Simple Exponential Smoothing

Lesson 2.2

Introduction

A popular forecasting method in business is exponential smoothing. It is popularity derives from its flexibility, ease of automation, cheap computation, and good performance.

Simple exponential smoothing is similar to forecasting with a moving average with weights decreasing exponentially into the past. The idea is to give more weight to recent information. Like the moving average, simple exponential smoothing should only be used for forecasting series that have no trend or seasonality.

Quick Review of Differencing to Remove Trend and Seasonality

A simple and popular method for removing a trend and/or a seasonal pattern from a series is differencing. Differencing means taking the difference between two values. A **lag-1** difference (also called first difference) means taking the difference between every two consecutive values in the series $y_t - y_{t-1}$. Lag-1 differencing results in a differenced series that measures the changes from one period to the next

Differencing at lag-k means subtracting the value from k periods back $y_t - y_{t-k}$. For example, for a daily series, lag-7 differencing means subtracting from each value y_t the value on the same day in the previous week y_{t-k}.

To remove trends and seasonal patterns we can difference the original time series and obtain a differenced series that lacks trend and seasonality

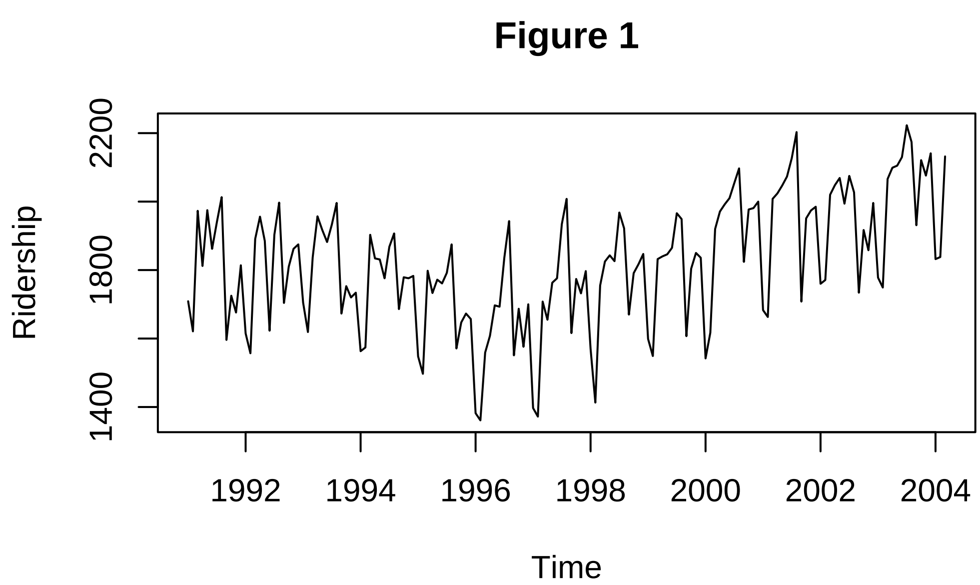
Consider the train ridership data *Amtrak.xlsx*.

```
library(readxl)
library(tidyverse)
library(patchwork)
library(forecast)
amtrak <- read_excel("Amtrak.xlsx")
```

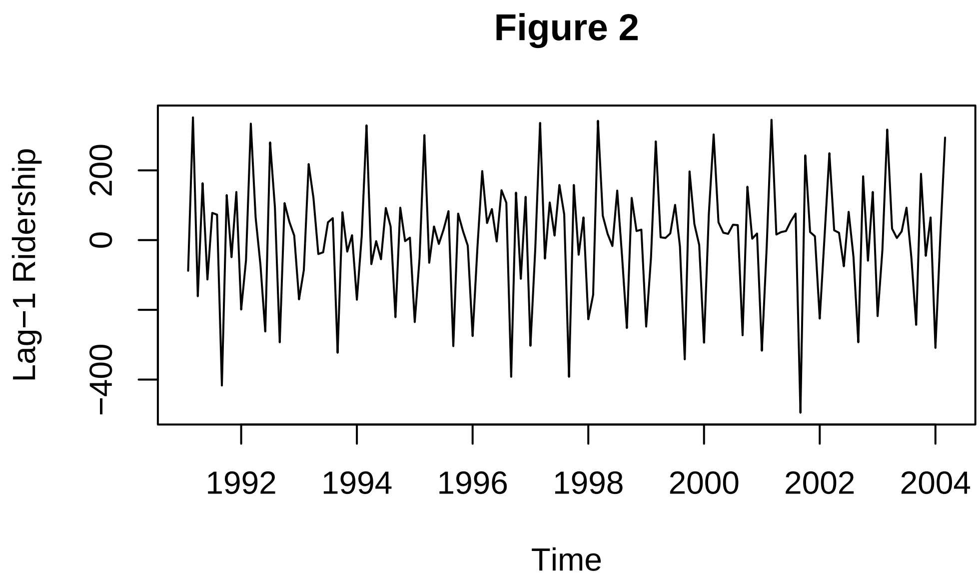
```r
amtrak.ts <- ts(amtrak$Ridership,
                frequency=12,
                start=c(1991,1),
                end=c(2004,3))

amtrak.ts.lag1 <- diff(amtrak.ts,lag=1)
amtrak.ts.lag12 <- diff(amtrak.ts,lag=12)
amtrak.ts.2d <-diff(amtrak.ts.lag12,lag=1)
```

