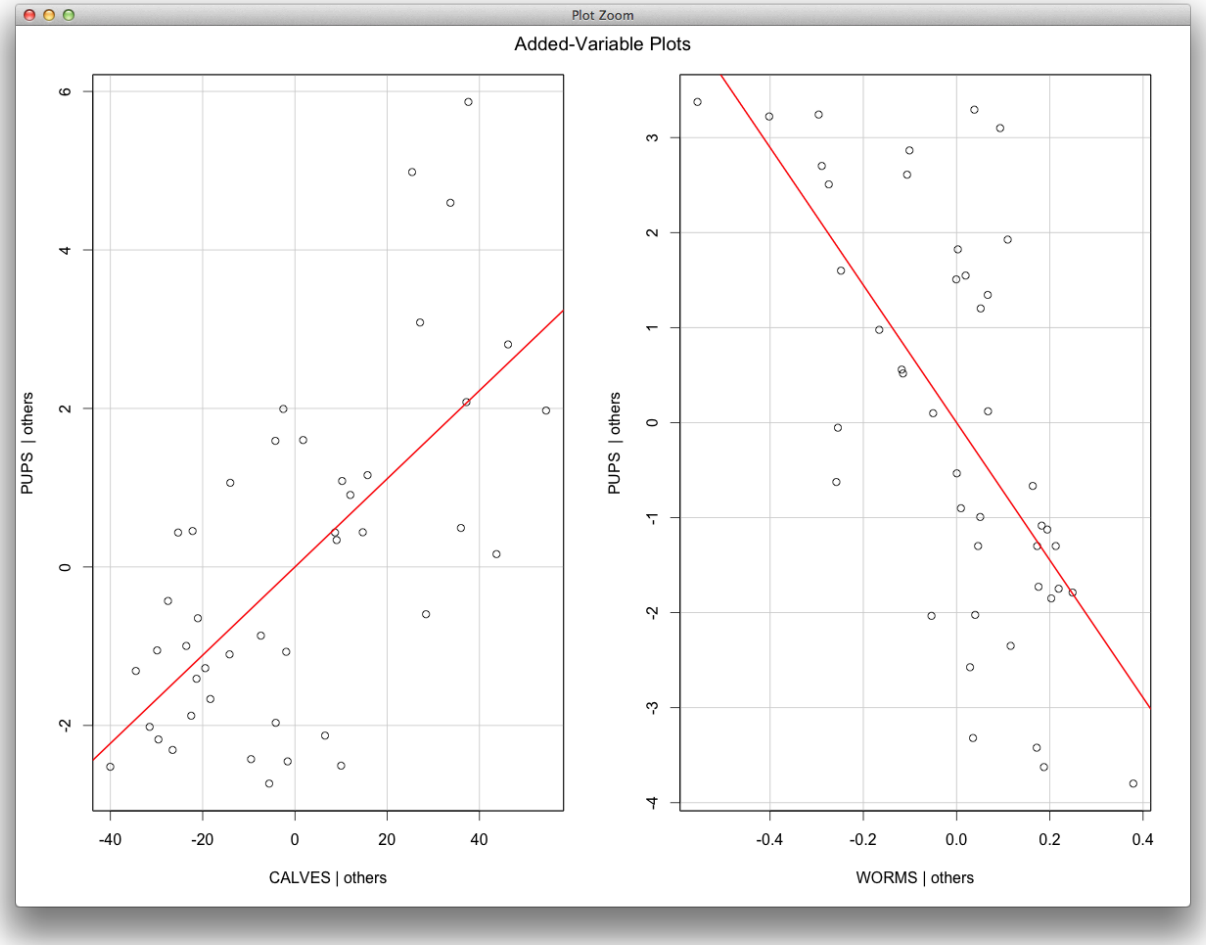
Advancing in R



Module 4: Multiple regression

Introduction

In this exercise, we will perform multiple regression analysis.

We will assume that you are familiar with the general syntax of coding in R, with making basic plots, with the general theory of linear regression, and that you have reviewed material covered in the practical exercises preceding this one. If you need to, review this material before attending the practical session.

Learning outcomes

Know how to produce panels of scatterplots to check for multicollinearity

Know how to build multiple regression models

Know how to check for variance inflation in a model

Know how to adjust models affected by multicollinearity

Know how to simplify models using F-tests or likelihood ratio tests

Know how to scale predictor variables

Know how to interpret standardized regression coefficients

Know how to produce added variable plots

Libraries

In addition to the basic libraries automatically loaded with R, you may wish to use the following additional packages. If necessary, make sure you either have internet access on your machine or that you install these packages prior to the practical.

