

# Forecasting 2016/2017 - Problem Set 1

*João Eira, Paola Di Nunzio, Esteban Miguélez*

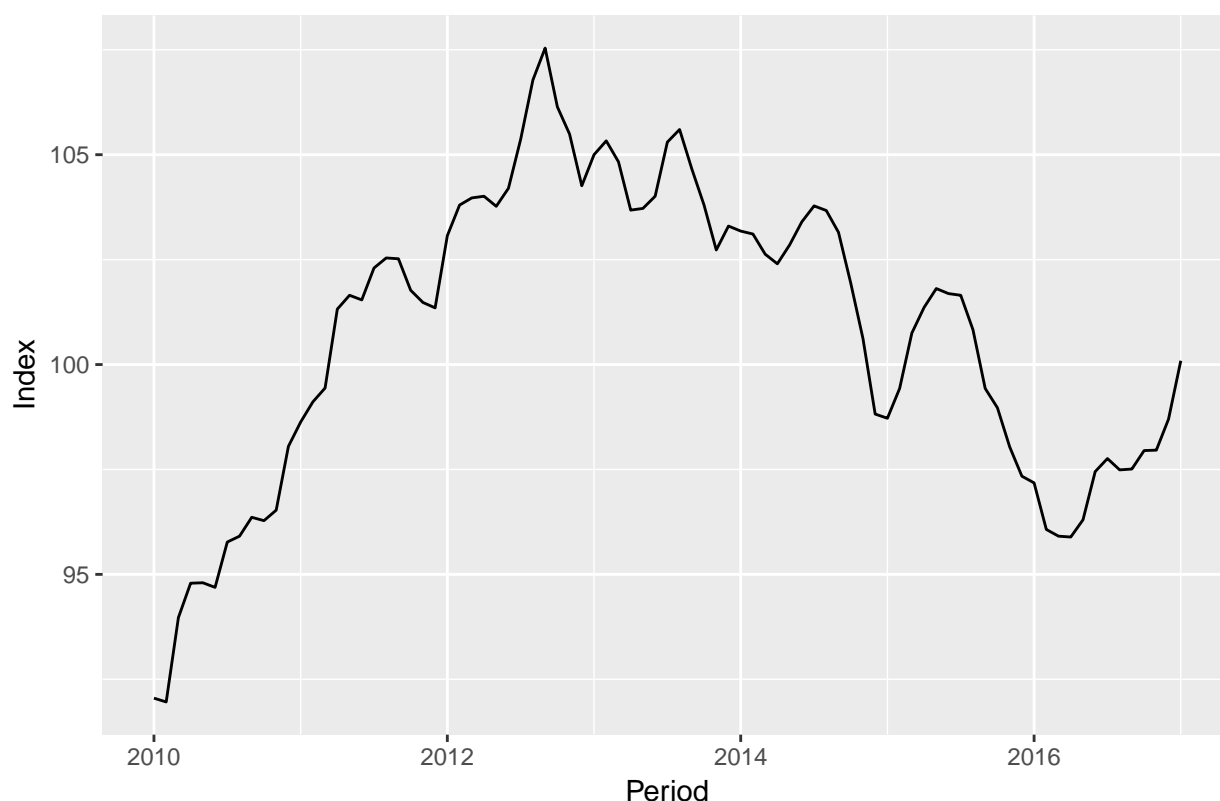
*March 2017*

The purpose of this report is to use the forecasting methods taught in class using time series data of Portugal's index of output prices, which was retrieved from Instituto Nacional de Estatística, to forecast likely future values of the index for the months of February, March, and April.

## Exploratory graphical analysis

In this first section we'll plot the available data in a variety of manners to try to understand what the data is telling us before proceeding to use any forecast method.

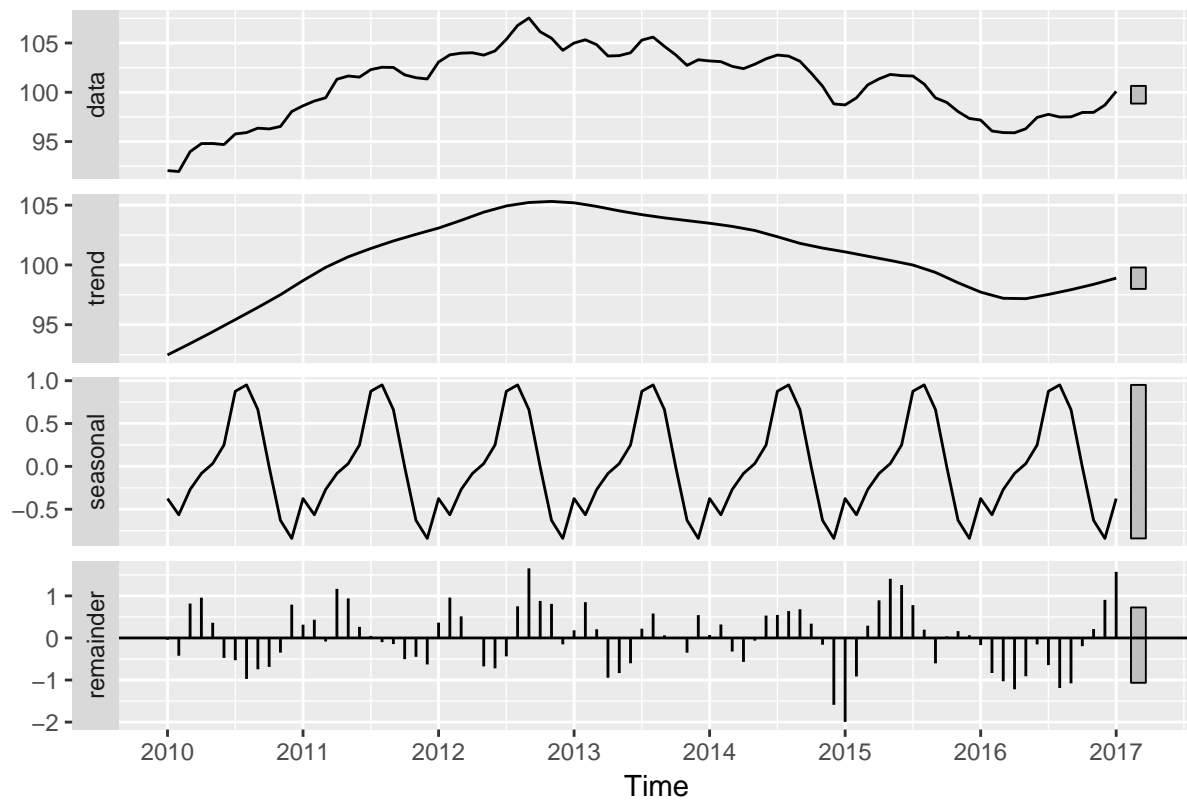
### Index of Output Prices 2010–2017



A cursory analysis of the plot above gives us indication of a structural shift happening around the middle of 2012 which inverted a previous upward trend. This new downward trend seems to have persisted until the first months of 2016 and it now seems to be trending upwards, though it is still unclear whether this apparent upward trend will be sustained.

