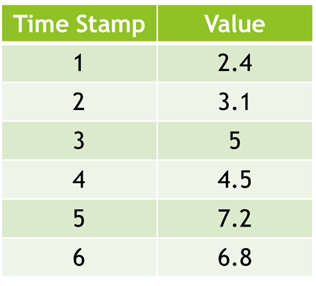
Often, you come across **time stamped data**. Time stamped data is basically a sequence of data that has time values attached to the sequence of values, such as



**Time Stamped Data**

Assume you have to **forecast** the value for timestamp 7. Now, based on the knowledge you gained in the previous modules, you may think, why not use regression for forecasting?

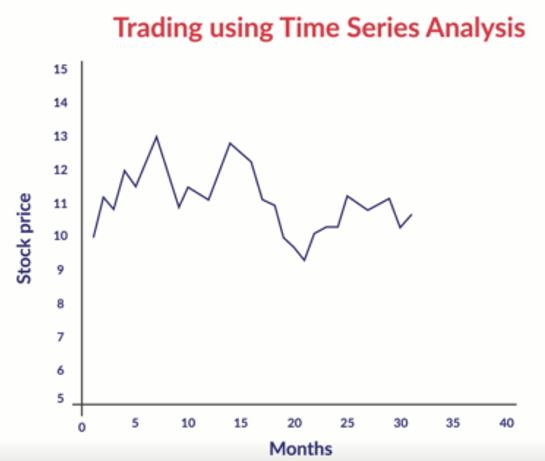
However, it is not enough to simply use regression to make the forecast. You have to do something more than that to make an accurate forecast

* How does time series analysis **differ from regression analysis**?
* What are the **basic features** and **characteristics** of a time series?
* What are the **basic components** of a time series?

So, a time series is a series of **time stamped** values. In other words, it is a sequence of values with time values attached to it.

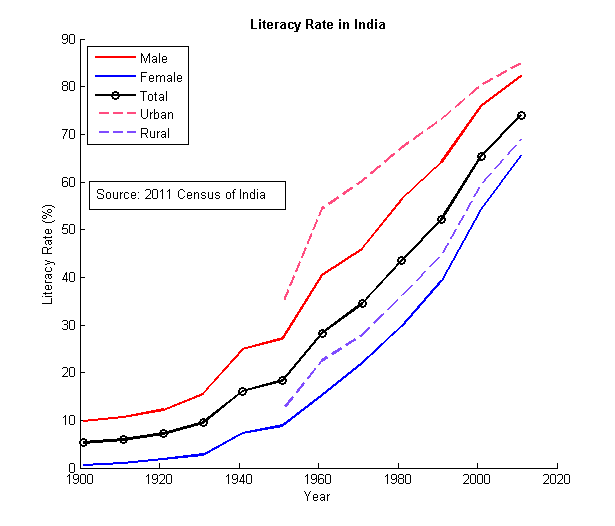
A few examples of such a series are

* Daily stock market figures



**Daily Stock Market Figures**

* Demographic/Development data (population, birth rate, infant mortality figures, literacy, per-capita income, school enrolment figures) by year



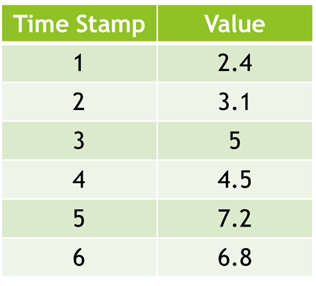
**Literacy Rate in India (1900 - 2020)**

Using time series analysis, you can **forecast**

* The value of the stock market index for a future month, or
* The value of the literacy rate for a future census

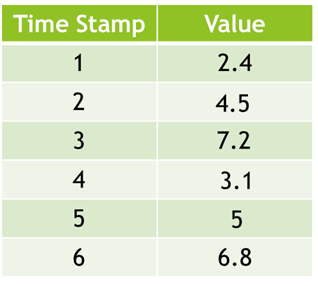
We will look into exactly how this is done, later in the module.

Now, you may think that you can just use regression or advanced regression to make a forecast, taking time as an independent variable. However, this will not work due to various reasons. One reason is that in a time series, the **sequence is important**. For example, let’s take the data from before:



**Original Time Stamped Data**

Using regression or advanced regression, let’s say you predict the value for timestamp 7. Now let’s say you shuffle the data around like this:



**Shuffled Time Stamped Data**

This data will also give you the same prediction for time stamp 7 if you use regression. However, a time series analysis will give you different forecasts for the original data and for the shuffled one.

Why does this happen? This happens because while forecasting using **time series**, your model predicts not only on the basis of the **values given**, but also on the basis of the **sequence** in which the values are given. Hence, the sequence is very important in a time series analysis and should not be played with.

**Time Series vs Regression**

Let’s say that you are trying to predict the literacy rate of India in the 2021 census. Which of the following statements about this time series model is correct?

1 - Unlike regression models, the value of literacy at a particular timestamp (say 2001) may be correlated to the value of literacy at a previous timestamp (say 1981).

2 - Unlike regression models, we are not concerned with what the reasons are for a rise in literacy rate. We just want to find out what the literacy rate will be in 2021, based on the literacy rates of the past.



Both statement 1 and 2 are correct

Feedback : *As you learnt earlier, the time series model will make forecasts for literacy rates in the future, based on the the literacy rates of the past, and the order they were in (that’s why you can’t randomise the order). Clearly, the past value of literacy affects the future value, and hence, is correlated.Also, as was taught recently, in a time series analysis you just try to predict future values, based on the past values and their order, you are not worried about the causal factors for rising literacy.*

So, the two most important **differences** between time series and regression are:

* Time series have a strong **temporal (time-based) dependence** — each of these data sets essentially consists of a series of time stamped observations, i.e., each observation is tied to a specific time instance. Thus, unlike regression, the order of the data is important in a time series.
* In a time series, you are **not concerned with the causal relationship** between the response and explanatory variable. The cause behind the changes in the response variable is very much a black box.

Out the following events, which one is an example of a local pattern variation -

A) Due to a recent terrorist attack in Britain, the number of tourists visiting the city of Manchester has decreased, suddenly. Now, the number is returning back to normal, steadily.

B) Due to countries like India focusing on their respective tourism industries, the number of tourists visiting the British city of Manchester is on the decline.

