Exercise

This practical has two exercises, the second one is optional extra.

Exercise 1: Linear model with additive continuous and categorical predictors (= explanatory variables)

This exercise builds on the linear model with one continuous predictor, and the linear model with one categorical predictor, by adding these two sources of variation in the same model.

1. As in previous exercises, either create a new R script (perhaps call it linear_model_3) or continue with your previous R script in your RStudio Project. Again, make sure you include any metadata you feel is appropriate (title, description of task, date of creation etc) and don't forget to comment out your metadata with a `#` at the beginning of the line.

2. Import the data file 'loyn.txt' into R and take a look at the structure of this dataframe using the `str()` function. We know that the abundance of birds `ABUND` increases quickly with the area of the patch `LOGAREA`, and more slowly for the larger patches (a saturating "log-linear relationship"). We now also know that bird abundance changes in a non-linear way with the grazing intensity `FGRAZE`. But how do these effects combine together? Would a small patch with low grazing intensity have more birds than a larger patch with high grazing intensity? Could the poor fit of the `ABUND ~ LOGAREA` model for the large patches be improved, if we accounted for grazing intensity in the patches?

3. As previously we want to treat `AREA` as a log-transformed area to limit the influence of the few disproportionately large patches, and `GRAZE` as a categorical variable with five levels. So the first thing we need to do is create the corresponding variables in the loyn dataframe, called `LOGAREA` and `FGRAZE`.

4. Explore the relationship between grazing and patch area, using a scatterplot. Hint: you may want to use `GRAZE` rather than `FGRAZE` for this. Is there any variability in patch area within each grazing level? Is the sampling design balanced, i.e., is the whole range of patch areas evenly represented at each grazing level?

5. You could explore the joint effect of `FGRAZE` and `LOGAREA` on `ABUND`, using panel plots. Hint: See the function `coplot` in the Data exploration lecture slide 24, and/or the help page for `coplot`. Factor levels increase from the bottom-left panel to the top-right panel. What pattern do you see? What do you expect your model results to look like?

6. Fit an appropriate linear model in R to explain the variation in the response variable, `ABUND` with the explanatory variables `LOGAREA` and `FGRAZE` acting additively. Hint: `+` is the addition symbol!. Remember to use the `data =` argument. Assign this linear model to an appropriately named object, like `birds.add.1` (yes, you could be right if you sense there are more of these to come).

7. Produce the ANOVA table using the `anova()` function on the model object. What null hypotheses are being tested? Hint: the `anova()` function performs sequential tests. Do you reject or fail to reject the null hypotheses? What percentage of variation does the model explain overall? Hint: (SST-SSE)/SST. How much do `LOGAREA` and `FGRAZE` explain respectively?

8. Since the `anova()` function does sequential tests of the effects, the results could be different if we put `FGRAZE` first. Run the corresponding model and its analysis of variance. What null hypotheses are being tested? Do you reject or fail to reject the null hypotheses? What percentage of variation does the model explain overall? Hint: (SST-SSE)/SST. How much do `LOGAREA` and `FGRAZE` explain respectively?

9. Use the `summary()` function on the first model object to produce the table of parameter estimates. Using this output take each line in turn and answer the following questions: (A) what does this parameter represent, mathematically? (B) What is the biological interpretation of the corresponding estimate? (C) What is the null hypothesis associated with it? (D) Do you reject or fail to reject this hypothesis?

10. Now that you have interpreted all the coefficients, let's check that it all fits together. Write down the equation of the model with the appropriate parameter estimates from the summary. By hand, calculate the predicted bird abundance (A) for a patch with `LOGAREA`= -0.5 and `GRAZE`= 1, and (B) for a patch with `LOGAREA`= -0.5 and `GRAZE`= 3. Can you predict the difference in expected abundance between (A) and (B) before doing the calculation? Hint: the difference between `GRAZE3` and `GRAZE1` for a given patch area. Now, predict (C) for `LOGAREA`= 0.5 and `GRAZE`= 3. What does the difference between (C) and (B) correspond to?

11. Now let's check the assumptions of your linear model by creating plots of the residuals from the model. Remember, that you can split your plotting device into 2 rows and 2 columns using the `par()` function before you create the plots. Check each of the assumptions using these plots and report whether your model meets these assumptions.

