4. Model fitting

Thomas Lumley
Ken Rice

Universities of Washington and Auckland

Seattle, June 2011
Regression commands

Two of the most important R commands;

• `lm()`: fits **Linear Models**

• `glm()`: fits **Generalized Linear Models**

(If you’ve used SAS, its glm is **not** the same as R’s)

‘Linear Regression’ and ‘Logistic Regression’ are special cases.

There’s a lot to learn here – entire graduate courses! – so the help files are huge. How are `lm()`, `glm()` used in genetics?
Linear regression, with SNPs

Many analyses fit the ‘additive model’

\[ y = \beta_0 + \beta \times \text{#minor alleles} \]
Linear regression, with SNPs

An alternative is the ‘dominant model’;

\[ y = \beta_0 + \beta \times (G \neq AA) \]
Linear regression, with SNPs

or the ‘recessive model’;

\[ y = \beta_0 + \beta \times (G = AA) \]
Linear regression, with SNPs

Finally, the ‘two degrees of freedom model’;

\[ y = \beta_0 + \beta_{Aa} \times (G == Aa) + \beta_{aa} \times (G == aa) \]
Use of \texttt{lm()} in genetics

The \texttt{lm()} command fits all of these, in the same way. Formally,

\begin{verbatim}
  lm(outcome \sim \text{genetic.predictor}, [...]
\end{verbatim}

estimates the association between \texttt{outcome} and \texttt{predictor}

The \textbf{optional} arguments [...] might be

- \texttt{data = my.data} – your dataset
- \texttt{subset = race=="CEPH"} – use partial data
- \texttt{weights =} – for advanced analyses
Use of \texttt{lm()} in genetics

How to make the \texttt{genetic.predictor} variable? Suppose you had genotypes stored as character strings ("AA"/"Aa"/"aa") in a vector \texttt{g}. You might use these commands:

<table>
<thead>
<tr>
<th>Chosen Model</th>
<th>Command to define variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Additive</td>
<td>\texttt{genetic.predictor &lt;- (g==&quot;Aa&quot;) + 2*(g==&quot;aa&quot;)}</td>
</tr>
<tr>
<td>Dominant</td>
<td>\texttt{genetic.predictor &lt;- (g==&quot;Aa&quot;)</td>
</tr>
<tr>
<td>Recessive</td>
<td>\texttt{genetic.predictor &lt;- g==&quot;aa&quot;}</td>
</tr>
<tr>
<td>2 degs of freedom</td>
<td>\texttt{genetic.predictor &lt;- factor(g)}</td>
</tr>
</tbody>
</table>

When \texttt{R} meets \texttt{FALSE} or \texttt{TRUE} in a ‘math’ setting, it will \texttt{coerce} them to be zero or one. So \texttt{1 + 2*TRUE} is 3, \texttt{TRUE + 2*FALSE} is 1, etc. Using \texttt{factor()} sets up several binary variables.

- There are \textit{many} other ways to do this!
  
  Use \texttt{table(g, genetic.predictor)} to check your method

- Often, genotypes may be stored as 0/1/2. This is easier to work with in \texttt{R} – but makes it harder to decide if A/C/G/T is the minor allele, or risk allele.
lm(): Estimates, Intervals, p-values

lm() produces **point estimates** for your model;

```r
> genetic.predictor <- (g=="Aa") + 2*(g=="aa") #using additive model
> my.lm <- lm( cholesterol ~ genetic.predictor )
> my.lm
Call:
  lm(formula = cholesterol ~ genetic.predictor)
Coefficients:
  (Intercept)   predictor
       0.2104       0.9507
```

– also available via `my.lm$coefficients` or `coef(my.lm)`.

The coefficients in the output tell you the **additive increase** in outcome associated with a **one-unit** difference in the genetic predictor.

The coefficient for `predictor` is in units of cholesterol per 'a' allele.
lm(): Estimates, Intervals, p-values

You will also want confidence intervals:

> confint.default(my.lm)

            2.5 % 97.5 %
(Intercept) 0.08391672 0.3368275
predictor 0.85279147 1.0486953

Remember to round these numbers to an appropriate number of significant figures! (2 or 3 is usually enough)

We are seldom interested in the Intercept.
\textbf{\texttt{lm()}: Estimates, Intervals, p-values}

Two-sided \textbf{p-values} are also available;

\begin{verbatim}
> summary(my.lm)

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.21037    0.06426   3.274  0.00119 **
predictor     0.95074    0.04977  19.101  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
\end{verbatim}

- In this data, we have \textbf{strong evidence} of an \textbf{additive effect} of the minor allele on cholesterol

- \texttt{summary(my.lm)} gives \textbf{many} other details – ignore for now

- Confidence intervals are just \texttt{Estimate \pm 2\times Std.Error}
**Use of `glm()` in genetics**

Logistic regression is the ‘default’ analysis for binary outcomes.

