Exercises

Exercise: Linear model with single continuous explanatory variable

1. Either create a new R script (perhaps call it linear_model_1) or continue with your previous data exploration 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 summary of the graphical data exploration you performed during the previous exercise. The aim of this exercise is to get familiar with fitting a simple linear model with a continuous response variable, bird abundance (`ABUND`) and a single continuous explanatory variable forest area (`AREA`) in R. Ignore the other explanatory variables for now.

3. Create a scatterplot of bird abundance and forest patch area to remind yourself what this relationship looks like. Remember you may have to transform the `AREA` variable, return to your data exploration if in doubt. Try to remember which is your response variable (y axis) and which is your explanatory variable (x axis). Now fit an appropriate linear model to describe this relationship using the `lm()` function. Remember to use the `data =` argument. Assign this linear model to an appropriately named object (`loyn.lm` if you imagination fails you!).

4. Obtain the ANOVA table by using the `anova()` function on your model object. What is the null hypothesis you are testing here? Do you reject or fail to reject this null hypothesis? Explore the ANOVA table and make sure you understand the different components. Refer back to the lectures if you need to remind yourself or ask an instructor to take you through it.

5. Now display the table of parameter estimates using the `summary()` function on your model object. Again, make sure you understand the different components of this output and be sure to ask if in doubt. What are the estimates of the intercept and slope? Write down the word equation of this linear model including your parameter estimates (hint: think y = a + bx).

6. What is the null hypothesis associated with the intercept? What is the null hypothesis associated with the slope? Do you reject or fail to reject these hypotheses?

7. Looking again at the output from the `summary()` function how much variation in bird abundance is explained by your log transformed `AREA` variable?

8. Now onto a really important part of the model fitting process. Let's check the assumptions of your linear model by creating plots of the residuals from the model. Remember, you can easily create these plots by using the `plot()` function on your model object (`loyn.lm` or whatever you called it). Also remember that if you want to see all plots at once then you should split your plotting device into 2 rows and 2 columns using the `par()` function before you create the plots (Section 4.4). Can you remember which plot is used to check the assumption of normality of the residuals? What is you assessment of this assumption? Next, check the homogeneity of variance of residuals assumption. Can you see any patterns in the residuals versus fitted value plots? Is there more or less equal spread of the residuals? Finally, take a look at the leverage and Cooks distance plot to assess whether you have any residuals with high leverage or any influential residuals. What is your assessment? Write a couple of sentences to summarise your assessment of the modelling assumptions as a comment in your R code.

9. Using your word equation from Question 5, how many birds do you predict if `AREA` is 100 from your model?

10. Calculate the fitted values from your model using the `predict()` function and store these predicted values in an object called `pred.vals`. Remember, you will first need to create a dataframe object containing the values of log transformed `AREA` you want to make predictions from. Refer back to the model interpretation video if you need a quick reminder of how to do this or ask an instructor to take you through it if you're in any doubt (they'd be happy to take you through it).

11. Now, use the `plot()` function to plot the relationship between bird abundance (`ABUND`) and your log transformed `AREA` variable. Also add some axes labels to aid interpretation of the plot. Once you've created the plot then add the fitted values calculated in Question 10 as a line on the plot (you will need to use the `lines()` function to do this but only after you have created the plot).

12. OK, this is an optional question so feel free to skip if you've had enough! (you can find the R code for this question in the exercise solutions if you want to refer to it at a later date). Let's recreate the plot you made in Question 11, but this time we'll add the 95% confidence intervals in addition to the fitted values. Remember, you will need to use the `predict()` function again but this time include the`se.fit = TRUE` argument (store these new values in a new object called `pred.vals.se`). When you use the `se.fit = TRUE` argument with the `predict()` function the returned object will have a slightly different structure compared to when you used it before. Use the `str()` function on the `pred.vals.se` to take a look at the structure. See if you can figure out how to access the fitted values and the standard errors. Once you've got your head around this you can now use the `lines()` function three times to add the fitted values (as before) and also the upper and lower 95% confidence intervals. Don't forget, if you want the 95% confidence intervals then you will need to multiply your standard error values by the critical value of 1.96. If you need a reminder then take a look at the video on confidence intervals on the course MyAberdeen site (in the 'Introductory statistics lectures' folder which you can find in the 'Course Information' learning module).

13. And another optional question (honestly, it's optional!). This time plot the relationship between bird abundance (`ABUND`) and the original untransformed `AREA` variable. Now back-transform your fitted values (remember you got these with the `predict()` function) to the original scale and add these to the plot as a line. Hint 1: you don't need to reuse the `predict()` function, you just need to back-transform your `LOGAREA` values. Hint 2: remember if you used a $\log_{10}$ transformation (`log10()`) then you can back-transform using `10^LOGAREA` and if you used a natural log transformation then use `exp(LOGAREA)` to back-transform. Comment on the differences between the plot on the transformed (log) scale and the plot on the back-transformed scale in your R script.

End of the linear model with single continuous explanatory variable exercise