Forecasting with Exponential Smoothing Models

Nicholas Bradley

# Forecasting with Exponential Smoothing Models

This analysis will document forecasting with various exponential smoothing models and will cover the following topics:

* Forecasting with moving average models
* Forecasting approaches with smoothing models
* Tuning parameters for smoothing models

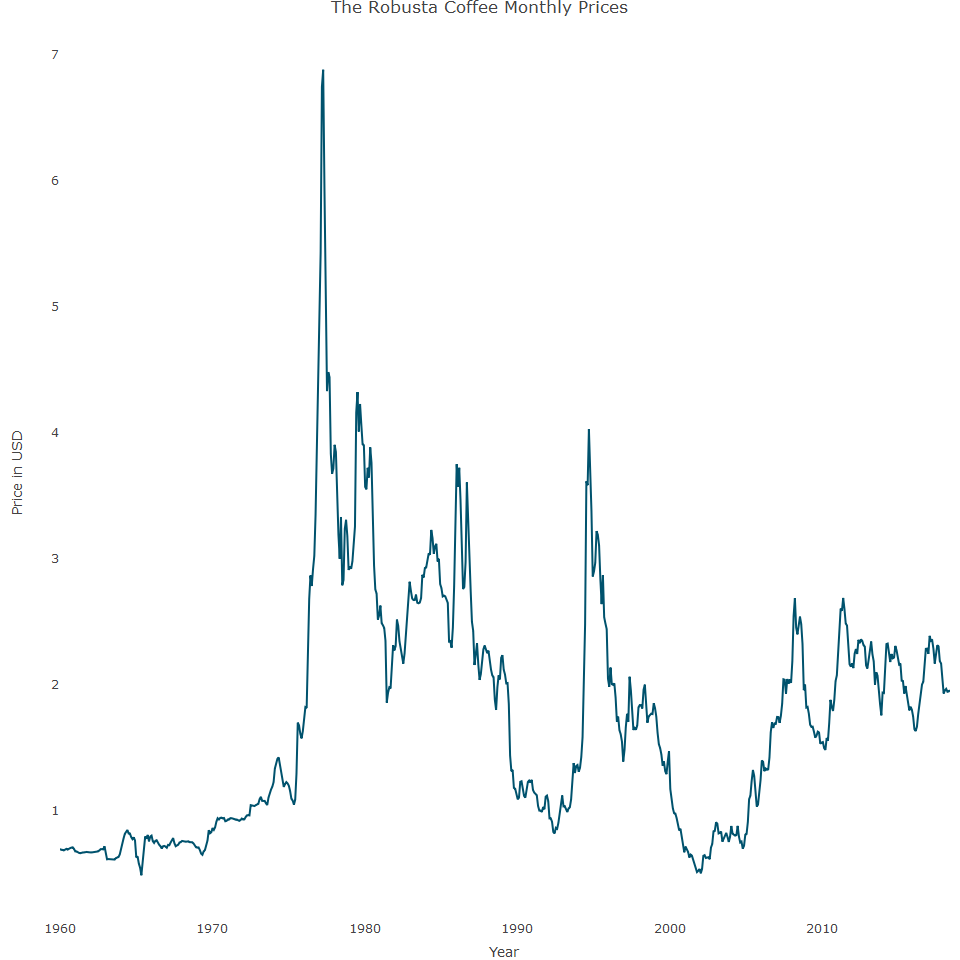
## Forecasting with moving average models

The first such model is the simple moving average and it calculates the average of a series of data points over a specified period. The algebraic formula for the SMA model can be represented as:

Where: - is the value of the moving average at time . - is the number of periods in the moving average. - is the value of the data point at time .

This formula computes the average of the most recent data points to obtain the current value of the moving average at time .

Forecasting with SMA is recommended when the input series has no structural patterns, such as trend and seasonal components. In this case, it is reasonable to assume that the forecasted values are relatively close to the last observations of the series. We can use data on coffee prices to create a customised SMA and forecast the coffee prices for the next 12 months.This series has no specific trend or seasonal patterns, but does have a cycle component, where the magnitude and length of the cycle keep changing from cycle to cycle. We can extract the monthly coffee prices



We can now create a SMA function

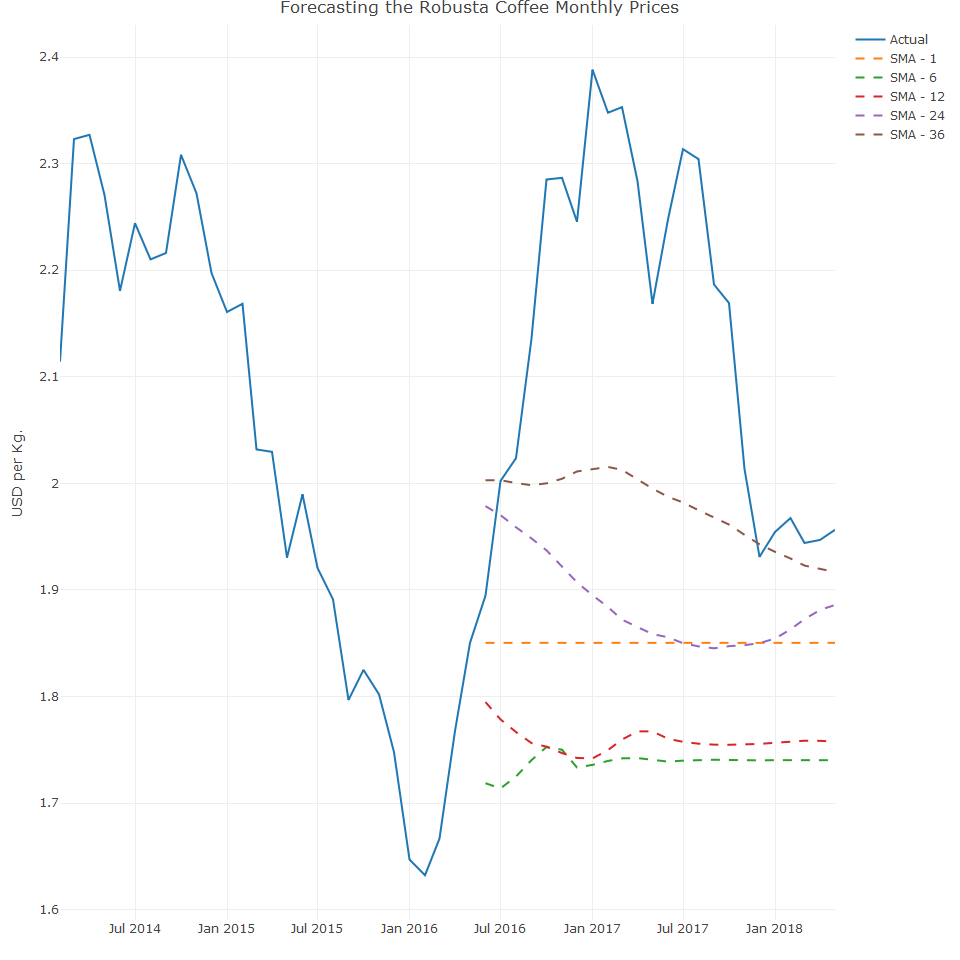
The main arguments in the function are:

* **df:** the input series in a two-column data frame format, where the first column is a Date object and the second one is the actual values of the series
* **h:** The horizon of the forecast.
* **m:** The length of the rolling window
* **w:** The weights of the average by default, using equal weights.

The sma\_forecast function has the following components

* **Error handling:** Test and veriy whether the input arguments of the function are valid. If one of the defined tests isn’t true, it will stop the function from running and trigger an error message
* **Data preparation:** This defines the data frame object based on the window length and forecast horizon.
* **Data calculation:** Calculates the simple moving average and returns the results

Lets use this function to forecast the last 24 months of the Robusta series using a rolling window of 3, 6, 12, 24 and 36 months:



The main observations from the plot are:

* If the length of the rolling window is shorter:
  + The range of the forecast is fairly close to the most recent observations of the series
* If the window length is longer:
  + The longer it takes until the forecast converges to some constant value
  + It can better handle shocks and outliers
* An SMA forecasting model with a rolling window of a length of 1 is equivalent to the naive forecasting model

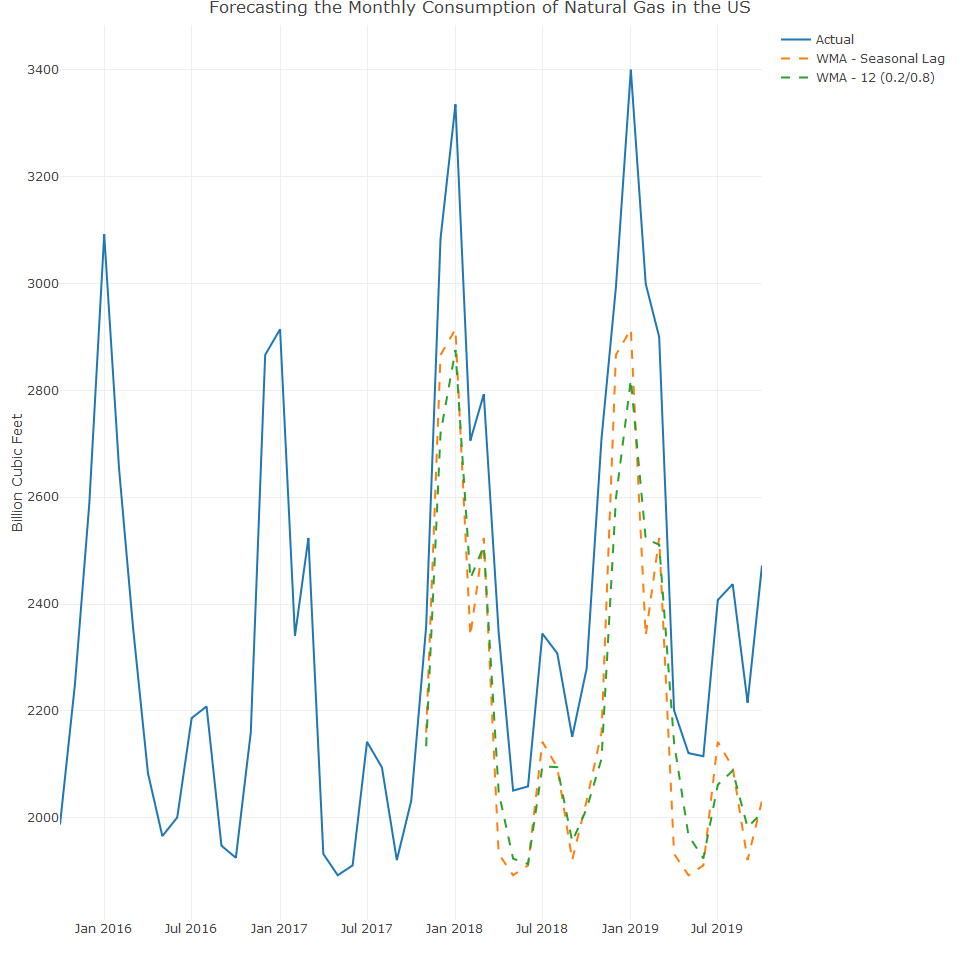
While the SMA function is fairly simple and cheap on computing power, it does have limitations:

* The forecasting power of the SMA function is limited to a short horizon and may have poor performance in the long run
* This method is for time series dat, with no trend or seasonal patterns. This mainly effects the arithmetic average that smooths the seasonal pattern and becomes flat in the long run.

## Weighted moving average

This is an extended version of the SMA and is based on use of the weighted average and the main advantage compared to SMA is that it enables the user to distribute the weight of the lags on the rolling window. This can be useful when the series has a high correlation with some of the lags. WMA provides more flexibility as it can handle time series with a seasonal pattern. In the following example, we use the sma\_forecast function to forecast the last 24 months of the US gas series. We utilise the w argument to set average weight and transform the SMA to WMA.

