Linear, Generalized Linear, and Mixed-Effects Models in R

John Fox

McMaster University

ICPSR 2018

Linear and Generalized Linear Models in R

Topics

To be covered as time permits:

- Part 1
  - Multiple linear regression
  - Factors and dummy regression models
  - Overview of the `lm()` function
  - The structure of generalized linear models (GLMs) in R; the `glm()` function
  - GLMs for binary/binomial data and count data
  - Mixed-effects models for hierarchical and longitudinal data
- Part 2
  - Visualizing statistical models
  - Tests and confidence intervals for coefficients
  - Diagnostics for linear and generalized linear models

Linear Models in R

Arguments of the lm function

- lm(formula, data, subset, weights, na.action, method = "qr", model = TRUE, x = FALSE, y = FALSE, qr = TRUE, singular.ok = TRUE, contrasts = NULL, offset, ...)
- formula

Expression	*Interpretation*	*Example*
A + B	include both A and B	income + education
A - B	exclude B from A	a*b*d - a:b:d
A:B	all interactions of A and B	type:education
A*B	A + B + A:B	type*education
B %in% A	B nested within A	education %in% type
A/B	A + B %in% A	type/education
A^k	effects crossed to order k	(a + b + d)^2

Linear Models in R

Arguments of the lm() function

- data: A data frame containing the data for the model.
- subset:
  - a logical vector: subset = sex == "F"
  - a numeric vector of observation indices: subset = 1:100
  - a negative numeric vector with observations to be omitted: subset = -c(6, 16)
- weights: for weighted-least-squares regression
- na.action: name of a function to handle missing data; default given by the na.action option, initially "na.omit"
- method, model, x, y, qr, singular.ok: technical arguments
- contrasts: specify list of contrasts for factors; e.g., contrasts=list(partner.status=contr.sum, fcategory=contr.poly))
- offset: term added to the right-hand-side of the model with a fixed coefficient of 1.

Generalized Linear Models in R

Review of the Structure of GLMs

- A generalized linear model consists of three components:

1. A *random component*, specifying the conditional distribution of the response variable, y_i, given the predictors. Traditionally, the random component is an exponential family — the normal (Gaussian), binomial, Poisson, gamma, or inverse-Gaussian.
2. A linear function of the regressors, called the *linear predictor*,
$$\eta_i = \alpha + \beta_1 x_{i1} + \cdots + \beta_k x_{ik}$$
on which the expected value μ_i of y_i depends.
3. A *link function* $g(\mu_i) = \eta_i$, which transforms the expectation of the response to the linear predictor. The inverse of the link function is called the *mean function*: $g^{-1}(\eta_i) = \mu_i$.

Generalized Linear Models in R

Review of the Structure of GLMs

- In the following table, the logit, probit and complementary log-log links are for binomial or binary data:

Link	$\eta_i = g(\mu_i)$	$\mu_i = g^{-1}(\eta_i)$
identity	μ_i	η_i
log	$\log_e \mu_i$	e^{η_i}
inverse	μ_i^{-1}	η_i^{-1}
inverse-square	μ_i^{-2}	$\eta_i^{-1/2}$
square-root	$\sqrt{\mu_i}$	η_i^2
logit	$\log_e \dfrac{\mu_i}{1-\mu_i}$	$\dfrac{1}{1+e^{-\eta_i}}$
probit	$\Phi(\mu_i)$	$\Phi^{-1}(\eta_i)$
complementary log-log	$\log_e[-\log_e(1-\mu_i)]$	$1-\exp[-\exp(\eta_i)]$

- Generalized linear models are fit with the `glm()` function. Most of the arguments of `glm()` are similar to those of `lm()`:
  - The response variable and regressors are given in a model `formula`.
  - `data`, `subset`, and `na.action` arguments determine the data on which the model is fit.
  - The additional `family` argument is used to specify a *family-generator function*, which may take other arguments, such as a link function.