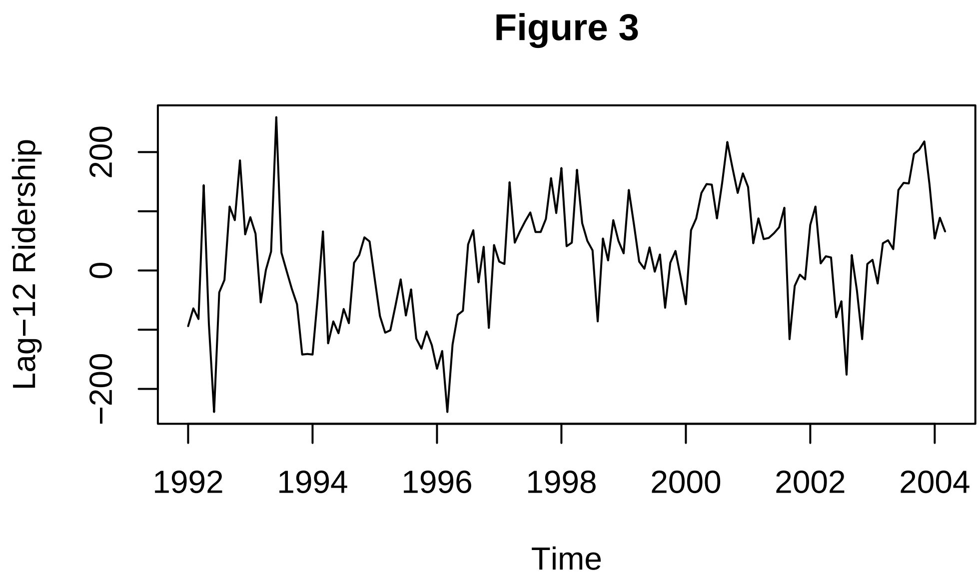
```r
#par(mfrow=c(2,2))
plot(amtrak.ts,
     ylab="Ridership",
     xlab="Time",
     main = "Figure 1")
```




```r
plot(amtrak.ts.lag1,
     ylab="Lag-1 Ridership",
     xlab="Time",
     main = "Figure 2")
```

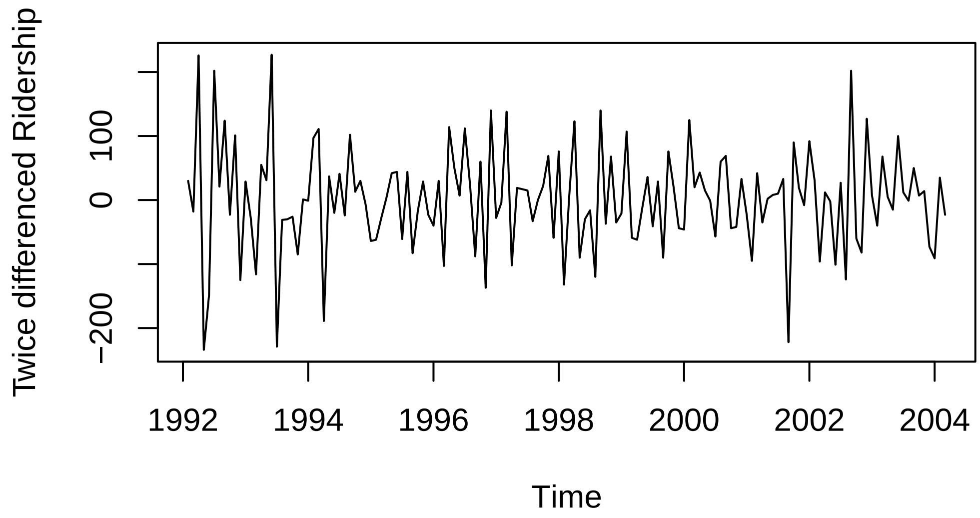



```
plot(amtrak.ts.lag12,
     ylab="Lag-12 Ridership",
     xlab="Time",
     main = "Figure 3")
```




```
plot(amtrak.ts.2d,
     ylab="Twice differenced Ridership",
     xlab="Time",
     main = "Figure 4")
```