{car}

{ggplot2}

Some useful commands

Below is a list of some (but not all) of the useful commands you may need to run at some stage of today's practical.

anova()
avPlots()
library()
lm()
pairs()
rep()
seq()
update()
vif()
summary()

Data quality control

Begin by loading the RStudio package. Open a new project and a new Rscript, and save each of these using meaningful and descriptive filenames. Include the usual annotation at the start of your script that identifies the purpose of the script and its author, as well as commands that clear the workspace. Find the data file called "WOLFPUPS.csv" in the folder for this practical, and read it into a well-named data frame. Verify that it has loaded properly, and examine the structure of the object.

This dataset is from a study of variables affecting the number of wolf pups born in a wildlife park across a number of years. You can assume for the sake of this exercise that wolf pup numbers are measured with a high degree of reliability. The predictor variables included in the dataset are the number of adult moose in the park (ADULTS), the number of moose calves born (CALVES), the average blood helminth load (WORMS) of trapped wolves, determined using blood samples from randomly selected wolves every year, and the number of moose hunting permits (PERMITS) issued by the park to human hunters.

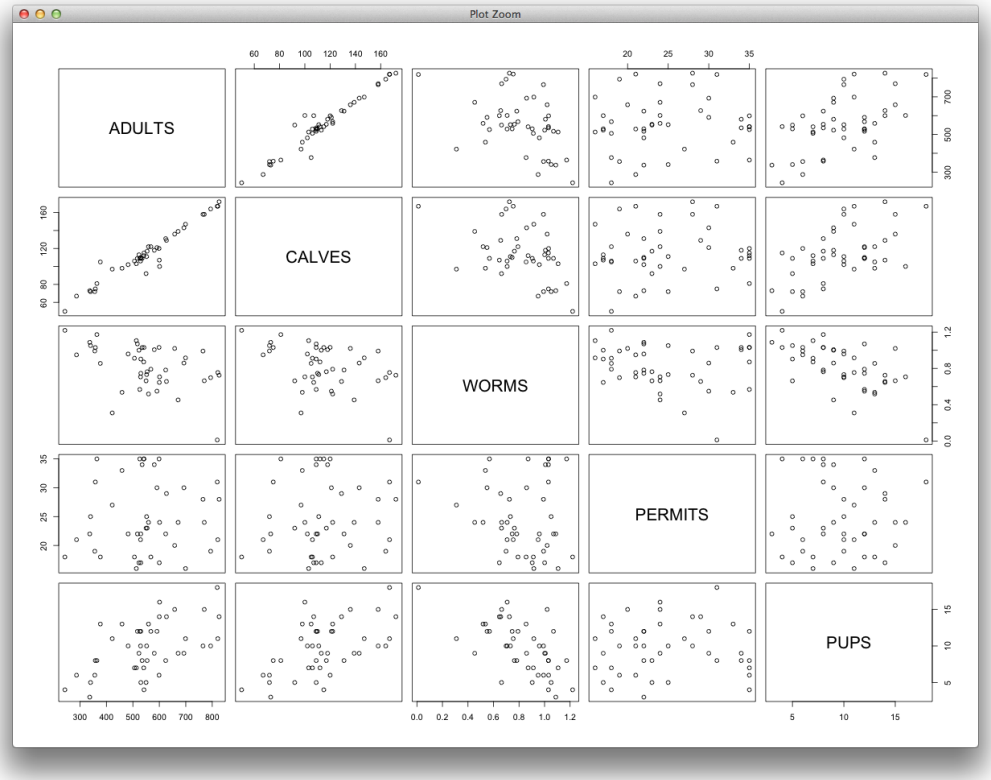
Use plotting commands to exercise data quality control. Check for outliers. If necessary, clean the data and recode any variables that need to be read differently than they currently are. Make sure you check distributions again after cleaning the data to ensure that you have fixed any problems.

Examine the distributions of all predictors and the response variable. Have you got any concerns about the data? Make a note of these using annotations.

Checking for multicollinearity and variance inflation

One of the special potential problems in multiple regression is the problem of multicollinearity (also known as collinearity), which we covered during this module's lecture. Very strong correlations can cause problems for parameter estimation in a couple of ways: they can make the analytic exercise of finding coefficients difficult, but in a practical sense they also produce incorrect coefficients even when the equations for estimating parameters work.

