Time Series Modelling and Forecasting with Applications in R

Prof. Sudeep Bapat

Shailesh J. Mehta School of Management

Indian Institute of Technology Bombay

Week 03

Lecture 15: Practical Session in R-3

Thank you. Hello all, welcome to this new lecture in this course on time series modeling and forecasting using R. So, again, just to give you a very short brief about what we discussed in the last session. So, we explored more about types of seasonality, right? So, we focused more on seasonal models per se because seasonality is a really important idea for capturing non-stationarity. And then, towards the end of the last session, we also explored the SARIMA model.

So, the model structure and then what parts a SARIMA model contains, right? And then, right towards the end of the last session, we kind of discussed some pros and cons of modeling any time series data using either an ARIMA model or a SARIMA model. So, we will kind of quickly finish off with one or two more slides in the same spirit, and then we will kind of move on to a practical session in R, where we will try to explore more about, let us say, fitting an ARIMA model on a dataset and so on and so forth. So, the slide that you see in front of you kind of gives you some use cases of ARIMA and SARIMA models, right? I mean, why exactly are we studying all the modeling per se, right?

So, let's say, be it ARIMA or SARIMA, right? But even though the structure of both models is difficult, where exactly can you apply such models, okay? So, all these are kind of research papers, by the way, and one can actually find out more information by searching the title. So, for example, the first one is an application of the ARIMA model to forecast the dynamics of the COVID-19 epidemic in India. So, somehow, people try to kind of model the COVID-19 dynamics.

So, probably let us say how the cases are behaving or how the cases are growing on a timeline, right? So, back in, let us say, 2020 or 2021, right? And then they try to fit a suitable ARIMA model to forecast that. So, the specification is that this research paper utilized ARIMA to forecast COVID-19 case numbers in India. And then, after this paper or simultaneously with this paper, there were several other papers where people tried to model different ideas about COVID-19 data using suitable ARIMA or SARIMA models.

And down the line, let's say back in 21 or 22, people tried to apply some SARIMA models also because, by that time, they found out whether COVID-19 cases were seasonal for some reason. Because, let us say, the number of cases in summers might have been more, or the number of cases in winter might not be, or the number of cases in winter might be slightly lower, and so on and so forth. So, all these ideas correspond to capturing seasonality. So, the first use case is regarding, let us say, forecasting the dynamics of COVID-19 data.

The second use case is a time series ARIMA model for the prediction of daily and monthly average global solar radiation. And this is a particular case study from Seoul in South Korea. So, this is a study that forecasts solar radiation in South Korea based on the hourly solar radiation data obtained from a particular site, which is the Korean Meteorological Administration, over the last 37 years by using an appropriate ARIMA model. Okay. Then there was one more use case, which is, let us say, disease management with an ARIMA model in time series.

So how can you forecast the spread of a certain disease, or how can you manage it properly? So, another example of using ARIMA in disease management utilizes the wide applicability of either ARIMA or SARIMA models. And then this research paper touched upon some real-life use cases of ARIMA. So, for example, let us say there was a hospital in Singapore that accurately predicted the number of beds they would be needing in three days during the SARS epidemic. So, again, this data sort of relates to COVID-19 days.

And then the idea was that you could forecast the number of beds that the hospital would require in Singapore to house all the patients suffering from COVID-19. And the last use case could be something like forecasting demand using a suitable ARIMA model. So, this is more from an economic point of view. So how can you forecast some demand and supply trajectories and so on and so forth? So, this use case focuses on modeling and forecasting demand in a food company using ARIMA.

So here, due to time constraints, I've listed just four use cases. But if somebody is really interested, they can actually Google thousands of different use cases of ARIMA and SARIMA models. So, just to close this session, I'll give a few comments that the ARIMA model itself is a very simple-looking model. And then the very first model that people try out to model or forecast any time series data is, in fact, ARIMA. Because what happens is, with ARIMA, you can actually capture the trend part, right?

