Generalised additive models

## Transcript

Video 2: <https://youtu.be/JrgG5E2NTrA>

Hello and welcome to video number 2 of the generalised additive model online material for NCRM. In this video we’re going to be fitting a generalised additive model using the R software package, and in particular we’ll be using the MGCV package, which is designed to deal specifically with generalised additive models. So, the first thing I’m going to do is load in the required packages so that I can use them in my current R session. This includes the tidyverse packages. These are really good to use to tidy the dataset. They also contain ggplot2, which we’re going to be using to plot our data. Next we’re going to attach the MGCV package, which, as I mentioned, is the main package in R that fits GAM models. And finally we’re going to use the marginal effects package, which allows us to fit the outcome and the results of GAMs without having to simulate data or extract them ourselves.

The next thing I’m going to do is to load the data that we’ll be dealing with, which is this bike dataset here. I’ll give you a quick preview of this dataset. It contains information about the total number of bikes that were rented every day in Seoul between the end of 2017 and 2018. This dataset contains information about the total number of bikes that were rented, as well as information about the weather conditions for each day. So, what we’re going to aim to do is to investigate whether the number of bikes that were rented was in some way linked to the temperature, which is measured in degrees Celsius, within Seoul that day.

So, the first thing I might want to do is visualise the data. So, I could look at how the relationship between the number of bikes rented and the temperature. So, both of these are numeric variables. I want to see every day separately, so I want to be able to see individual observations, so I’m going to produce a scatter plot. I’m going to use the ggplot function, because I find it easier to customise and to add layers to our plots than the base R version of plot. I’m going to first of all specify the dataset and plotting, which is bike underscore daily. I’m going to specify the information from this dataset that I want to show in my plot, so my X axis is going to be temperature underscore C, and the Y axis will be determined by the total number of bikes that were rented daily. I’m then going to use a plus symbol to add an extra layer to my ggplot, and I’m going to specify the visual markings of my data using points, so creating a scatter plot. On top of this I want to make my plot look as nice as possible, I want to make sure that the text is large enough to be seen, so I’m also going to change the theme and I’m just going to use the classic theme which is prebuilt into R and I’m going to set the base size, so the smallest text size to be 12 so we can make sure it’s accessible.

So, here is the scatter plot looking at the relationship between temperature and number of bikes rented, and we can see quite clearly that there appears to be some kind of strong relationship between temperature. It seems as though very few bikes were rented when it was very cold, so it’s a steady increase up to around maybe 5 degrees, and then between 5 and around 23, 25 degrees, there’s a much steeper increase in the number of bikes rented, and that seems to tail off and reduce between 25 and 30 degrees, which makes sense because above 25 degrees, maybe it’s not particularly pleasant to ride bikes, so we can see there is a very strong relationship here between bike rental and temperature, but we can also see that appears to be non-linear. This is the perfect situation where we might want to use our generalised additive model.

So, to fit a generalised additive model in R within the MGCV package, the first thing I’m going to do is to give this model a name. So, I’m just going to call it GAM underscore bike underscore temp so I know what it’s containing, and I’m using this arrow symbol to save it as an object to call upon later on in my analysis. The function we use to fit a GAM is simply GAM. And the first thing we need to specify is the model formula, and this looks very similar to the GLM function or the LM function if you’re familiar with that, I first of all have to tell it what my outcome is from my dataset and that was the rented underscore bike underscore daily variable. And then follow this with a tilde symbol, that is dependent on or that afterwards is going to be my covariates. And then I enter the covariates in, so if I have any linear relationships I’m including in here, I just simply enter them in separated by a plus in exactly the same way we would do with a GLM or LM function. If I want to add a smooth function and non-linear relationship, however, I’m going to surround that covariate with an S, so S for smooth. And I’m going to enter in temperature as I can see that there’s this non-linear relationship here.

When I’ve specified my model formula, the next thing I’m going to tell it is the dataset I’m using, so it’s this bike underscore daily. And the final thing I want to change from the default setting is the method that the GAM function is going to use to fit this model. By default it uses an approach called GCV, which is generally thought of as less stable than the penalised approach that we saw in video 1. So, I’m going to tell it to use that approach, which in the GAM function is represented by the REML method, so this is restricted maximum likelihood.

When I’m happy with this formula, the data, the method, I’m going to run this piece of code and I will get the object, the GAM object up here in my environment, which usually, if I fit a model, I could use the summary function to extract some information about the model results. Now, in GLMs or LMs, what we get is parametric coefficients of the coefficients of our model. So, here we have the intercept, we get an estimate for that, we also get a standard error and a P value associated with it, but notice that temperature isn’t up here, and that’s because this is further down, so this is the smooth term. This is not telling us information about the coefficients, because remember we get many coefficients for a single smooth function. The number of coefficients is going to be determined by the number of knots or turning points that are as determined this smooth function required. So, I could actually extract those coefficients if I wanted to using the coef function. So, I simply put in my GAM object in there, and I can get information about those nine coefficients, which means I had nine basis functions extracted from the model fit.