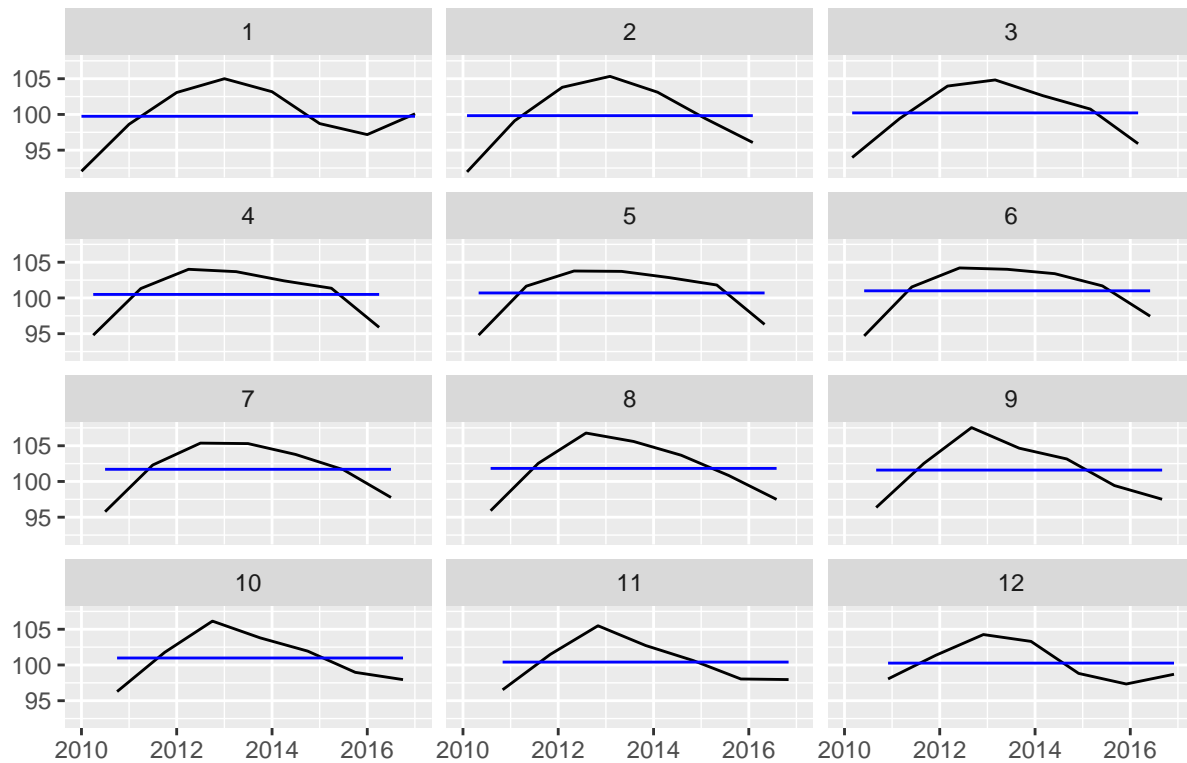
Decomposing the original time series into a trend and seasonal components allows us to better peer into its innards.

## Time Series Decomposition



As noticed before, there is a shift in the direction of the trend component around 2012. The upwards trend starting in 2016 noticed before does seem to be due to a change in the underlying trend of the series and not noise. Furthermore, the plot of the seasonal component provides evidence for seasonal variation in this time series, but of diminute importance when compared to the trend component. The time series also does not seem to have multiplicative seasonality.

## Monthly Visualization of Time Series Data



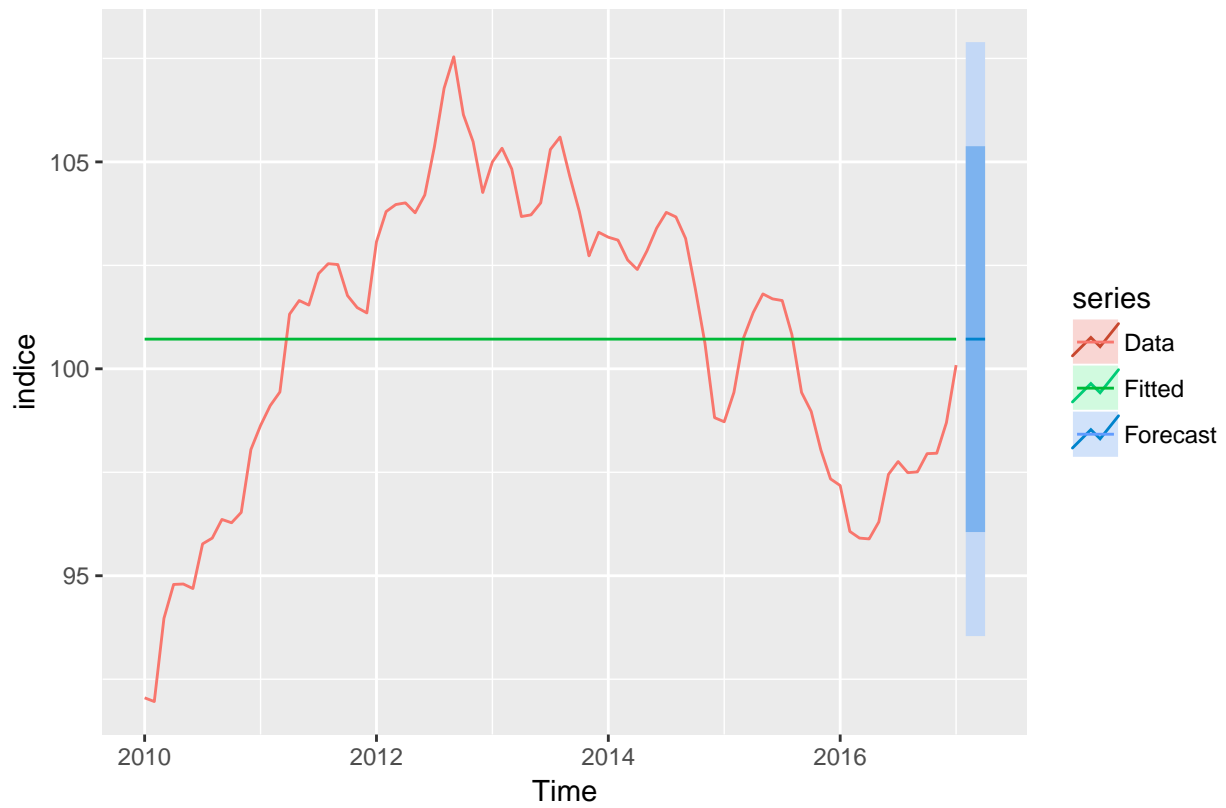
A plot of the yearly values for each month doesn't seem to provide more information about the time series than the decomposition above. Again there's an inflection point in 2012, and a seemingly change upwards starting in 2016, plus a seasonal component that is not much noticeable.

## Forecasts

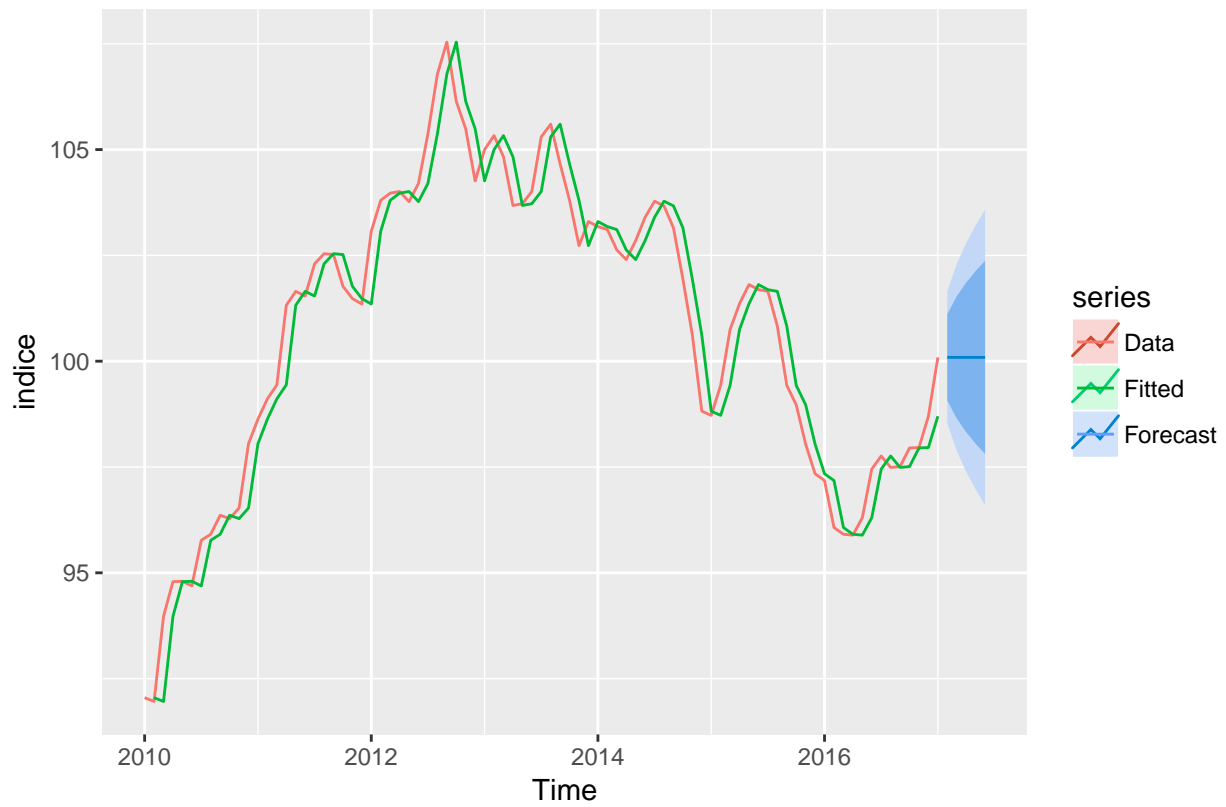
### Simple forecast methods

We'll start our use of the forecast methods introduced in class by trying out the simplest methods available: Mean, Naive, Naive Seasonal, and Drift.