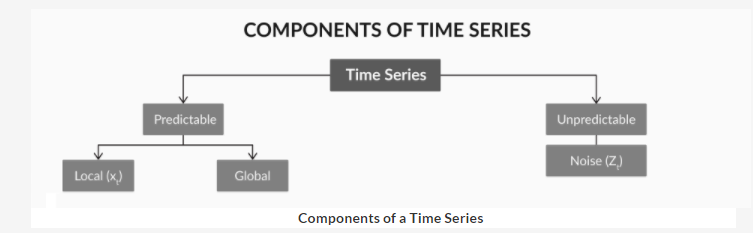


**Only A**

**Feedback :***Local pattern variations are those that show up in the short-term and balance out in the long-term, unlike global variations. Statement A looks like an example of a short-term change that will even out after a few days; then more people will start visiting the city again*

So far, you have learnt about the following components of a time series:

**Components of a Time Series**

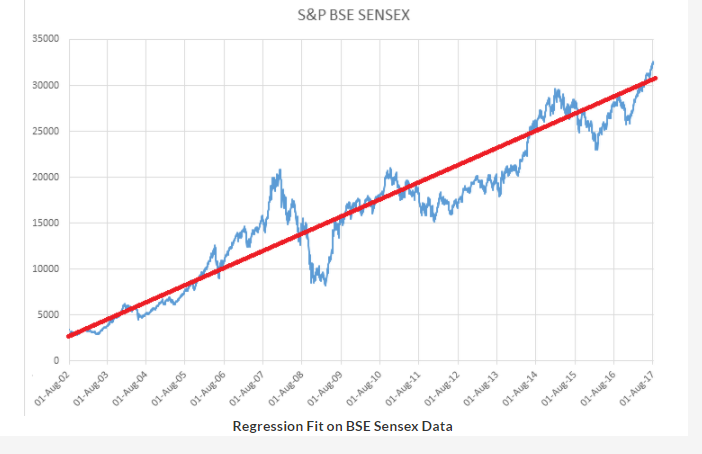


So you have the **globally predictable** part and the **locally predictable** part. You also have the **unpredictable part**, which is something you cannot predict.

To understand these better, let’s look at the **BSE data** for the last 15 years.

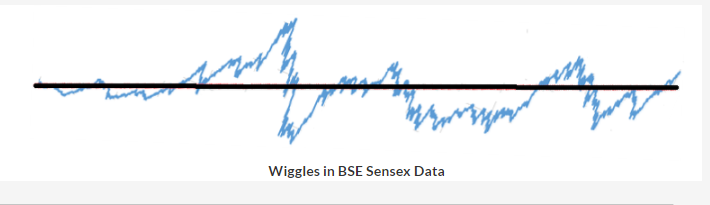


So in general, you can see that the value of the index has been increasing over the years. Hence, if you fit a **regression model** on to it, like you saw earlier, you will get a line like this:



he red line here, the one that shows the regression fit, gives the **overall pattern** of the data. So overall, you expect the index to go up every year. Hence, this red line is indicative of the overall pattern of the data, or in other words, the global pattern.

But then, you see that there are a lot of **wiggles** in the data, on top of the simple regression fit. Let’s plot these wiggles, the ones that would be left after removing the regression fit.



In a time series, you model this part of the data as well. This is the **locally predictable** part of the time series, or in other words, the local pattern in the time series. It is called local because the values here don’t show any long-term pattern. The values don’t increase regularly or repeat with a set frequency; so there is **no overall global pattern**. However, locally, there is some predictability: the value of the wiggle today depends somewhat on what the value of the wiggle was yesterday.

Look at it this way: because of the **overall global pattern**, you expect the stock market index to go up every year as India grows economically. So the underlying cause of the global pattern is **India’s long-term economic growth**.

However, other than the general overall growth, you do experience some **up-and-down variations** if you zoom in on a day-by-day or a week-by-week level. These variations happen mostly because of what the index value was the previous day. If the market starts to go up, you generally expect it to go up the next day too. If it has been going up for a long time, you expect it to decrease soon. The underlying causes of these local wiggles are the **market sentiment**, opinions of stock market experts, and various other factors that pop up in a sporadic fashion.

In short, there are three main components of a time series: **global, local and noise**. In the upcoming lecture, we will break down the global component further into two subcomponents: **trend** and **seasonality**.

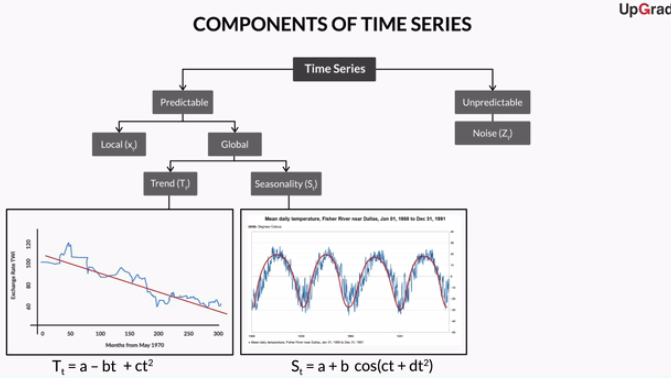
**Components of Time Series - I**

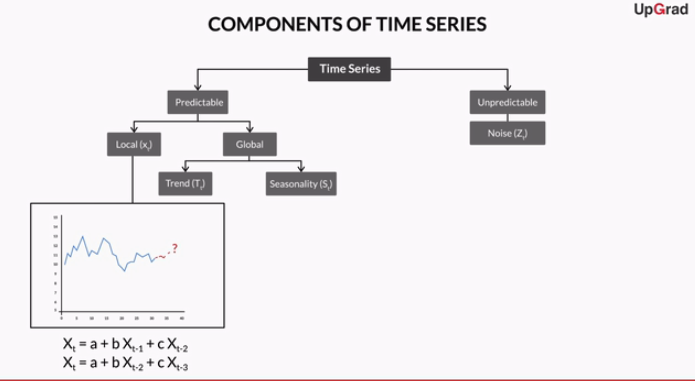
A restaurant has been experiencing higher sales during the weekends, as compared to the weekdays. Select the correct component of the time series that explains the daily restaurant sales patterns for this restaurant.

