Linear Models, Marked Practical

Week 4, MT 2024

- This practical handout contains two sections. **Write a report on the Exercise in Section 2 only.**
- **The report has a word limit of 2000 words.** This word limit is on the main body of the report. Equations, tables, figures, captions, appendices to your report and computer code do not contribute to the word count.
- **You should use your anonymous practical ID (of the form P123, and not your name)** for the cover page of the report, and you should name the PDF file you upload using that same ID (e.g. `P123.pdf`).
- **You should submit your report via the Canvas system.** There will be instructions about how to do this on the Canvas Practicals page next to this practical handout.

You are welcome to ask questions about the examples in Section 1 during the practical session. If this was an assessed practical I would not be able to answer questions regarding the exercise in Section 2, with the sole exception of questions relating to a limited number of programming issues. In order to mimic the setup for an assessed practical, only limited help will be available for the exercise in Section 2.

1 Examples for practice, NOT MARKED

(a) A quadratic term

If you want e.g. $y = \beta_0 + \beta_1 x + \beta_2 x^2 + \epsilon$, then you should use `y ~ x + I(x^2)`. It is important to use `I()` here. If you don't use `I()` then R will treat the `^` in `x^2` as a formula operator (see `?formula`) and you will not get the $\beta_2 x^2$ term that you want.

```
## one of the introductory examples
plot(dist ~ speed, data = cars)

cars0.lm <- lm(dist ~ speed, data = cars)
cars1.lm <- lm(dist ~ speed + I(speed^2), data = cars)
cars2.lm <- lm(dist ~ speed + speed^2, data = cars)

summary(cars0.lm)
summary(cars1.lm)
summary(cars2.lm)
## cars2.lm is the same as cars0.lm and is probably not what was intended
```

(b) Box-Cox transformation

Suppose a normal linear model applies not to y, but to some power of y, say to y^λ. We can use the Box-Cox method to find the best value of λ. Where possible we might hope for an interpretable value of λ. Faraway (2015): "If explaining the model is important, you should round λ to the nearest interpretable value."

As λ varies in the range $(-2, 2)$ we get the inverse transformation ($\lambda = -1$), square and cube roots ($\lambda = \frac{1}{2}, \frac{1}{3}$), the original scale ($\lambda = 1$), as well as the squared case ($\lambda = 2$). We want a sensible $\lambda = 0$ case as well, so the method actually works with the transformation to $y^{(\lambda)}$ where

$$y^{(\lambda)} = \begin{cases} \frac{y^\lambda - 1}{\lambda} & \lambda \neq 0 \\ \log y & \lambda = 0. \end{cases}$$

Note this is consistent because $\lim_{\lambda \to 0} \left(\frac{y^\lambda - 1}{\lambda} \right) = \log y$.

We assume all y_i values satisfy $y_i > 0$ (if not we could add a small constant to all y_is).

We can treat λ as a parameter and find the MLE: see Davison (2003, p389–390), or Faraway (2015, p134–137) for details.

Example (i)

```
## the boxcox() function is in the MASS package
library(MASS)

trees0.lm <- lm(Volume ~ log(Height) + log(Girth), data = trees)
par(mfrow = c(2, 2))
plot(trees0.lm)

## see ?boxcox
boxcox(Volume ~ log(Height) + log(Girth), data = trees,
       lambda = seq(-0.25, 0.25, length = 10))
abline(v = 0, col = "red")
```

```
trees1.lm <- lm(log(Volume) ~ log(Height) + log(Girth), data = trees)
par(mfrow = c(2, 2))
plot(trees1.lm)
```

Examples (ii)

```
lmod <- lm(sr ~ pop15 + pop75 + dpi + ddpi, data = LifeCycleSavings)
boxcox(lmod)
boxcox(lmod, lambda = seq(0.5, 1.5, by = 0.1))
## Faraway: "no good reason to transform"

## need the faraway package for the gala data
library(faraway)

lmod <- lm(Species ~ Area + Elevation + Nearest + Scruz + Adjacent, data = gala)
boxcox(lmod, lambda = seq(-0.25, 0.75, by = 0.05))
## Faraway:
## "... perhaps a cube root transformation might be best here.
## A square root is also a possibility ...
## Certainly there is a strong need to transform."
```

