Lecture 6

Modelling Seasonal Time Series Data

In the previous lectures, we had seen how to model time series data with trend (increasing/decreasing) using the method of differencing. A *seasonal* time series data is a collection of observations made every month or every quarter of a calendar year; that is, usually, it is either a collection of monthly data or quarterly data. Suppose that $\{Y_t\}$ is a seasonal time series with number of seasons, denoted by S, as $S = 4$ or $S = 12$. This time series could have trend, and also possibly the effects of different seasons on the observations, called *seasonal factors (or effects)*.

Decomposition procedures are used in time series to describe the trend and seasonal factors in a time series. One of the main objectives for a decomposition is to estimate seasonal effects that can be used to create and present seasonally adjusted values. For example, the sale of cold beverages. air conditioners, fans tend be higher during the summer than in the other three seasons. Thus, the average (or mean) sales during summer would be different from that during the other three seasons, which makes the sales data non stationary. So, in what follows, we present methods to capture these seasonal effects and remove them from the original data. We call such data (after removing the seasonal effects) the **de-seasonalised** or **seasonally adjusted** data. This again could be non stationary as the trend could still be present in the seasonally adjusted data. But, we know how to model and forecast for the time series data with trend. To get the forecasts for the original data, we just "put" back the seasonal factors in the forecasts of the seasonally adjusted data.

The following two structures are considered for basic decomposition models:

Additive = Seasonal + Trend + Irregular

Multiplicative: = Seasonal * Trend * Irregular

The additive model is useful when the seasonal variation is relatively constant over time.

The multiplicative model is useful when the seasonal variation increases over time.

Now, we present two candidate examples for the additive and the multiplicative model structures.

Figure 1: The graph below shows the quarterly production of beer in Australia for 18 years (72 observations). As we can notice clearly, the quarterly variation in beer sales remains relatively constant across years, and hence an additive model is recommended for the analysis.