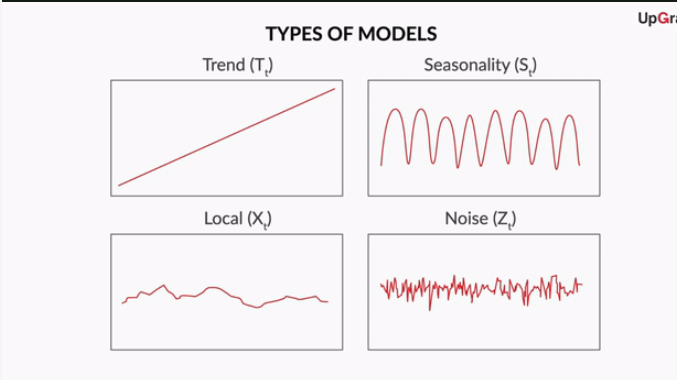


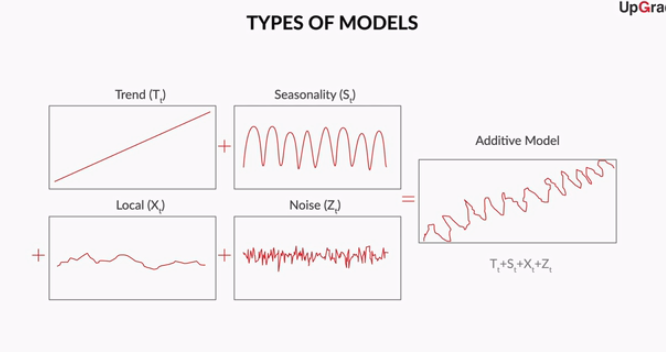
**Seasonality**

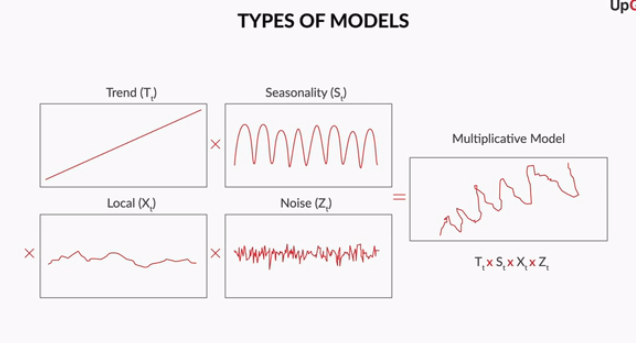
**Feedback :***When there are patterns in a data set that repeat over known, fixed periods of a time set, such patterns are known as seasonality. A season may refer to a time period as denoted by the calendar seasons, such as summer or winter, and commercial seasons, such as the holiday season. Companies that understand the seasonality of their business can time their inventories, staffing and other decisions to coincide with the expected seasonality of the associated activities.*







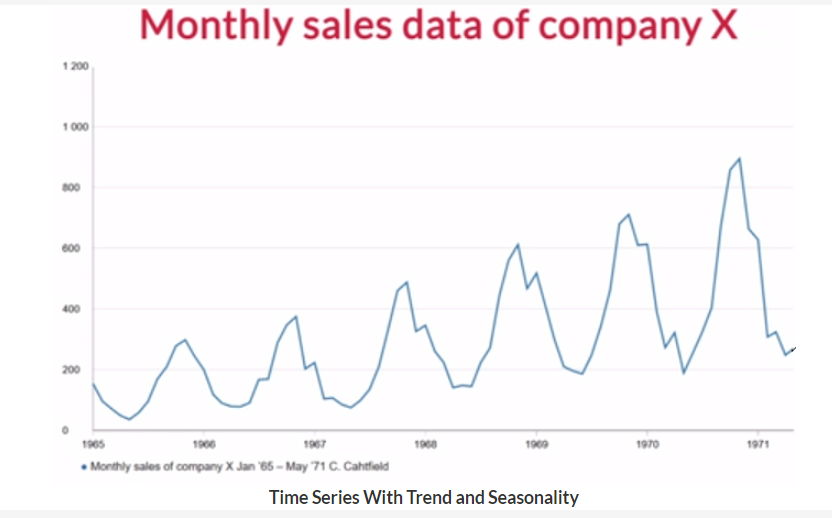




So the globally predictable component of a time series can be of two types — **trend** and **seasonality**

Here, the **trend** refers to any pattern that talks about the **overall increase or decrease** in the values, whereas **seasonality** refers to a **repeating pattern** of values seen in the data.

Trend and seasonality can also both appear in a time series. For example, let’s look at the **sales data**example from the lecture:

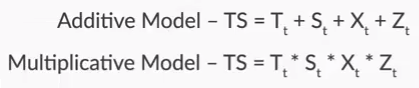


Now, clearly this data has both types of patterns; it has a **trend**, as the overall pattern suggests an **increase in sales**. However, the sales are also **seasonal**, as a similar up-and-down pattern is **repeating** itself **every 12 months**.

A similar example would be the sales of an ecommerce start-up such as Flipkart. You would expect Flipkart’s sales to increase every year, but you would also expect a seasonal variation on top of that, i.e. sales will be higher in the months around Diwali and lower in the off-seasons.

However, a time series can also have only a trend or only a seasonality. For example, the stock market data you saw in the previous section has only a trend in it and no seasonality.

You also saw the different ways in which these components of the time series can be related to each other. Their combination can either be **additive** or **multiplicative**.



**Components of Time Series - II**

Consider a time series which is represented as Y = T\*S\*X\*Z, T:Trend; Seasonality; X: Local, Z:Noise. How can you convert this multiplicative model to an additive model?

Top of Form



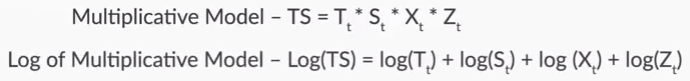
**Taking logarithm of the variables on both side of the equation**

**Feedback :***Recall that log(a\*b\*c) = log a + log b + log c. So, if you observe that a time series can be modelled as a multiplicative model, you can take logarithm of all the attributes at the onset and then model it as an additive model.*

As a broad rule of thumb,

* When the **magnitude** of the seasonal pattern in the data **increases** with an increase in data values, and decreases with a decrease in the data values, the **multiplicative model** may be a better choice.
* When the **magnitude** of the seasonal pattern in the data **does not directly correlate** with the value of the series, the **additive model** may be a better choice.

You typically work with additive models. An easy way to **transform** a multiplicative model to an additive model is to carry out the analysis on the **log** of the values in the time series.



Below are given two sets of information (A - C, I - IV) - examples and the components of time series. Select the correct option which maps the examples with the appropriate time series component.

A) Death rate decreased due to advance in science

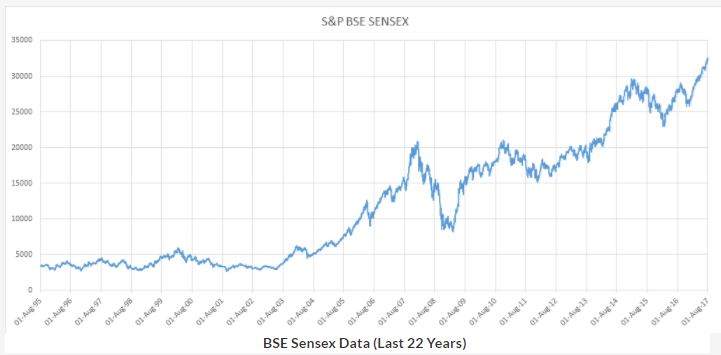
B) A fire in a factory delaying production for some weeks.

C) The sale of air condition increases during summer

**A = I, B = IV, C = II**

**Feedback :***Death rate decreased due to advance in science: It is a pattern which has taken place over time gradually and hence a trend. A fire in a factory delaying production for some weeks: It is an unpredicted event causing some unpredictability in the data and hence it is noise. The sale of air condition increases during summer: It is a periodic event occurring every summer.*

Let’s revisit the stock market data example from an earlier session. However, this time, let’s look at the **BSE data** for the last 22 years, instead of the last 15 years.



Clearly, you can see that the data not only has a **trend**, but it also has some **cyclicity**. Clearly, the BSE index goes through some up-and-down **cycles**; it can possibly be related to economic depressions, booms, etc. However, the **period** for these up-and-down cycles is **not fixed**, and this is where cyclicity is different from seasonality.

In general, you will see such cyclicity in most macroeconomic contexts, provided you have the data for a sufficient number of years.

We will not go into the modelling of cyclicity in this course, as it is too complex and will not be needed in many time series analyses.