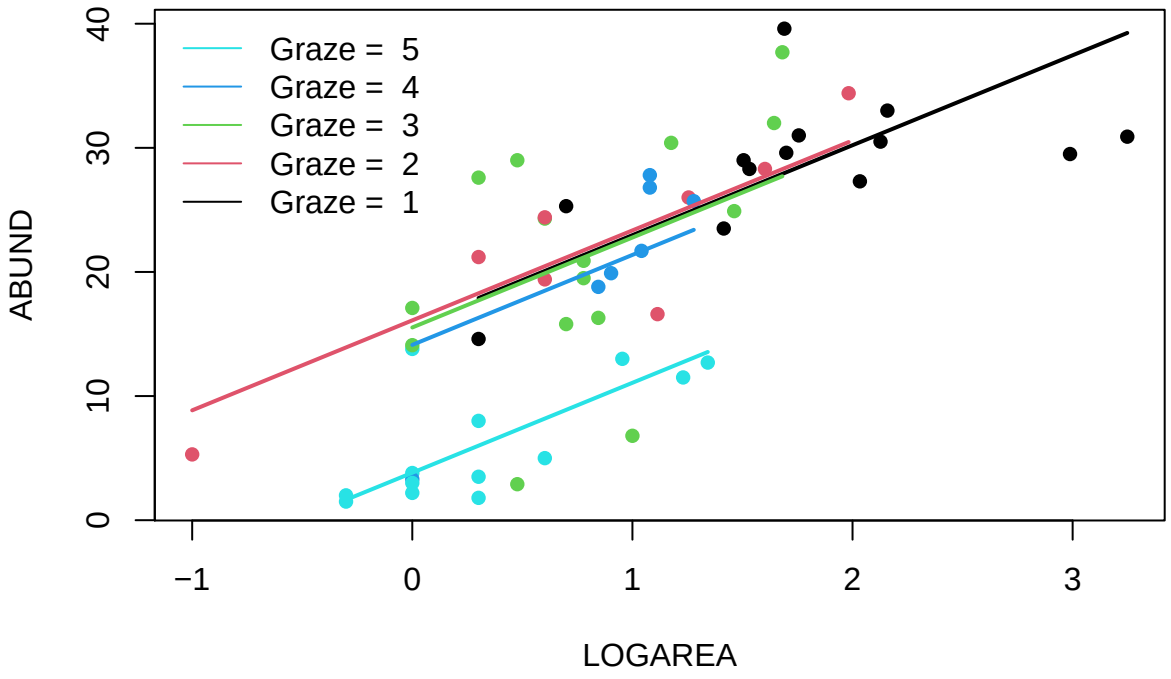
12. Let's plot the predictions of your initial model to figure out how it really fits the data. Here's a recipe, using the `predict()` function.

- plot the raw data, using a different colour per `FGRAZE` level
- for each `FGRAZE` level in turn,
- create a sequence of `LOGAREA` from the minimum value to the maximum within the grazing level (unless you wish to predict outside the range of observed values)
- store it in a data frame (e.g. `dat4pred`) containing the variables `FGRAZE` and `LOGAREA`. Remember that `FGRAZE` is a factor, so it requires double quotes.
- add a predicted column containing the predictions of the model for the new data frame, using `predict()`
- plot the predictions with the appropriate colours

See the script below, for one of many ways of doing this.