(c) Interactions

```
# an example from lectures
data(whiteside, package = "MASS")
gas2.lm <- lm(Gas ~ Temp * Insul, data = whiteside)
```

The term `Temp * Insul` in `gas2.lm` is shorthand for `1 + Temp + Insul + Temp:Insul`

In particular, the term `Temp:Insul` is the interaction between `Temp` and `Insul`

If we had three variables, say `a`, `b` and `c`, then for a model involving the main effects of `a`, `b` and `c`, plus all of the two-way interactions `a:b`, `a:c` and `b:c`, we can use the shorthand

```
y ~ (a + b + c)^2
```

which is the same as

```
y ~ 1 + a + b + c + a:b + a:c + b:c
```

(and for a model that also includes the three-way interaction `a:b:c` as well, use `y ~ a * b * c`).

In a similar way `y ~ a * (b + c)` is the same as

```
y ~ 1 + a + b + c + a:b + a:c
```

2 MARKED EXERCISE

The data in `swim.csv` are the competitors' times in some swimming races. The times are from the finals of individual events at the 2016 Olympics, and from the finals of similar events at the 2016 World Championships. The Olympic events were "long course" events – swum in a 50 metre pool; the World Championships were "short course" – swum in a 25 metre pool. The strokes swum in these events were freestyle, backstroke, breaststroke, or butterfly, and also medley. In a medley race, all four of the other strokes are swum, an equal number of lengths of each stroke.

For each event, the times of the finalists are recorded as well as some other information about the event. The variables recorded are:

- `event`, the name of the event, e.g. "50 m Freestyle"
- `dist`, the length of the event, in metres
- `stroke`, the stroke swum in the event
- `sex`, to indicate whether an event is women's or men's
- `course`, to indicate whether an event is short course or long course
- `time`, the time of one of the swimmers in the final, in seconds.

The file `swim.csv` is on Canvas.

You can load the data using something like:

```
swim <- read.csv("swim.csv")
```

Exercise:

Investigate and write a report on how race times depend on the other variables.

The main goal here is to obtain a suitable interpretable model and to give a full interpretation of that model.

You are also asked, using the same model, to predict times for four additional races.

(a) Perform an exploratory analysis of the data and summarise your findings. As well as producing suitable plots that examine the relationship between race times and the available explanatory variables, you should also present some numerical summaries. The text of your report should say what you want the reader to notice from the numerical summaries and plots.

(b) Model the relation between race times and the other variables that are available using an appropriate normal linear model. Stick to normal linear models, with fixed effects. Choose and fit an appropriate initial model.

(c) Assess the quality of model fit using suitable methods. Use the results obtained to consider whether any revisions to the model are appropriate. Make any appropriate revisions and carry out model selection.

In doing (b) and (c) above:

- You are welcome to consider transformations of variables where appropriate.

- Carry out model selection to examine the relationship between race times and the possible explanatory variables.
- Explain any model revisions that you make and your reasons for making them, or explain your reasons for not needing to make any revisions.
- You may want to consider some models that include two-way interactions, but remember when selecting your model that the main aim here is interpretation.

(d) Interpret your final normal linear model carefully, preferably on the original scale rather than on a transformed scale should your model involve transformed quantities.

(e) Using the model obtained above, obtain predicted times and prediction intervals for the four additional races below (prediction intervals, not confidence intervals).

Comment on the predictions you obtain.

name	dist	stroke	sex	course
RaceA	400	Freestyle	F	Long
RaceB	50	Backstroke	F	Long
RaceC	400	Butterfly	F	Long
RaceD	100	Medley	F	Long