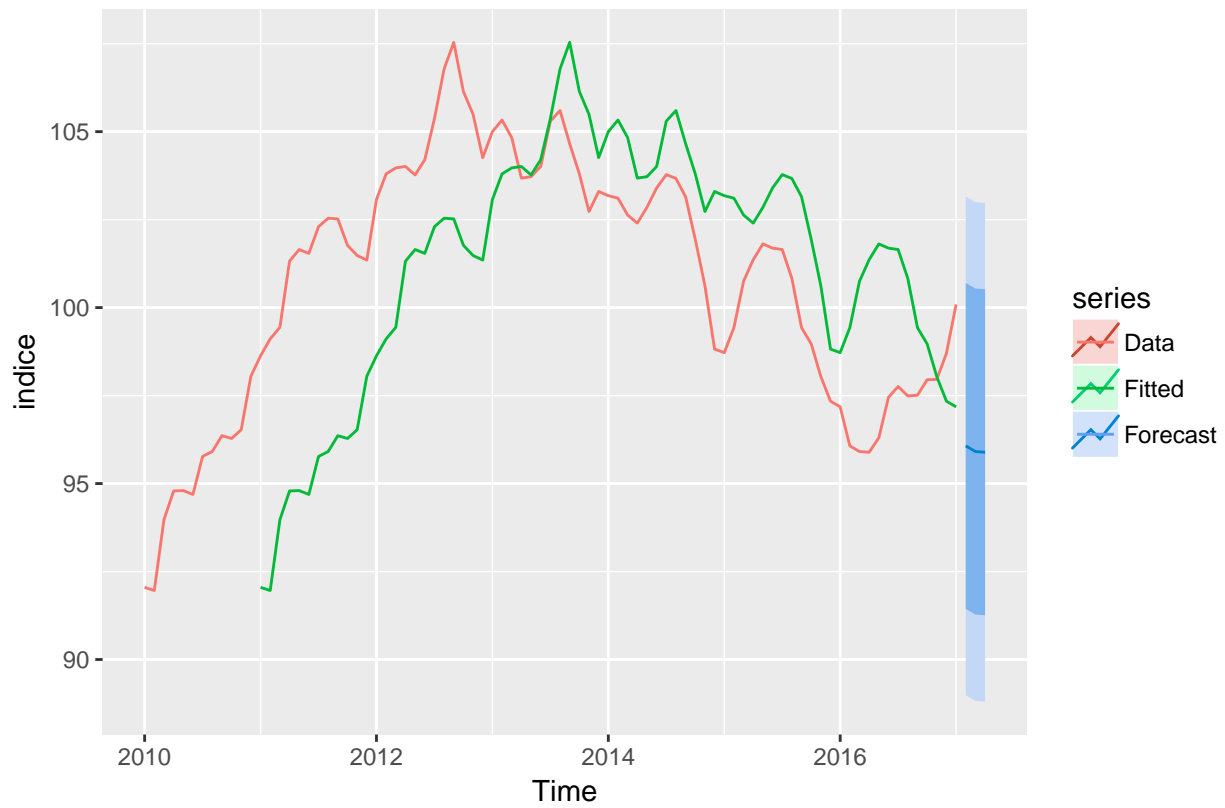
Mean Method



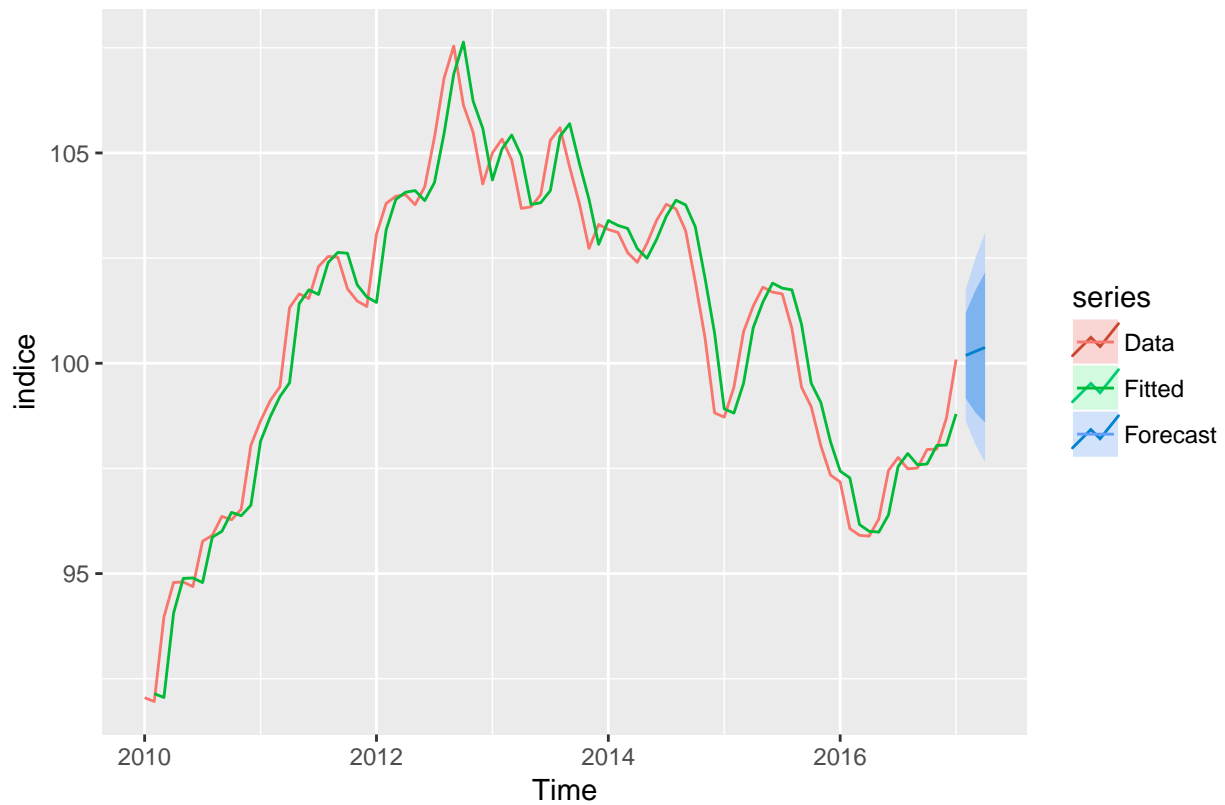
Naive Method



Seasonal Naive Method



Random Walk with Drift Method



## Assessing model accuracy

Table 1: Forecast Accuracy Measures

	ME	RMSE	MAE	MPE	MAPE	MASE
Mean	0.0000	3.5664	3.0557	-0.1275	3.0578	0.9970
Naive	0.0957	0.7961	0.6140	0.0966	0.6080	0.2003
Seasonal Naive	0.3825	3.6147	3.0649	0.3336	3.0287	1.0000
Drift	0.0000	0.7903	0.6157	0.0015	0.6099	0.2009

Both the drift and the naive method have the best performance in this group of forecasting methods, with the drift beating the naive method when it comes to its mean error and mean percentage error. However, the similarity of their values in the remaining error measures, the RMSE for example, indicate that either one will likely perform roughly the same.

