

Time Series Modelling and Forecasting with Applications in R

Prof. Sudeep Bapat

Shailesh J. Mehta School of Management

Indian Institute of Technology Bombay

Week 10

Lecture 50: Practical Session in R – 10

Hello all welcome to this course on time series modeling and forecasting using R ok. Now, we are almost through this week and then again just to give you a very faint idea as to what we covered this week is that. So, the broader topic throughout this week was modeling volatility using different kinds of time series processes. And then throughout this week, it has been slightly heavy.

I can understand that, again, if you're not very confident about some of the material we've covered this week. So, again, a very strong suggestion would be, again, to go back and then watch those videos diligently, sincerely, maybe take some notes on a separate sheet of paper, wherever required, because all these modeling techniques are slightly complicated. on the newer side as compared to all the models we have seen so far ok. And again just to press on the fact that we have seen number of different models and then many of them were kind of extensions of that GARCH model ok. But at the base level two very important models were studied this week which were ARCH and GARCH.

And both of these models had capabilities of handling volatility and modeling volatility you know and then trying to sort of capture the changing variance pattern which is seen in the practical data set. Now, this is as you see in front of you. So, we have an R session open up and then we will be simulating a few R's processes initially, we will be simulating a few GAR's processes initially and down the line we will try to fit R's and GAR's processes on some real data sets as well, ok. But again just to again ensure One important point is that you should keep in mind that where exactly can you implement ARCH processes or where exactly can you implement GARCH processes etc.

So there has to be some changing variance pattern which is seen evidently in the underlying time series. If you do not have any changing variance and then you see seeing that the variance is kind of constant then obviously applying any of the ARCH models or GARCH models or their extensions does not make sense. And of course just again a very

quick refresher as to Towards the end of last session or maybe last to last session, we have covered number of extensions of GARCH processes. If you remember, so, GJR GARCH, exponential GARCH, threshold GARCH, right.

So, all these have capabilities of, you know, controlling the symmetricities or asymmetricities which are seen in the dataset, ok. So, here we will open up a new fresh R worksheet and then initially as discussed, we will sort of try to implement some of the ARCH processes and GARCH processes on some simulated datasets, ok. So, as always we will start with incorporating some packages. So, the first one is FGARCH. So, obviously FGARCH has not been used earlier because FGARCH is sort of persistent or only focuses on capturing the GARCH capabilities of a time series.

involve that in the R environment now again just one disclaimer that if you have not installed the packages again you to go to tools and then click on install packages and then you can write down the name of the package here and then click on install ok so assuming that you have all these three packages installed in the R environment then obviously you can call them using library or require command Alright. So, the first package is FGARCH. Second package is RUGARCH. So, again pertaining to a GARCH modeling setup.

And then the third package I think we have used in some of the earlier R sessions as well which is called as QUANTMODEL. So, we will again make sure that again one very important point is even if you call some of these packages you have to again make sure that you do not get any errors in the console. I mean once a while due to some connectivity issues or some mismanagement in the package itself it might happen that R might throw some errors here. And then again, you do sort of Google as to what that error means, you know.

You do a little back and forth. But if you don't see any errors here, then you can assume that all the three packages are successfully loaded. So, as discussed a short while back, so the first initial task for us today will be simulating a bunch of ARCH processes and GARCH processes. So, again we will specify some set seed. So, again just to make sure that all of you understand the role behind set seed is that if you want to replicate this a number of times, then R will ensure that every time you get a randomized kind of an output.

So, you will start with the same thread or rather the same number 1, 2, 3, but you will produce a randomized output every time. Alright, and then we will specify how many

observations we want. So, n is 1000 here. Now, again one small point is that you can give any random number here. So, we have given 1, 2, 3, but then you can give any random number.

So, once you give any number here, let us say 4, 5, 6 or whatever you want. So, R will keep that number in memory and then every time you try to replicate the same code it will start from the same set or same seed value which is 1 2 3 in this case and then try to replicate the entire experiment as if a randomized trial is happening ok. So, hopefully this part is clear I mean `set.seed(1 2 3)` or for that matter any number is more from a syntax point of view. So, now once you specify how many observations you want, we have to specify all the parameter values.

So, let us say ω is 0.1, then α_1 is 0.5, β_1 is 0.4 and then α_2 is 0.3. So, we are keeping some values of all the relevant parameters beforehand. So, so until line 11 we have not specified the model right, but again if you are not very confident about the model structure you may have to again go back to some of the earlier sessions this week and then try to understand as to how the arch model looks like or how the general GARCH model looks like and so on and so forth ok. Alright, but just in case we are trying to specify all these parameters beforehand so that later on once we start specifying some of the models let us say ARCH 1 or GARCH 1 etc. So, you need not specify all the underlying parameters again and again ok.