And then generally, we see that if you want to model any time series dataset, then there is a very drastic trend which is present, right? So, let us say either quadratic or linear or some power trend, something like that. So, the presence of a trend kind of supports the usage of a particular ARIMA model. And on the other hand, if you have a combination of a trend and seasonality, then one can't shy out by using a SARIMA model. Because SARIMA is nothing but an extension of ARIMA, which also incorporates the seasonality and the trend.

So, I think now what we will do is we will shift our attention to a practical use case. So, we will discuss some dataset and so on and so forth in R. So again, this is a very typical-looking R window that you see in front of you. And then for today's session and for this particular worksheet, you require these two packages. So, the first one is T-Series, right?

And the second one is Forecast. Now again, if you sort of forgot as to how do you install the packages. So, one can actually go to Tools and then click on Install Packages. And then in this window here, you simply write down T-Series or Forecast. So, you simply write down the name of the package, right?

So here... For this session, we require two packages. So, T-series and then forecast. And again, one small point to note here is to make sure that this repository CRAN is selected, right? And then not the other one, okay?

So, one has to install both these packages from the web, right? So they actually have to select the repository. So, for our packages, you have a repository containing all these packages. So, if you want to install a package from the repository, make sure that this is selected here. And then click on install.

So, since we already installed both these packages, I will not install them again. And the next important step is, once you install a package, make sure that you call those packages in the R environment. By using the library command. So library T-series, we will run this. And then the second one is library forecast.

And then this is the console window, right? So, if you have any errors regarding any of the packages, they'll basically show up here. So here in our case, we don't have any errors. So, the library forecast has been called in the R environment. And even here, if you see, the library T-series has been unpackaged successfully, right?

So, once you install any packages, make sure that you call the packages in the R environment using the library command, okay? Alright, now the data set that we will be using for the current worksheet is an interesting one coming from a banking domain. So, this data set gives you the monthly volume of commercial bank real estate loans in billions of dollars. So, let us say you have some commercial bank, and that bank kind of gives out some real estate loans. And then, what exactly is the monthly volume of such loans in billions of dollars?

So, this is a very practical use case coming from the banking domain, which sort of focuses on, let's say, modeling the bank real estate loans or, let's say, forecasting the bank real estate loans on a monthly basis. So again, this data set is stored in a text file, by the way, on my desktop. So, this would be the correct idea to call that. So, as dot ts, right? So, what we are doing is we are kind of converting all the observations to a time series framework.

So, this as dot ts ensures that all the values are converted to a time series kind of data structure, right. And then the scan command. So, the scan command is reading the data in the R environment. So, if you have a data set, be it a text file or an Excel file, which is stored on the desktop, the scan command ensures that you are bringing that data set or basically loading the data set into the R environment. So, if you run this, then we are bringing the data here.

And then here, if you see in the console, it says read 70 items. And then, if you want to see briefly what exactly your data is or how exactly your data looks like, you can simply type in the data name, right. So, in this case, we have given the data a particular name, right. So, bank underscore case, all right. So, you can actually input that here and then try seeing how exactly the data set looks like, okay.

So, precisely this is exactly how the data set looks like. So, this is a particular time series. The start value is 1, the end value is 70, and then the frequency is 1. So again, how many observations are there in the data set? You have 70 observations.

And then each one of these observations is a monthly observation, right? Because these are nothing but monthly volumes of real estate loans for that commercial bank. So let's say in the first month it was 46.5. Then in the second month it was 47. Then it rose to 47.5, and so on and so on.

Now again, one point to note here is that since this is a monthly kind of data set, the values from 1 to 12, which you see in the first row, coincidentally by the way, correspond to a particular year. And then the 13th value is the same value in the same month as the first one, basically. Now again, which month is the first value and the 13th value, we don't know. So, we have to dig deeper into the data set to find it out. But whichever month gives you 46.5, that same month gives you the value of 55.6 in the next year.