## Cross-validated Accuracy Measures

An interesting way to choose between different forecasts is to perform cross-validation and calculate their respective RMSE and MAE. This provides us an estimate of how performant each forecast is in actual forecasting future time series values for different forecast horizons by iteratively restricting the sample and comparing forecast values with actual values to generate forecast residuals. As such this error measure will prove useful to not only select the best forecast method from the same group, but across families of methods as well, e.g. exponential smoothing vs ARIMA models.

Table 2: Cross-validated RMSE and MAE for h=1

	RMSE	MAE
Mean	3.6549	3.1881
Naive	0.7961	0.6140
Seasonal Naive	3.6201	3.1010
Drift	0.8249	0.6486

Table 3: Cross-validated RMSE and MAE, h=2

	RMSE	MAE
Mean	3.8117	3.3520
Naive	1.3493	1.1106
Seasonal Naive	3.6704	3.1630
Drift	1.4162	1.1605

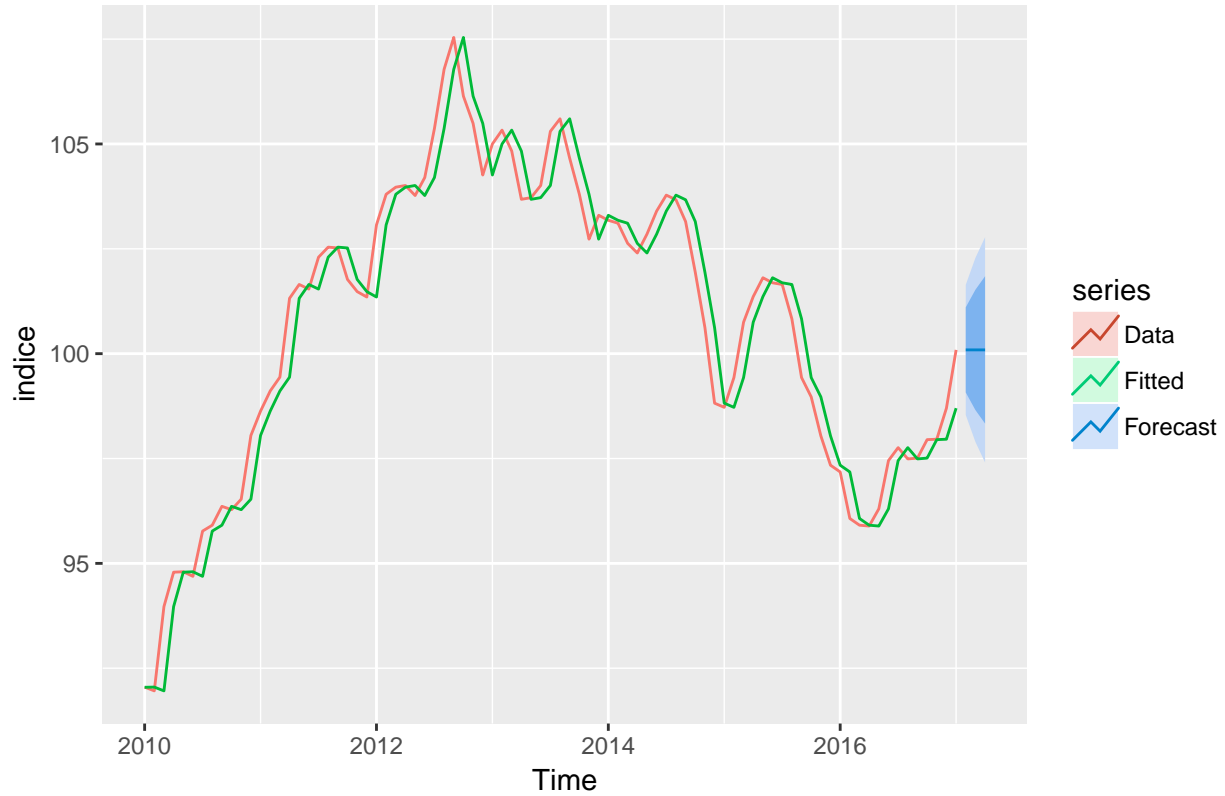
Table 4: Cross-validated RMSE and MAE, h=3

	RMSE	MAE
Mean	3.9647	3.4964
Naive	1.7566	1.4882
Seasonal Naive	3.4448	2.9451
Drift	1.8749	1.5193

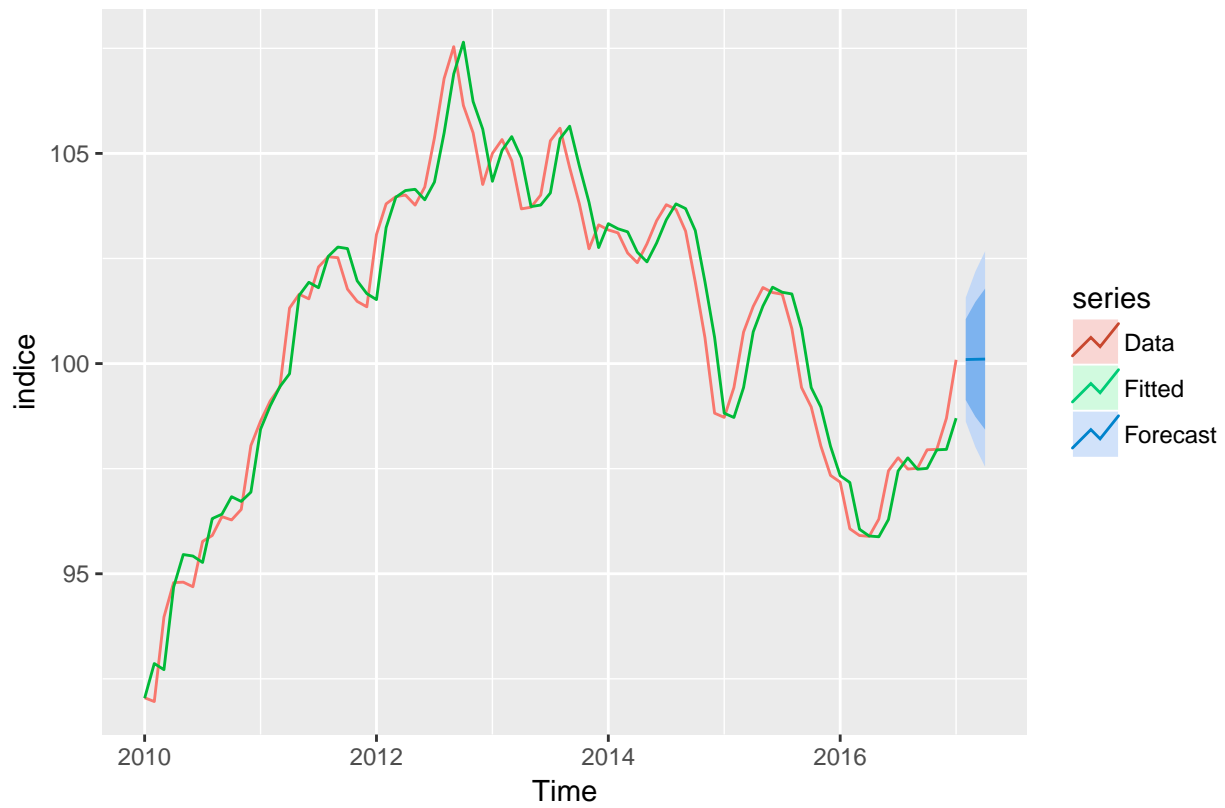
## Exponential Smoothing Methods

In this section we'll explore the use of exponential smoothing methods to forecast future values for this time series.