- The following table gives family generators and default links:

Family	*Default Link*	*Range of* y_i	$V(y_i \mid \eta_i)$
`gaussian`	`identity`	$(-\infty, +\infty)$	ϕ
`binomial`	`logit`	$\frac{0, 1, \ldots, n_i}{n_i}$	$\mu_i(1-\mu_i)$
`poisson`	`log`	$0, 1, 2, \ldots$	μ_i
`Gamma`	`inverse`	$(0, \infty)$	$\phi\mu_i^2$
`inverse.gaussian`	`1/mu^2`	$(0, \infty)$	$\phi\mu_i^3$

- For distributions in the exponential families, the variance is a function of the mean and a dispersion parameter ϕ (fixed to 1 for the binomial and Poisson distributions).

Generalized Linear Models in R

Implementation of GLMs in R

- The following table shows the links available for each family in R, with the default links as ■:

family	link: identity	inverse	sqrt	1/mu^2
gaussian	■	□		
binomial				
poisson	□		□	
Gamma	□	■		
inverse.gaussian	□	□		■
quasi	■	□	□	□
quasibinomial				
quasipoisson	□		□	

Generalized Linear Models in R

Implementation of GLMs in R

family	link: log	logit	probit	cloglog
gaussian	□			
binomial	□	■	□	□
poisson	■			
Gamma	□			
inverse.gaussian	□			
quasi	□	□	□	□
quasibinomial		■	□	□
quasipoisson	■			

- The `quasi`, `quasibinomial`, and `quasipoisson` family generators do not correspond to exponential families.

- The response for a binomial GLM may be specified in several forms:
  - For binary data, the response may be
    - a variable or an R expression that evaluates to `0`'s ('failure') and `1`'s ('success').
    - a logical variable or expression (with `TRUE` representing success, and `FALSE` failure).
    - a factor (in which case the first category is taken to represent failure and the others success).
  - For binomial data, the response may be
    - a two-column matrix, with the first column giving the count of successes and the second the count of failures for each binomial observation.
    - a vector giving the *proportion* of successes, while the binomial denominators (total counts or numbers of trials) are given by the `weights` argument to `glm`.

Generalized Linear Models in R
GLMs for Binary/Binomial and Count Data

- Poisson generalized linear models are commonly used when the response variable is a count (Poisson regression) and for modeling associations in contingency tables (loglinear models).
- The two applications are formally equivalent. Poisson GLMs are fit in R using the `poisson` family generator with `glm()`.
- Overdispersed binomial and Poisson models may be fit via the `quasibinomial` and `quasipoisson` families.
- The `glm.nb()` function in the **MASS** package fits negative-binomial GLMs to count data.

The Linear Mixed-Effects Model

- The *Laird-Ware form* of the linear mixed model:

$$
\begin{aligned}
y_{ij} &= \beta_1 + \beta_2 x_{2ij} + \cdots + \beta_p x_{pij} + b_{1i} z_{1ij} + \cdots + b_{qi} z_{qij} + \varepsilon_{ij} \\
b_{ki} &\sim N(0, \psi_k^2), \mathrm{Cov}(b_{ki}, b_{k'i}) = \psi_{kk'} \\
&\quad b_{ki}, b_{k'i'} \text{ are independent for } i \neq i' \\
\varepsilon_{ij} &\sim N(0, \sigma^2 \lambda_{ijj}), \mathrm{Cov}(\varepsilon_{ij}, \varepsilon_{ij'}) = \sigma^2 \lambda_{ijj'} \\
&\quad \varepsilon_{ij}, \varepsilon_{i'j'} \text{ are independent for } i \neq i'
\end{aligned}
$$

The Linear Mixed-Effects Model

- where:
  - y_{ij} is the value of the response variable for the jth of n_i observations in the ith of M groups or clusters.
  - $\beta_1, \beta_2, \ldots, \beta_p$ are the fixed-effect coefficients, which are identical for all groups.
  - $x_{2ij}, \ldots, x_{pij}$ are the fixed-effect regressors for observation j in group i; there is also implicitly a constant regressor, $x_{1ij} = 1$.
  - $b_{1i}, \ldots, b_{qi}$ are the random-effect coefficients for group i, assumed to be multivariately normally distributed, independent of the random effects of other groups. The random effects, therefore, vary by group.
    - The b_{ik} are thought of as random variables, not as parameters, and are similar in this respect to the errors ε_{ij}.
  - $z_{1ij}, \ldots, z_{qij}$ are the random-effect regressors.
    - The z's are almost always a subset of the x's (and may include *all* of the x's).
    - When there is a random intercept term, $z_{1ij} = 1$.