And then similarly, 60.1, then 59.7, 63.9, and then lastly 75.2. So essentially, this entire first column of observations you see. Starting from 46.5, 55.6, all the way up to 75.2. These correspond to the monthly volumes of commercial bank real estate loans corresponding to different years. And then similarly, the second column, the third column.

So essentially, each column gives you a data set for that particular month over all the years. So, this is exactly how the data set looks. So again, going back to the coding window. So, essentially, the first thing we have to do is plot the data, right. So, here what we will do is we will sort of create three plots, right.

We will create three plots. So, let me just show you the first plot without running the par command. So, if you run this plot command, what you have. So, this is nothing but a very simple plot of the data set. So, if you zoom in.

This is exactly how it looks. So, something like that. Okay. So initially, the volumes are very low, and then the volumes are higher. Right.

So, you can see a trend here as well. Right. Isn't it? So, you can see a trend. And then, how many months of data do you have?

You have 70 months, which you see on the x-axis. And then the y-axis gives you the monthly volumes. Right. So, you do have a trend here, which is very evident, and hence the dataset is not stationary in the first place. So again, the idea of plotting the data in the first place is to visually check if the dataset is stationary or not.

So here, clearly, one can observe that you have a very strong trend here, right? And then, that trend may not be a straight line either, right? Because if you try to fit a straight line, that may not be very good, so you might require a slightly curved kind of relationship, all right? Okay. So the dataset contains some trend, and now I'll show you two other plots. The next plot is the ACF plot. Now, remember what exactly is ACF? So, ACF is nothing but the autocorrelation function, right? And then the plot after that is the PACF, or partial autocorrelation function. So now, the first plot is the ACF plot of the actual data. Let me zoom in.

So, this is the ACF plot of the actual data. And then here, clearly, you can make out that for any lag, any given lag all the way from 0 to 10, all the correlation values are outside the limits and hence significant, right. So, one can actually see significant correlations for each and every lag, which again points to non-stationarity in the data, right. Because remember, if you again go back to the sessions where we discussed AR models, ARMA models, MA models, right. So, there, how did we decide on whether a data set is stationary or not.

So, there should be some break in the correlations after some point, right? So, the correlations can either cut off or they should tail off at least, right? But here, over the entire spectrum of all the lags from 0 to 10, we can see that each and every correlation is significantly outside the limits. So again, these blue lines are nothing but the limits, right? So, such a structure of ACF is a very typical structure which amounts to non-stationarity. So, if you are kind of analyzing any time series data, let us say later on for some project or for some research paper and so on and so forth, then if you are analyzing a real data set and if you try to plot the ACF and the PACF of the actual data and there if you observe such a thing. So, such a thing kind of amounts to non-stationarity, which is a very, very standard thing, okay.

Now, at the same time, if you decide to plot the PACF, so PACF is what? So, PACF is the partial autocorrelation function, right? So, if you decide to plot the PACF, then the PACF plot is much more promising to us. So, let me zoom in, right? Because here you can see that besides the first lag, right?

All the other correlations are very, very small. So, hence again, if you remember what we discussed earlier, if you want to analyze or put forward a particular model for a time series, analyzing both ACF and PACF plots is important. In other words, one should not conclude simply based on one of them. So, either the ACF plot or the PACF plot. Because here, if you see in the PACF plot, one can actually conclude that the series is stationary because, after the first spike, all the other subsequent spikes are really small.

So, if you only see the PACF plot, one can have a slightly different kind of notion, right? But when you see the ACF plot, we saw that all the correlations were significant, and hence the dataset is not stationary, okay? So, the pattern that you saw in the ACF plot, so again, let me rerun the ACF plot one more time and then show you the ACF plot once more. So, if you remember, this was the structure of the ACF plot, right? Where all the correlations are significantly outside the limits.