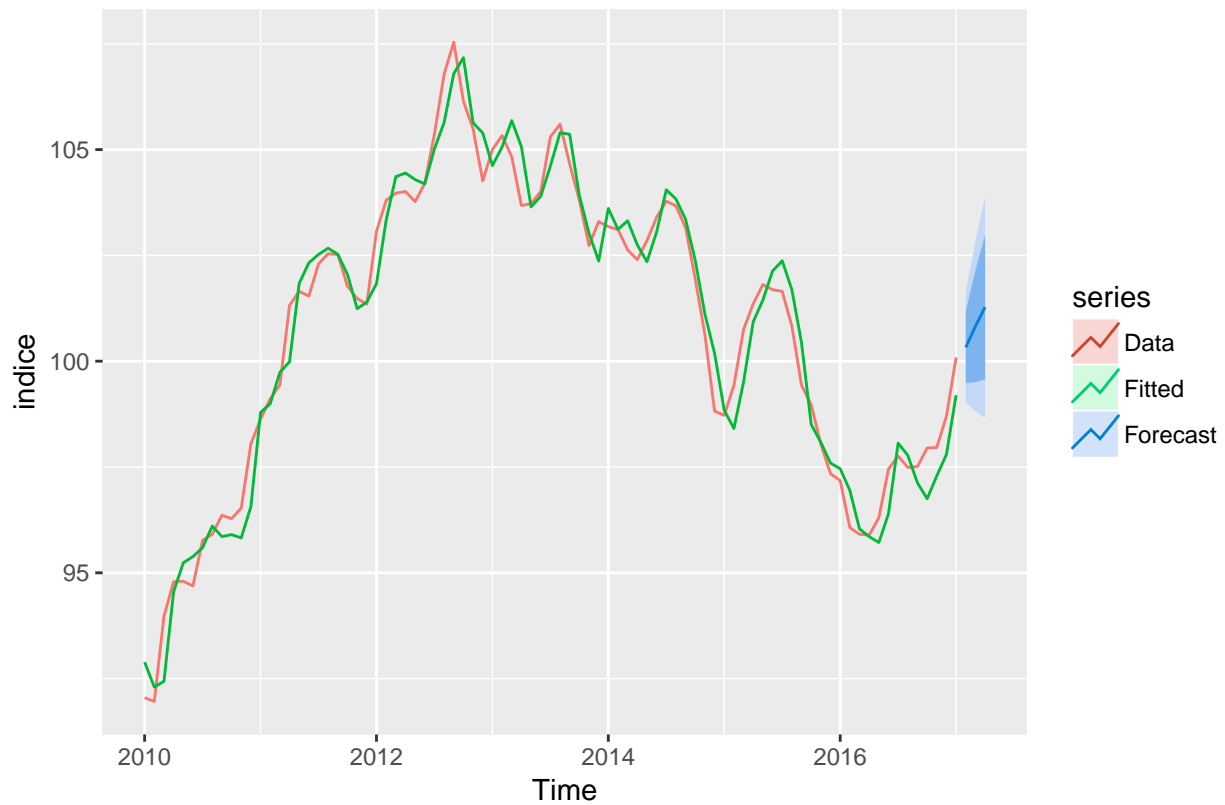
### Simple Exponential Smoothing with Additive Errors



