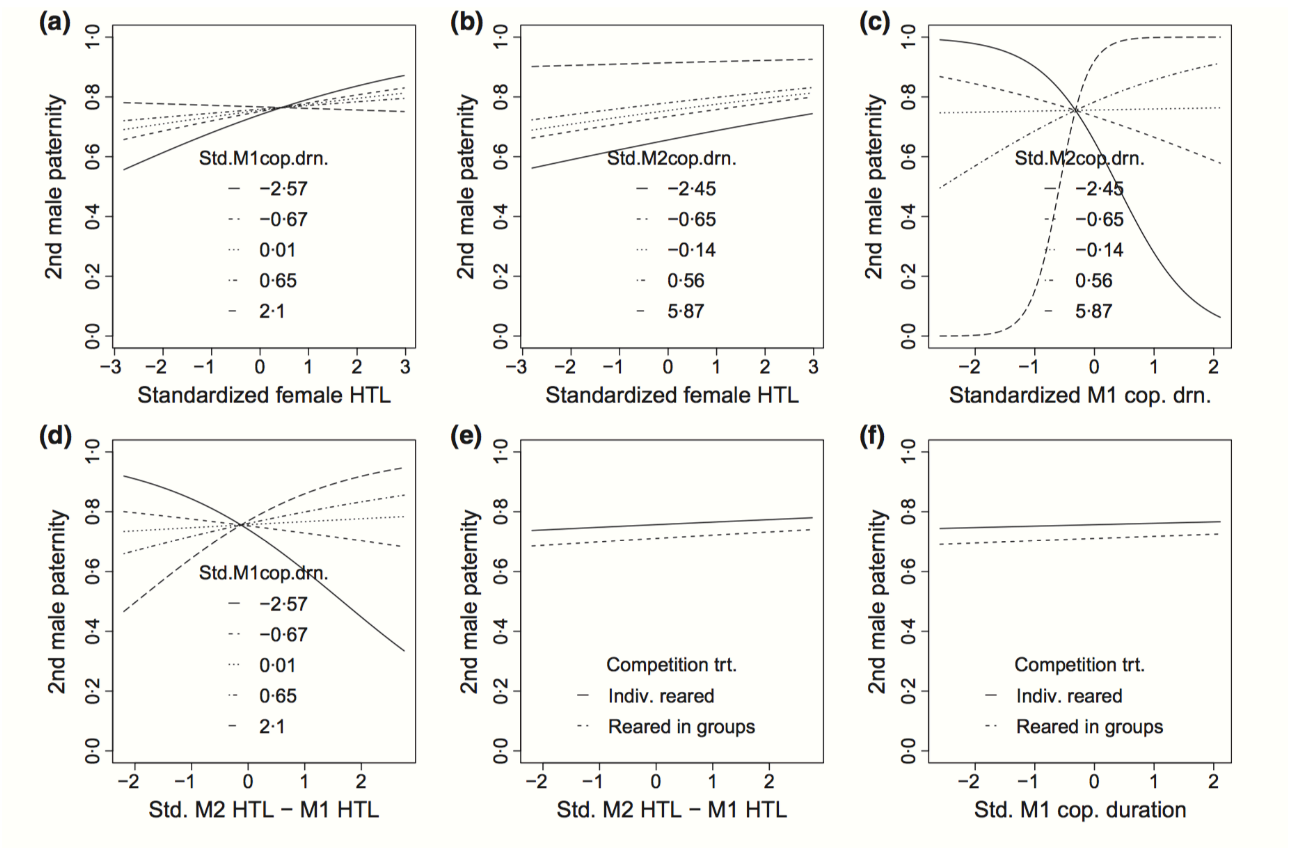
Advancing in R



Module 7: Predicting fits from model coefficients

Introduction

In this practical, we will start with some practice using model coefficients to predict fitted lines and confidence regions, an exercise that will be useful the next time you encounter a table of coefficients in the literature, or in the work of colleagues or students. Then we will use some familiar examples from previous topics as well as some new datasets to practice using several approaches to fitting lines to linear models. The practical assumes that you are familiar with the general syntax of coding in R, with making basic plots, with the theory and practice of linear and multiple regression, with the general and generalized versions of the linear model, and that you have reviewed material covered in the preceding practical exercises. If you need to, review this material before attending the practical session.