So, you can keep all the fixed parameter values initially itself along with n ok. So, again just to recap very quickly we are specifying these 5 quantities predominantly. So, n , ω , α_1 , β_1 and α_2 . So, now the very first exercise rather the very first model we will try to simulate is a simple ARCH 1 model ok. So, autoregressive conditional heteroscedasticity with order of 1.

Now, again just for time brevity or time pausity I will not discuss the model structure here. So, we we have discussed each and every model structure which you see here in detail right or or it might happen that we might have sort of talked about arch m in general where m is some generalized order, but again whenever you have m if you replace m with 1 it will give you arch 1 ok or if you have if you want the order to be 2. Right. So, you extend that model up to that point. OK.

So, obtaining some specialized models. So, by specialized, I mean obtaining some fixed order model. Let us say ARCH 1, ARCH 2, ARCH 3, ARCH 4, etc. Shouldn't be a

problem if you have the general ARCH M model ready with you. And the exact same story with GARCH model.

So, if you have the general model structure for GARCH PQ. So, wherever you have P, if you replace 1 and wherever you have Q, if you replace 1, it will give you a GARCH 1 1 model. So, hopefully writing down the model structure or understanding the model structure should not be a problem here at this stage. So, the first model we try to simulate is simulating a ARCH 1 model and then we have this very convenient sort of a function which is inbuilt in R or rather inbuilt in one of the packages we have loaded earlier which is called as GARCH sim. So, GARCH sim of course, stands for GARCH simulator ok.

So, this would be a simulated sort of an ARCH 1 model. Now, again, one point to notice here is even if you are simulating an ARCH 1 model, the function name is GARCH, ok. Because probably, as all of you might know, that ARCH is one special case of GARCH, right. So, GARCH is sort of an extension of ARCH, or rather, ARCH is a special case of GARCH. So, R does not kind of differentiate between ARCH sim and GARCH sim, ok.

So, using this GARCH sim, you can still simulate from an ARCH 1 model. Make sense so far? Okay. So, initially, we will give it a name, let us say arch1, and then we will try to run this. And again, here, all this is part of the syntax. So, let us say model equals, and then here, if you see, you would specify the parameters.

So, omega, alpha1, beta, and then lastly, how many observations you want. So, I want 1000 observations, which is nothing but n. Okay. So, once you specify this entire chunk of code and give it a name, let us say arch1, I will show you the plot of the simulated object. So, let me zoom in. So, this is exactly how the simulated ARCH 1 model looks like, ok.

And clearly here, you might not be able to sort of see the changing variance, but again, if you look slightly deeper, you may find out that this is highly volatile, right. So, again, by highly volatile, I mean that you have some abrupt ups and downs throughout, okay. So, one cannot find out some fixed horizontal bands in which the entire series could be encapsulated. So, in a way, the changing variance problem is there. Make sense?

So, this is, and again and again, the next time you try to simulate the exact same model, it might give you a different picture because this is an entirely simulated sort of exercise, okay. But in any case, this is exactly how an ARCH 1 simulated model looks like. So, which is highly volatile, right, and then changing variance could also be seen here to

some extent, okay. So, we will close the plot again and go back to the code. Now, the next model we will try to simulate is an ARCH 2 model.

So, ARCH 2 stands for, again, we will use the same function 'guard_sim,' and then again, we will specify a couple of parameters here. So, omega, alpha 1, alpha 2, etcetera, okay, and then we will see how the plot of ARCH 2 looks like and whether you see any differences between ARCH 2 and ARCH 1, okay. So, if you trade ARCH 2, and then we can give it a name, let us say ARCH 2, okay. And then this is exactly how the plot of ARCH 2 looks like. So, again, I am using a slightly different color.

So, as to differentiate between let us say arch 1 and then arch 2. Now, again if you look from a very broader perspective, you might not be able to spot any differences between arch 1 and arch 2 and then again needless to say that again all these exercises are simulated based and then arch 2 is nothing but one extra order than arch 1. So, we should not hope that any drastic changes in changing variance would be seen. Now, again one similar point to what we seen earlier is that this even arch 2 is highly volatile.

So, you see abrupt ups and downs everywhere throughout the series. And hence changing variance problem could be clearly seen. Make sense? So, arch 2 is nothing but one extra order than arch 1. So, there is not a whole lot of a difference that you see here.