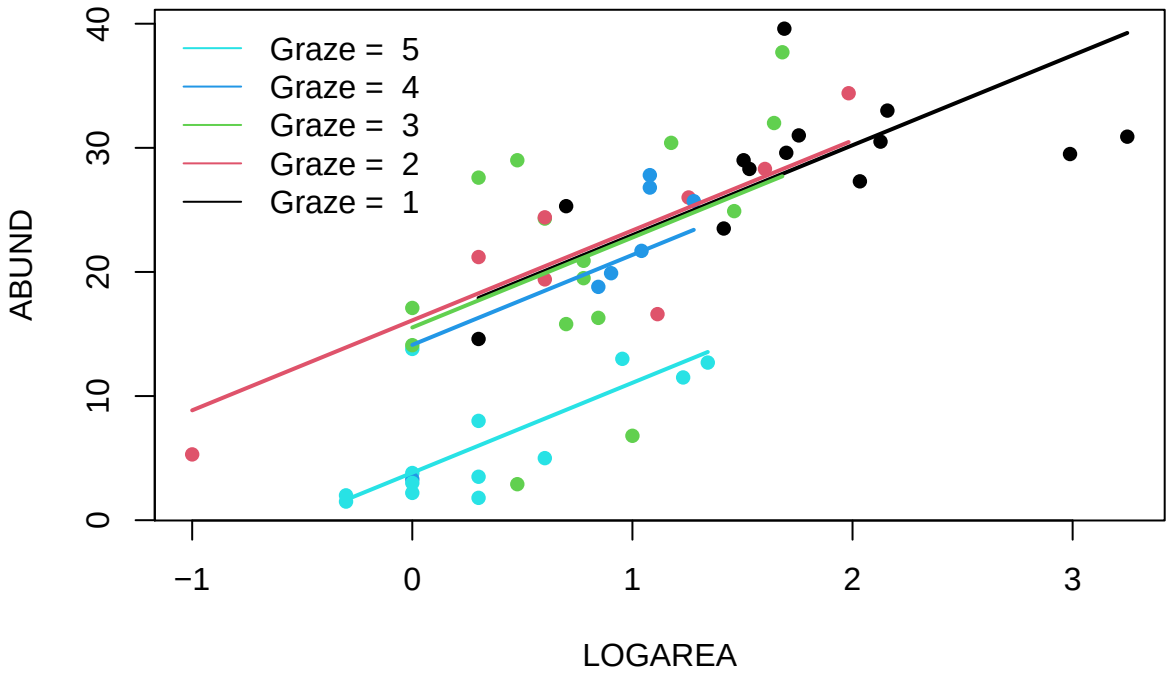
```
par(mfrow= c(1, 1))
plot(ABUND ~ LOGAREA, data= loyn, col= GRAZE, pch= 16)
# Note: # color 1 means black in R
# color 2 means red in R
# color 3 means green in R
# color 4 means blue in R
# color 5 means cyan in R

# FGRAZE1
# create a sequence of increasing Biomass within the observed range
LOGAREA.seq<- seq(from= min(loyn$LOGAREA[loyn$FGRAZE == 1]),
                  to= max(loyn$LOGAREA[loyn$FGRAZE == 1]),
                  length= 20)
# create data frame for prediction
dat4pred<- data.frame(FGRAZE= "1", LOGAREA= LOGAREA.seq)
# predict for new data
```

```r
dat4pred$predicted<- predict(birds.add.1, newdata= dat4pred)
# add the predictions to the plot of the data
lines(predicted ~ LOGAREA, data= dat4pred, col= 1, lwd= 2)

# FGRAZE2
LOGAREA.seq<- seq(from= min(loyn$LOGAREA[loyn$FGRAZE == 2]),
                  to= max(loyn$LOGAREA[loyn$FGRAZE == 2]),
                  length= 20)
dat4pred<- data.frame(FGRAZE= "2", LOGAREA= LOGAREA.seq)
dat4pred$predicted<- predict(birds.add.1, newdata= dat4pred)
lines(predicted ~ LOGAREA, data= dat4pred, col= 2, lwd= 2)

# FGRAZE3
LOGAREA.seq<- seq(from= min(loyn$LOGAREA[loyn$FGRAZE == 3]),
                  to= max(loyn$LOGAREA[loyn$FGRAZE == 3]),
                  length= 20)
dat4pred<- data.frame(FGRAZE= "3", LOGAREA= LOGAREA.seq)
dat4pred$predicted<- predict(birds.add.1, newdata= dat4pred)
lines(predicted ~ LOGAREA, data= dat4pred, col= 3, lwd= 2)

# FGRAZE4
LOGAREA.seq<- seq(from= min(loyn$LOGAREA[loyn$FGRAZE == 4]),
                  to= max(loyn$LOGAREA[loyn$FGRAZE == 4]),
                  length= 20)
dat4pred<- data.frame(FGRAZE= "4", LOGAREA= LOGAREA.seq)
dat4pred$predicted<- predict(birds.add.1, newdata= dat4pred)
lines(predicted ~ LOGAREA, data= dat4pred, col= 4, lwd= 2)

# FGRAZE5
LOGAREA.seq<- seq(from= min(loyn$LOGAREA[loyn$FGRAZE == 5]),
                  to= max(loyn$LOGAREA[loyn$FGRAZE == 5]),
                  length= 20)
dat4pred<- data.frame(FGRAZE= "5", LOGAREA= LOGAREA.seq)
dat4pred$predicted<- predict(birds.add.1, newdata= dat4pred)
lines(predicted ~ LOGAREA, data= dat4pred, col= 5, lwd= 2)

legend("topleft",
 legend= paste("Graze = ", 5:1),
 col= c(5:1), bty= "n",
 lty= c(1, 1, 1),
 lwd= c(1, 1, 1))
```