### Holt's Linear Method with Additive Errors

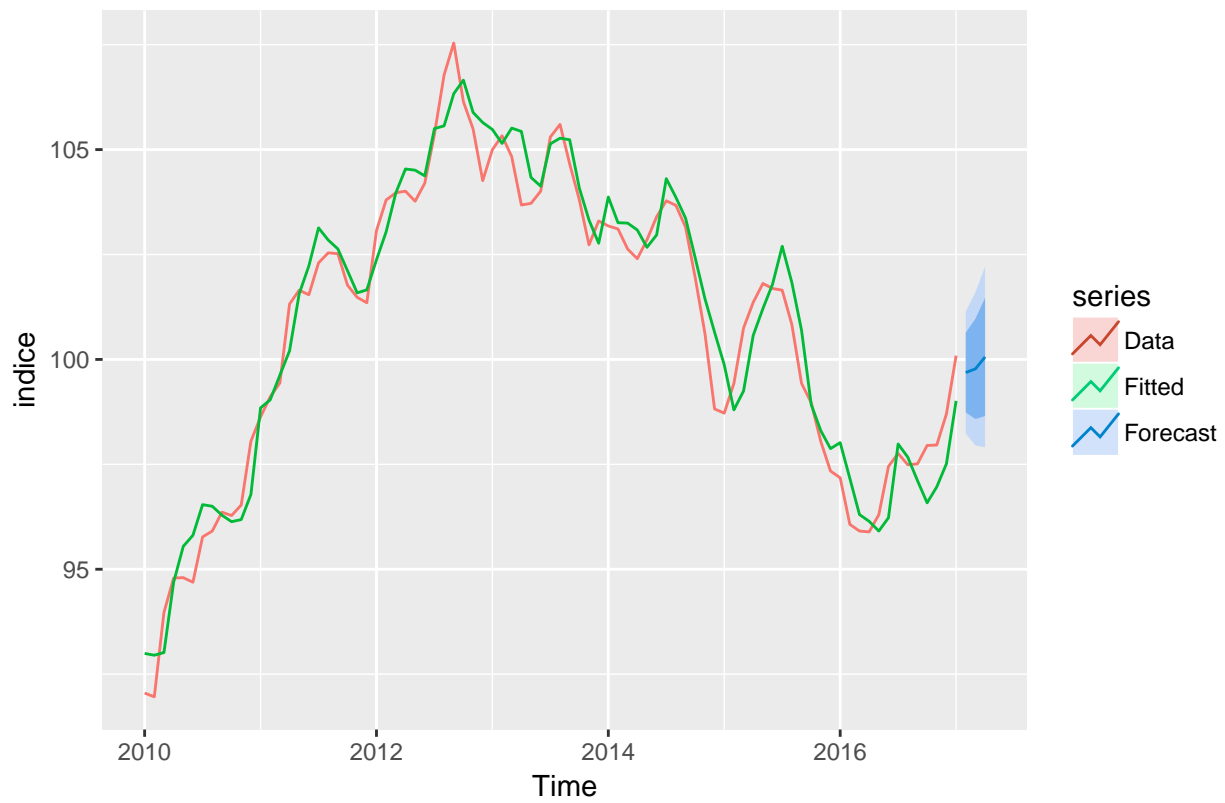


### Additive Holt–Winter's Method with Additive Errors

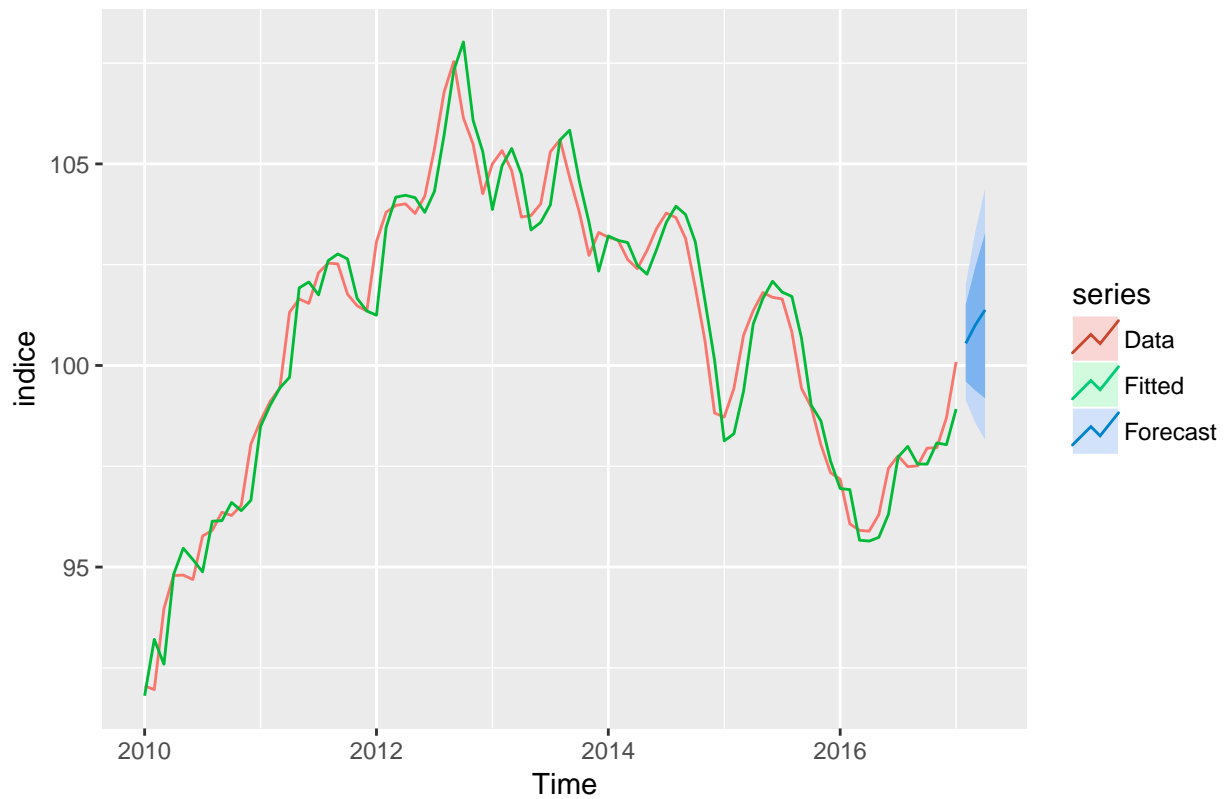




Multiplicative Holt–Winter's Method with Multiplicative Errors

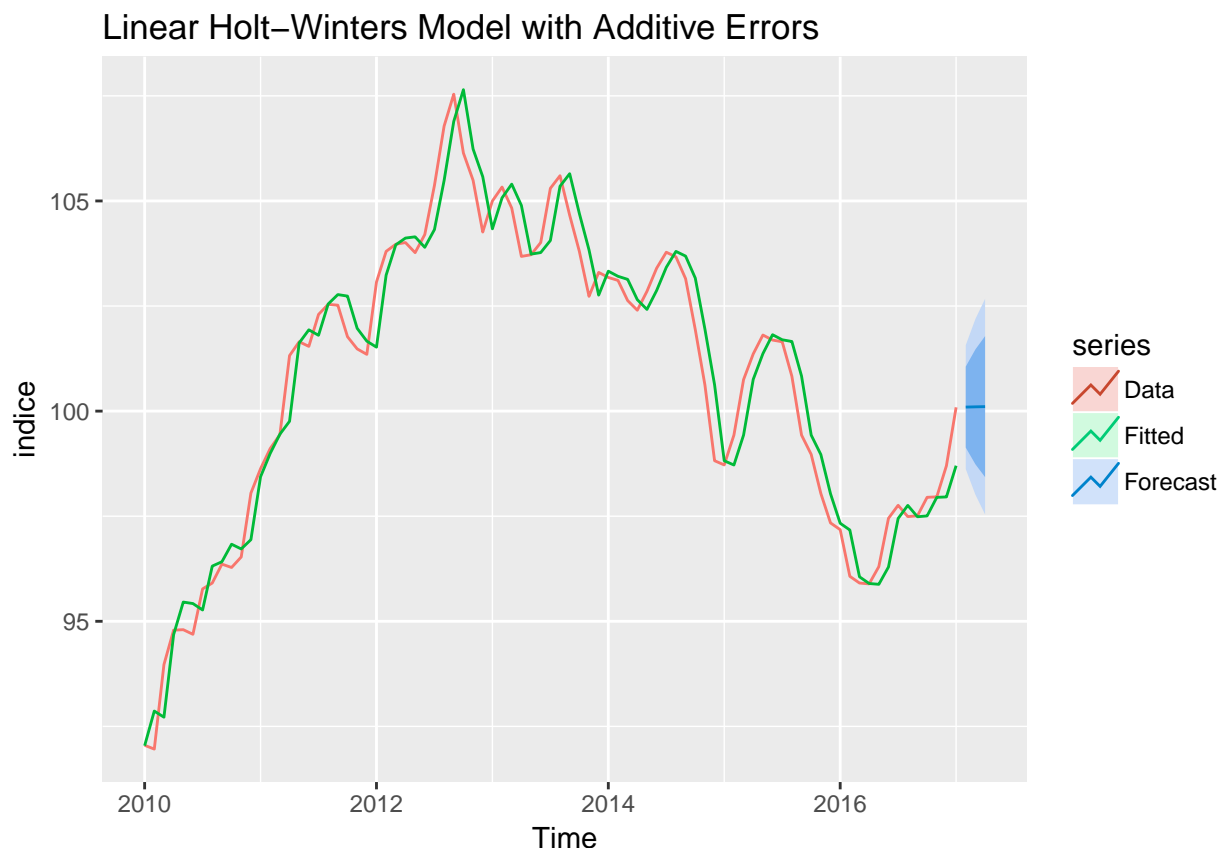


Multiplicative Damped Trend Method with Multiplicative Error Terms



Another possible way to forecast future values using exponential smoothing methods is to call the `ets()` function from the **forecast** R library developed by Rob Hyndman without passing on any model specifications. This instructs the function to go through the 30 possible ETS models and select the best based on their respective AIC and BIC values.

```
indice %>% ets() %>% forecast(h=3) -> ets1
autoplot(indice, series="Data", main = "Linear Holt-Winters Model with Additive Errors") +
  autolayer(fitted(ets1), series="Fitted")+
  autolayer(ets1,series="Forecast")
```



This seems to suggest that a linear Holt-Winters model with additive errors is the best exponential smoothing method for this time series.

## Assessing forecast accuracy

Table 5: Forecast Accuracy Measures

	ME	RMSE	MAE	MPE	MAPE	MASE
Simple Exponential Smoothing	0.0947	0.7914	0.6069	0.0955	0.6010	0.1980
Holt's Linear Method	-0.0492	0.7515	0.5877	-0.0508	0.5823	0.1917
Holt-Winter's Method	0.0245	0.6579	0.5265	0.0257	0.5241	0.1718
Holt-Winters Damped Method	-0.1378	0.7464	0.6098	-0.1384	0.6083	0.1990
Exponential Trend Method	0.0230	0.7482	0.5843	0.0230	0.5793	0.1906

The usual forecast accuracy measures seem ambiguous in this situation. Some models perform better on some accuracy measures while being beaten by other models in other accuracy measures.

### Cross-validated Accuracy Measures

Same as before, we'll estimate the cross-validated RMSE and MAE for the considered models for a forecast horizon of  $h = 1, 2, 3$ .

Table 6: Cross-validated RMSE and MAE,  $h=1$

	RMSE	MAE
Simple Exponential Smoothing	0.8433	0.6359
Holt's Linear Method	0.8858	0.6918
Holt-Winter's Method	0.9581	0.7699
Holt-Winters Damped Method	0.9909	0.7796
Exponential Trend Method	0.8964	0.6885

Table 7: Cross-validated RMSE and MAE,  $h=2$

	RMSE	MAE
Simple Exponential Smoothing	1.3776	1.1327
Holt's Linear Method	1.4505	1.1965
Holt-Winter's Method	1.4631	1.2192
Holt-Winters Damped Method	1.4705	1.2159
Exponential Trend Method	1.4789	1.2180