But at least you should understand that how do you simulate let us say ARCH 1 model, ARCH 2 model or let us say for that matter some higher order let us say ARCH 5 model etcetera. Alright, now the next module we will try to simulate is GARCH 1 1 alright and then this is the name we will give it. So, GARCH 1 1 and again exactly I specified a short while back that the function name does not change. So, even here we have this exact same function name which is GARCH SYN ok and we will sort of apply this GARCH SYN. The only difference here is that

Apart from specifying omega and alpha 1, you might have to specify let us say beta 1 as well. So, these are the underlying parameters of a GARCH 1 1 model. Now, just one small point to note if you go up to where we simulated this ARCH 2. So, again since ARCH 2 is an extension of ARCH 1 it requires two orders ok. So, here it is specify alpha 1 and then alpha 2 right because you are you are sort of extending the order up to 2.

But again if you notice we are we are not specifying any value of beta. So, we are kind of removing the beta from the equation because it is not a GARCH model to start with it is an ARCH 2 model. But, the moment you come down to GARCH 1 1. So, in GARCH 1 1

what happens is you have one order for the ARCH model and one order which captures the GARCH model ok. So, here you have to specify alpha 1 and then beta and then of course, n and then omega.

So, remember one thing that omega stays the same because omega is nothing but the intercept portion or the constant part of the model ok. So, this is exactly how we can specify or rather simulate from a GARCH 1 1 setting. So, again one important point to note here is that depending on which model you want to simulate from you can actually specify those many parameters inside the GARCH spec function ok. So, just to give an example let us say if you want to simulate from an ARCH 5 model instead ok. So, again you can pause the video and try to try to answer my next question is that how many alpha values would you require?

Again, just to repeat the question: if you want to simulate from an ARCH 5 model, then the question is how many alpha values you will need to specify. So, again, the answer is 5, right? So, one would require alpha 1, alpha 2, alpha 3, up to alpha 5. And again, since the model is ARCH 5, you will not have any beta. So, beta would be 0.

Does this make sense so far? So, we have tried to simulate from the GARCH 1 1 model. So, we will run that, and then again, I will show you the plot. So, again, as before, hopefully, the plot would not be a whole lot different from, let us say, either ARCH 1 or ARCH 2, because again, GARCH 1 1 is just an extension of, let us say, ARCH 1, okay? So, again, you can see that the series is volatile.

So, probably an ARIMA kind of structure would not work well here because you have lots of random ups and downs throughout the entire series, which suggests that the variance is not constant or rather is changing over time. So, this is the simulated data from a GARCH 1 1 model, and we can extend this further. So, just to show you one last simulated data set. So, what would happen if you want to simulate from a GARCH 2 1 model? So, again, before showing you the sort of parameters we require here, you may pause the video and try to understand what parameters you would need.

So, here for simulating from a GARCH 2 1 model, of course, it requires omega, then alpha 1, alpha 2, and then beta 1, ok? Because this is GARCH 2. So, the first order stands for the ARCH component, and then this 1 stands for the GARCH component. So, there have to be 2 alpha values. So, alpha 1, alpha 2, and then since the GARCH order is 1, you only require 1 beta 1, ok? So, alpha 1 again—if you scroll towards the right-hand side, you can see that omega, alpha 1, alpha 2, and then beta 1, and then, of course, N.

Make sense? All right. So, we will simulate from this GARCH 2 1 model again using the exact same function, which is GARCH_sim, and then we will see how the plot looks. So, this is exactly how the plot looks. So, let me zoom in slightly.

And again, so probably here you can see one small difference is that the volatility might have increased slightly. So, if you compare GARCH 1 1 with GARCH 2 1 or other ARCH 1 1, sorry. So, not ARCH 1 1, but then ARCH 1 with, let us say, something like GARCH 2 1, you can see that underlying this model, you can see a lot more volatility as compared to, let us say, either GARCH 1 1 or ARCH 1 or ARCH 2, etc. Now, again, how do you judge that? So, again

A lot more volatility means that you have lots and lots of ups and downs, let us say, of increased heights. So, the peaks and troughs are slight. So, in this GARCH 2-1 model, the highs and lows are much higher and lower, correspondingly, as compared to, let us say, something like GARCH 1-1 or ARCH 1 kind of a model. And, of course, this would be one slightly different characteristic because, as you go on increasing the orders, you are trying to sort of control for that much complexity in the model. So, obviously, if you compare ARCH 1 and then GARCH 2-1, obviously GARCH 2-1 is an extension of ARCH 1.