So, if such a pattern is observed in either the ACF plot or the PACF plot, such a thing kind of amounts to non-stationarity in a sense. So, hence the conclusion is that one should not jump to any conclusion based on a single plot. So, one should always observe both the ACF plots and the PACF plots to come to some concrete conclusion. So, here again, clearly the dataset is not stationary, as suggested by the time series plot as well as the ACF plot, okay. Now, again, from a couple of sessions earlier, we discussed a hypothesis testing for testing whether the series is stationary or not.

So, if you remember the ADF test. So, what is the full form? So, ADF is Augmented Dickey-Fuller, okay. So, in R, you have a very simple command to apply that test. So, adf.test, right?

So, we will see what happens if you apply this ADF test on the actual data or the initial data, which is bank_case. So, we will run this, and again, the output is shown in the console. So, the name is Augmented Dickey-Fuller. And again, if you remember, this ADF test's whole idea is to check whether the time series is stationary or not. So, what conclusion are you getting here?

So, this gives you the name of the data. So, the data is bank_case. Then, the test statistic value of this Dickey-Fuller test is -0.25591. Then, the lag order is 4. So, up to how many lags have they checked?

So, up to lag 4, they have made the checks. And lastly, the p-value. So, here clearly you can see that the p-value is almost equal to 1, which is 0.99, and that is really high. So, if you have such a high p-value, what does that mean? So, if you have a high p-value, you fail to reject the null hypothesis, right?

If you have a smaller p-value, let us say 0.01 or 0.002, something like that, which is less than your alpha, for example, let us say alpha is 5%, then in that case, you can actually reject the null. But here, since the p-value is really high, which is 0.99, you actually fail to reject the null. And then the last line actually tells you what the null hypothesis is or what the alternative hypothesis is. So, in this case, my alternative hypothesis is stationary. Which means that you are basically going with the null hypothesis, which is not stationary.

So, again, just to summarize, since you are failing to reject the null, and in our case, the null hypothesis amounts to non-stationarity. So, the dataset is non-stationary. So, non-stationarity has been confirmed through all these things. So, the usual plot, the ACF plot, and the ADF test. So, now what could be done?

So, if you have a trend, right? So, we saw that you had a trend in the dataset from the actual plot, right? So, whenever you have a trend, it is always better to kind of difference, right?

So, it is always better to difference the data, right? So, again in R, we can do a, we can apply this very easy function called DIFF.

So, DIFF stands for differencing. So, we will try differencing the data once, right, because we have to do it sequentially. So, we cannot jump to, let us say, differencing the data three times in the first go, right. So, we have to differentiate the data once, then again check, then if not, again differentiate one more time, then again check, right. It is more like a recursive kind of exercise, right.

Okay. And then again, we are giving a different name to it. So, let us say bank underscore case underscore D1 is DIFF. So, we are differencing the actual data. So, we will do this right and then we will again perform the same test, which is the ADF test, on the differenced data now, right, to see if the differenced data itself is stationary or not.

So, we run this ADF test again, and you can actually see the conclusion in the console. So, the Augmented Dickey-Fuller test. And then, if you look at the p-value, the p-value is still large, right, which is 0.67. I mean, of course, the p-value is reduced from the earlier case. So, earlier you had 0.99, which you even see here.

So, now at least it is reduced. So, we are kind of moving towards stationarity, but then still, the first difference is not able to give you a completely stationary process. Because still, the p-value is 0.67, and then, since the p-value is large again, the same story that one fails to reject the null hypothesis, and remember, for any ADF test, the null hypothesis is what? That the series is not stationary. So, if you are not able to reject that, then it kind of tells you that the series indeed is not stationary.

So then what? So, if the first difference is not working, we will difference it one more time, right? So, the new name is bank underscore case underscore d2. So, d1 was the first difference, d2 is my second difference. So, here what we are doing, we are differencing the first difference, right? We are differencing the first difference, so you get something like that, and then you again perform the ADF test on the second difference. And now we will see what happens. Surprisingly, if you see, the p-value happens to be now less than 0.05.