Table 8: Cross-validated RMSE and MAE,  $h=3$

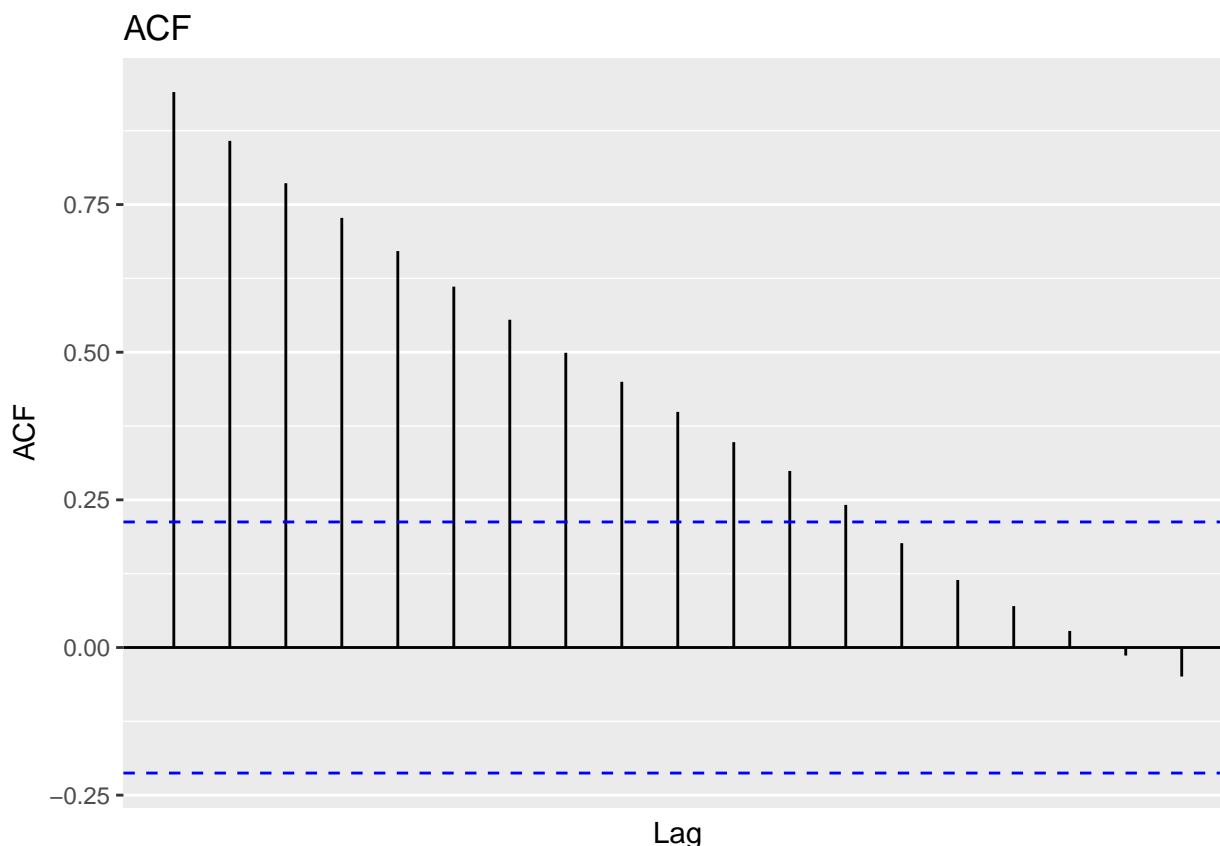
	RMSE	MAE
Simple Exponential Smoothing	1.7772	1.5105
Holt's Linear Method	1.9077	1.5752
Holt-Winter's Method	1.8342	1.5815
Holt-Winters Damped Method	1.8225	1.5944
Exponential Trend Method	1.9190	1.5881

Additionally, We can compute the cross-validated RMSE and MAE for the 30 possible ETS models and choose the one that minimizes those values. For purposes of space and legibility we refrain from showing the final table and instead report which model got the lowest scores:

	Model	RMSE	MAE
$h=1$	Simple exponential smoothing with multiplicative errors	0.8433	0.5359
$h=2$	Simple exponential smoothing with additive errors	1.3776	1.1327
$h=3$	Multiplicative Holt-Winters' method with additive errors	1.652	1.336

## ARIMA Models

Before estimating an ARIMA model for the time series, it is perhaps useful to study the time series further so that the results we later get are in accordance to our understanding of the time series. We start by looking at the ACF plot for the original time series, which we can predict right away will have most of its values outside the significance limits as the series is not stationary. We'll then try to estimate the order of the differences we must do before we're able to estimate an ARIMA model, and then we'll proceed to estimate the ARIMA model proper.



As expected, most of its values are outside the significance levels, indicating non-stationarity. A more formal way to prove this is to use the *Augmented Dickey-Fuller (ADF)* test. For this test, the following regression model is estimated:

$$y'_t = \phi y_{t-1} + \beta_1 y'_{t-1} + \dots + \beta_k y'_{t-k}$$

If the original series,  $y_t$  is already stationary, then  $\hat{\phi} < 0$ . If it needs differencing, then the coefficient should be approximately zero. The null-hypothesis for an ADF test is that the data are non-stationary, so large p-values are indicative of non-stationarity, and vice-versa.

```
adf.test(indice)

##
## Augmented Dickey-Fuller Test
##
## data: indice
## Dickey-Fuller = -2.5232, Lag order = 4, p-value = 0.3612
## alternative hypothesis: stationary
```

The large p-value does not allow us to reject the null-hypothesis, leading us to conclude that the time series is indeed non-stationary. An alternative method would be to compute the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test for the null hypothesis that the time series is level or trend stationary.

```
kpss.test(indice)
```

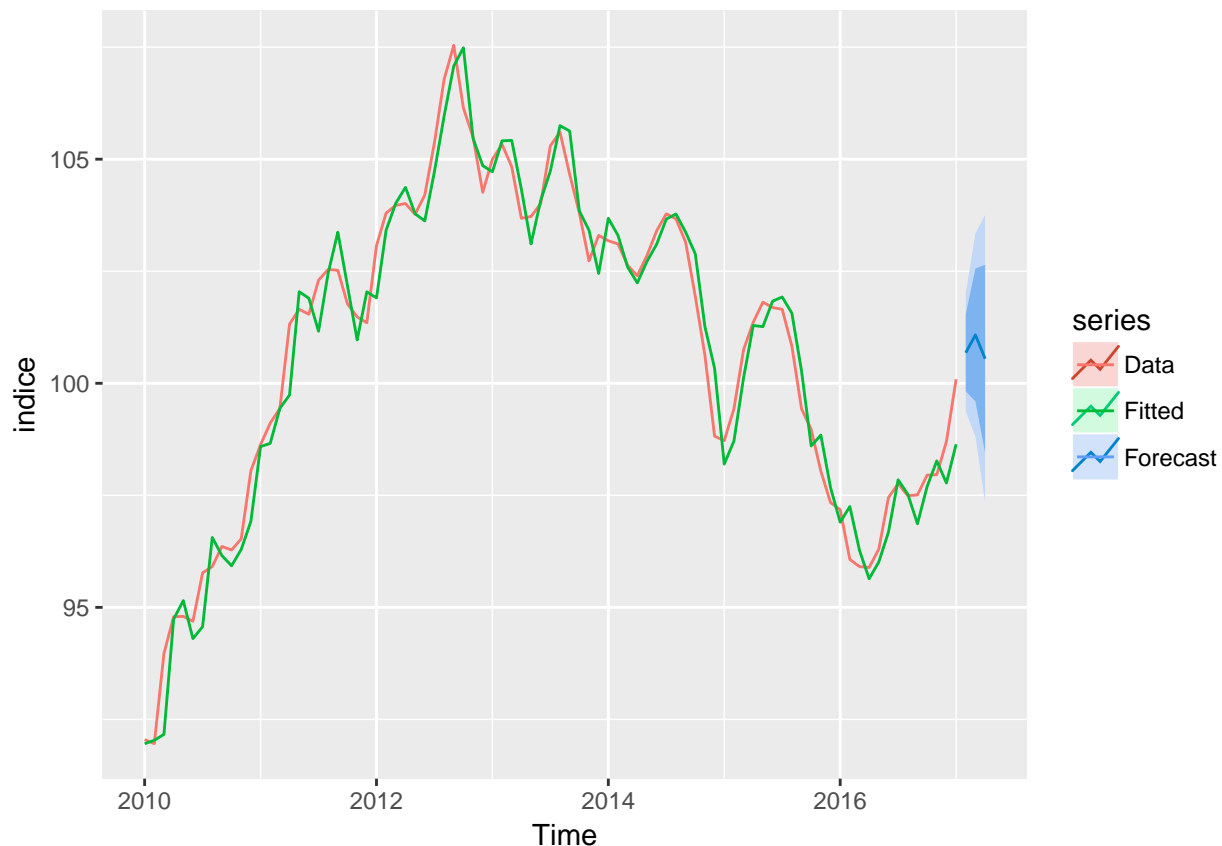
```
##
## KPSS Test for Level Stationarity
##
## data: indice
## KPSS Level = 0.63453, Truncation lag parameter = 2, p-value =
## 0.0195
```