There are two complementary approaches to exploring whether multicollinearity is a problem. First, we can produce a series of pairwise plots that illustrate bivariate relationships between all of the variables in your dataset, and search visually for strongly correlated predictors. You could do this using the plot() command, but the pairs() command is much quicker. Make sure that the object to which you apply pairs() has only numeric variables, as pairs() usually won't work on any dataframe containing characters. If you're not sure how to use pairs(), try the help file (?pairs, passed to the console). If you can't figure this out after a few tries, see below the figure on the next page for a hint.



(HINT: The pairs() command requires only a single object: a dataframe containing only numeric variables. You can either enter your entire dataframe, or clip the variables of interest by selecting the relevant columns with square brackets.)

The figure above shows a number of quite strong relationships, but most of these are not so strong to raise a concern. However the relationship between ADULTS and CALVES is so strong that we can predict the models will have trouble distinguishing them. Before we start the modelling, let's look closely at the relationships between PUPS and the other variables – these are the ones that are of interest for our focal scientific question. You may wish to create new versions of these scatterplots using the plot() command so that you can see the axis units more clearly. Which of the predictors appear to be related to PUPS? What is the sign of each relationship? Which is the strongest predictor, and which the weakest? Using your plots, write down educated guesses for the value of the slope coefficient for each predictor.

Now that you're armed with predictions, let's build a model and see if there is variance inflation. Use the lm() command as we did last in previous exercises, but add all the terms that could conceivably be important to the predictor side of the equation, separated by plus symbols (+), as follows:

```
> MOD.1<-lm(PUPS~ADULTS+CALVES+WORMS+PERMITS,data=WP)
```

What's the first thing you should do after building your model? Make a guess before turning the page to see if you're right.

Examine the diagnostic plots for your model! Always do this before looking at the model summary – if you're model is terrible, there's no point getting excited about the parameter estimates, as they will probably be wrong anyway.

How are the diagnostics for your current model? If you have serious concerns, then try using some of the transformations you tried earlier to see if they help. If there are data points with a high influence, just make a note of which ones they are for now – we'll come back to this after we've settled on a reasonable model. As you'll see, this initial model is a long way from being ideal, so there's no point in getting too obsessive about the diagnostics just yet, even if it is still necessary to look at them and see how your model is performing.

Now we should check for variance inflation. To do this, we need to load an R package called {car}, which stands for "companion to applied regression". This is a very useful package, and we'll use two of its functions today. Install and load {car} now if you haven't done so already.

Once {car} is loaded, run the vif() function (for variance inflation factors) on your model:

```
> vif(MOD.1)
```

You should see an output of one variance inflation factor for each of the four predictors. Variance inflation factors measure the following quantity:

$$\frac{1}{1-R_j^2}$$

where R^2_j is the multiple R^2 of a regression of the focal predictor on all other predictors. What does the multiple R^2 value measure? You can see that as the R^2 value gets very high (because other predictors covary strongly with the focal one), the vif also gets very high, indicating multicollinearity. A handy rule of thumb is that vifs over 10 require model adjustments to deal with variance inflation, and vifs over 4 or 5 should at least be investigated by building a model without the problematic term and asking whether the coefficients for other terms change substantially. Are any of your vifs over these thresholds?

If yes, you will want to build an alternative model excluding one of the offending high vif predictors. The easiest way to do this is using the update() command. The notation is as follows:

```
> MOD.2<-update(MOD.1,~. - ADULTS)
```

The syntax here tells update() which model to work with, then the tilde and fullstop (~.) tell R to use MOD.1 "as-is", except for what follows, in this case a minus sign and the term ADULTS. Full translation: update MOD.1 with the same structure as before, but excluding ADULTS.

Examine the vifs for the new model. Has the problem been solved? Why? Just for fun, examine the parameter estimates for the new model, and compare them to the original one that suffered from variance inflation. Are any of the parameters strikingly different across models? Are they the same terms that suffered variance inflation? If you like, check whether removing CALVES instead of ADULTS affects the outcome. Although the two models will give almost identical results in

principle, there is an objective way to decide which of the models is better. Can you think of what that objective basis for model evaluation is?

Model simplification

The principal of parsimony can be paraphrased as "the simplest adequate explanation is the best". In selecting among various different models, we'll want the one with the fewest terms and least complex formulation, subject to the constraint that it adequately explains most of the variation in the data. Crawley's treatment of model simplification is very helpful in clarifying the principals behind model simplification and the issues that one should keep in mind while conducting it, so I recommend that you consult one of his books (ideally *The R Book*, Ch. 9, which is conveniently provided as an e-book by the library). We'll study another method for model selection in the module on ANCOVA.

