GLMM

Nick Syring

2023-02-02

Contents

Chapter 1

Introduction

Chapter 2

Introduction

Chapter 3

Poisson Regression

3.1 Children Ever Born Data

The "Children Ever Born" (CEB) dataset consists of grouped data on the number of births of Fijian women. The women are described according to their marriage duration in years in ordinal levels: (0-4, 5-9, 10-14, 15-19, 20-24, 25-29); their place of residence (Suva—the capital city—Urban, or Rural); and, their level of education (none, lower primary, upper primary, secondary or greater). The count, mean, and variance of the number of children ever born, and the group size, is given for each group of women by cross-classified factorial level. These summaries are sufficient to model counts of children ever born by a Poisson distribution (each individual woman's count is not needed).

The CEB data is an example of an observational dataset — the characteristics of the individuals are inherent rather than set by experimenters as in an experimental/controlled trial—but, that should be clear from the context. Several interesting questions may be answered using this data, such as: are more or fewer born children associated with higher or lower education among Fijian women; does an urban versus rural living location influence the number of children ever born; and, do the number of children ever born steadily increase with marrige duration, or tend to plateau?

```
ceb <- read.table('ceb.dat')
head(ceb)
```

```
##   dur   res  educ mean  var  n     y
## 1 0-4  Suva  none 0.50 1.14  8  4.00
## 2 0-4  Suva lower 1.14 0.73 21 23.94
## 3 0-4  Suva upper 0.90 0.67 42 37.80
## 4 0-4  Suva  sec+ 0.73 0.48 51 37.23
```

```
## 5 0-4 urban  none 1.17 1.06 12 14.04
## 6 0-4 urban lower 0.85 1.59 27 22.95
```

A statistician (or student statistician) familiar with multiple linear regression and/or ANOVA for factorial experiments may instinctively choose to fit a Gauss-Markov linear model to the CEB data, treating the responses as independent normal random variables. However, since the responses are counts, a Poisson model is more reasonable. But, just how does one perform *Poisson regression*? — as opposed to the familiar multiple linear regression described by the Gauss-Markov model:

$$Y = X\beta + \epsilon, \quad \epsilon \sim N_n(0_{n\times 1}, \sigma^2 I_n).$$

That is the motivation for this chapter, in which we will explore the family of *Generalized Linear Models* from defining and fitting the model, to performing inference and model diagnostics, all within the context of the CEB example.

3.2 Defining GLMs

For the CEB data we naturally want to model the CEB grouped counts as realizations of Poisson r.v.'s with means $n_j x_j^\top \beta$ where n_j is the number of women in the j^{th} factorial group, x_j is the vector of their common covariates, and β is the common regression coefficient vector. Then, the likelihood of the model is

$$L(\beta; \text{data}) = \prod_{j=1}^{70} \frac{(n_j x_j^\top \beta)^{y_j} e^{-n_j x_j^\top \beta}}{y_j!}$$

and the loglikelihood is given by

$$\ell(\beta; \text{data}) = \text{constant} + \sum_{j=1}^{70} y_j \log(n_j x_j^\top \beta) - n_j x_j^\top \beta.$$

The Poisson likelihood is a member of the Exponential Family, which contains all distributions with PDFs that may be expressed as

$$f(y; \theta, \phi) = \exp\{[y\theta - b(\theta)]/a(\phi) + c(y, \phi)\}.$$

Looking ahead, we will apply the exponential family model above to independent but not identically distributed responses, similar to the data we encounter in multiple linear regression and the Gauss-Markov model, so we will allow θ as well as the forms of the a, b, and c functions to vary over observations, but we will fix ϕ, so that the loglikelihood for a sample of size n may be written as follows:

$$\ell(\beta; \text{data}) = \sum_{i=1}^{n} \{[y_i\theta_i - b_i(\theta_i)]/a_i(\phi) + c_i(\phi, y_i)\}.$$

The Poisson regression model for grouped data is a fairly simple member of this family, having $\theta = \log(n_j x_j^\top \beta)$, $\phi = a(\phi) = 1$, and $b(\theta) = \exp(\theta) = n_j x_j^\top \beta$. In fact, it is very often the case that GLMs satisfy $a(\phi) \propto \phi$ up to a known constant. In general, GLMs satisfy

$$E(Y) = b'(\theta) \quad \text{and} \quad V(Y) = b''(\theta)a(\phi).$$

For the Poisson regression model, in particular, we have

$$\theta = \log(n_j x_j^\top \beta); \quad b(\theta) = \exp(\theta); \quad \text{and} \quad a(\phi) = 1$$

$$E(Y) = b'(\theta) = \frac{\partial}{\partial \theta} \exp(\theta) = \exp(\theta) = n_j x_j^\top \beta; \text{ and,}$$

$$V(Y) = b''(\theta)a(\phi) = \frac{\partial^2}{\partial \theta^2} \exp(\theta) = \exp(\theta) = n_j x_j^\top \beta.$$

3.3 Fitting GLMs

Like any other model defined by a likelihood, GLMs may be fit by maximizing the (log)likelihood. But, it is generally not the case that the maximizers (MLEs) are available in closed form. Instead, they are computed iteratively using Newton's method or a similar iterative procedure. Refer again to the exponential family loglikelihood using the usual representation $a_i(\phi) = \phi/w_i$ where w_i are known constants:

$$\ell(\beta; \text{data}) = \sum_{i=1}^{n} \{w_i[y_i\theta_i - b_i(\theta_i)]/\phi + c_i(\phi, y_i)\}.$$