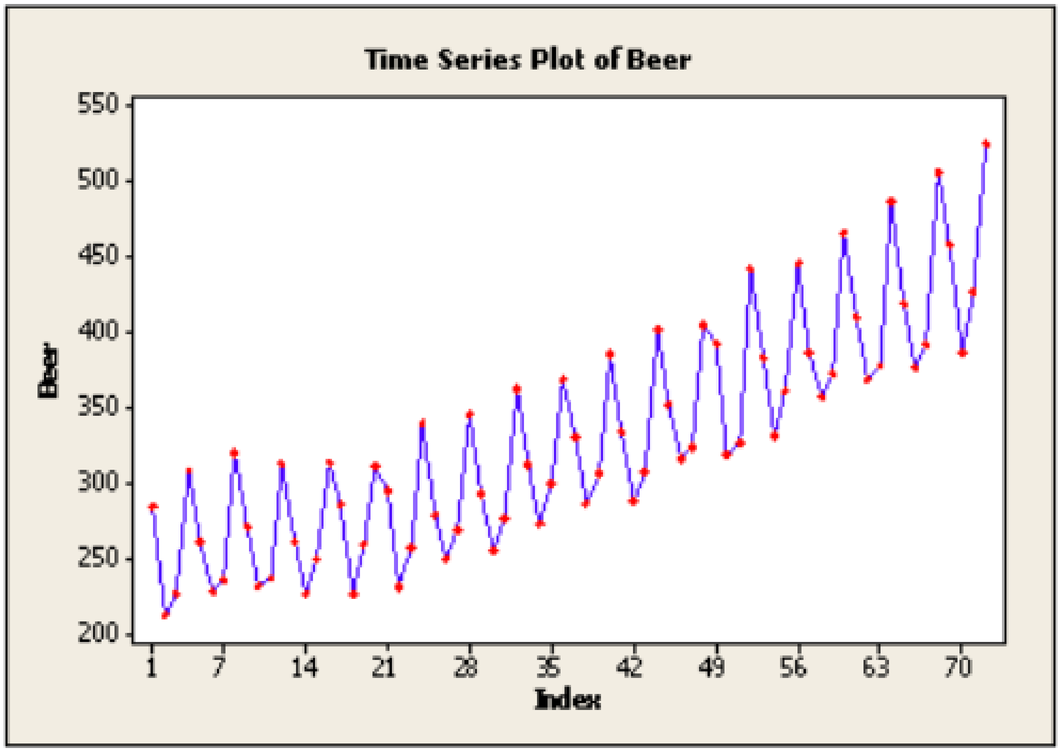
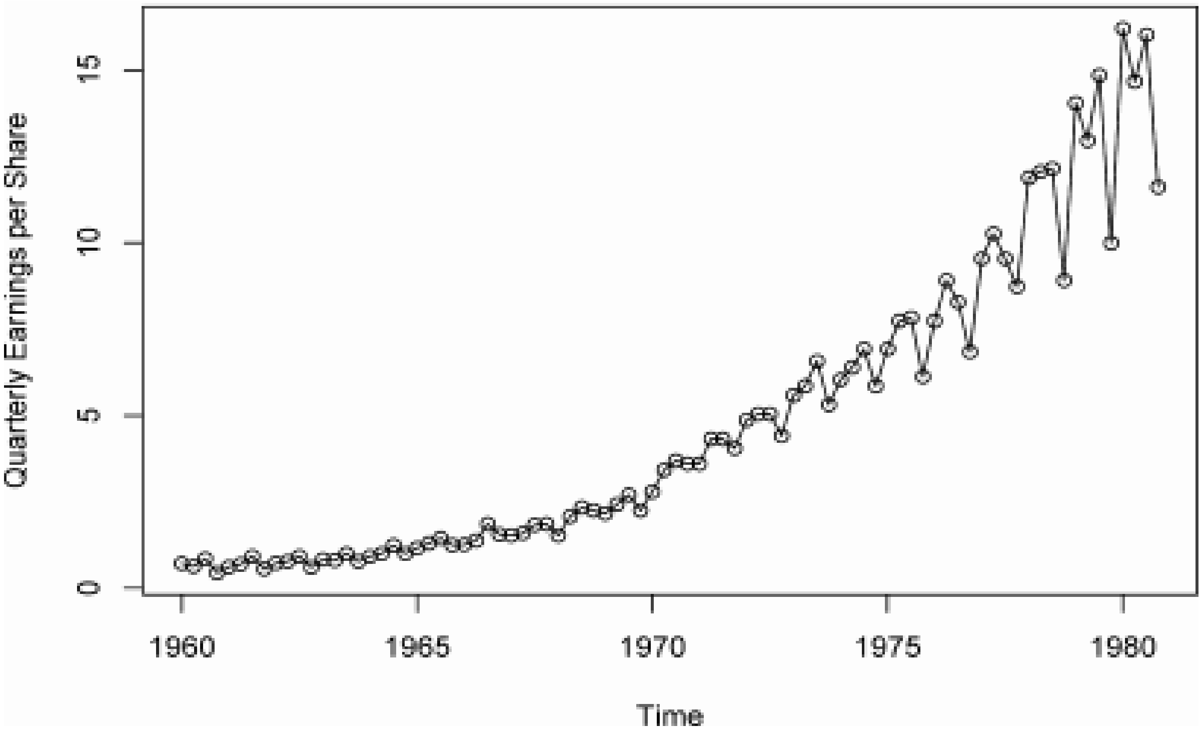


Figure 2: (*Johnson & Johnson* data from **Lecture 1**) The next graph shows the quarterly earnings per share for the U.S. company *Johnson & Johnson*. The data is for 84 successive quarters (21 years), starting from the first quarter of 1960 till the last quarter of 1980.

Here, we see clearly that the quarterly variation keeps increasing across the quarters. Hence, a multiplicative model is to be recommended for the analysis.



Forecasting Seasonal Time Series Models

For the time series $\{Y_t\}$, letting S_t, T_t and I_t denote the seasonal, trend, and the irregular components respectively, we have the additive and the multiplicative decompositions given by

$$Y_t = S_t + TR_t + I_t,$$

and

$$Y_t = S_t * TR_t * I_t$$

respectively.

Additive Models

Let us consider the problem of forecasting for the additive model first; that is,

$$Y_t = S_t + TR_t + I_t.$$

Let $\{Y_t, t = 1, 2, ..., T\}$ be the data and we want to forecast Y_{T+1}.