Figure 4



As we have seen lag-1 differencing is useful for removing a trend. As shown in the time plots for the Amtrak data, compared to the original series (Figure 1), which exhibits a U-shaped trend, the lag-1 differenced series contains no visible trend (Figure 2).

For quadratic and exponential trends, often another round of lag-1 differencing must be applied in order to remove the trend. This means taking lag-1 differences of the differenced series

For removing a seasonal pattern with M seasons, we difference at lag M. For example, to remove a day-of-week pattern in daily data, we can take lag- 7 differences. For example, the plot (Figure 3) illustrates a lag-12 differenced series of the Amtrak monthly data. The monthly pattern no longer appears in this differenced series. We call this process as *seasonal adjustment* or *deseasonalizing*

When both trend and seasonality exist, we can apply differencing twice to the series in order to de-trend and deseasonalize it.

For example, we performed double differencing on the Amtrak data, which contain both trend and seasonality. The plot (Figure 4) shows the result after first differencing at lag-12, and then applying a lag-1 difference to the differenced series.

Note that differencing is often used as a pre-processing step before applying a forecasting model to a series.

Simple Exponential Smoothing

The exponential smoother generates a forecast at time $t + 1$ as follows:

$$F_{t+1} = \alpha y_t + \alpha(1-\alpha)y_{t-1} + \alpha(1-\alpha)^2 y_{t-2} + \cdots$$

where α is a constant between 0 and 1 called the smoothing constant. The above formulation displays the exponential smoother as a weighted average of all past observations, with exponentially decaying weights

We can also write the exponential forecaster in another way, which is very useful in practice:

$$F_{t+1} = F_t + \alpha e_t$$

where e_t is the forecast error at time t

We have seen in both formulations that we need to store and use only the forecast and forecast error from the previous period. Forecasting further into the future yields the same forecasts as a one-step-ahead forecast, that is, the k-step-ahead forecast is given by

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

The smoothing constant α, which is set by the user, determines the rate of learning. An α value close to 1 indicates fast learning (that is, only the most recent values influence the forecasts), whereas a value close to 0 indicates slow learning (past observations have a large influence on forecasts). Default values that have been shown to work well are in the range 0.1-0.2. Trial and error can also help in the choice of α.

To evaluate the quality of the forecast we examine the time plot of the actual and predicted series, as well as the predictive accuracy (e.g., MAPE or RMSE of the validation period).

In R, forecasting using simple exponential smoothing can be done via the *ets* function in the forecast package. The three letters in **ets** stand for *error*, *trend*, and *seasonality*, respectively.

Applying this function to a time series will yield forecasts and forecast errors (residuals) for both the training and validation periods. You can use a default value of $\alpha = 0.2$, set it to another value, or choose to find the optimal α in terms of minimizing RMSE over the training period.

Let us illustrate forecasting with simple exponential smoothing using the twice-differenced Amtrak ridership data. To make forecasts in the validation period, we fit the simple exponential smoothing model to the training set (February 1992 to March 2001) with $\alpha = 0.2$.

The simple exponential smoothing model in the *ets* framework is the model with additive error (A), no trend (N), and no seasonality (N).

```r
nValid <- 36
nTrain <- length(amtrak.ts.2d) - nValid
train.ts <- window(amtrak.ts.2d,
                   start = c(1992, 2),
                   end = c(1992, nTrain + 1))
valid.ts <- window(amtrak.ts.2d,
                   start = c(1992, nTrain + 2),
                   end = c(1992, nTrain + 1 + nValid))
ses <- ets(train.ts, model = "ANN", alpha = 0.2)
ses.pred <- forecast(ses, h = nValid, level = 0)
```

```r
plot(ses.pred, ylim = c(-250, 300),
     ylab = "Ridership (Twice-Differenced)",
     xlab = "Time",
     bty = "l", xaxt = "n",
     xlim = c(1991,2006.25),
     main = "", flty = 2)

axis(1, at = seq(1991, 2006, 1),
     labels = format(seq(1991, 2006, 1)))
lines(ses.pred$fitted, lwd = 2, col = "blue")
lines(valid.ts)
```