Exercises for multiple regression

Ananthan Ambikairajah

June 12th, 2019

Table of Contents

Exercise 1

1.1 Pub data – Visualising your data/models

Load the dataset "pubs.dat" and save the output to a variable (i.e. pubs_data). The dataset consists of two variables including "pubs" – the number of pubs in a region and "mortality" – the number of deaths in a region.

1. Inspect the data using head(), tail(), summary() and str().
2. Construct a baseline linear model with no predictors to examine "mortality". Save the output to a variable. Inspect the output of the model using summary().
3. Construct a simple linear regression of "mortality" as a function of "pubs". Save the output to a variable. Inspect the output of the model using summary().
4. Create a scatterplot of the data (i.e. the relationship between "pubs" and "mortality") with the predictor (i.e. pubs coefficient) as the predictive line.
5. Extract the residuals from the simple linear regression model. What do you notice? Are residuals enough for assessing a model?

1.2 Extension activity

1. Run the same simple regression as 1.1, without the outlier. Save the output to a variable. Inspect the output using summary().
2. Draw a new regression line over the previous plot that you have made and compare the two regression models on the scatterplot (i.e. with and without the outlier). Notice whether the intercept and slope changes.

Exercise 2

2.1 Album data - Building Models

Load the dataset "Album Sales 2.dat". It consists of four variables including "adverts" – the amount of money spent on advertisements in the week before the release of the album (thousands of dollars); "sales" – the number of record sales in the week after the release of the album (thousands); "airplay" – the number of times the song is played on the radio in the week before the release of the album and "attract" – the

attractiveness of the band on a scale of 0 (hideous potato heads) to 10 (gorgeous sex objects).

1. Inspect the data using head(), tail(), summary() and str().
2. Construct a baseline linear model with no predictors to examine "sales". Save the output to a variable. Inspect the output of the model using summary().
3. Construct a simple linear regression of "sales" as a function of "adverts". Inspect the output of the model using summary().
4. Construct a multiple regression model of "sales" as a function of "adverts", "airplay" and "attract". Inspect the output of the model using summary().
5. Compare your models using anova() and AIC() to assess which model fits the dataset the best.
6. Use confint() to obtain confidence intervals for the parameters in your chosen model.
7. Write out the multiple regression equation, using the beta-coefficients produced by your chosen model.
8. Write out the interpretation for each predictor within your chosen model in words.
9. Use lm.beta() from the "QuantPsyc" package to obtain standardised beta estimates from your chosen model.

Exercise 3

3.1 Model Diagnostics – Outliers and Influential cases

Conduct model diagnostics on your chosen model (from the previous exercise) to detect any potential outliers/influential cases. Store the output of your model diagnostics into the "album_data" dataframe. Use the information below to help conduct model diagnostics:

Residuals can be obtained with the resid() function, standardised residuals with the rstandard() function and studentised residuals with the rstudent() function. Cook's distances can be obtained with the cooks.distance() function, DFBeta with the dfbeta() function, DFFit with the dffits() function, leverage (hat) values with the hatvalues() function and the covariance ratio with the covratio() function.

Inspect your dataframe to examine the output of the model diagnostics. Hint, use head() to inspect the first 6 rows and round() to round the values to 2 digits.

3.2 Model Diagnostics – Plotting Outliers and Influential cases

Plot your model diagnostics by wrapping the output of the functions listed above in plot() to visually detect any potential outliers/influential cases.

1. Check how many standardised residuals are greater than 1.96 or less than -1.96. Given the sample size, how many standardised residuals outside of these limits are reasonable to expect?

Hint: Remember that R stores "TRUE" values as 1 and "FALSE" values as 0.

2. Can you use subsetting techniques to determine which exact cases were "TRUE" for the condition specified above? Include the standardised residuals of these cases, when subsetting.
3. How many standardised residuals are greater than 2.56 or less than -2.56. Given the sample size of the initial dataset, how many standardised residuals are reasonable to expect?
4. Are there any particular residuals that are a cause for concern? Which residual(s) should you investigate further, if any? Why/Why not?
5. Use subsetting techniques to investigate the casewise diagnostics for the cases that you would like to investigate further, including cooks distance, leverage values and covariance ratios.
6. Calculate the average leverage values. Are there any values that exceed twice (or three times) the average?
7. Examine the covariance ratios. Are there any cases that cause concern? What do the cook's distance of these cases reveal?

Note: Casewise diagnostics are most informative when viewed together using different tests, opposed to viewing the results of individual tests in isolation. An understanding of what each diagnostic test is measuring will help you make an informed choice about whether it is an outlier/influential case.

Exercise 4

4.1 Model Generalisability – Testing Assumptions

1. Test the assumption of independent errors using the Durbin-Watson test. This statistic can be obtained along with a measure of autocorrelation and a p-value in R using the dwt() function.
2. Test the assumption of multicollinearity using the variance inflation factor i.e. vif() function. Check the tolerance statistic and the value of the mean variance inflation factor.
3. Check assumptions relating to residuals by using the hist() and plot() functions.