So the p-value is 0.01157, which is less than a standard alpha of 5%. So now, in such a case, we can actually reject the null hypothesis and then go with the alternative, which says that the series is stationary. So, you have to actually perform this exercise until you get a stationary process, basically. So, you have to perform this differencing exercise until you get a stationary process. So, in this case, we started with the actual data and then we kind of confirmed using three things.

So, the actual plot of the data, the ACF of the data, and the ADF test all were kind of pointing to non-stationarity. So, hence, since you have a trend in the data, we thought of differencing the data. So, we tried differencing it once, but it did not work because the data is still not stationary. Then we differenced it twice, and then eventually the data became stationary. So, now let us see how the plot looks like.

So, the plot of the second difference. So, this is the plot of bank underscore case underscore D2. So, D2 is the second difference. So, we will see how this plot looks like. And now, essentially, the plot looks something like that.

And then here, clearly, you see that you have a fixed mean kind of thing. You don't have any trend. There is no seasonality also. And then the variance is almost equal. Because if you draw horizontal bands, then almost all the observations are falling inside those hypothetical bands.

So visually, at least, one can safely say that the data is kind of stationary. And one has already performed the ADF test. So, but then even visually, one can sense that the data indeed has become stationary, okay. So, once the data becomes stationary, then again for a visual check, we can perform or we can run the ACF plots and the PACF plots on the second difference data, okay. So now, how exactly does the ACF plot look like?

Is there any improvement from before? Of course, yes. So, if you see the ACF plot, then barring the first two correlations, all the other correlations are between the bounds, right. This is a clear indication that the series is stationary, but still, you have to look at the PACF plot, right. So do not jump to conclusions based on a single plot, like we discussed.

But then, the ACF plot is giving us a good story. Now, how about the PACF plot? So, if you draw the PACF plot of the second difference data, then one can actually see the same structure here, right? So, again here, barring the first correlation, all the other correlations are not that significant, okay? So, here, what is the conclusion?

So, based on both the ACF plots and the PACF plots, right, we came to the conclusion that the series is indeed stationary visually. And, of course, we tested it much more formally using the ADF test a short while back. So now, we started with the actual loans data initially. We differenced it once. It didn't work.

We differenced it twice. And then, the twice-differenced data is stationary. So, once you make any data stationary, then you can actually fit some model on it. So, line 27 onwards is nothing but we're trying to fit an ARIMA model or a suitable ARIMA model on the dataset. Now, we are giving it one name here.

So, let us say bank fit, bank underscore fit, and then in R, in the R framework, you have this very easy command called arima. So, one can actually apply arima to any data set. So, here essentially, if you take a note, you can pause the video and then see what line 28 is giving you. So, in line 28, we are trying to fit a particular arima model. What are the orders of the arima model?

So, orders are given by this command. So, C 0 comma 2 comma 1. So, C in R means concatenate. So, concatenate means combine, basically. So, if you want to combine any individual items in a vector form, then one has to use C, the C command.

So, here, what are we combining? We are combining the orders of the arima model. And now you may wonder why exactly these orders? So, I will tell you exactly why. So, for that, you have to first understand that we are trying to fit the arima model on which data set.

So, are we fitting on the initial data set, the first difference data set, or the second difference data set, right? So, here you can see that we are actually fitting the ARIMA model on the initial data set, right? Because bank_underscore_case is the very initial data set that we started with. So, if you are fitting a particular ARIMA model on the non-stationary initial data, then this middle order should be the number of times you are differencing the data, which is 2 in this case. So, we started with bank_underscore_case, then we differentiated once, and we got bank_underscore_case_underscore_d1, which did not work. So, we differentiated one more time, and then ultimately, we got hold of bank_underscore_case_underscore_d2.

