data

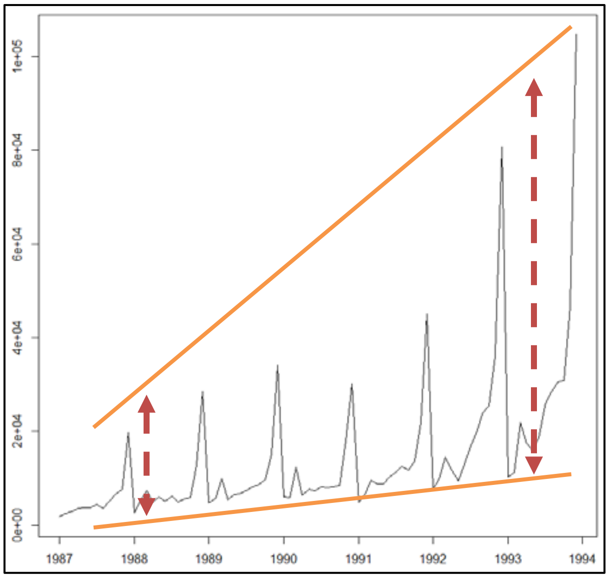
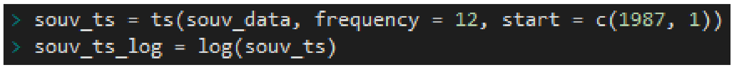


Figure 36 Souvenir Sales: Multiplicative Data

Look at Figure 35. The orange lines show the amplitude is consistent over time. This means the amplitude, measured from peak to peak, is approximately the same over time, as shown by the red, dashed arrows.

Figure 26 presents a different scenario. This data is the sales of souvenirs at a shop in Queensland, Australia. Notice the amplitude changes drastically from the start of the time series to the end. The red, dashed arrows reveal the amplitude’s have varying distance.

The exponential smoothing models presented in this tutorial cannot handle multiplicative data. In order to use any of the presented models with multiplicative data, the data must be transformed into additive. Many transformations can be used including log-based, inverse, or squared. The code snippet in Figure 37 shows how to apply a log transformation on the souvenir data. Figure 38 presents a plot of the log-transformed souvenir data. Comparing it to Figure 36, the amplitudes are more uniform throughout the time series.



```
> souv_ts = ts(souv_data, frequency = 12, start = c(1987, 1))
> souv_ts_log = log(souv_ts)
```

Figure 37 Applying Log Transformation

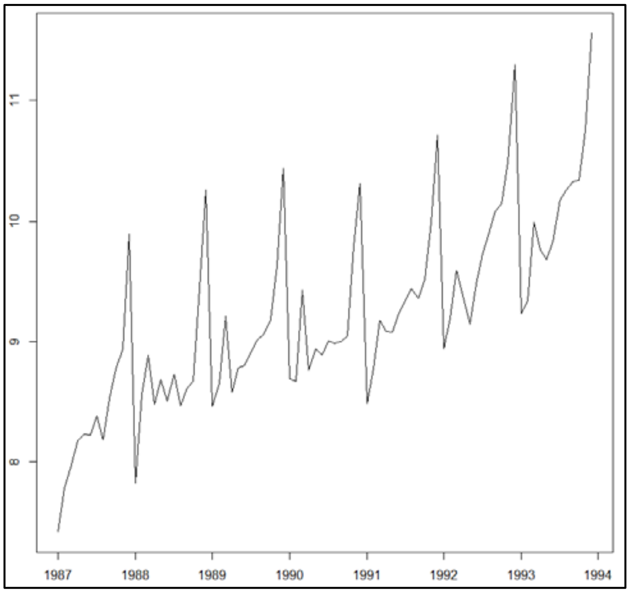


Figure 38 Plot of Log-Transformed Data

6. Correlations in Time Series

The last concept in time series is that of autocorrelation. The idea behind autocorrelation is that each data point may be influenced by a previous data point. That is, the event is correlated with an event in the past. Exponential Smoothing models assume there is no correlation in the data. Often, however, data do exhibit autocorrelation.

When speaking about correlated time periods, the term “lag” is used. If the correlated time periods are successive, one right after the other, then the lag is 1. If the correlation occurs two time periods apart, then it is a lag of 2.

Many phenomena exhibit autocorrelation. One of the best examples include stock prices. Stock prices are heavily influenced by previous evaluations. If the price was high a few days ago with an increasing trend, then the current price will more than likely be higher; if the price was decreasing, then the current price will be lower. Another example includes fly populations. The population of the past and the proliferation of breeding heavily influences the future population of the flies.

Autoregressive Integrated Moving Average (ARIMA) models are a class of models designed to take advantage of the autocorrelation. Unfortunately, this tutorial will not cover ARIMA models or how to assess autocorrelation in your time series data (this tutorial is very long already). If you would like to become more expert in using time series models, it is important to learn more about these topics. Using exponential smoothing models on data with autocorrelation would be akin to using linear regression with non-linear data.