Once you’ve seen the time series once, it is time to model the global pattern and then, look at the local pattern of the data.

So first, let’s see how the global pattern would be modelled.

Let’s say that you are modelling the global pattern as a+bt. Using R, the equation you get for the global pattern will be



**Exchange Rate = 111.82 - 0.22\*Month**

**Feedback :***Remember that the global trend is captured using regression. If you create a linear regression model on the original data frame, with the exchange rate as the dependent variable and the month as the independent variable, you will get 111.82 as the intercept and -0.22 as the coefficient of the variable 'Month'.*

Which of the following components does the time series have, based on what was discussed earlier?

**Feedback :**Note that the value of the exchange rate, as a whole, only declines as time increases. Other than the steady decline, there is no clear seasonal pattern that repeats itself regularly.

Now that you’ve modelled the globally predictable part of the time series, you can move on to the locally predictable part.

First, create the variable globalpred, which contains all the predictions made using the global model (Y = 111.82 - 0.22T). Then, subtract the global predictions from the original values, and store them in a variable called localpred.

The value of localpred for timestamp 300, rounded to 2 digits after the decimal point, is

After creating the variable localpred, you can find the value for timestamp 300 using the command localpred[300]. This value is 3.13.

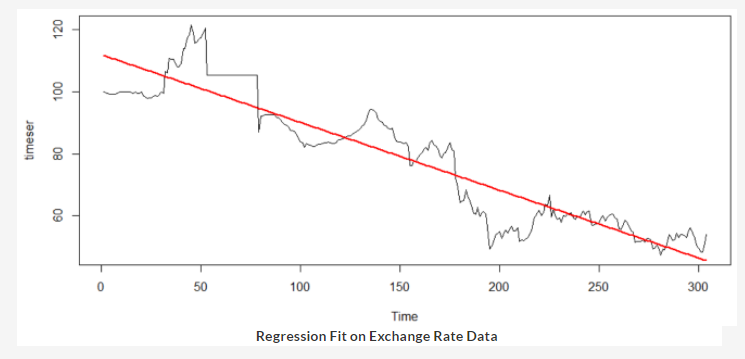
So when you started off with this time series, the exchange rate TWI was from 1970 to 1985.

**Exchange Rate TWI (1970-1985)**

Now, after **fitting a regression line** on it, you found that the **global trend** in this series can be represented by the equation Tt=111.82−0.22tTt=111.82−0.22t. This means that, on average, the exchange rate decreases by 0.22 every month.

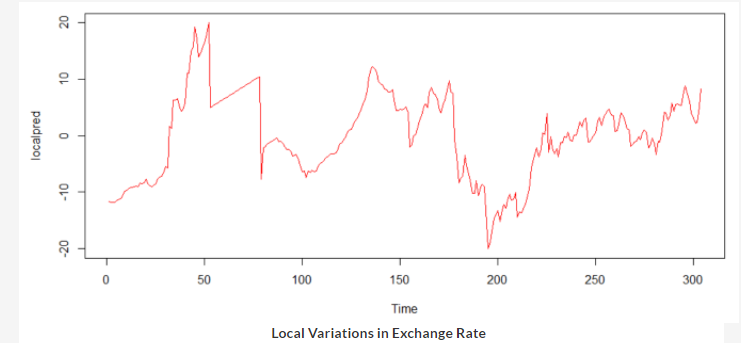
By the way, you only modelled the trend in this series, because as discussed earlier, this time series only has a trend. It does not have any seasonal component. If a seasonal component was present, you would have modelled that too, by fitting a sin or a cos function.

**Regression Fit on Exchange Rate Data**



**Regression Fit on Exchange Rate Data**

However, you can also have some **local variations**on top of the global prediction. These cannot be modelled simply by using linear or advanced regression.



These are the variations that are **dependent on the immediate past**. For example, if the exchange rate has been falling too steeply lately, you expect it to increase soon and become consistent with the global prediction.

So, the time series you have (Exchange Rate) can be broken down into two parts:

**Two Parts of a Time Series**

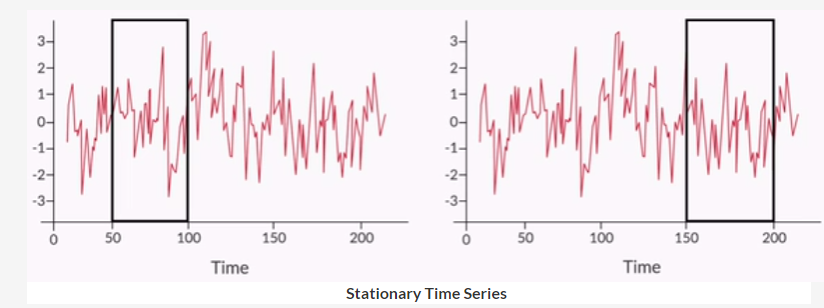
Here, the left graph tells you about the global trend of the data, while the right graph is what is left after you subtract the global trend from the series.

So the series on the right, when **modelled**, gives you the **locally predictable part** of the model. But how is this locally predictable part modelled? You'll learn this in the next session.

Also, assuming that everything has been modelled correctly, if you subtract your prediction (global + local) from the original values, you will get white noise. Now what is white noise? How can you identify if a given time series is white noise? These are also questions we will address in the next session.

In general, the different steps of the time series modelling process can be summarised as follows:

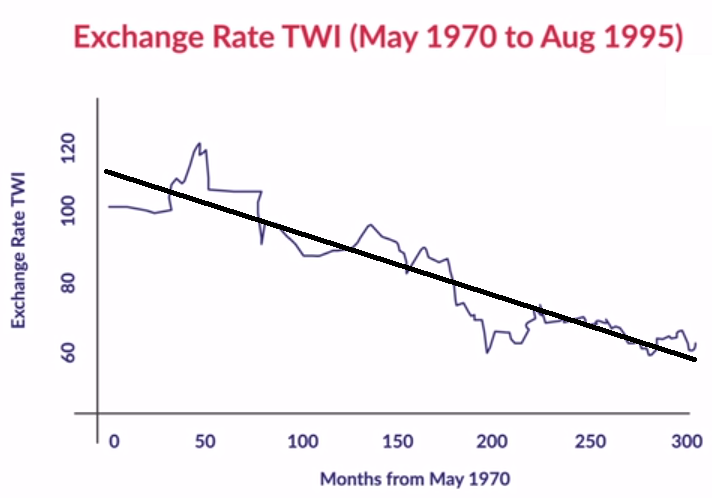
1. Visualise the time series.
2. Recognise the trend and seasonal component.
3. Apply regression to model the trend and the seasonality.
4. Remove the trend and seasonal component from the series. What remains is the stationary part: a combination of the autoregressive, and white noise.
5. Model this stationary time series.
6. Combine the forecast of this model with the trend and seasonal component.
7. Find the residual series by subtracting the forecasted value from the actual observed value.
8. Check if the residual series is pure white noise.
9. So as the professor said, if a time series is stationary, its **statistical properties**will be the **same throughout the series**, irrespective of the time at which you observe them.
11. In other words, for a stationary time series, properties such as **mean, variance, etc.** will be the **same for any two time windows** that you pick.