Learning outcomes

Know how to interpret tables of model coefficients

Know how to call a predict function for a linear model in R

Know how to use the packages {broom} and {visreg} to assist model interpretation and visualization

Know how to export broom data frames to .csv files

Know how to illustrate interactions between continuous variables

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.

{broom}
{dplyr}
{ggplot2}
{visreg}

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.

augment()
ggplot()
glance()
tidy()
visreg()

Interpreting a table of coefficients

Get into groups of two or three for the following exercise. Imagine that you both have measured the body mass (g) and the length (mm) of asp viper (*Vipera aspis*) in three populations in France – in the Pyrenees (N=10), the Massif Central (N=10), and the Jura mountains (N=10). Assume that you have conducted all of the required initial stages of data analysis in R, including quality control and assessing variable distributions, and that the key variables of the data set conform to expectations (i.e., mass and length are both nicely Gaussian). Population (called "pop" below), is a categorical variable with three levels. The summary statistics of the total mass and length data are as follows:

```
> summary(mydat$mass)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  276.2   385.8   959.0   996.6  1536.0  1988.0 
> summary(mydat$length)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  45.20   50.19   53.29   54.83   59.99   69.69 
```

Pretend you are interested in whether length is a good predictor of body mass, and that you particularly want to know how any such relationship differs across populations. You have fitted a linear model of body mass as a function of length and population, including an interaction between length and population. When you call of a summary of the minimal adequate model, you get the following output:

```
Call:
lm(formula = mass ~ pop * length, data = mydat)

Residuals:
     Min       1Q   Median       3Q      Max 
-20.0662  -2.5770   0.8962   5.0093  15.6037 

Coefficients:
                         Estimate Std. Error t value Pr(>|t|)    
(Intercept)               -0.2737    23.8061  -0.011    0.991    
popMassif Central         17.5304    31.8089   0.551    0.587    
popJura                    6.7285    38.6962   0.174    0.863    
length                     6.1459     0.4220  14.564 2.06e-13 ***
popMassif Central:length  11.7764     0.5746  20.495  < 2e-16 ***
popJura:length            24.1139     0.6954  34.679  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9.359 on 24 degrees of freedom
```

```
Multiple R-squared:  0.9998,      Adjusted R-squared:  0.9997
F-statistic: 2.113e+04 on 5 and 24 DF,  p-value: < 2.2e-16
```

Challenge: using only the table of coefficients above, draw a graph by hand that illustrates the predicted relationships between body mass and length. In addition, calculate the predicted body mass of individuals of average length in each of the three populations.

Hint: you should get three predicted lines, one for each population. What do the coefficients suggest are the differences between them? Can you derive the best-fit equations for each population? You may need to carefully consider the ranges of the observed data.

When you think you have a good answer to the question, compare it to the candidate answer on the last page of this practical handout (where you will also see a rough plot illustrating the relationships across the three populations). Try not to consult this answer before making a guess yourself! Discuss with your classmate(s) any discrepancies between your guesses and the suggested answer.

Using {broom} to manipulate model summaries and plot fits
Start up the RStudio software. 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.

Now retrieve the dataset called “tannin.csv” from the practical exercise on univariate regression. You may also wish to copy over the lines of code from that exercise that you used to build a regression, examine it, and plot the line of best fit.