Now, unlike linear coefficients, these don’t actually have a meaningful interpretation because, remember, they are essentially acting as weights for this smooth function so it’s determining the nature of this relationship. So, instead of simply looking at the P value associated with the coefficients and the coefficient estimates as many of you would do for a linear model, what we’re going to do instead is to extract this linear relationship and we’re going to plot the prediction of that on a plot over here, and then we’re going to interpret that rather than the coefficient themselves. Now, I could do this using the base R function plot, which if I enter in a GAM object, I’m going to set the argument shade to be true simply because it’s going to show me the confidence interval of this line just in a nicer way to look at, and that’s going to show me the estimated non-linear smooth function between temperature and the smooth function itself, so notice that this isn’t actually an estimate on the outcome scale, but what it’s doing instead is that when these models are fitted, by default R is going to centre them around zero in an effort to make them easier to interpret. So, zero could kind of be thought of as the average effect, so this is kind of where the average number of bikes occurred. Below that is a reduction from that average and above that is an increase. So, I can make that reference a bit easier to see by adding a line, a reference line, abline, horizontal so H is equal to zero, and I’m just going to change this to a dashed line to make it easier to see.

So, I can see, as expected, from the plot that we had earlier, between below minus 10 up to maybe about 2 degrees, we have this really steady increase, which picks up speed between about 2 or 3 up to maybe 25, and then this reduces between there. So, we can clearly see this estimated function of that relationship, we can see a confidence interval around that line, but as we’re comparing it to this kind of standardised function, we might prefer to see it on the actual outcome scale, putting it into the context of the problem that we’re dealing with. So, that’s where the marginal effects package comes in as that includes the function plot underscore predictions, which takes the model that we fit, so GAM bike temp, takes the covariate that I’ve entered as a smooth function, which was temperature underscore C. I can also show the observed points on this plot, so basically overlapping it with the original data, the original scatter plot, and I can add that using the points argument, which takes a value between zero, where the points are completely transparent, in other words we can’t see them, to one where they’re completely solid. So, if I put 0.5, that’s going to be a little bit see-through but still visible.

Another benefit of using this plot predictions is it’s actually plotted within the ggplot framework, so I could add on extra layers here and I could change the theme to make it consistent with my previous plot earlier. So, this here shows that relationship, it shows the trend that’s been estimated based on that GAM. And it’s allowing us to look at that relationship without assuming that it is constant, it’s linear.

However, there are a few points down here that this is not particularly good at capturing. There appears as though, despite the general trend increasing as temperature increases, the have kind of the optimal temperature within this model, but there’s very few bikes rented. So, we might want to extend this model to improve it and take account of other variables within our dataset, try and capture exactly what’s going on with the data. Now, if we look back at this dataset and think about which other variables may actually be having quite a strong impact on this, something like rainfall could be playing quite a significant role because we can see here that perhaps these are days where it's warm but it’s raining so it’s still not going to be very pleasant to ride your bike.

So, let’s investigate that. We’re going to plot this scatter plot we had earlier. I’m going to copy that and paste it down here. But I’m going to add on the rain variable and I’m going to change the colour of these points so that it takes the value of whether it rained or not on that day. And just to make it easier to see, I’m going to change the colour theme, the colour palette, and I’m going to manually set this colour palette. So, the values, and I want yes to be blue, so it’s blue when it’s rained, and no to be maybe brown because it’s dry. So, I’m going to have blue and then I’m going to use a nice chocolate4 colour to show the no. So, when I plot this I can see quite clearly that that does appear to be the case, where there is a lower-than-average number of bikes rented despite the temperature being optimal, these are days where it rained. So, it’s not explaining everything because there are still higher values where it did rain, but it definitely seems to explain a lot of what is going on at this bottom end.

So, what I might want to do is to add rain in here either as a covariate itself, so if I’m expecting the trend between temperature and number of bikes to be equal where it rained or not, I could include that separately as a linear predictor, or if I expect that this relationship between temperature and number of bikes rented may actually differ between days where it rained and where it didn’t, I may want to include it as an interaction term. And because I’m only looking at one continuous covariate and a categorical, I could actually try and plot this using the geom smooth layer in my ggplot because the method Gam actually exists within this geom smooth framework, so this is going to fit, because I’ve got this categorical variable set for colour, I’m going to produce two smooth estimates of the relationship between temperatures and bikes rented, one for when it did rain and one for when it didn’t, and if these look significantly different then potentially I want to include it as an interaction.

And I can see here it looks very different. The nature of this relationship changes quite a lot. So, I’m going to take my GAM from earlier, I’m going to change the name of it, I’ll call it bike rain so I don’t overwrite the original, and if I want to include an interaction term where there is a smooth function created separately for each of the temperatures, for each of the days when it rained and when it didn’t, I can add an extra term inside this smooth function, which is by equal and then the name of that factor or the interaction term, rain. And that is now when I look at the summary for this, I can see there’s actually two smooth functions fitted here, one for where it rained, one for where it didn’t, still looking at the relationship between temperature and number of bikes rented, and then if I take my plot reductions from earlier, I change the model to the updated model with rain included in there, and I put two conditions including rain as well, that’s going to fit two separate lines, but I can see something strange is going on in this model, which is suggesting to me that something about it is not quite right.

So, besides fitting our model, it’s very important to check that the model’s actually valid and it’s actually doing the job that we want it to. So, the next video is going to go through some model diagnostics to ensure that the model that we fit is valid, which here it doesn’t appears as though it is, and some kind of approaches that we can use to improve our model and to rectify any of these issues that arise to ensure that our model is valid when we’re interpreting it and that it is robust to these assumptions that we’re making about the data.

National Centre for Research Methods (NCRM)  
Social Sciences  
Murray Building (Bldg 58)  
University of Southampton  
Southampton SO17 1BJ  
United Kingdom

**Web** www.ncrm.ac.uk   
**Email** info@ncrm.ac.uk  
**Tel** +44 23 8059 4539  
**Twitter** @NCRMUK