In general, a stationary time series will have **no long-term predictable patterns** such as trends or seasonality. Time plots will show the series to roughly have a **horizontal trend** with the **constant variance**.

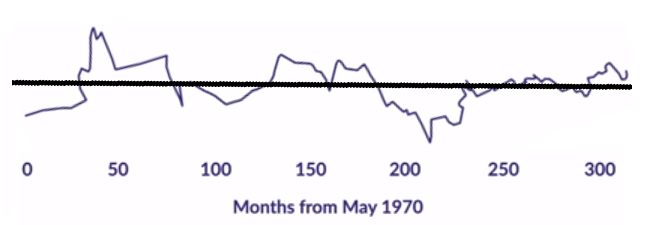
So why is stationarity so important? To understand this, let’s go back to the exchange rate TWI example. If you remember, you were trying to predict what the exchange rate’s value would be in the near future.

For that, you modelled the global trend of the data using linear regression, which is shown here using the black line.



**Modelling Global Trend**

Then, after removing the global pattern from the data, you got the following series:



**Series Left After Removing Global Trend**

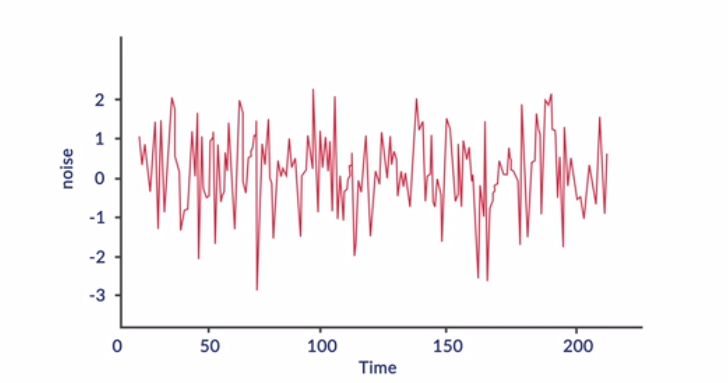
The regression model should capture all the variations caused in the exchange rate due to the trend and seasonality. When you model the residual series — the one you get after subtracting the regressed series from the original one — there should be **no trend or seasonality left** in it. Using the residual series, you only want to model the local patterns, i.e. the dependence of values on past values. You don’t want any global patterns to seep in there.

But how do you model this stationary (de-trended and de-seasonalised) series? We will look into this in later sections.

**White Noise**

So far, you learnt that a time series has three main components: the **global pattern** (which captures both the trend and seasonality), the **local** (predictable) pattern which should be stationary, and the (unpredictable) **noise**.

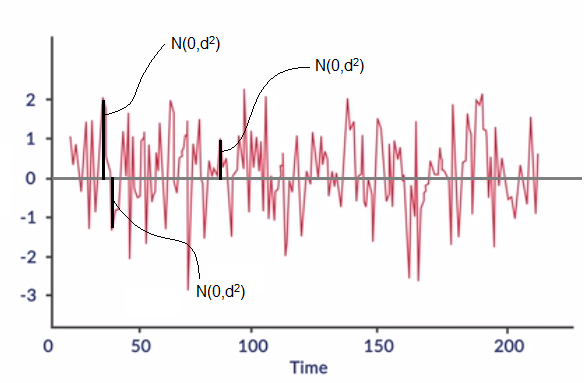
Let's understand noise in detail. Remember that noise (or **white noise**) is what remains when all the predictable parts of a time series have been modelled and extracted from it. It is a set of **independent and uncorrelated** values. If you plot white noise over time, it will look something like this:



**White noise**

Notice that there are no identifiable trend, seasonal or cyclical components. So a white noise series is basically an example of a stationary series. Let's learn more about it.

So **white noise** is basically a series of values that are all **independent**. Typically, for noise, you can assume that all these values come from a **Gaussian distribution** with a **zero mean**.



**White noise**

Remember the methodology for modelling a times series. At the **end of the analysis**, you will need to know if the series you have is **pure noise** or not. However, since the graphs for stationary series and white noise look so similar, it may actually be a little difficult to tell the two apart.

As you saw above, only performing a visual inspection of the time series plot will not help you confirm if the series is white noise. This is because any stationary series would resemble a pure white noise series.

Hence, you will have to perform some **concrete tests** to check whether the series is noise, or if it is just a stationary series. As you learnt earlier, if the series is white noise, the values in it will belong to a **normal distribution**. Hence, you can test if the series’ values belong to a normal distribution or not. For this, you’ve learnt two tests:

* Histogram test
* Q-Q plot test

You can download the R-codes used by Prof. Raghavan, for the testing of white noise, from the link given below.

Other than the tests mentioned above, i.e. the histogram test and the q-q plot test, there are a few more popular tests that can be used to understand whether a series is white noise or not. Some of them are listed here:

1. Ljung-Box (Portmanteau) test
2. Turning point test
3. Difference sign test
4. Runs test
5. Rank test
6. ACF and PACF
7. As discussed earlier, there are some specialised tests that can help you distinguish white noise from an ordinary stationary series. These tests are necessary because visually, a stationary time series does not really look very different from white noise.
9. One such test is called an **autocorrelation function (ACF)**. Let’s listen to Prof. Raghavan as he explains what ACF is.

Now let’s look at strong and weak stationarity again.

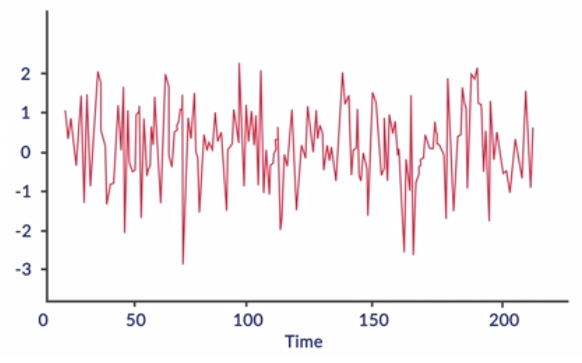
**Strong stationarity:** If a series is stationary, then the shape of the time series plot will look identical, irrespective of the time, t, at which you start collecting/observing the data. The time series 'shifted' to the right or left makes little difference to the shape of the plot.

**Weak stationarity:** If a series is stationary, then the pairwise relationships are preserved. In other words, the time series 'shifted' to the right or left makes a fixed difference to the shape of the plot.

The only two series that are stationary in the strongest sense are

1. White noise
2. Constant function

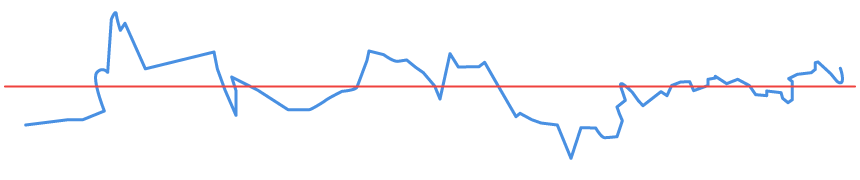
Remember the exchange rate time series example. At the end of the entire modelling exercise, when you have modelled both the globally and locally predictable parts of the time series, you will have to check whether the final residue is white noise or not.