Let’s now plot the WMA models



Both models captured the seasonal oscillation to some extent. While it can capture seasonality, it cannot capture the series trend (due to the average effect) and therefore, will start to lose its effectiveness once the forecast horizon crosses the length of the series frequency (for example, more than a year for monthly series). The Holt Winter model can handle time series with both seasonal and trend components

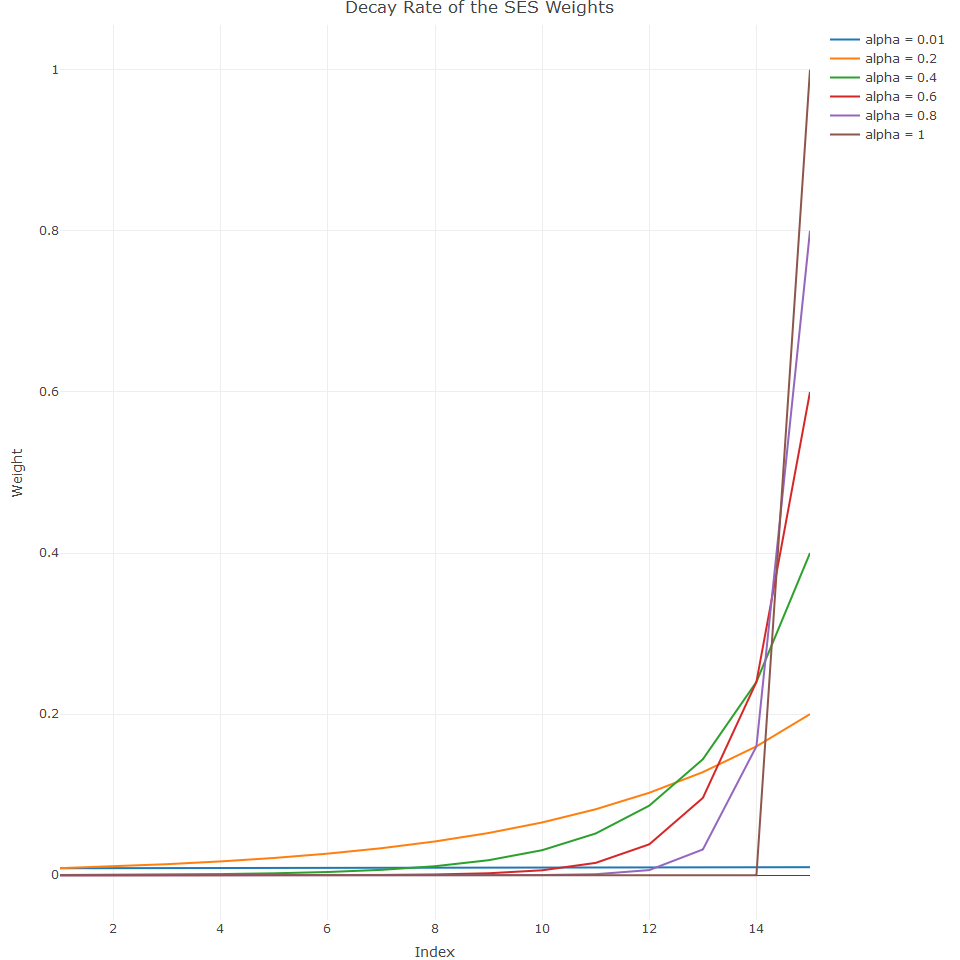
## Forecasting with exponential smoothing

These models are among the most popular forecasting methods and a key distinction with SMA models is that it takes a subset of observations rather than average all observations. Such modelling can also handle series with trend and seasonal components and the main model types are:

* Simple exponential smoothing model
* Holt model
* Holt Winters model

## Simple exponential smoothing model

The main assumption of the model is that the series stays at the same level (that is, the local mean of the series is constant) over time and therefore, the model is suitable for series with neither trend nor seasonal components. The SES model shares some of the attributes of the WMA model as both models forecast the future values by a weighted average of the past observations of the series. The main distinction is that the SES model is utilizing all the previous observations, whereas the WMA model is using only the most recent m observations (for a model with a rolling window of length m). The main attribute of the SES model is the weighted average, which is based on the exponential decay of the observations weights according to their chronological distance (that is, series index or timestamp) from the first forecasted values. The following example demonstrates the decay of the weights of the observations on the most recent 15 observations for values between 0.01 to 1:

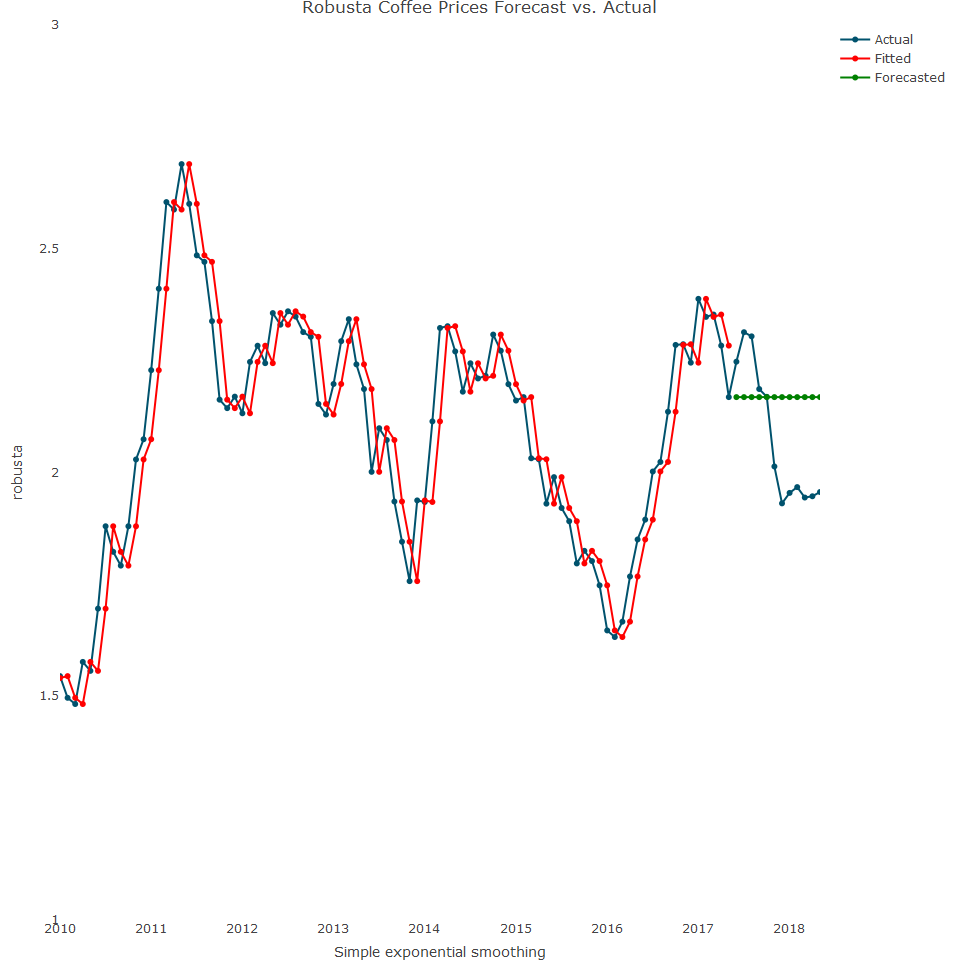


## Forecasting with the ses function

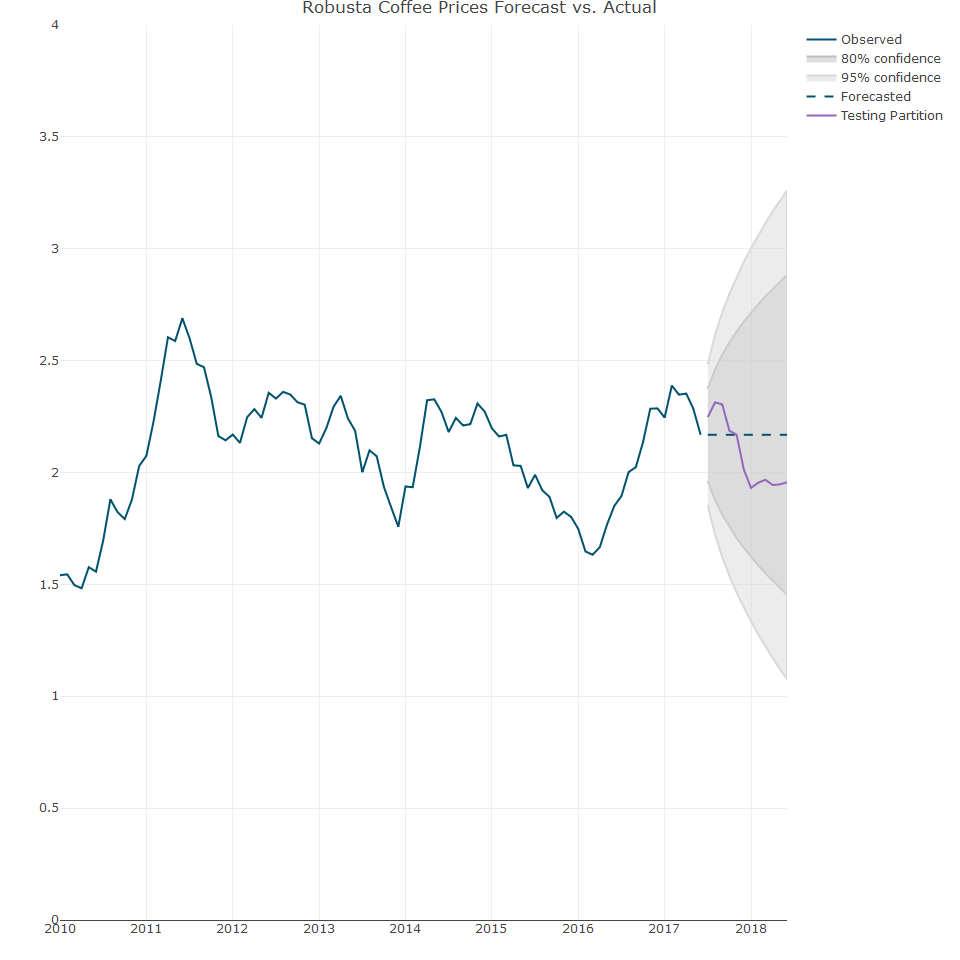
The main arguments of this function are:

* **initial:** Defines the method for initialising the value of , which can be calculated by using the first few observations of the series by setting the argument to simple, or estimating it with ets model (an advanced version of the Holt-Winters model) when setting it to optimal
* **alpha:** Defines the value of the smoothing parameter of the model. If set to NULL, the function will estimate it
* **h:** Sets the forecast horizon

Lets use the ses function to forecast the monthly prices of the Robusta coffee. We leave the last 12 months of the series as as testing set for benchmarking the model’s performance.



The forecast in the above plot is flat, and so confidence intervals play a critical role, since the level of uncertainty is higher. Therefore, we need to assess whether the forecast is within the model confidence intervals.



The forecast plot indicates that the forecast is within the 80% confidence interval.