The current model includes three of the four original predictors (one having been removed to prevent multicollinearity). Is there a simpler model that explains the data just (or very nearly) as well? Examine the table of coefficients for your model. Which of the estimates contributes the least to explaining variation in PUPS? Use the update command as above to build a newly named model without the term in question.

Before examining the new model, let's ask R whether it is significantly worse than the previous, more complex one. This is very simply done: use the anova() command with two arguments, which are the names of the two models you want to compare.

```
> anova(MOD.2,MOD.3)
Analysis of Variance Table

Model 1: PUPS ~ CALVES + WORMS + PERMITS
Model 2: PUPS ~ CALVES + WORMS
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1     40 279.05
2     41 280.85 -1   -1.8028 0.2584  0.614
```

The output is an ANOVA table (similar to the kind you might generate for a simple linear regression), but here the F-ratio is derived from examining the increase in residual SS that inevitably arises when a term is excluded from a model. If the increase in residual SS is sufficiently high (adjudicated by observing the theoretical F-distribution, and observable as a p-value that is less than 0.05), this means that removing the term in question causes an unacceptably high increase in model deviance. In this case, there is only a small increase in SS when one removes PERMITS from the model. This is a non-significant increase in SS, and consequently we should prefer the simpler model. You will get more practice doing model simplification in the rest of the course, and while the details of the test may change slightly, the principle is always the same: if the change in model deviance is not large (i.e., non-significant), then you should prefer the simpler model. If however excluding a term causes a large change in unexplained variation, you should retain the more complex model.

Do you think you can remove any of the terms in the simpler model you have just created? Try it just to see what happens. The model that you end up with after rejecting all terms that do not substantially decrease deviance is your minimal adequate model. It is almost always the model for which you should report parameters in your paper. In some cases, you may also want to cite statistics to defend excluding a predictor from the minimal adequate model. If you want to do this, you can cite the test-statistic (usually either an F-value or a χ^2 value) and p-value associated with the model simplification step at which the term in question was excluded.

IMPORTANT: Note that we did not use the anova() command to compare MOD.1 and MOD.2. Why not? The order of operations here is extremely important. *First*, we found a model that was not troubled by collinearity, *then* we used likelihood ratio tests to find the minimal adequate model. Never use anova to decide whether to retain highly collinear variables, as their collinearity is more than enough reason to get rid of them, and is expected to artificially and inappropriately decrease model deviance.

Centering and scaling

Have a look at the table of coefficients again. Can you write the equation for the line of best fit? Which of the two remaining predictors is more important in explaining variation in PUPS?

This is a difficult question to answer now because the units for the coefficients are in totally different scales. To derive *standardized regression coefficients*, we will centre each predictor (subtract all observations from the mean predictor value) and then scale them by the variance (divide each observation by the standard deviation). You could work out how to do this easily enough, but R already has a built in function, scale() that will do it for you. You can easily nest scale() into the call for a linear model as follows:

```
> MOD.3s<-lm(PUPS~scale(CALVES)+scale(WORMS),data=WP)
```

Examine the table of coefficients for the new centred and scaled model. Note that while the model R-squared value is identical, the coefficients now are expressed in terms of the variation in your dataset. Consequently, your coefficients now represent the expected change in the y-variable for every standard deviation change in each predictor. They are now fully comparable! Which of the two coefficients has a stronger effect on PUPS, based on the existing variation in predictors that is present in your dataset? How confident are you that the difference in standardized coefficients represents a real difference as opposed to an accident of sampling?

Visualizing partial effects

As usual, there are several ways to visually represent the results of a multiple regression analysis. I suggest two productive options:

1) Use the predict command and low level {graphics} package to add the lines and 95% confidence intervals to plots of the data. The advantage of this approach is that it is already familiar to you, and will likely also be so to readers. Unfortunately, you will need to make arbitrary decisions about the value of the "other predictors" that are not being illustrated. This is often done by simply setting the other predictors to the mean value, as below. Note you'll need to create two new x variables, one for WORMS, and one for CALVES or ADULTS (depending on which of these you retained in

your minimal adequate model; predict needs new values for every single predictor in the model it is using to generate new fits). For each plot, you'll want the focal variable to span the full range of the variation in the data, while you'll want the other variable to be fixed at its mean value. Use the seq() and rep() commands to make the non-focal vector the same length as your other variable, as follows:

```
> # create new predictor values in in NEWWORMS
> NEWWORMS<-seq(0.01,1.22,length=100)
> # the next line will create a vector with the same number of rows, but
all set to the mean value for CALVES
> MEANCALVES<-rep(mean(WP$CALVES),100)
```