**Final Residue (Original Series - Global Predictions - Local Predictions)**

In other words, you will have to check whether the **final residual series** is **strongly stationary** or not.

Also, in the intermediate step, you will check whether the series obtained after subtracting the global predictability from the original values is stationary or not. Actually, here, you check for **weak stationarity** and not strong stationarity.



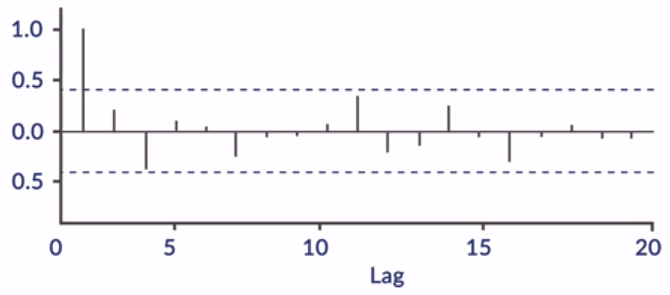
**Intermediate Residue (Original Series - Global Predictions)**

Think about it: if the series is strongly stationary, then that means the exchange rate’s values are actually not dependent on past values. Whereas, if the series is weakly stationary, then it means that the exchange rate’s values are dependent on previous exchange rate values, and they are dependent on them in a fixed way. There is no long-term trend or seasonality, but you do have some local predictability in the exchange rate data. The values of the past are dependent on the future, and you now model this relationship.

The **ACF** can actually help you **identify** and **distinguish** between **strong** and **weak stationarity**. First, let’s look at how it can help you identify strong stationarity, i.e. the presence of white noise.

So for white noise, the autocorrelation function (**ACF**) is **zero** for all **non-zero lags**. In other words, past values are not correlated with present values at all, making the series a sequence of independent, uncorrelated values.

However, in reality, the value of correlation will not exactly be equal to zero. Hence, using hypothesis testing, you can check if it is significantly different from zero.



**ACF Plot for White Noise**

So if the correlation value is **between the blue lines** (that signify the upper and lower limits of the confidence interval), you can say that it isn't significantly different from zero; and hence, you can **take it to be zero**.

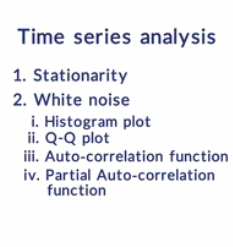
However, the autocorrelation function is not the only way to differentiate between white noise and a simple stationary series. You also have the **partial autocorrelation function (PACF)** method.

Play Video

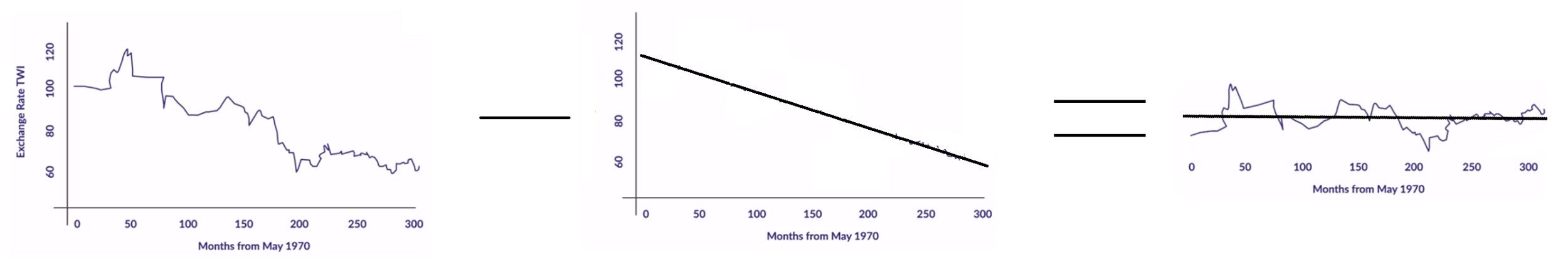
Notice that the ACF looks at the correlation between values at timestamps that are 'h' unit steps away — say, XtXt and Xt+hXt+h. However, this correlation is a sum result of several '**partial**' **correlations**between Xt+hXt+h and other XiXis (such as Xt+1,Xt+2,Xt+3,Xt+1,Xt+2,Xt+3, …. up to Xt+h−1Xt+h−1). It may be useful to try '**isolating**' the **direct correlation** between just Xt+hXt+h and XtXt, without the influence of any of the intermediate values. This is how **PACF** functions.

Now let’s say you want to understand how the exchange rate in month 12 is related to the exchange rate in month 8. Here, when the ACF defines their relationship for you, it is actually the incremental effect, i.e. the effect of months 8, 9, 10, and 11 on month 12. However, PACF simply isolates the effect of month 8 out, and it tells you that the exchange rate in month 8 affects the exchange rate in month 12.

For a more elaborate and mathematical understanding of these concepts, please refer to section 4.1 of the lecture notes.

By now, you have a basic understanding of white noise and the different tests you can carry out to confirm if a series given to you is, in fact, white noise.

Now, let's go back to the locally predictable pattern, i.e the weakly stationary time series. Remember the exchange rate example:



**Intermediate Residue (Original Series - Global Predictions)**

Once you model the global pattern of the data and subtract it from the original series, you should be left with a time series that has no trend and seasonality in it, i.e. a time series that is stationary. Now if it is stationary, it could be weakly stationary or strongly stationary. **Weak stationarity** would imply that the values of this time series **depend on past values** in a fixed manner. So there is **local predictability** of some kind that is present.

This raises two questions:

1. How can you **identify** if a given time series is weakly stationary or not?
2. If a time series is weakly stationary, how do you **find its equation**?

First, let’s look at how the second question, i.e. modelling the weakly stationary series and finding the correct equation, is addressed.

We have two basic types of time series models:

**Autoregressive (AR) model**

An autoregressive time series is one where the value at time 't' depends on the values at times (t-1)....(t-h) superimposed on a white noise term. You define an autoregressive time series AR(h) of the order 'h' as a series \{X_t\}_0^T where \[ X_t = \mu + \sum_{i=1}^h \alpha_i X_{t-i} + Z_t,~~~~Z_t=\mathcal{W}(0,\sigma^2) \] for some constants {\mu}and {\alpha_i}, 1\leq i\leq h. The coefficient \alpha_i  represents the influence (weight) of the value of the time series 'i' steps in the past, on the current value.