## Model optimization with grid search

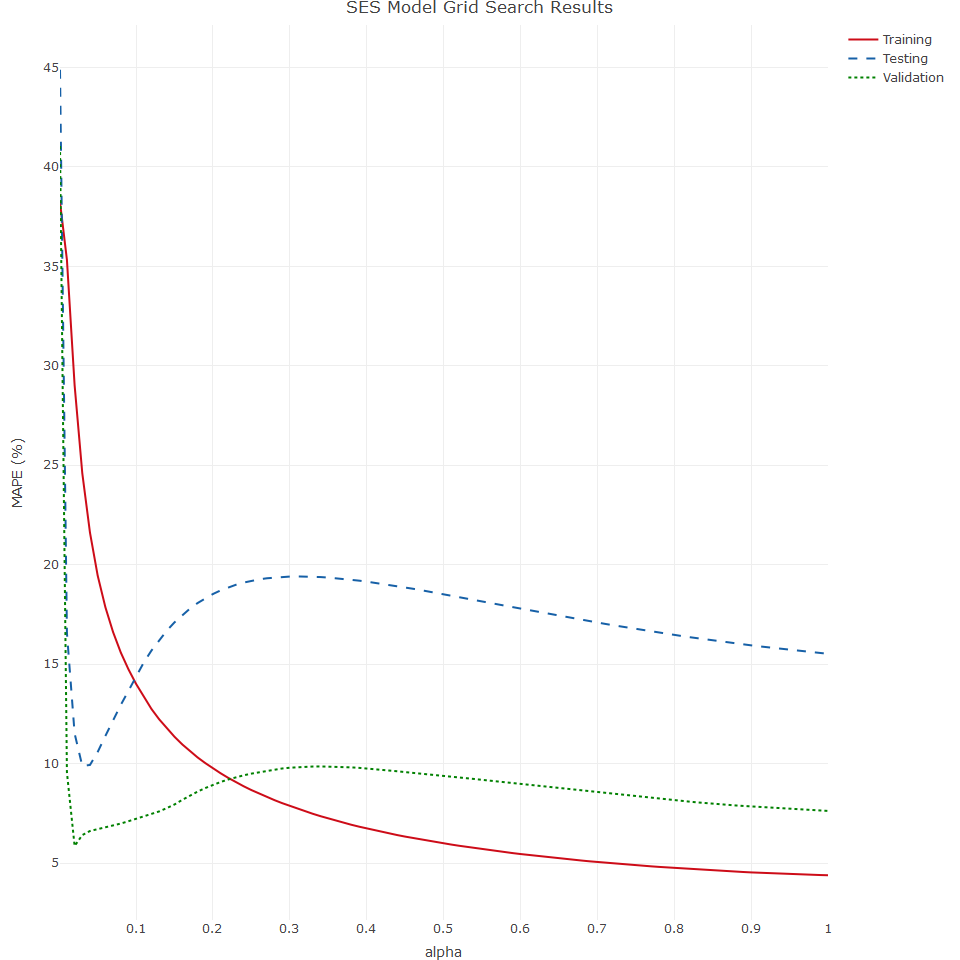
An alternative optimization is to use a grid search, a simple but powerful approach that is used to identify values of the model’s parameters that minimize model error. In the case of the SES model, we apply a grid search to identify the optimal value that minimizes some error metric of the model (MAPE< RMSE etc). In this example, we use a grid search to tune the model parameters, which minimise the MAPE for the Robusta coffee.

Before the function is created, we set the training, testing and validation partitions

The following alpha value defines the search range and we assign a range of values between 0 and 1 with an increment of 0.01

We can now create the grid

We can plot the grid search results



## Holt Method

The Holt method is a forecasting technique that extends exponential smoothing to capture both level and trend in a time series. It is defined by the following equations:

## Level Equation

## Trend Equation

## Forecast Equation

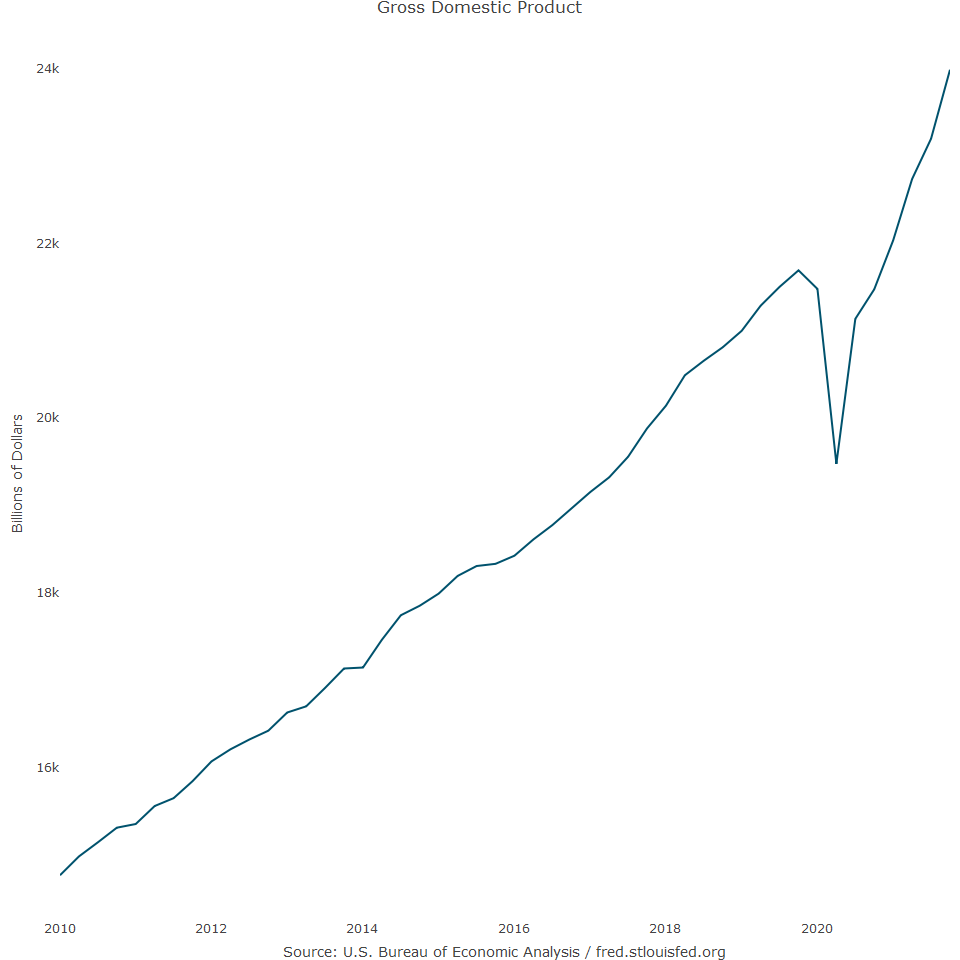
where: - is the level at time . - is the trend at time . - and are smoothing parameters (0 < < 1). - is the actual value at time . - is the forecasted value periods ahead.