The Linear Mixed-Effects Model

- and:
  - ψ_k^2 are the variances and $\psi_{kk'}$ the covariances among the random effects, assumed to be constant across groups.
    - In some applications, the ψ's are parametrized in terms of a smaller number of fundamental parameters.
  - ε_{ij} is the error for observation j in group i.
    - The errors for group i are assumed to be multivariately normally distributed, and independent of errors in other groups.
  - $\sigma^2\lambda_{ijj'}$ are the covariances between errors in group i.
    - Generally, the $\lambda_{ijj'}$ are parametrized in terms of a few basic parameters, and their specific form depends upon context.
    - When observations are sampled independently within groups and are assumed to have constant error variance (as is typical in hierarchical models), $\lambda_{ijj} = 1$, $\lambda_{ijj'} = 0$ (for $j \neq j'$), and thus the only free parameter to estimate is the common error variance, σ^2.
    - If the observations in a "group" represent longitudinal data on a single individual, then the structure of the λ's may be specified to capture serial (i.e., over-time) dependencies among the errors.

Fitting Mixed Models in R
with the **nlme** and **lme4** packages

- In the **nlme** package (Pinheiro, Bates, DebRoy, and Sarkar):
  - `lme()`: linear mixed-effects models with nested random effects; can model serially correlated errors.
  - `nlme()`: nonlinear mixed-effects models.
- In the **lme4** package (Bates, Maechler, Bolker, and Walker):
  - `lmer()`: linear mixed-effects models with nested or crossed random effects; no facility (yet) for serially correlated errors.
  - `glmer()`: generalized-linear mixed-effects models.
- There are also Bayesian approaches to modeling hierarchical and longitudinal data that offer certain advantages; see in particular the **rstan** package that links R to the state-of-the-art STAN software for Bayesian modeling.

A Mixed Model for the Exercise Data

Longitudinal Model

- A level-1 model specifying a linear "growth curve" for log exercise for each subject:

$$\log\text{-exercise}_{ij} = \alpha_{0i} + \alpha_{1i}(\text{age}_{ij} - 8) + \varepsilon_{ij}$$

- Our interest in detecting differences in exercise histories between subjects and controls suggests the level-2 model

$$\begin{aligned}\alpha_{0i} &= \gamma_{00} + \gamma_{01}\text{group}_i + \omega_{0i}\\ \alpha_{1i} &= \gamma_{10} + \gamma_{11}\text{group}_i + \omega_{1i}\end{aligned}$$

where group is a dummy variable coded 1 for subjects and 0 for controls.

A Mixed Model for the Exercise Data

Laird-Ware form of the Model

- Substituting the level-2 model into the level-1 model produces

$$\begin{aligned}\log\text{-exercise}_{ij} &= (\gamma_{00} + \gamma_{01}\text{group}_i + \omega_{0i})\\ &\qquad + (\gamma_{10} + \gamma_{11}\text{group}_i + \omega_{1i})(\text{age}_{ij} - 8) + \varepsilon_{ij}\\ &= \gamma_{00} + \gamma_{01}\text{group}_i + \gamma_{10}(\text{age}_{ij} - 8)\\ &\qquad + \gamma_{11}\text{group}_i \times (\text{age}_{ij} - 8)\\ &\qquad + \omega_{0i} + \omega_{1i}(\text{age}_{ij} - 8) + \varepsilon_{ij}\end{aligned}$$

- in Laird-Ware form,

$$Y_{ij} = \beta_1 + \beta_2 x_{2ij} + \beta_3 x_{3ij} + \beta_4 x_{4ij} + \delta_{1i} + \delta_{2i} z_{2ij} + \varepsilon_{ij}$$

- Continuous first-order autoregressive process for the errors:

$$\text{Cor}(\varepsilon_{it}, \varepsilon_{i,t+s}) = \rho(s) = \phi^{|s|}$$

where the time-interval between observations, s, need not be an integer.

A Mixed Model for the Exercise Data

Specifying the Model in `lme`

- Using `lme()` in the **nlme** package:

```
lme(log.exercise ~ I(age - 8)*group,
            random = ~ I(age - 8) | subject,
            correlation = corCAR1(form = ~ age |subject)
            data=Blackmoor)
```