**Moving average (MA) model**

A moving average time series is one where the influence of the noise at some time step 't' carries over to the value at t+1, or possibly up to t+h for some fixed 'h'. Formally, a moving average time series \{X_t\}$ of the order 'h' (denoted MA(h)) is \[ X_t = \mu + Y_t + \sum_{i=1}^h \alpha_i Y_{t-i} \], where \{Y_t\}=\mathcal{W}(0,\sigma^2)} for some constants {\mu} and {\alpha_i}, 1\leq i\leq h. Note that this gives you a process that is centred around the constant value {\mu} . The value at time 't' in an MA(h) process is, therefore, the noise at the current time, t, superimposed on the cumulative weighted influence of the noise at 'h' previous timestamps.

Having looked at MA(q) and AR(p) separately, you can now combine the two into a composite ARMA(p,q) model.

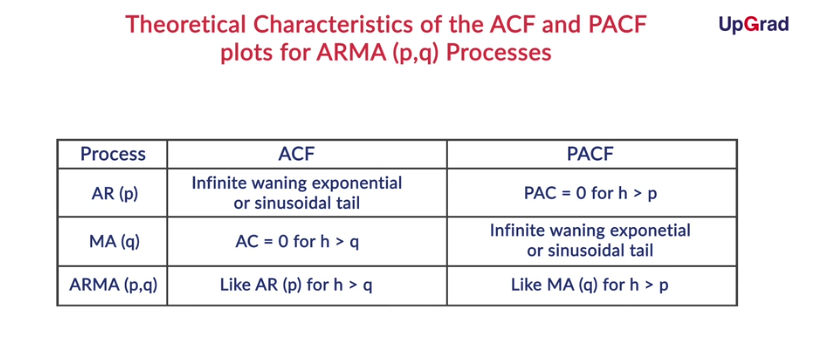
**Autoregressive - Moving average (ARMA) model**

A time series that exhibits the characteristics of an AR(p) and/or an MA(q) process can be modelled using an ARMA(p,q) model. Formally, you define an ARMA(p,q) process as a series $\{X_t\}$, where

## \[ X_t = \mu + \sum_{i=1}^p \alpha_i X_{t-i} + \sum_{i=1}^q \beta_i Z_{t-i} + Z_t \].

So as you saw, there are two formal tests for checking the stationarity of a time series, both with opposite null and alternate hypotheses.

Therefore, you look for the cutoff value in the **PACF** plot for the most optimal p in the **AR(p)** model, and the **ACF** plot for q in the **MA(q)** process. For the ARMA(p,q) process, you need to find the cutoff lag values from both the ACF and the PACF plots. This can be summarised in the following table:



* **ADF test:** Null hypothesis assumes that the series is not stationary
* **KPSS test:** Null hypothesis assumes that the series is stationary

Here’s a summary of all the tests you saw so far:

1. Tests for **normality (white noise/QQ plot)**: Tests for strong stationarity (white noise)
2. **ACF/PACF**:
   1. Test for **strong stationarity (white noise)**: ACF/PACF should not be significantly different from zero for non-zero lags
   2. Test for **weak stationarity (local predictability):** Check ACF/PACF plot patterns for patterns similar to the AR, MA, and ARMA processes
3. **Formal tests (ADF, KPSS):** Tests for strong stationarity (white noise)

A brief explanation of each of these measures is given below:

* **Log likelihood:** This refers to the log of the likelihood (probability) of the given data set being generated by the chosen model. The higher the likelihood, the better the fit of the model to the data. Note that under the Gaussian assumption, the model that maximises the likelihood also minimises the cumulative square error. Since probabilities have to be < 1, the log likelihoods are all negative.
* **Akaike information criterion (AIC):** This is a theoretical measure of the information in the model. It also takes into account the model complexity. In this case, as the order (p+q) of the model increases, the model becomes more complex. As you saw in the module on regression, in general, the more complex the model, the more prone it is to overfitting. Therefore, you try to pick a model that has an AIC that's as small as possible.
* **AIC corrected (AICc):** This is the AIC measure corrected for the size of the data set, i.e. in this case, the length of the time series 'n'.
* **Bayesian information criterion (BIC):** This is a measure that's similar to the AICc; however, it's arrived at using Bayesian methods. Again, the lower the BIC measure, the better the fit of the model.

So the MAPE (mean absolute percentage error) is given by

MAPE=100n∑ni=1∣∣yi−^yiyi∣∣MAPE=100n∑i=1n|yi−yi^yi|.

It is **similar to R-squared but still slightly different.** Some pros and cons of using the MAPE are

1. R-squared punishes big deviations very strictly compared to the MAPE.
2. The MAPE will not perform well if one of the actual data points is equal to zero.
3. It is difficult to interpret as a percentage.
4. The MAPE is known to favour models that consistently predict lower values. This may introduce bias.

# Summary

You covered a lot of conceptual ground in this session. Let's revisit some of those topics:

1. Stationarity
2. The two components of a stationary time series: white noise and the autoregressive part
3. ACF and PACF functions and their plot
4. Making a time series stationary
5. Modelling a stationary time series:
   1. AR model
   2. MA model
   3. ARMA model
6. Evaluating a time series model
7. So, the ACF and PACF plots already gave you some hints about the fact that the series you're looking at is an AR(1) series.
9. After that, you could have tried to find the best fit ARMA (p,q) by finding the log likelihood, AIC, etc., as discussed at the end of the session. Another way is that you just let R do it for you. Using auto.arima, you can find the best fit p and q for the ARMA (p,q). It gives you AR - 1, I - 0, and MA - 0. Ignore the I part for now; we will look into it later (anyway, it is 0 here, so it is absent). So this is an ARMA (1,0) model, or, in other words, it's an AR(1) model.
11. Now that the globally and locally predictable parts of your time series have been modelled, you can make forecasts by combining them. With that done, you can move on to the final part. In the final part, you will check whether the residue that you get after subtracting the forecasts from the original values is white noise (strongly stationary) or not.

So all the indications point to the fact that the final residual series is strongly stationary, i.e. white noise. The **ACF plot of the residuals** for the variable armapred is **entirely within the blue lines** that define the confidence intervals. Also, the **ADF and KPSS tests** suggest that the series is strongly stationary, i.e white noise.

So, you have gone through the methodology for modelling time series. In general, the different steps involved in time series modelling can be summarised as follows:

1. Visualise the time series.
2. Recognise the trend and seasonal components.
3. Apply regression to the model trend and seasonality.
4. Remove the trend and seasonal components from the series. What remains is the stationary part: a combination of the autoregressive part and the white noise.
5. Model this stationary time series.
6. Combine the forecast of this model with the trend and seasonal component.
7. Find the residual series by subtracting the forecasted value from the actual observed value.
8. Check if the residual series is pure white noise.

In the next section, you will go through another method of time series analysis, which uses **ARIMA modelling**. But what is ARIMA modelling, and how is it different from ARMA modelling? Let’s explore this in the next session.

You will look at another approach that can be used to make the time series stationary. You can use the in-built **ARIMA function**, which uses the method of **differencing** to make the data stationary, before modelling it as an ARMA process.

You will look at another approach that can be used to make the time series stationary. You can use the in-built **ARIMA function**, which uses the method of **differencing** to make the data stationary, before modelling it as an ARMA process.

