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`

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

- `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.
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 \( \mu_i \) 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**: \( g^{-1}(\eta_i) = \mu_i \).

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

<table>
<thead>
<tr>
<th>Link</th>
<th>( \eta_i = g(\mu_i) )</th>
<th>( \mu_i = g^{-1}(\eta_i) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>identity</td>
<td>( \mu_i )</td>
<td>( \eta_i )</td>
</tr>
<tr>
<td>log</td>
<td>( \log_e \mu_i )</td>
<td>( e^{\eta_i} )</td>
</tr>
<tr>
<td>inverse</td>
<td>( \mu_i^{-1} )</td>
<td>( \eta_i^{-1} )</td>
</tr>
<tr>
<td>inverse-square</td>
<td>( \mu_i^{-2} )</td>
<td>( \eta_i^{-1/2} )</td>
</tr>
<tr>
<td>square-root</td>
<td>( \sqrt{\mu_i} )</td>
<td>( \eta_i^2 )</td>
</tr>
<tr>
<td>logit</td>
<td>( \log_e \frac{1}{1-\mu_i} )</td>
<td>( \frac{1}{1 + e^{-\eta_i}} )</td>
</tr>
<tr>
<td>probit</td>
<td>( \Phi(\mu_i) )</td>
<td>( \Phi^{-1}(\eta_i) )</td>
</tr>
<tr>
<td>complementary log-log</td>
<td>( \log_e[-\log_e(1-\mu_i)] )</td>
<td>( 1 - \exp[-\exp(\eta_i)] )</td>
</tr>
</tbody>
</table>
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 $y_i$ | $V(y_i|\eta_i)$       |
|------------------|--------------|----------------|------------------------|
| gaussian         | identity     | $(-\infty, +\infty)$ | $\phi$                |
| binomial         | logit        | $0, 1, ..., n_i$ | $\mu_i(1 - \mu_i)$    |
| poisson          | log          | $0, 1, 2, ...$ | $\mu_i$                |
| 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 $\phi$ (fixed to 1 for the binomial and Poisson distributions).
The following table shows the links available for each family in R, with the default links as □:

<table>
<thead>
<tr>
<th>family</th>
<th>link</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>identity</td>
</tr>
<tr>
<td>gaussian</td>
<td></td>
</tr>
<tr>
<td>binomial</td>
<td></td>
</tr>
<tr>
<td>poisson</td>
<td></td>
</tr>
<tr>
<td>Gamma</td>
<td></td>
</tr>
<tr>
<td>inverse.gaussian</td>
<td></td>
</tr>
<tr>
<td>quasi</td>
<td>□</td>
</tr>
<tr>
<td>quasibinomial</td>
<td>□</td>
</tr>
<tr>
<td>quasipoisson</td>
<td>□</td>
</tr>
</tbody>
</table>

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.

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:

\[
\begin{align*}
y_{ij} &= \beta_1 + \beta_2 x_{2ij} + \cdots + \beta_p x_{ pij} + b_{1i} z_{1ij} + \cdots + b_{qi} z_{qij} + \varepsilon_{ij} \\
b_{ki} &\sim N(0, \psi^2_k), \text{ Cov}(b_{ki}, b_{k'i}) = \psi_{kk'}
\end{align*}
\]

- \( b_{ki}, b_{k'i} \) are independent for \( i \neq i' \)

\[
\begin{align*}
\varepsilon_{ij} &\sim N(0, \sigma^2 \lambda_{ijj}), \text{ Cov}(\varepsilon_{ij}, \varepsilon_{ij'}) = \sigma^2 \lambda_{ijj'}
\end{align*}
\]

- \( \varepsilon_{ij}, \varepsilon_{i'j'} \) are independent for \( i \neq i' \)

where:

- \( y_{ij} \) is the value of the response variable for the \( j \)th of \( n_i \) observations in the \( i \)th 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 \( \varepsilon_{ij} \).
- \( z_{1ij}, \ldots, z_{qij} \) 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:
  - $\psi^2_k$ are the variances and $\psi_{kk'}$ the covariances among the random effects, assumed to be constant across groups.
    - In some applications, the $\psi$’s are parametrized in terms of a smaller number of fundamental parameters.
  - $\varepsilon_{ij}$ 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, $\sigma^2$.
    - If the observations in a “group” represent longitudinal data on a single individual, then the structure of the $\lambda$’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):
  - lm() : 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:

  \[ \log\text{-exercise}_{ij} = \alpha_0 + \alpha_1 (\text{age}_{ij} - 8) + \epsilon_{ij} \]

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

  \[ \begin{align*}
  \alpha_0_i &= \gamma_0 + \gamma_{01}\text{group}_i + \omega_0_i \\
  \alpha_1_i &= \gamma_0 + \gamma_{11}\text{group}_i + \omega_1_i
  \end{align*} \]

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

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

  \[ \begin{align*}
  \log\text{-exercise}_{ij} &= (\gamma_0 + \gamma_{01}\text{group}_i + \omega_0_i) \\
  &\quad + (\gamma_0 + \gamma_{11}\text{group}_i + \omega_1_i)(\text{age}_{ij} - 8) + \epsilon_{ij} \\
  &= \gamma_0 + \gamma_{01}\text{group}_i + \gamma_0 (\text{age}_{ij} - 8) \\
  &\quad + \gamma_{11}\text{group}_i \times (\text{age}_{ij} - 8) \\
  &\quad + \omega_0_i + \omega_1_i(\text{age}_{ij} - 8) + \epsilon_{ij}
  \end{align*} \]

  in Laird-Ware form,

  \[ Y_{ij} = \beta_1 + \beta_2x_{2ij} + \beta_3x_{3ij} + \beta_4x_{4ij} + \delta_1i + \delta_2iz_{2ij} + \epsilon_{ij} \]

- Continuous first-order autoregressive process for the errors:

  \[ \text{Cor}(\epsilon_{it}, \epsilon_{i,t+s}) = \rho(s) = \phi^{|s|} \]

  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)
  ```