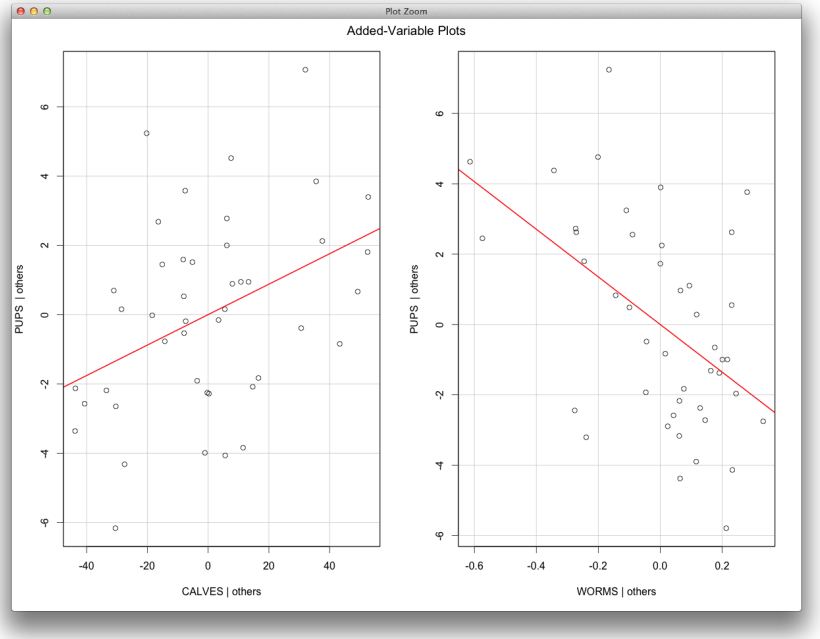
Now use the predict command to generate the predicted number of PUPS for this particular combination (in doing so, we're illustrating the partial effect of WORMS after controlling for CALVES, since all CALVES values are set to the mean):

```
> NEWYUSINGWORMS<-
predict(MOD.3,newdata=list(WORMS=NEWWORMS,CALVES=MEANCALVES),int="c")
```

The code for plotting NEWWORMS, its predicted response, as well as for plotting confidence intervals, is the same as in the previous practical exercise.

2) Alternatively, you can use the avPlots() command from the {car} package to automatically generate "added variable plots". These are simpler to generate and arguably preferable to the approach above because they illustrate the real partial effects, even if they are sometimes more difficult to interpret in terms of the original scales of the data.

Try calling avPlots() with the only argument being the name of your minimal adequate model. You should get something that looks like this:



Added variable plots illustrate residuals from a model that includes all terms except the predictor of interest (on the y-axis) regressed on residuals of a model of the focal predictor on all other predictors. Conceptually, then, this is illustrating "the remaining unexplained variation in y, having controlled for other factors" against "the fraction of the predictor variable that is also not explained by other factors". If this sounds a bit contorted, it is: welcome to the logic of multiple regression. If you wanted, you could check that you understand what the avPlots is doing by making two new models, and plotting the residuals against one another, as follows:

```
> RESWORMSONOTHERPREDICTORS<-lm(WORMS~CALVES,data=WP)
> RESPUPSNOWORMS<-lm(PUPS~CALVES,data=WP)
> plot(RESWORMSONOTHERPREDICTORS$residuals,RESPUPSNOWORMS$residuals)
```

This is no different from what avPlots() is doing, but avPlots() has some nice optional arguments that make it even more valuable than its economy of code. See if you can work out how to add more meaningful axis labels to the plots, to make them more suitable for publication. To test your knowledge, you might also try writing a figure legend explaining to your reader what the figure is illustrating.

Interpreting results

Examine the parameters for the minimal adequate model (and the added variable plots if they assist your interpretation). How do the results compare to your predictions for the coefficients based on univariate exploratory plots? Can you explain any discrepancies? What, if anything, can you say about the *relative* contributions of CALVES and ADULTS to variation in wolf PUPS? Compose a few sentences of Results text in which you explain the significant predictors of PUPS, citing statistics, tables of coefficients, or figures as you see fit.

Remember to save your project and R script, and if you want, your figure as well.

~ End of Practical ~