[Optional] Alternative method:

```
# Okay, that was a long-winded way of doing this.
# If, like me, you prefer more compact code and less risks of errors,
# you can use a loop, to save repeating the sequence 5 times:
par(mfrow= c(1, 1))
plot(ABUND ~ LOGAREA, data= loyn, col= GRAZE, pch= 16)

for(g in levels(loyn$FGRAZE)){# `g` will take the values "1", "2",..., "5" in turn
    LOGAREA.seq<- seq(from= min(loyn$LOGAREA[loyn$FGRAZE == g]),
                                        to= max(loyn$LOGAREA[loyn$FGRAZE == g]),
                                                length= 20)
    dat4pred<- data.frame(FGRAZE= g, LOGAREA= LOGAREA.seq)
    dat4pred$predicted<- predict(birds.add.1, newdata= dat4pred)
    lines(predicted ~ LOGAREA, data= dat4pred, col= as.numeric(g), lwd= 2)
}
legend("topleft",
 legend= paste("Graze = ", 5:1),
 col= c(5:1), bty= "n",
 lty= c(1, 1, 1),
 lwd= c(1, 1, 1))
```



13. What have we learned from this analysis so far?

End of the Linear model with additive continuous and categorical predictors exercise

Exercise 2 (optional extra): Linear model with interactive continuous and categorical predictors

This exercise is optional. You will get further opportunities to work with interactions in other practicals. You will probably not have enough time during the timetabled session, but if you wish to do it at a later time and have questions pertaining to this exercise, you are welcome to discuss them with a demonstrator in one of the regular or one of the catch-up sessions.

Let's now explore this question left unanswered: "is it okay to assume the effect of `LOGAREA` to be the same for all grazing levels?" This is effectively asking if we should let the slope of `LOGAREA` vary across `FGRAZE` levels, which is the definition of an interactive effect.

14. Fit the corresponding linear model in R to explain the variation in the response variable, `ABUND` with the explanatory variables `LOGAREA` and `FGRAZE` acting in interaction. Hint: this time, the acting will be done by stars `*`. Remember to use the `data =` argument. Assign this linear model to an appropriately named object, like `birds.inter`.

15. Produce the ANOVA table using the `anova()` function on the model object. What null hypotheses are being tested? Hint: the `anova()` function performs sequential tests. Which of these hypotheses are relevant to us? Do you reject or fail to reject the null hypotheses?

16. Use the `summary()` function on the model object to produce the table of parameter estimates. Using this output take each line in turn and answer the following questions: (A) what does this parameter represent, mathematically? (B) What is the biological interpretation of the corresponding estimate? (C) What is the null hypothesis associated with it? (D) Do you reject or fail to reject this hypothesis?

17. Check if you can make sense of the model structure, by writing down the equation of the model with the appropriate parameter estimates from the summary. Then, calculate again the predicted bird abundance (A) for a patch with LOGAREA= 2.5 and GRAZE= 1, and (B) for a patch with LOGAREA= -0.5 and GRAZE= 5.

18. Now let's check the assumptions of your linear model by creating plots of the residuals from the model. Remember, that you can split your plotting device into 2 rows and 2 columns using the `par()` function before you create the plots. Check each of the assumptions using these plots and report whether your model meets these assumptions.

19. Let's now plot the predictions of the interactive model to figure out how it really fits the data. Hint: the script from question 12 should work all the same, since the predictor variables involved in the equation are the same. Don't forget to update the name of the model!

20. Do you think the model is biologically plausible? Is it supported statistically?

End of the Linear model with interactive continuous and categorical predictors exercise