But then, since you are applying or trying to model the initial data using the suitable ARIMA model, this middle order has to be 2 because this 2 suggests how many times you are and then, how did you choose 0 and 1? So again, I will tell you. That if you look at the ACF and PACF plots here,

So what do you observe? So, let me again rerun the ACF plot. So, this is the ACF plot. Which looks like that. And then from here, you can clearly see.

That is after the first lag. Right. After the first lag, the ACF plot is cutting off. Isn't it? So, after the first lag, the ACF plot is cutting off.

And if you observe the PACF plot now. Okay. So, what can one see? So, if you observe the PACF plot, we can see that after the first lag, the PACF is also cutting off. Right.

So, after the first lag, both ACF and PACF show some tailing off or cutting off tendencies. Right. So, ideally speaking, the AR order and the MA order of this ARIMA model cannot be very different from 1-1. So, essentially, we can actually take, let us say, 1, 1 here or at the max 2, 1. So, hence, we can actually play around with all these combinations.

So, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 2 at the max or 2, 1. So, 1 need not check beyond 2; it would not make sense. So, how did you achieve these orders? So, you have to look at both the ACF plot and the PACF plot. To kind of fix the AR orders and MA orders.

So, I will show you what happens if you run this command first. So, if you run this command, it has fitted that ARIMA model. If you want to see the details of the ARIMA model, then these are the details. So, essentially, which model are you fitting? So, we are fitting an MA model here because you do not have any AR orders, right?

So, this is an MA model, and here you see the MA coefficient. So, the MA coefficient is minus 0.37. And then here, you should watch out for this value. So, AIC. So, AIC is a particular information criterion.

It is called the Akaike information criterion. And then, the lower the AIC, the better the fit. Right. So, remember this AIC. So, how much is the AIC?

The AIC for this model, this particular ARIMA model, is 26.1. Right. Now, let me show you what happens if you change the order slightly. So, what would happen if you change 0 to 1? Right.

So, which model is this? This is ARIMA 1, 2, 1. So, basically, the idea is I am trying out different combinations, right? So, as we discussed. So, let us say if you rerun the fit and then again look at the AIC.

So now essentially the AIC has increased, right? So earlier the AIC was 26.1. But then for this particular model, 1, 2, 1, the AIC has increased to 27.9. So, this is no good for us, right? Because remember, you want to minimize the AIC as much as possible.

So, a lower AIC is better. So, let me again go back to 0, and you can actually check this for any combination you want. So, for any other combination other than this, the AIC would be higher. So, hence we will move forward with this particular ARIMA model, which is 0 to 1. So, then we can actually get hold of the fitted values.

So let us say all these are the fitted values as per the ARIMA 0, 2, 1 model. And now the very last thing is to forecast. So again, in R, we can make use of the forecast command. So, using which fits do you want to forecast? We want to use bank_underscore_fit data, which is nothing but our ARIMA model, right?

And then we want to forecast it. So, h being 24 means 24 months in the future or two more years in the future, so I'll show you how exactly the plot looks. So, this is exactly how the forecast looks for the particular ARIMA 121 model for the bank loan data. And here you can see that the blue line is the forecast, and the shaded regions are given in grey. So, dark grey and light grey give you some confidence bands along the forecasts.

But here, clearly, you can see that the forecast is making very, very good sense, right? Because it is kind of following the same path. And the forecast is kind of retaining the trend aspect also. So, this is essentially an ARIMA model fitted on the bank loans data. So, my suggestion is that if you want to practice more, try googling for some other very, very basic data sets coming from time series.

And then see what happens if you try to fit some very basic ARIMA model on that. So, are you getting some good forecasts? Are you initially able to kind of reduce any non-stationarity to stationarity by, let us say, differencing, right? And then, eventually, are you able to pinpoint some model, for example, 0 to 1 in this case, right? And then the last thing is forecasting.

So, this was a very simple exercise where you tried to fit an ARIMA model on a practical use case coming from the banking domain, all right? Thank you.