## Forecasting with the holt method

We can use the US gross domestic product quarterly data to model the holt forecasting

## The gdp series is a ts object with 1 variable and 48 observations  
## Frequency: 4   
## Start time: 2010 1   
## End time: 2021 4

In the below plot, we can see that the GDP series has a strong linear trend and no seasonal component (since the series is seasonally adjusted)



We can now split the data into train and test data and we leave the last eight quarters for the testing data set. Once we have built the model, let’s check its parameters

## Holt's method   
##   
## Call:  
## holt(y = train, h = 8, initial = "optimal")   
##   
## Smoothing parameters:  
## alpha = 0.9985   
## beta = 0.0941   
##   
## Initial states:  
## l = 14603.0487   
## b = 161.9411   
##   
## sigma: 80.9869  
##   
## AIC AICc BIC   
## 504.8838 506.6485 513.3282

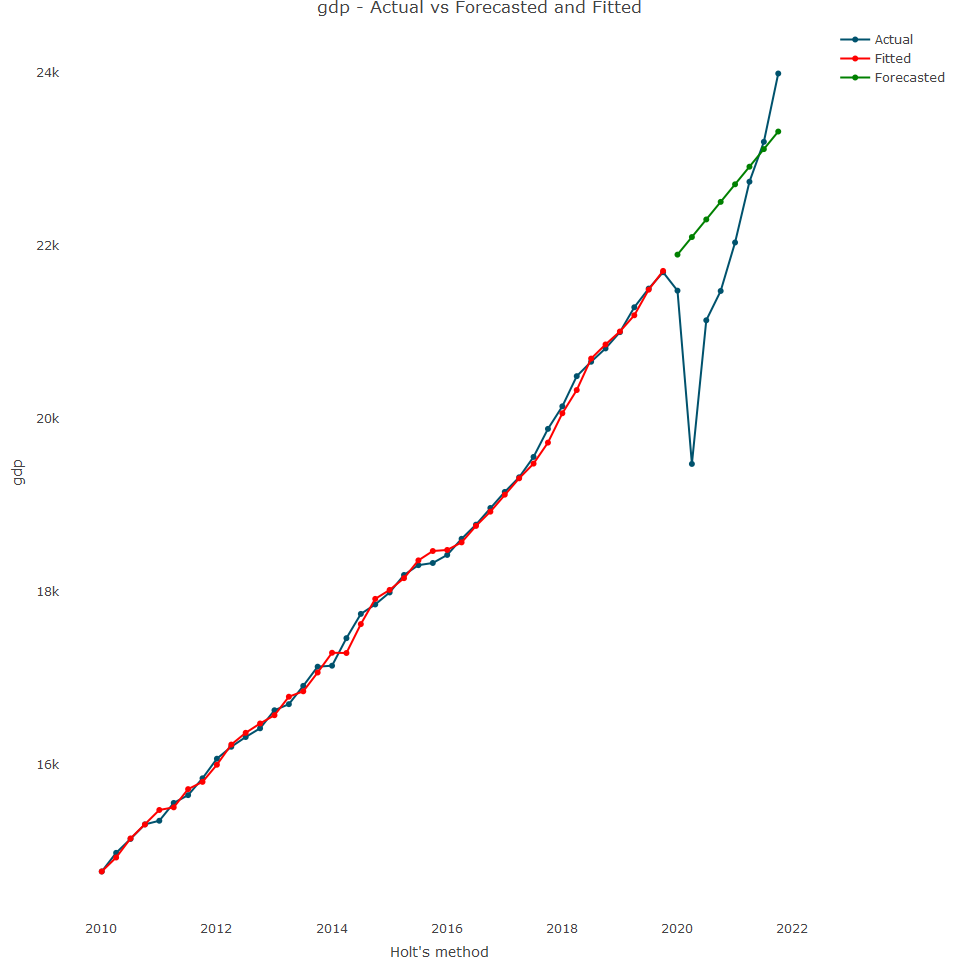
The initialized values of l and are close to the values of the first observation of the series (14721.35) and the average difference between each quarter. Additionally, the selected beta is close to zero, which indicates that updating the trend value from period to period doesn’t take into account the change in the level. Let’s compare the model’s performance in the training and test partitions:

## ME RMSE MAE MPE MAPE MASE  
## Training set 10.99164 76.83093 61.7626 0.05241543 0.3460477 0.08790283  
## Test set -665.87355 1141.19097 854.8069 -3.29699460 4.0874667 1.21659297  
## ACF1 Theil's U  
## Training set 0.0192117 NA  
## Test set 0.2975704 1.08055

As seen from this output, the ratio between the error rate on the testing and training set is more than 5 times for MSE and nearly 4 times for MAPE, which is caused mainly by he following two reasons:

* The fitted values of the model for the training set are not bound by a linear line (as opposed to the forecast output)
* The growth of the trend in the last few quarters shifted from a linear rate of growth to an exponential rate.