So, GARCH 2-1 would be much more capable of capturing highly volatile time series or where variance is changing abruptly and much more predominantly as compared to ARCH 1. So, hopefully, one should keep all these points in mind. So, again, this is a simulated sort of exercise, but how do you compare between processes? So, tomorrow, if somebody asks you what exactly the difference is between, let us say, ARCH 1 and then GARCH 2-1. So, again, this exactly should be the answer.

So, now the next data is slightly different, and again, if you are not very confident about its model structure. So, we have covered this in the very last lecture itself. So, again, this is an extension of GARCH, and then you have two different components. So, ARMA plus GARCH. So, ARMA 1-2 plus GARCH 1-1 model to capture the volatility.

So, you have an ARMA model inbuilt to capture the mean structure of the model and then you have this added a GARCH 1 1 model to capture the changing variance. So, how do you simulate something from an ARMA 1 2 plus GARCH 1 1 model? So, again as always or rather as assumed it would have a two leveled kind of a model. So, the first kind of a model would be ARMA orders. So, we specify the ARMA model and then here since the model being ARMA 1, 2.

So, the AR part corresponds to only one order. So, AR equals 0.7 whereas the MA side corresponds to two orders. So, minus 0.4 and then 0.3. So, all these are fixed by the way. So, let me run the first level and then the second level is to specify the GARCH structure.

So, ARMA GARCH is nothing but we will make use of this GARCH sim and then again since the model is GARCH 1 1 we have to specify let us say omega plus alpha 1 and then beta 1 and then lastly n. Make sense so far? Okay. But then just one small part here is that since you are sort of in building the GARCH 1 1 structure on top of this ARMA 1 2. So, here you to actually specify the ARMA structure beforehand.

So, ARMA underscore order, you extract the AR part, and then MA is nothing but ARMA underscore order, and then you extract the MA part. Okay. And then you specify these two orders here inside the GARCH sim model, and then the usual GARCH parameters. Or rather, the coefficients. Make sense?

Okay. So, let me run the second-level model, which is ARMA-GARCH, and then probably we will see the plot. So, this is exactly how the plot of ARMA(1,2) plus a GARCH(1,1) model looks like. Now, again, one small disclaimer is that you might not see huge differences between, let us say, ARCH, ARCH(1), ARCH(2), GARCH(1,1), or ARMA plus GARCH. So, the whole idea is that, as you are slowly extending the models, they should be capable of capturing highly volatile processes and, on the other hand, where the variance is changing abruptly.

And much more predominantly. So, this is the whole idea of all these extensions. Alright. So, this is again just the simulated path of ARMA(1,2) plus a GARCH(1,1) model structure. Alright.

So, now the second exercise would be to analyze some real data. Alright. And then, in this spirit, the first data we will try to analyze is the New York Stock Exchange data. Okay. So, the very first thing is we have to fetch the real data, and then we have a way of fetching the real data from the R repository or rather some online repository directly in R. Now, again, all these are syntax.

So, this would—probably line 36 all the way up to line 38—be sort of important if you want to extract any stock market-related data directly into R. Okay, by using this `getSymbols` command—by the way, this command is inbuilt in the `quantmod` package—right? And then, for that matter, you require that `quantmod` package, which we installed and then called earlier. Alright, so now the first exercise is fetching the NYIC composite

index data, and then, for that matter, we will run this `getSymbols`—and by the way, this is the symbol for NYIC data—and then, if you see here, the source is Yahoo. Okay, so this data would be borrowed directly from Yahoo markets. Okay, and then here you can specify 'from,' and then you can write down some date we want.

So, I want the data from the 1st of January 2020 all the way up to the system date. So, the system date is today's date or rather the current date. Okay. So, I want the data from that time period. So, let me run that.

So, get symbols. Okay. And then here, if the line is being processed successfully, it will give you the code like this. So, NYE. So, this means that the R has sort of brought in the data from the Yahoo repository.

Okay. But again, so we may not want the entire data, right? Because the entire data might contain lots of sort of columns. So what exactly we want? We only want the closing prices, okay?

So obviously we can extract the closing prices. So, and then we'll give it a name, let's say NYSE underscore close. which would be close of operated on NYE. So, extracting the closing prices and now again we will not play around with prices itself right we want to play around with the returns ok. So, NYSE underscore returns is nothing but the first difference of log prices into 100 ok.

So, this small formula ensures that we are converting the price data to returns data. So, as mentioned here, it computes the daily returns of the data. So, let me run this NYSE_returns, and then the last thing we do. So, all this is part of the data filtering process. So, the last thing we will do is to extract, or rather remove, all the NA values because we do not want any NA values before we proceed with modeling and forecasting.

So, this is exactly how one can remove the NA values by using the `na.omit` command. So, we will try to run this again. So, NYSE_returns, and now plot the returns. So, initially, I will just show you how the plot looks. Let me zoom in.