<table>
<thead>
<tr>
<th>Outcome</th>
<th>Type</th>
<th>Regression</th>
<th>Scale</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cholesterol</td>
<td>Continuous</td>
<td>Linear</td>
<td>Difference in Outcome</td>
</tr>
<tr>
<td>Blood Pressure</td>
<td>Continuous</td>
<td>Linear</td>
<td></td>
</tr>
<tr>
<td>BMI</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Death</td>
<td>Binary</td>
<td>Logistic</td>
<td>Ratio of odds</td>
</tr>
<tr>
<td>Stroke</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMI $&gt;30$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

What are **odds**? Really just **probability**...
Use of `glm()` in genetics

Odds are a [gambling-friendly] measure of chance;

```r
# Prob of survival, 10yrs
0.0 0.2 0.4 0.6 0.8 1.0
AA  Aa  aa
33% 2
66% 1
50% 1
50% 1
10% 1
90% 9
```
Use of `glm()` in genetics

Odds are a [gambling-friendly] measure of chance;

<table>
<thead>
<tr>
<th></th>
<th>Prob of survival, 10 yrs</th>
<th></th>
<th>Prob(death)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AA</td>
<td>66%</td>
<td>2</td>
<td>33%</td>
</tr>
<tr>
<td>Aa</td>
<td>50%</td>
<td>1</td>
<td>50%</td>
</tr>
<tr>
<td>aa</td>
<td>50%</td>
<td>1</td>
<td>10%</td>
</tr>
</tbody>
</table>

- 90% | 9
- 10% | 1
Use of \texttt{glm()} in genetics

Odds are a [gambling-friendly] measure of chance;

\begin{itemize}
  \item AA: 66\%
  \item Aa: 50\%
  \item aa: 33\%
\end{itemize}

\begin{itemize}
  \item 0.0 0.2 0.4 0.6 0.8 1.0
  \item Odds of death:
  \begin{itemize}
    \item 33\%: 1
    \item 66\%: 2
    \item 50\%: 1
    \item 50\%: 1
    \item 10\%: 1
    \item 90\%: 9
  \end{itemize}
  \item Prob of death:
  \begin{itemize}
    \item 90\%: 9
    \item 10\%: 1
  \end{itemize}
\end{itemize}

– so what are \textbf{odds ratios}?
Use of `glm()` in genetics

Using the data from slide 4.12:

```r
> genpred2 <- factor(g)  # the 2df model
> glm1 <- glm( dead10yrs ~ genpred2, family=binomial)
> coef(glm1)
pred2Aa     pred2aa
     0.6931     2.8904
```

- These are log odds ratio estimates; to transform to OR, use $e^{0.6931} = 2, e^{2.8904} = 18$

- They are given relative to the baseline group – ‘AA’ in this case

- Don’t forget the family=binomial argument!
Use of \texttt{glm()} in genetics

Confidence intervals, p-values as with \texttt{lm()}, \textbf{for the log odds ratios};

\begin{verbatim}
> confint.default(glm1)
2.5 %   97.5 %
genpred2Aa 0.1201986 1.2660957
genpred2aa 2.1148912 3.6658523
\end{verbatim}

\begin{verbatim}
> summary(glm1)

          Estimate Std. Error  z value Pr(>|z|)
genpred2Aa 0.69310   0.29232  2.3710   0.0177 *
genpred2aa 2.89042   0.39566  7.3049 2.77e-13 ***
\end{verbatim}

Use \texttt{exp()} to get odds ratio estimates, intervals; p-values are \textbf{scale-independent}
The formula syntax

We saw `lm(y ~ genetic.predictor)` and `glm(y ~ genpred2)`. To see how phenotype depends on several covariates, we specify e.g.

\[ y \sim \text{genotype.pred + age + sex} \]

– formally, this gives multivariate regression; the genotype.pred coefficients reflect the genotype effects adjusted for age and sex

- Separate covariates with ‘+’. This is not addition!
- For now, make predictor variables first, then do regression. It’s possible to do everything in one step, but use of e.g. ‘+’ will confuse R – unless you’re careful.
- For keen people; in the formula syntax, * indicates that interactions should be fitted, I() insulates mathematical operations, -1 removes the intercept... see ?formula
- For very keen people; `vcovHC()` in the sandwich package provides ‘robust’ standard errors; `coeftest()` in the lmtest package can use them to give ‘robust’ tests.