The changes in the trend growth and forecast can e observed with the test\_forecast function:



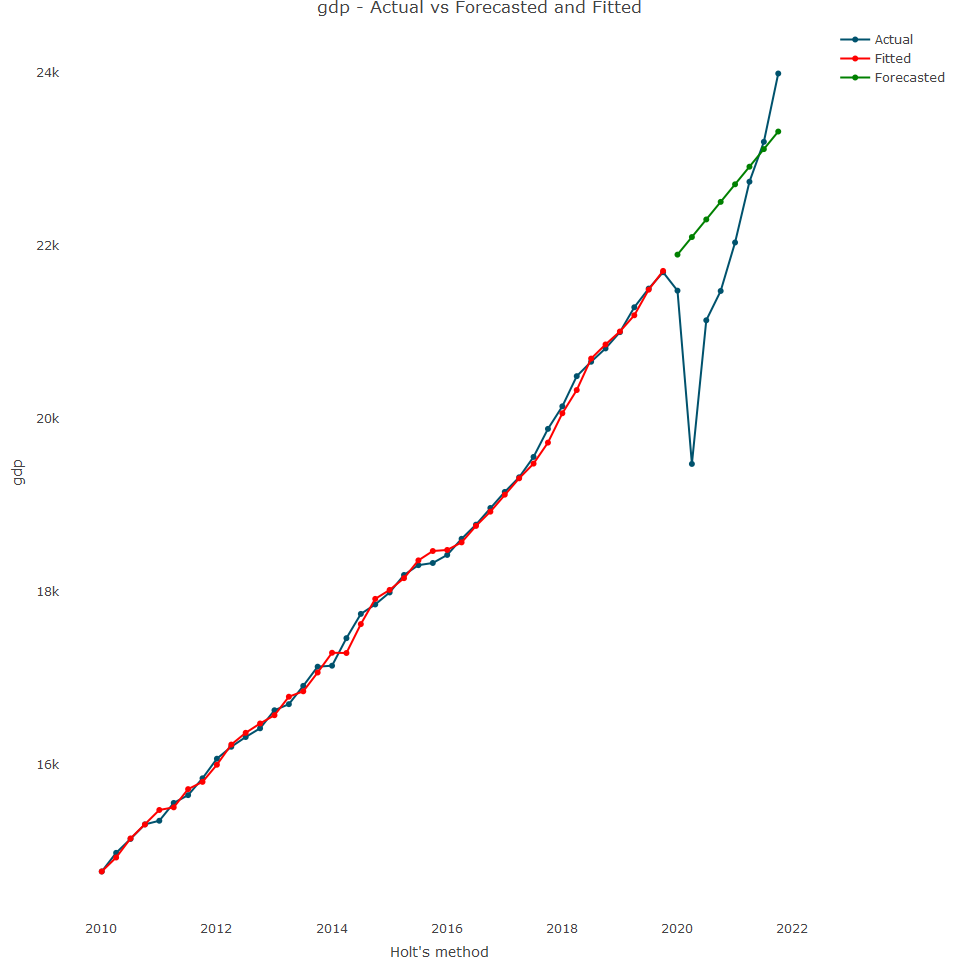
While the Holt model was designed to handle time series with the linear trend, the exponential argument in the holt function provides the option to handle series with exponential or decaying trends when set to TRUE. In the preceding example, we can utuilize the exponential argument to modify the growth pattern of the trend. In this case, we want a higher weight for the trend, so we set to 0.75 (a more robust means to identify would be a grid search):

## Holt's method with exponential trend   
##   
## Call:  
## holt(y = train, h = 8, initial = "optimal", exponential = TRUE,   
##   
## Call:  
## beta = 0.75)   
##   
## Smoothing parameters:  
## alpha = 0.781   
## beta = 0.75   
##   
## Initial states:  
## l = 14572.2291   
## b = 1.0142   
##   
## sigma: 0.0053  
##   
## AIC AICc BIC   
## 515.9291 517.0720 522.6847

Let’s review the accuracy of the training and testing set:

## ME RMSE MAE MPE MAPE MASE  
## Training set -2.792636 90.20753 74.14323 -0.01756593 0.4165413 0.1055234  
## Test set -703.014806 1151.21262 858.97402 -3.46212317 4.1130726 1.2225238  
## ACF1 Theil's U  
## Training set -0.1541725 NA  
## Test set 0.2834645 1.090637

There is still a large ratio difference between training and test for RMSE and MAPE. We can also plot it:

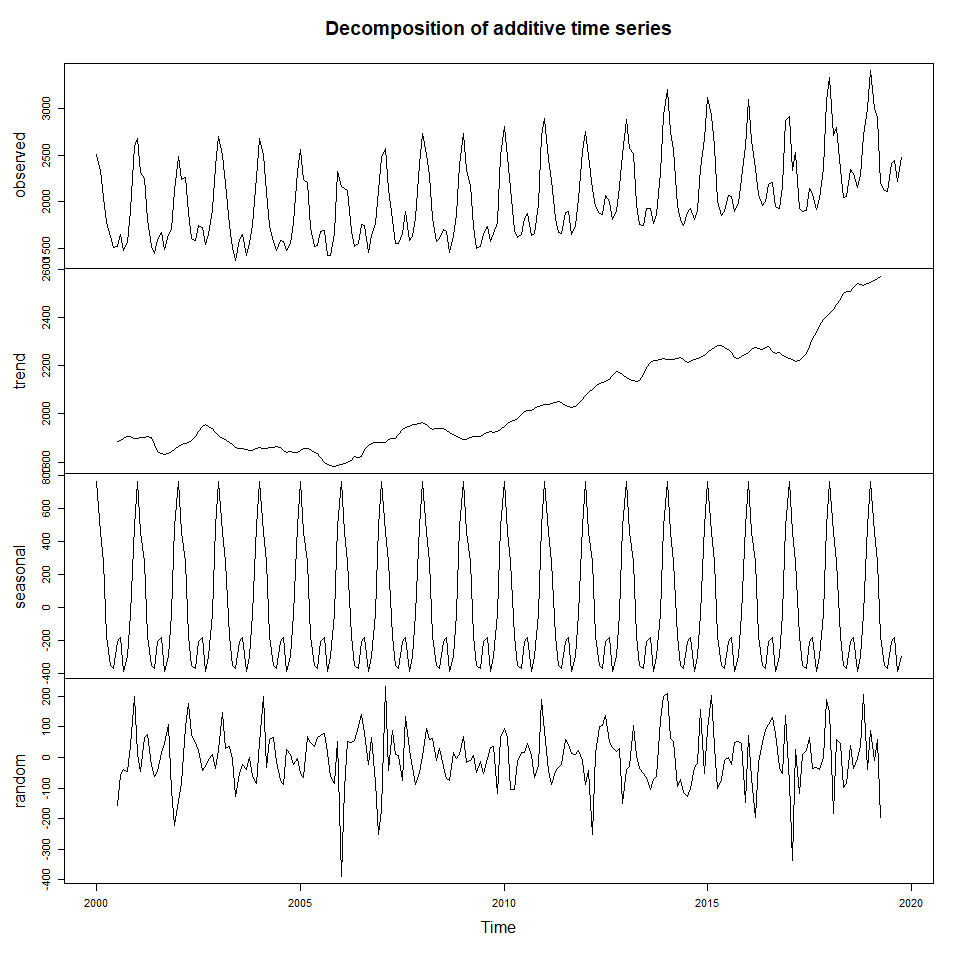


## Holt Winters Model

This model is an extended version of the Holt model and can handle time series data with both trend and season components. Forecasting the seasonal component requires a third smoother parameter and equation, in addition to the ones of the level and trend. Both of the trend and seasonal components could have either an additive or multiplicity structure, which adds some complexity to the model as there are multiple possible combinations:

* Additive trend and seaosonal components
* Additive trend and multiplicative seasonal components
* Multiplicative trend and additive seasonal components
* Multiplicative trend and seasonal components

Therefore, before building an HW model, we need to identify the structure of the trend and the seasonal components. We can use HW to forecast the last 12 months of the US gas series



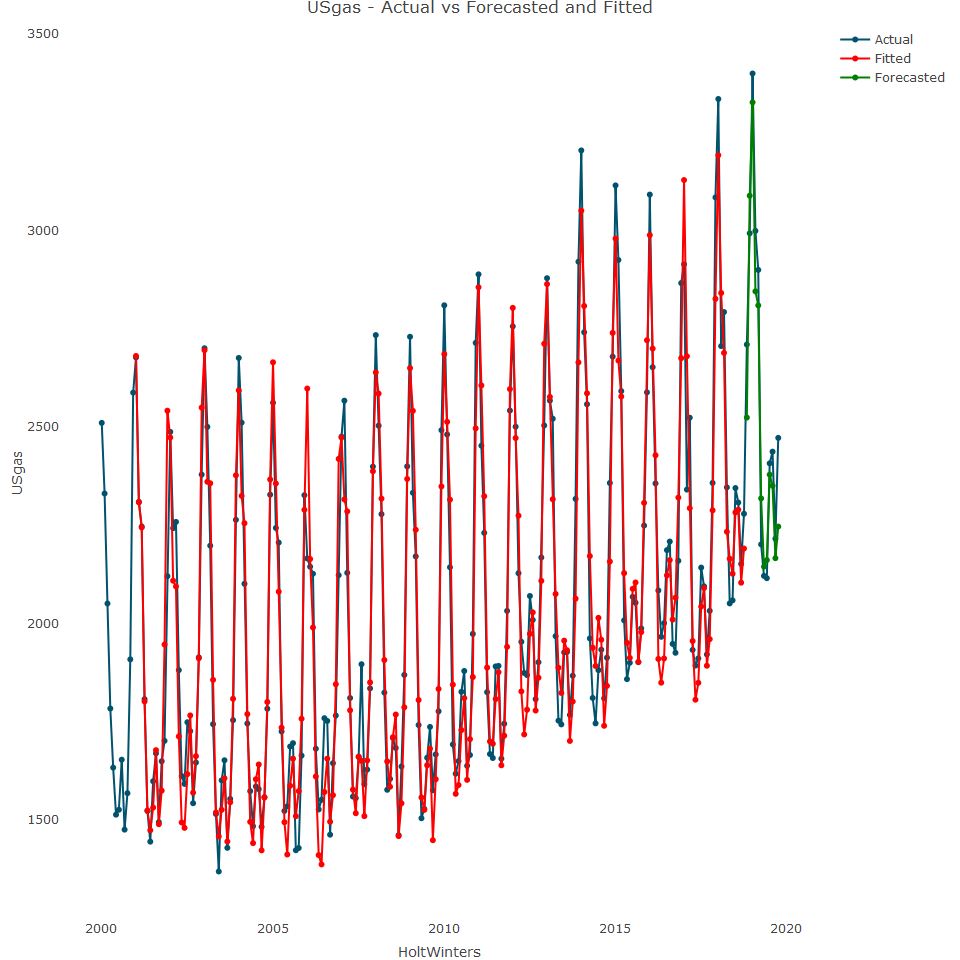
As seen in the preceeding plot, both the trend and seasonal components of the series have an additive structure. We can next create the training and test data sets to evaluate the model performance

## Holt-Winters exponential smoothing with trend and additive seasonal component.  
##   
## Call:  
## HoltWinters(x = train)  
##   
## Smoothing parameters:  
## alpha: 0.371213  
## beta : 0  
## gamma: 0.4422456  
##   
## Coefficients:  
## [,1]  
## a 2491.9930104  
## b -0.1287005  
## s1 32.0972651  
## s2 597.1088003  
## s3 834.9628439  
## s4 353.8593860  
## s5 318.1927058  
## s6 -173.0721496  
## s7 -346.6229990  
## s8 -329.7169608  
## s9 -112.1664217  
## s10 -140.3186476  
## s11 -324.5343787  
## s12 -243.9334551

The model output indicates that the model is learning from the level and seasonal update(with = 0.35 and = 0.44). However, there is no learning from the trend initialised = 0. The next step is to forecast the 12 months and evaluate the model’s performance.

## ME RMSE MAE MPE MAPE MASE  
## Training set 7.828361 115.2062 87.67420 0.2991952 4.248131 0.7614222  
## Test set 51.013877 115.1555 98.06531 1.7994297 3.766099 0.8516656  
## ACF1 Theil's U  
## Training set 0.21911103 NA  
## Test set -0.01991923 0.3652142

The accuracy metrics of the model are quite balanced, with a MAPE of 4.3% and 3.7% for training and test respectively.



As we can see in the plot, the HW model does a good job of capturing both the series seasonal patterns, but it misses the peak of the year during the month of January in most cases.