Tutorial 1

STAT3015/4030/7030 Generalised Linear Modelling

The Australian National University

Week 1, 2017

Overview

Welcome

Welcome to the GLM tutorial

- Yang Yang (CBE 3.53)
- Consultation hours: 12:00 to 3:00 pm every Friday
- Consultation venue: CBE 3.09
- Extra consultation: by appointment

A brief introduction . . .

- This course is an extension to STAT2008/STAT6038 Regression Modelling.
- In the previous course, we mainly consider continuous data such as weight, age, rainfall, etc.
- We used the conventional 0 and 1 coding to treat gender (random variable).

A brief introduction . . .

- This course is an extension to STAT2008/STAT6038 Regression Modelling.
- In the previous course, we mainly consider continuous data such as weight, age, rainfall, etc.
- We used the conventional 0 and 1 coding to treat gender (random variable).
- **What if we have categorical variables with more than two levels? e.g. age groups of 20s, 30s, 40s, etc.**
- **What if we have more than one categorical variable? e.g. we need to consider age group, gender, social-economic status, etc.**

Learning outcomes

- We are going to learn "generalised linear modelling methods, with emphasis on, but not limited to, common methods for **analysing categorical data**".
- We will learn theories (assumptions, formulas, diagnostics) underlying multiple linear regression, log-linear models for contingency tables, logistic regression for binary response, etc.
- We will also learn how to perform statistical analysis using R/RStudio.

- Assessments include a Wattle quiz and two assignments.
- Two assignments asking you to analyse given data using R. Results should be submitted as a report consisting relevant R outputs and necessary interpretations.
- Detailed assignment specification will be handed out later.
- We will firstly do a brief review of R basics.

Adjust RStudio settings on ANU PCs

- Find a place to store your working files. H: drive or a USB drive.
- Installing a package can be done by clicking or `install.packages(...)`
- Loading a particular package using `library(...)` or `require(...)`
- Save occasionally while working on a large project, e.g. assignments.

Some basic commands

- Simple linear regression command `lm(Y ~ X)`
- `summary(...)` and interpretation of coefficients
- `plot(...)`
- Checking assumptions using diagnostic plots

R code to (a)

- `attach(child.iq)`
- `child.iq.lma <-lm(ppvt~momage)`
- `plot(momage, ppvt, pch=20)` and `abline(child.iq.lma)`
- `par(mfrow=c(2, 2))` and `plot(child.iq.lma)`
- `summary(child.iq.lma)`

Explanation to (a)

- **No obvious evidence** of any dependence structure in the residuals, any non constant variance or major departures from normality.
- Note that the **maximum** of mother's age in the data is 29. It is dangerous to use this model to **extrapolate** outside this range.
- Our model assumes that the true underlying relationship between a mother's age and a child's IQ is **linear**. Also, we are not controlling for **other factors**.

A preview of GLM

- To build GLM models we need to deal with categorical variables.
- The first type of GLM we are going to learn would be the analysis of variance (ANOVA) model.
- In the "Brick" we can find the following sentence: the ANOVA is seen as a method of testing whether the mean responses at **different values**, or **levels** of a categorical predictor, or *factor*, are all equal.

Then the question would be how to input (numerically coded) categorical variables into linear models.

R may misinterpret numerically coded variables as **continuous** if we include them directly.

The factor(...) command

- We use the `factor(...)` command to tell RStudio the number of levels contained in categorical variables.
- Explanation of `factor(...)` in R
  http://www.stat.berkeley.edu/classes/s133/factors.html

Indicator variables

- We often create indicators for categorical variables in GLM modelling.
- How should we generate indicator/dummy variables?

Indicator variables

- We often create indicators for categorical variables in GLM modelling.
- How should we generate indicator/dummy variables?
- We can manually assign values to observations. (time consuming)
- `ifelse(...)` function is very useful, especially for variables that can take several unique values.