# Time Series Smoothing

You would have noticed, from the examples of time series shown earlier, that visually 'spotting' a trend or seasonality is made unclear by the local 'spikiness' of the data; the 'spikiness' could be due to noise or a very strong autoregressive behaviour. From earlier examples, you have also seen that visually autoregressive data is hard to distinguish from pure noisy data.

**Smoothing** is the process of making the curve smoother by 'averaging' out the noise to make the trend and seasonality more apparent. We will explore a few common ways of smoothing, in this segment.

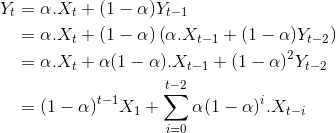
So, a **moving average smoothing** algorithm with a window size of say, 5, would replace the original value xtxt with ytyt, where yt=(xt−2+xt−1+xt+xt+1+xt+2)/5yt=(xt−2+xt−1+xt+xt+1+xt+2)/5.

However, the moving average is not the best smoothing technique available to you. Let’s listen to Prof. Raghavan to understand why the moving average smoothing method is not the best and how its shortcomings can be dealt with.

The **exponential smoothing** technique is the most popular way of smoothing. In this scheme, the current value is replaced by a weighted sum of the current value and the previous smoothed value. Exponential smoothing takes the following form:

\[ Y_t = \alpha.X_t + (1-\alpha)Y_{t-1},~~~~Y_1 = X_1 \] for some parameter 0<\alpha<1

'Unrolling' the recursion in the expression above, you get



Notice that the 'smoothed value' $Y_t$, in this case, is just a weighted sum of all the historical values in the time series, with the **weight dropping exponentially**as you go farther away from the current time.

So in R, you can implement the **moving average** smoothing method using the command **filter()**, and you can implement **exponential** smoothing using the command **HoltWinters()**.

Also, you saw how the value of $\alpha$ affects the level of smoothing. Small values of $\alpha$ result in higher levels of smoothing (this may result in 'distorting' the original time series), and large values (close to 1) will not result in any smoothing at all.

The model you will use here will be a composite combination of the various components of a time series. The broad expression for the model is

TS=Tt∗St+Xt+ZtTS=Tt∗St+Xt+Zt,

where TtTt, StSt, XtXt, and ZtZt refer to the trend, seasonality, local part (autoregressive behaviour), and noise respectively.

But you're probably wondering why you're using such a weird combination. Well, by hit-and-trial, you were able to observe that the model above was able to perform better than a pure multiplicative model.

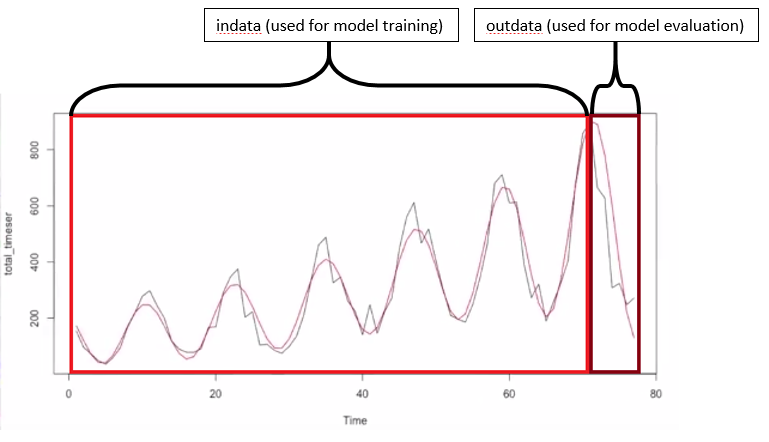
So now, let’s move on to the model fitting process.

So, the**first residue** (Tt∗St−TSTt∗St−TS) is supposed to give you the **locally predictable component** of the series, since it is obtained after subtracting the trend and seasonality from the original time series (TS). However, here, it turned out to be an AR(0,0,0) model. This simply means that the model is something like this:

TS=Tt∗St+ZtTS=Tt∗St+Zt

XtXt is not present at all. However, you still went through the entire modelling process as if a non-zero local component (XtXt) is present. You've done this so that the coding exercise can serve as a template for other cases where that component will actually be present.

So that concludes the classical decomposition part of the end-to-end analysis. As you can see, even visually, it looks like the model did a decent job of fitting the data.



**Classical Decomposition Forecast**

# End-to-End Analysis - ARIMA

Now let’s start the second part of your end-to-end analysis. Here, you will see how the **auto ARIMA**model can be used to create a time series model for the same monthly sales data: the one that you just modelled using classical decomposition.

Notice that for a time series model, the predicting is done differently from a regression model. For a regression model, predictions are made using a given set of monthly values.

global\_pred <- predict(lmfit, Month = timevals\_in)

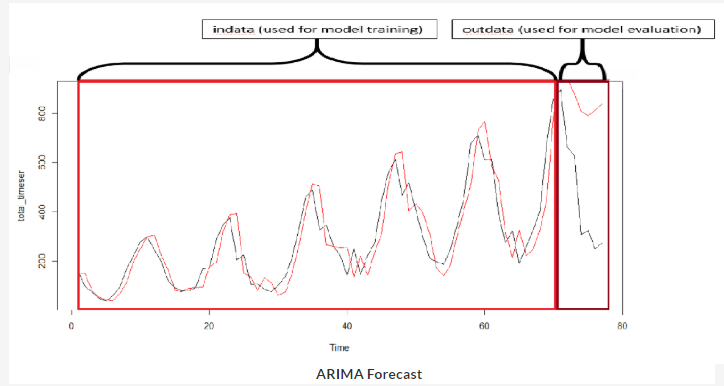
However, for a time series model, predictions are made using the number of months ahead for which they are needed.

fcast\_auto\_arima <- predict(autoarima, n.ahead = 6)

So that concludes the auto ARIMA part of your end-to-end analysis. Clearly, the steps involved in this method are fewer than in the previous one. It is a method that is very easy to implement, and this is a huge advantage over the classical decomposition method for creating time series models.

However, it does not fit the data very well. This can be observed visually through the following image:

image:



**ARIMA Forecast**

Notice how much the model deviates from the original values towards the end. This is clearly an indication of a bad fit.

Also, the MAPE value here is very high (137.1%) compared to the classical decomposition (39.1%).

So, to summarise, this is what you did:

* You plotted the original time series.
* You smoothened the time series using a simple moving average of a window size of 3.
* You fitted a trend line to the smooth time series, using linear regression.
* You removed the predicted values from the linear regression line in the time series, to get a stationary time series.
* You modelled the stationary series using the auto.arima() function.
* Finally, you used the auto.arima() function on the original time series itself.

You saw, in this case, that the predictions from the linear regression were significantly better than the predictions from the ARIMA() method. However, such results vary from data set to data set and must not be taken as a rule of thumb.

However, if you want a numerical figure that will summarise the quality of the fit for you, you can use MAPE, which, in this case, is 39.9%. Remember this value; you will compare it to the MAPE for the model created using auto ARIMA.

Bottom of Form