ST2304 Exercises Week 7: Multiple Regression

Bob O'Hara

19 February 2018

Problem 1: Life Expectancy (again)

In the last set of exercises, you looked at life expectancy, and transformations to try to get a good model. Here we will try another approach, using polynomials and multiple regression. First, read in the data and remove South Africa:

```
rawdata <- read.csv("../Data/LifeExpectancy.csv") # NOTE: the file path might be different!
NoSA <- rawdata[rawdata$Country!="South Africa",]
```

Now, we want to fit polynomial models:

$$y_i = \alpha + \sum_{p=1}^{P} \beta_p x_i^p + \varepsilon_i = \alpha + \beta_1 x_i + \beta_2 x_i^2 + \cdots + \beta_p x_i^p + \varepsilon_i$$

(optional problem: explain why this is equivalent to the model $y_i = \alpha^* + \sum_{p=1}^{P} \beta_p^* (x_i - \bar{x})^p + \varepsilon_i$)

To fit a polynomial, we need to fit a model with terms for all of the powers (linear, quadratic, cubic etc.). We could do this by creating new variables X, X^2, X^3 etc, but it is a bit easier to fit them in the model just using the X. Unfortunately just writing Y ~ X + X^2 does not work (for good but obscure reasons), instead we have to write Y ~ X + I(X^2):

```
mod.does.not.work <- lm(Y ~ X + X^2)
mod.does.work <- lm(Y ~ X + I(X^2))
```

So, you should fit a series of polynomial models to the data, up to order 10, i.e. up to $p = 10$ (this is either a lot of copying and pasting, or if you want to play a bit more with R, look at the update() function: https://tinyurl.com/y8yswwqo).

1. For each model, find the R^2, and look at and describe how the R^2 changes with the order of the polynomial (hint: plot them).
2. There are several ways to decide which order of polynomial is best (e.g. quadratic, cubic etc.), but without trying them, what order do you prefer, and why? What factors are did you consider when making this decision? (there is no right or wrong answer to this, rather I am hoping you will think about what factors other than model fit might be important)
3. My conclusion from looking at the data last week was that log-transforming the x-axis (health spending) was the best alternative. So fit that model and compare how well that model to the polynomial models.
4. Check how well the model fits - are there any outliers, any influential points, any bigger problems (e.g. heteroscedasticity)?
5. Plot the fitted model with the data (to do this, use predict() to predict new data over the range of the data, and lines() to plot the fitted model, You might also have to use seq(): the help page is actually helpful here). Use this to summarise (in words and possibly numbers) what the model says about the relationship between health spending and life expectancy.

Problem 2: When regression goes bad

Multiple regression has more variables, so it gets more complicated. These simulations show some of the issues that can arise. All the models are bivariate, i.e.

$$y_i = \alpha + \beta_1 x_{1i} + \beta_2 x_{2i} + \varepsilon_i$$
$$\varepsilon_i \sim N(0, \sigma^2)$$

But the x's and β's will change. For all of these examples we will use a sample size of 50, $\alpha = 0$ (if you want to change α, go ahead and see what happens), and $\sigma^2 = 1$. We will want to simulate the x's so that they are correlated, which we can do using the mvrnorm() function in the MASS package. The following code does this (with a correlation between the x's of 0.5), and puts the x's into a matrix with 2 columns:

```
N <- 50
library(MASS)
muX <- c(0,0) # mean of bivariate distribution
Corr <- 0.5 # correlation
sigmaX <- matrix(c(1,Corr,Corr,1), nrow=2) # covariance matrix
x <- mvrnorm(N, muX, Sigma=sigmaX) # 2 columns: x[,1] & x[,2]
```

1. Simulate x_1 and x_2 from a standard normal distribution with no correlation. Then simulate the model (above) with $\beta_1 = 1$ and $\beta_2 = 1$. Plot y against each x, and regress y against each x separately. Explain the results. Then regress y against both x's, and again explain the results. This is a straightforward example, and here is some code to simulate the regression, but with the wrong parameters:

```
N <- 50; alpha <- 0; sigma <- 1 # same throughout
beta1 <- -20
beta2 <- 5000

mu <- alpha + beta1*x[,1] + beta2*x[,2]
y <- rnorm(N, mu, sigma)
```

2. Simulate x_1 and x_2 from a standard normal distribution with correlation 0.7. Then simulate the model (above) with $\beta_1 = 1$ and $\beta_2 = 0$, i.e. where there is no effect of β_2. Plot y against each x, and regress y against each x separately. Explain the results. Then regress y against both x's, and again explain the results.
3. Simulate x_1 and x_2 from a standard normal distribution with correlation 0.7. Then simulate the model (above) with $\beta_1 = 1$ and $\beta_2 = 1$. As before, plot y against each x, and regress y against each x separately. Explain the results. Then regress y against both x's, and again explain the results.
4. Now simulate x_1 and x_2 from a standard normal distribution with correlation -0.8. Then simulate the model (above) with $\beta_1 = 5$ and $\beta_2 = 5$. As before, plot y against each x, and regress y against each x separately. Try to explain the results. Then regress y against both x's, and again explain the results. This might be more tricky to understand, but helpful for understanding what can happen.

Problem 3: Clutch Size

We can use the BirdBrains data to look at more than brains. Here we will try to explain clutch size - why do some species tend to lay more eggs in a clutch than others? First download the data from Blackboard. Then we extract the data we want, and scale the variables so that they have mean zero and variance 1. We want to explain clutch size ("Clutch.size") with the other covariates: Maximum.lifespan, Age.at.first.reprodction, Incubation.length, Mean.latitude, logBodyMass, logBrainMass.

```
BirdBrains <- read.csv('../Data/BirdBrains.csv') # beware the file path: this is for my computer!

Names <- c("Order", "Family", "Species.name.")
Variables <- c("Maximum.lifespan", "Age.at.first.reprodction", "Incubation.length",
               "Clutch.size","Mean.latitude", "logBodyMass", "logBrainMass")
```

```
BirdClutch <- BirdBrains[,c(Names,Variables)] # select the variables we want
BirdClutch[,Variables] <- scale(BirdClutch[,Variables]) # standardise the variables
```

1. Why is scaling like this a good idea? It might help to this about how the coefficient is interpreted (it is the change in the response if the covariate is changed by 1 unit: scaling changes what the unit is).
2. Fit univariate models, i.e. explain clutch size by each of the covariates individually. Summarise the effect - how much of the variation in clutch size does each variable explain, and in what direction (e.g. does a bigger body mass mean more or less eggs?)?
3. Fit a single model with all of the variables in it. Again, summarise the effects - how much of the variation in clutch size does each variable explain, and in what direction?
4. How (and why) do the results from fitting the individual models and from fitting one model with all variables differ?
5. Check the model fit (residuals etc.), and comment on it, and what you could try to improve the model. I don't expect you to spend time finding the best model, just show that you appreiate what might be problematic.