The p-value is smaller than 0.05, which allows us to reject the null hypothesis at a 5% significance level that the time series is level stationary, as expected. A way to proceed would be to calculate the differences of the time series and see if we get a new time series that is indeed stationary. Fortunately, the `ndiffs()` function from the **forecast** package automates the estimation of the number of differences required to make a given time series stationary.

```
ndiffs(indice, test = "kpss")
```

```
## [1] 1
```

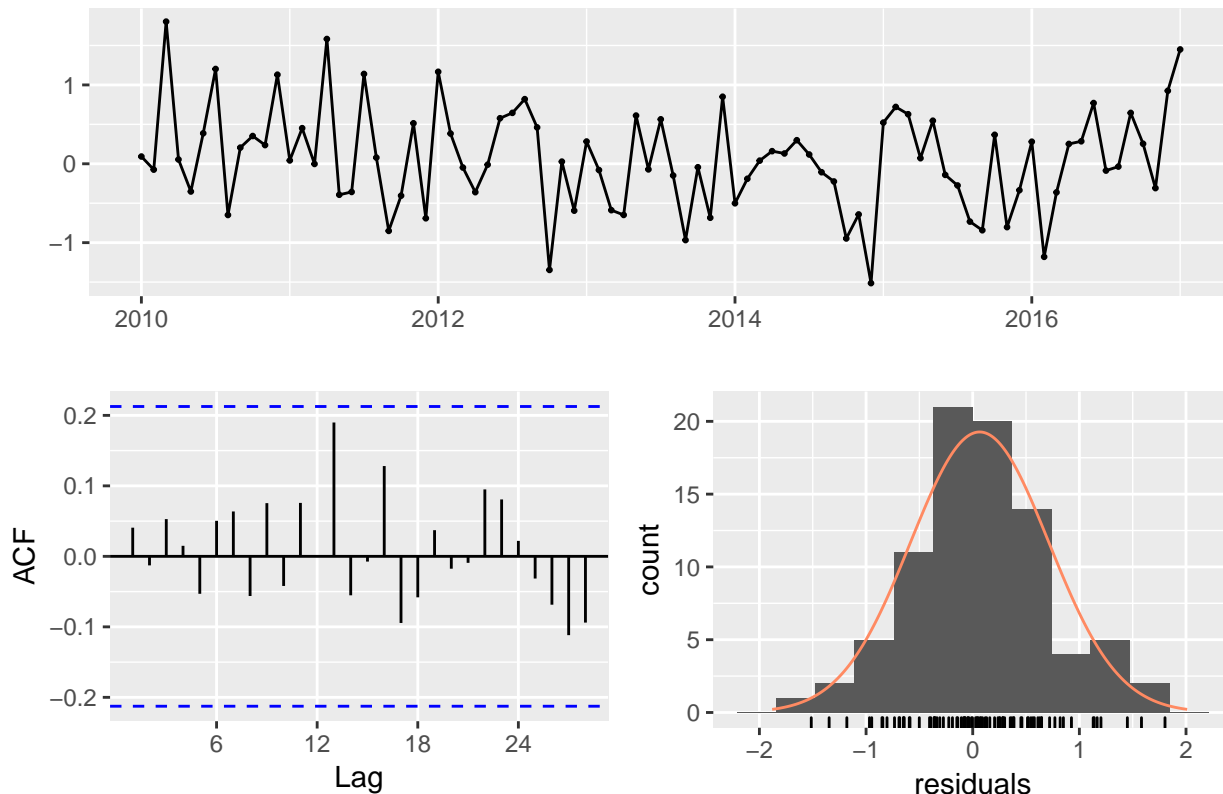
We now know the value of  $d$  in our  $ARIMA(p, d, q)(P, D, Q)_m$ , which we assume to be a seasonal model given the evidence for seasonality explored above. We could hand-code the estimation of a number of ARIMA models and choose the one where the small-sample-size corrected version of the *Akaike Information Criterion* ( $AIC_c$ ) is minimized, but we can automate the process by using the `auto.arima()` function provided by the **forecast** package.



```
## Series: .
```

```
## ARIMA(2,1,2)(0,0,1)[12]
##
## Coefficients:
##          ar1      ar2      ma1      ma2      sma1
##          0.1170 -0.6000  0.2806  0.8784  0.2491
## s.e.      0.2031   0.1498  0.1581  0.0830  0.1275
##
## sigma^2 estimated as 0.4517:  log likelihood=-84.5
## AIC=181.01   AICc=182.1   BIC=195.59
```

### Residuals from ARIMA(2,1,2)(0,0,1)[12]



```
##
## Ljung-Box test
##
## data: residuals
## Q* = 12.148, df = 19, p-value = 0.8792
##
## Model df: 5. Total lags used: 24
```

### Cross-validated Accuracy Measures

Estimating the cross-validated RMSE and MAE value of a sample of ARIMA models can be done but it is computationally infeasible to explore every possible model since, for univariate seasonal models, that would imply estimating the cross-validation errors of  $4^6$  models, or 4096 models. Instead what we choose to do is to use the fact that the  $ARIMA(2,1,2)(0,0,1)$  seems to be the best model for this time series to restrict the parameter space of our models such that  $(p,d,q) \in [0,2]$  and  $(P,D,Q) \in [0,1]$ . This leaves us with  $3^3 \times 2^3$  models, or 216 models, for which we will estimate their respective RMSE and MAE cross-validation values. We do not include the table with the entirety of the values for every model and instead report what model

had the lowest scores for both error measures.

- The minimum cross-validated RMSE for a forecast horizon  $h = 1$  was 0.6886 for the model  $ARIMA(2, 1, 2)(1, 0, 0)$ . The minimum cross-validated MAE for a forecast horizon  $h = 1$  was 0.5290 for the model  $ARIMA(2, 0, 0)(1, 0, 0)$ .
- For a forecast horizon  $h = 2$ , the model  $ARIMA(2, 0, 1)(1, 0, 0)$  performs the best, with a RMSE and a MAE of 1.1787 and 0.9681, respectively.
- For a forecast horizon  $h = 3$ , again the model  $ARIMA(2, 0, 1)(1, 0, 0)$  performs the best, with a RMSE and a MAE of 1.5334 and 1.2711, respectively.

## Conclusion

Having estimated the cross-validated RMSE and MAE for all models considered we're in a position to tentatively select which forecast method will perhaps supply us with the 'best' forecast value for Portuguese index of output prices, for different forecast horizons

### February

There was no single model that obtained the lowest values for both RMSE and MAE. The lowest RMSE was obtained using the model  $ARIMA(2, 1, 2)(1, 0, 0)$  while the minimum MAE was obtain using the model  $ARIMA(2, 0, 0)(1, 0, 0)$ . However, the drift method having obtained values for the RMSE and MAE of 0.7961 and 0.6140, respectively, pushes to consider it a viable forecasting method for so short a forecasting horizon.

Table 10: Forecast for February using the Drift Method

	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
Feb 2017	100.1857	99.1668	101.2046	98.6275	101.7439

### March and April

The model  $ARIMA(2, 0, 1)(1, 0, 0)$  performs the best, considering its values for the cross-validated *RMSE* and *MAE* for both a forecast horizon  $h = 2$  and  $h = 3$ .

Table 11: Forecast for March using  $ARIMA(2,0,1)(1,0,0)$

	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
Mar 2017	100.5682	99.0343	102.1021	98.2223	102.9141
Apr 2017	100.5616	98.5178	102.6053	97.4358	103.6873