In the original practical, I asked you to use the predict() command to plot the line of best fit. predict() is both versatile and instructively useful because it forces you to consider what is needed to generate a prediction: an equation for a line (as provided by a model object), as well as a series of predictor variables. In the original practical, and the several exercises that followed it, I asked you to generate a new vector for each predictor every time you produced a line of best fit. Your “newdata” vectors either spanned the full range of the predictor in your original dataset (when the focal variable was the one of interest), or they consisted instead of repeated values held constant (to allow you to illustrate the partial effects of another variable). Remind yourself of how the predict command works by working through your old code now.

As usual, there are R packages available that provide helpful shortcuts in working with model outputs, and allow you to circumvent some of the tedious coding involved in visualizing effects using predict(). These shortcuts don’t always work, and some of them allow less customization than predict(), so you may find that the “old-fashioned” approach to producing vectors of fitted values and confidence intervals is sometimes still quite useful.

We can use the package {broom} to convert model object contents to “tidy” data frames. Load the broom package now, as well as {dplyr} and {ggplot2}, which we will both use soon enough. The {broom} function tidy() converts the table of coefficients for any model to a data frame so that you can more easily access and manipulate the key components of the model summary. Try it now on the model that regressed caterpillar growth on dietary tanning levels. (If you prefer to practice rather than copy old code, you can build a new linear model of the form GROWTH ~ TANNIN.) Another {broom} function, glance(), similarly extracts information from the model summary and converts it to a single row data frame. Try it now on your model, and see what it does.

The functions tidy() and glance() are useful for organizing coefficients and summary statistics, but augment is more versatile still. Use it on your model object, and try to discern what it does, using the help file if necessary. Can you see why such a function might be useful for plotting model fits?

Let’s try to exploit this information now, using {dplyr} to chain together commands, and {ggplot2} to visualize the output. If you are feeling confident, try to do this yourself before consulting the code below.

```
tanfit %>%
  augment() %>%
  ggplot(data = .,
         aes(x = TANNIN,
             y = GROWTH)) +
  geom_point() +                  # points from original data frame
  geom_line(aes(y = .fitted)) +   # line from model fit
  geom_ribbon(aes(ymin = .fitted - (2 * .se.fit),
                  ymax = .fitted + (2 * .se.fit)),   # CIs from model fit
              alpha = 0.3) +
  theme_bw()
```

Can you work out the function for each line of code? Refer to the output for the augment() command on its own if the syntax for the lines in the geom_ribbon() command are confusing to you.

It is even possible to chain the model building itself into the code above, although I normally recommend that you spend some time inspecting model diagnostics and output before launching into visualizing fits.

An alternative approach to plotting fits involves the {visreg} package, which contains many useful functions for plotting lines of best fit. Load {visreg} now, and pass the function visreg() to the console with the model name as the only argument in the function call, as follows:

```
visreg(tanfit)
```

Can you figure out how to adjust graphical parameters for this plot? Consult the help file for advice. See if you can adjust the x- and y-axis labels as well as the point size and colour. Hint: the argument points.par allows you to adjust parameters such as character size (cex) and colour for the points alone. Feel free to experiment with adjusting the characteristics of this plot until you are satisfied with it before moving on.

Visualizing models with more than one predictor

Most of your models will be more complex than univariate regression. As we explained in the previous exercises, visualizing the outputs from these models requires a bit more thought. Most of the time, there will be several lines instead of just one, and you will need to use your judgment to find the graphical representation that most effectively communicates the model's message. Nevertheless we would ideally like to ensure that we are always illustrating the actual fitted values of our models rather than simply plotting univariate best-line fits on one predictor at a time. Consequently, we will need to consider carefully how to ensure that the partial effects of other predictors are controlled in our plots.

In the ANCOVA practical exercise, we considered data on the yield (FRUIT) of a crop subjected to two different grazing regimes (GRAZING=Grazed, and GRAZING=Ungrazed), for which we had access to a continuous covariate that measured plant size (ROOT) prior to grazing. Import the data from the .csv file "compensation.csv" once again, and save it as a memorably-named object. You may wish to copy the code specifying the minimal adequate model from the ANCOVA practical to save time (the minimal model had both predictors but no interaction term). Examine the model summary to reacquaint yourself with the patterns revealed by these data.

