Supplemental Instructions for the Generalized Additive Model Program

This document provides additional information for using the GAM program. I describe the fundamentals of the GAM in my book.

When you enter the equation of interest, you have two choices for a given predictor, (1) treat it as a linear predictor of Y, much like you would in traditional regression, or (2) treat it as a non-linear predictor whose relationship with Y is modeled using a smoother. A predictor is treated with a smoother if it is enclosed in s() when you specify the equation, such as

Y ~ s(X)

You can expand the notation within s() for a given predictor to take advantage of options provided by the GAM function from the R package mgcv that I make use of. The program offers a variety of regression based spline smoothers for modeling the relationship between X and Y. Examples include P-splines, B-splines, thin plate splines, and tensors. The default smoother is a penalized thin plate regression spline but you can force the program to use a different spline type for any given predictor. For example, to use a P spline for predictor X, I would specify

Y ~ s(X,bs='ps')

where bs is the keyword to signify the type of spline and 'ps' stands for a P spline. Options to specify after bs include 'tp' for thin plate regression splines, 'ds' for Duchon splines, 'cr' for cubic regression splines ('cs' specifies a shrinkage version of 'cr' and 'cc' specifies a cyclic cubic regression spline), 'sos' for splines on the sphere, 're' for random effects, 'mrf' for Markov random effects 'gp' for Gaussian process smooths, 'so' for soap film smooths, 'ad' for adaptive smoothers, and 'fs' for factor smooth interactions. The strengths and weaknesses of each are discussed in Wood (2017).

As discussed in my book, the overall smoother for X and Y is based on fitting a smaller number of functions to the data, called basis functions, and then adding the results of those functions up to yield the overall smoother. The more basis functions you use, the better you can account for the data but there always lurks the risk of overfitting. GAMs include a penalty function for too much wiggliness in the overall smooth as it searches for a balance between accounting for the variability in the data versus overfitting by treating as meaningful what is essentially random noise in the data. The GAM program uses default algorithms to help find the right balance, but sometimes we need to override the defaults. These overrides occur in the context of the equation you tell the program to use. See the mgcv program manual and my book for details.

GAMs also can explore non-linear interactions between continuous predictors. The most common approach is to use either a full tensor product smooth (called 'te' in the mgcv package) or a tensor product interaction (called 'ti'), with the latter used in the spirit of the classic philosophy of adjusting for or removing main effect influences from the product term. For predictors X and Z, the equation would be specified as

Y ~ ti(X) + ti(Z) + ti(X,Z)

For details, see Wood (2017) and my book. If Z is a nominal moderator, then the interaction often is modeled using the default penalized thin plate regression spline smoother in conjunction with the 'by' command, like this:

Y ~ Z + s(X) + s(X, by = Z)

See my book for details.

Sometimes we want to test if a continuous predictor X can be treated as a linear predictor or if a smoother for it is needed to accommodate non-linearity. This is often tested using the following model (see Wood, 2017, for details):

```
Y ~ X + s(X, m=c(2,0))
```

In this model, I include the linear term X and a smooth of X. The argument m tells the program that the usual penalty for wiggliness on the second derivative of the smooth should be used but with no such penalty on the constant function nor the linear function. By doing so, the s(X) term will now represent only the non-linear aspects of X so its test of significance is a test of the need to incorporate non-linearity. If it is statistically non-significant, one would just model X as linear, assuming the test is adequately powered.

GAMS can be difficult to interpret, so I like to pursue profile analyses within them. See my book and the program video for details. The general point here is that you will need to learn the ins and outs of overriding defaults in the mgcv package to achieve some of your modeling goals.