Firstly, to get the values of the seasonal factors, use the command *decompose* in R and *seasonal decompose* in Python. Get the values of the seasonal factors S_t's and subtract them from the respective Y_t's to get the seasonally adjusted Y_t's; that is,

$$U_t = Y_t - S_t = TR_t + I_t \quad t = 1, 2, ..., T \quad (U_t = \text{seasonally adjusted } Y_t).$$

Fit an appropriate $ARIMA$ model for the U_t's and get the required forecast of U_{T+1}, denoted by $\hat{U_T}(1)$, for this model. Now, $T+1$ is a time index that represents a unique season (as the original data is seasonal data) - one of the 4 quarters if it is quarterly data, or one of the 12 months if it is monthly data. For clarity, let us assume that our data is monthly and that $T+1$ is $November$. Now the forecast of Y_{T+1} is given by

$$\hat{Y_T}(1) = \hat{U_T}(1) + S_{Nov},$$

where, S_{Nov} is the *latest* S_{Nov} value in the set of seasonal factors gotten from the decomposition.

Multiplicative Models

In the case of multiplicative models, we have that

$$Y_t = S_t * TR_t * I_t$$

or equivalently (when Y_t is positive),

$$log(Y_t) = log(S_t) + log(TR_t) + log(I_t).$$

Letting V_t denote the seasonally adjusted $log(Y_t)$ (that is, $V_t = log(Y_t) - log(S_t)$), and using the method of forecasting for additive models, we get the forecast for $log(Y_{T+1})$ as (when we have monthly data and $T+1$ is November, as an example)

$$log(\hat{Y_T})(1) = \hat{V_T}(1) + log(S_{Nov})$$

or equivalently,

$$\hat{Y_T}(1) = \exp\{log(\hat{Y_T})(1)\}.$$

Important Note:

(a) When we invoke the command for the decomposition of our data, we need to mention the type (additive or multiplicative) and the number of seasons in a year (4 for quarterly data and 12 for monthly data) clearly, both in R and in Python.

(b) The seasonal factors computed by decomposition, both in the additive and the multiplicative models, is a collection of constants (one for each season) across years; that is, for example, if we have monthly data, then the seasonal factor for each month repeats itself for every year as presented below in the screen shot of the output:

```
> birthstimeseriescomponents$seasonal # get the estimated values of the seasonal component
            Jan        Feb        Mar        Apr        May        Jun        Jul        Aug        Se
 1946 -0.6771947 -2.0829607  0.8625232 -0.8016787  0.2516514 -0.1532556  1.4560457  1.1645938  0.
 1947 -0.6771947 -2.0829607  0.8625232 -0.8016787  0.2516514 -0.1532556  1.4560457  1.1645938  0.
 1948 -0.6771947 -2.0829607  0.8625232 -0.8016787  0.2516514 -0.1532556  1.4560457  1.1645938  0.
 1949 -0.6771947 -2.0829607  0.8625232 -0.8016787  0.2516514 -0.1532556  1.4560457  1.1645938  0.
 1950 -0.6771947 -2.0829607  0.8625232 -0.8016787  0.2516514 -0.1532556  1.4560457  1.1645938  0.
 1951 -0.6771947 -2.0829607  0.8625232 -0.8016787  0.2516514 -0.1532556  1.4560457  1.1645938  0.
 1952 -0.6771947 -2.0829607  0.8625232 -0.8016787  0.2516514 -0.1532556  1.4560457  1.1645938  0.
 1953 -0.6771947 -2.0829607  0.8625232 -0.8016787  0.2516514 -0.1532556  1.4560457  1.1645938  0.
 1954 -0.6771947 -2.0829607  0.8625232 -0.8016787  0.2516514 -0.1532556  1.4560457  1.1645938  0.
 1955 -0.6771947 -2.0829607  0.8625232 -0.8016787  0.2516514 -0.1532556  1.4560457  1.1645938  0.
 1956 -0.6771947 -2.0829607  0.8625232 -0.8016787  0.2516514 -0.1532556  1.4560457  1.1645938  0.
 1957 -0.6771947 -2.0829607  0.8625232 -0.8016787  0.2516514 -0.1532556  1.4560457  1.1645938  0.
 1958 -0.6771947 -2.0829607  0.8625232 -0.8016787  0.2516514 -0.1532556  1.4560457  1.1645938  0.
 1959 -0.6771947 -2.0829607  0.8625232 -0.8016787  0.2516514 -0.1532556  1.4560457  1.1645938  0.
```