Practice using {broom} to obtain tidy data frames containing model coefficients, summary statistics, and augmented data frames. Can you adopt the code from the linear regression above to plot the fitted effects for both grazing treatments? Hint: you will want to specify additional aesthetics for line colour (colour) and confidence interval fill colour (fill) in the main ggplot() command.

Now try using {visreg} to illustrate the fitted effects for your model of how grazing treatment and root diameter affect fruit yield. In the simplest call (in which the brackets following the function visreg contain only the model name), you will need to interact with the console to see all the plots. Consult with the help file, which has many useful examples. Use these examples to experiment with the arguments for specifying the x-variable of interest (xvar), adjusting the space between treatments (whitespace), specifying the value of other variables besides the x-variable (cond) and allowing the presentation of alternate values of other variables across panels (by). You may also wish to experiment with overlaying multiple lines on a single panel (overlay).

Import the data that you used to explore the effect of dietary grain and supplement on cattle growth in the supplementary exercise for the ANCOVA practical (growth.csv). Once again, you may wish to import code that specifies one or more of the models in order to practice illustrating the model effects. To help illustrate the power of visreg, it may be useful to try plotting the maximal model in this case (including an interaction and all four supplement levels), even though we know that the best model is somewhat simpler than that. Experiment with the arguments as above, and see which of the presentations (e.g., across panels vs. overlaid) you prefer. Contrast the output of visreg with simple boxplots, or with the barchart you may have built to illustrate the model in the ANCOVA practical. Which best illustrates your model coefficients?

Using visreg for generalized models and models featuring interactions

Import the data on the incidence of a bird species on islands as a function of isolation and island area, which we used in the generalized linear model practical (isolation.csv). Once again, reacquaint yourself with the main patterns in the data set by reviewing the model coefficients. You may be annoyed to learn that visreg can very efficiently visualize the effects we painstakingly coded during the practical. We hope that the exercise of backtransforming was instructive in illustrating where the fits and confidence regions come from! But in the future you may be able to save yourself a lot of the coding tedium by leaning on visreg.

First, plot the effects in the link scale by calling visreg as usual, perhaps by specifying one x-variable at a time as below:

```
visreg(MOD.2, xvar = "ISOLATION")

visreg(MOD.2, xvar = "AREA")
```

Remember that the y-axis scale in this example is the logit scale. You could relabel the plot if you wanted to present these figures, but generally they are most useful for inspection, for example to ensure that your coefficients match the plots.

You can also plot the fitted effects on the original scale by specifying scale=”response” in the visreg call. Note that by default these plots contain a “rug”, which illustrates the location of sampled observations using small ticks along the x-axis (note how the argument rug=2 helps indicate whether the focal observation was 1 or 0). You can remove these using the argument rug=FALSE, but then your reader will have no sense of which parts of the plot are well supported by your data.

Finally, let’s try a new challenge: how might you illustrate interactions between continuous variables using 2-dimensional plots? To explore this question, import the data from the file “FLYSEX.csv”, located within the predict folder of the course materials. This file contains counts of male approaches (ATTR, short for attractiveness) for artificial female fly silhouettes that were manipulated to exhibit all possible combinations of LEG and ABDOMEN ornaments representing the species mean as well as 3 standard deviations on either side of the mean. The biological question concerns the relative importance of the two kinds of ornamental trait, as well as the potential for the two ornaments to be more or less than additive when combined (as tested by the interaction term).

Inspect the data structure and the distribution of all variables. What form does the attractiveness variable take, and what modelling approach do you therefore prescribe? Make preliminary plots that illustrate the effects of each predictor on the response.