So, this is exactly how the NYSE plot looks. And then, clearly, you can see that you have a changing variance problem here. Because if you observe here in this period, which is the COVID period, you can see a highly volatile nature in the NYSE daily returns. But as the returns progress down the line, you do not see any drastic changing variance problem. But if you analyze this overall, you do have some changing variance problem.

So, this is called as a funneling in kind of a pattern, where initially you have high variance in this time point and then later on the variance sort of reduces. Okay. So, how do you control this changing variance pattern? So, we will implement we will try to implement some of the ARCH models or GARCH models or their extensions etc. So, now the first model we try to fit on this NYSE returns data is fitting some GARCH model.

So, we will try to specify some GARCH 1 1 model and this is one function to do that. So, we will specify we will try to use a slightly different model which is UGARCH spec, UGARCH spec. We will try to run this entire chunk and then here if you see we have to specify all the parameters. So, we would tell R that which model we want. So, since we want a GARCH 1 1 model, you can specify the orders here.

So, C 1 comma 1 and then if you do not want any ARMA models to be inbuilt beforehand, you can specify 0 0. So, mean model is you do not have any mean model here. So, ARMA orders are 0 0, but include dot mean equals true. So, this would tell R that there should be some overall constant or there should be some overall mean. Make sense?

So, all these are nitty-gritties when it comes to specifying a particular GARCH 1-1 model using this UGARCH spec function. Alright, now the next thing is fitting the model. So, how do you fit the model? So, you can fit the model using the UGARCH fit command. I will show you the summary.

So, the summary does not contain a whole lot, right? But now, the plot. So, this is exactly how the plot looks. Let me zoom in. Okay.

And then, this is exactly how the conditional standard deviation behaves vis-à-vis the returns. And as you can clearly see, the blue line is the conditional SD, and the light grey line is nothing but the absolute returns. And then, can you see that the underlying changing variance component using the GARCH 1-1 model is able to capture the changing variance pattern in the dataset? So, whenever you have high peaks, the model itself jumps to that peak, and whenever you have troughs, the model itself is on the lower end, okay? So, in short, the GARCH 1-1 model is doing a very decent job, okay?

And now, the next thing—or rather, the last thing—is forecasting. So, how do you forecast this to a later date? Again, you have a built-in function called UGARCH forecast, and then you specify which model to use. So, fit underscore NYSE, and then

how much ahead in the time point. So, which is tetan. So, we will run the forecast, and then we will run the plot.

So, this is exactly how the plot of the forecast looks like, along with the confidence bands. So, the blue line is the actual data, alright? And then, this red line is nothing but the forecasted data, OK? And hence, you can extend this further. So, you can specify—let us say—other models, let us say ARMA 1 2 plus GARCH 1 1, right? Now, again, the whole thing is that you would specify.

So, again, since you are in building, you only—you also have a mean model here—you can specify the ARMA order to be those orders. So, 1 2 along with the GARCH orders, OK? So, we will try to fit this model instead now. So, ARMA 1 2 plus GARCH 1 1, and then summary, and then plot. Now, again, the plot would not look a whole lot different again, right?

So, broadly speaking, and again, if you scroll down. So, since we do not have time in this session, I will sort of stop here. But just to mention that since you have access to the code. So, probably, the next exercise is using some other real dataset, and here we have Bitcoin data. So, again, if you are more interested in modeling some ARCH-GARCH process on Bitcoin data, you can go ahead and try to complete this worksheet.

So, initially, so far, we have seen how to simulate from RSVS processes, how to model them, how to forecast them, and how to play around with some real data. In this context, we took an example of the NYSE returns data and tried to fit a GARCH(1,1) model and an ARMA(1,2) plus GARCH(1,1) model on the NYSE returns data, okay. But again, just for the sake of it, if you want. access to play around with another real dataset, which is the Bitcoin data, you can, of course, scroll down and do the exact same thing on the Bitcoin data as well, if you want. Makes sense, alright? Okay, so broadly speaking, this is the idea as to how you can simulate and then apply some GARCH processes, ARCH processes on some real data, try to forecast, eventually compare the forecasts, right? Try to specify some of the plots or how to visualize the fits, how to visualize the forecasts, right. So, we have tried to cover as much as possible, and hopefully, this entire week would have been slightly more interesting because modeling volatility is kind of predominant, or rather important.

more from a financial literature, a financial point of view, right. And then, from next week, we will try to extend this entire idea—not volatility modeling, of course, but

slightly different kinds of time series models, which are called non-linear time series, starting from next week.

Thank you.