What kind of plot might illustrate the combined effects of LEG and ABDOMEN on attractiveness? There are doubtless a few alternatives, but the following ggplot code is one approach, using a colour aesthetic to illustrate the z-dimension and jittering the points to avoid the otherwise substantial overlap:

```
ggplot(FLYSEX, aes(x = LEG, y = ABDOMEN, colour = ATTR)) +
  geom_jitter(alpha = 0.8, size = 3)
```

Can you predict the sign and relative magnitudes of the coefficients based on this plot? Is there any evidence for an interaction?

Use your knowledge of generalized linear modelling to find the minimal adequate model that best predicts fly attractiveness. Make sure you check the assumptions of the generalized linear model as you go, and that you deal with any problems according to your previous instructions.

When you are satisfied with your model, it will be time to illustrate the effects. Your best model should include an interaction between two continuous variables. What does this term imply about the relationships between the predictors and the response variable? How can you illustrate it to readers?

Use {visreg} to plot the effects of your model on both the link and the response scales. You will want to use the "by" argument to plot the effects of each predictor at several levels of the other predictor, which may help to convey the interaction effect. Experiment with overlaid plots to see if this presentation facilitates comprehension or not.

If you are feeling a bit masochistic, feel free to try coding this visualization using base graphics and predict as well. Compare the outputs and judge which is the most satisfying and clear presentation for yourself. Remember to save your script and any plots you want to keep as well.

Suggested answer to challenge at the start of the practical:

There is an interaction between length and population, indicating that the relationship between mass and length is different in the three populations. In the Pyrenees, mass increases by 6.1g for every mm increase in length (main effect of length). This effect is significantly stronger in individuals from Massif Central, so that the increase in mass is 6.1+11.8g for every increase in mm, and stronger again for individuals from Jura (increase in mass of 6.1+24.1g for every increase in mm. Thus, a graph would have three lines showing increasing mass with increasing body length, with the slope smallest for Pyrenees, steepest for Jura, and intermediate for Massif Central.

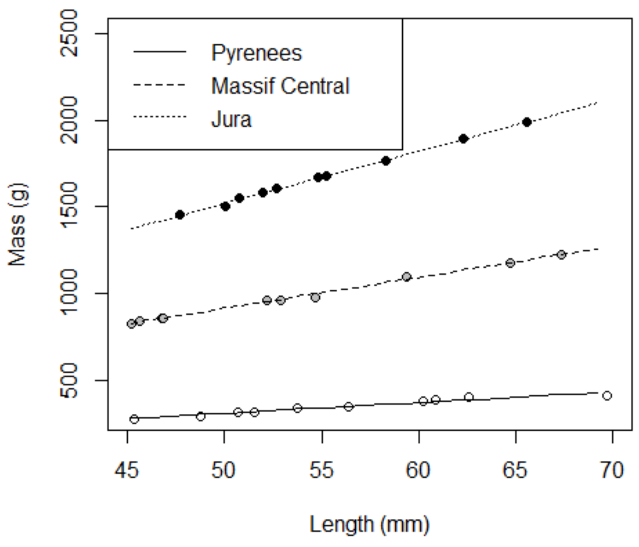
The interpretation of the main effects of population require careful thought because of the presence of the interaction. At the intercept (length = 0mm), individuals from Pyrenees are lightest, followed by individuals from Jura, and individuals from Massif Central are heaviest. But it's not terribly meaningful to compare snakes at length = 0mm. When we make predictions for a meaningful body length, e.g. at the mean body length measured (55mm), the predicted mass in the three populations is as follows:

*Pyrenees: -0.27+6.15*55mm = 338g.*

*Massif Central: -0.27+17.53+6.15*55mm+11.78*55mm = 1003g.*

*Jura: -0.27+6.73+6.15*55mm+24.11*55mm = 1671g.*

So, the order of predicted weights at the average measured length is Jura>Massif Central> Pyrenees. If you included the x(Length)=0mm in your scale, you will have the lines of best fit for Jura and the Massif Central crossing before they reach the y-axis.



~ End of Practical ~