Let $\mu_i = E(Y_i)$. Then, $b'(\theta_i) = \mu_i$, or, equivalently, $g_c(\mu_i) = \theta_i$ where g_c is termed the *canonical link*; for example, $g_c := \log$ for the Poisson distribution. Additionally, let g link the mean to the linear function of covariates, i.e., $g(\mu_i) = \eta_i = x_i^\top \beta$; e.g., g is the identity function for the Poisson model. Since $b_i'(\theta_i)$ is also equal to μ_i in the exponential family, we may differentiate the loglikelihood with respect to the regression parameter β using the chain rule:

$$\frac{\partial \ell}{\partial \beta_j} = \sum_{i=1}^{n} \left\{ \frac{w_i}{\phi} \left[y_i \frac{\partial \theta_i}{\partial \beta_j} - \frac{\partial b_i(\theta_i)}{\partial \beta_j} \right] + c_i(\phi, y_i) \right\}$$

using

$$\frac{\partial \theta_i}{\partial \beta_j} = \frac{\partial \theta_i}{\partial \mu_i} \frac{\partial \mu_i}{\partial \beta_j}.$$

Since $\mu_i = b_i'(\theta_i)$ we have $\partial\theta_i/\partial\mu_i = 1/b_i''(\theta_i)$. But, in light of $\mu_i = b'(\theta_i)$ we may always write $b_i''(\theta_i)$ as a function of μ_i, i.e., $V(\mu_i) = b_i''(\theta_i)/w$ so that $V(Y_i) =$

$V(\mu_i)\phi$. Moreover, since $\mu_i = g^{-1}(x_i^\top \beta)$ we have $\partial\mu_i/\partial\beta_j = x_{ij}/g'[g^{-1}(x_i^\top\beta)]$. Substituting, we can write the score function using only μ_i as follows:

$$\frac{\partial \ell}{\partial \beta_j} = \frac{1}{\phi}\sum_{i=1}^{n} \frac{y_i - \mu_i}{g'(\mu_i)V(\mu_i)} x_{ij}.$$

The second (mixed partial) derivative may be written

$$\frac{\partial^2 \ell}{\partial \beta_j \partial \beta_k} = -\frac{1}{\phi}\sum_{i=1}^{n} \frac{x_{ij}x_{ik}h(\mu_i)}{g'(\mu_i)^2 V(\mu_i)}$$

where $h(\mu_i) = 1 + (y_i - \mu_i)\{V'(\mu_i)/V(\mu_i) + g''(\mu_i)/g'(\mu_i)\}$. The expectation of the second derivative (which when multiplied by -1 appears in the Fisher information matrix) is the same quantity with $h(\mu_i)$ replaced by $E[h(\mu_i)]$, which simply equals 1 because $E(Y_i - \mu_i) = 0$. The Hessian of the loglikelihood is clearly a quadratic form $\phi^{-1}X^\top WX$ where X is the $n \times p$ design matrix of covariates and $W = [h(\mu_i)/\{g'(\mu_i)^2 V(\mu_i)\}]$ is an $n \times n$ diagonal matrix of "weights". Less obvious, we may define $G = \text{diag}\{g'(\mu_i)/h(\mu_i)\}$ so that the gradient of the loglikelihood equals $\phi^{-1}X^\top WG(y - \mu)$. With this clever rewriting, Newton's method updates take on the form of a weighted least squares solution:

$$\begin{aligned}
\beta^{[k+1]} &= \beta^{[k]} + (X^\top WX)^{-1}X^\top WG(y-\mu) \\
&= (X^\top WX)^{-1}X^\top W\{G(y-\mu)X + \beta^{[k]}\} \\
&= (X^\top WX)^{-1}X^\top Wz
\end{aligned}$$

where $z := G(y - \mu) + X\beta^{[k]}$ is sometimes referred to as the "pseudo-data". Repeating the weighted least squares update, iteratively, until convergence, is termed *iteratively re-weighted least squares* (IRLS) since, of course, the weights in W are updating with each iteration.

For our Poisson regression based on the grouped CEB data we have the following likelihood, gradient, and Hessian:

$$\begin{aligned}
\ell(\beta; \text{data}) &= \sum_{j=1}^{70} \left[y_j x_j^\top \beta - n_j e^{x_j^\top \beta} \right] \\
\nabla_s \ell &= \sum_{j=1}^{70} \left[y_j x_{js} - n_j x_{js} e^{x_j^\top \beta} \right] \\
\nabla^2_{s,t} \ell &= -\sum_{j=1}^{70} n_j x_{js} x_{jt} e^{x_j^\top \beta}.
\end{aligned}$$

Rewriting the Hessian and gradient as above for the general exponential family GLM we have

$$W_{k,k} = n_k \mu_k \quad \text{and} \quad G_k = (n_k \mu_k)^{-1}$$

so that the IRLS updates are given by

$$(X^\top W X)^{-1} X^\top W z$$

with $z_k = (n_k \mu_k)^{-1}(y_k - n_k \mu_k) + x_k^\top \beta$.

3.3.1 IRLS for the CEB data

Below we compute the MLEs for the Poisson regression of the grouped CEB data "by hand" using IRLS—and, also compare to the glm function in R. For our calculation we initialize the elements of the parameter vector μ by the sample means $\mu_j = y_j/n_j$. We set the pseudo data equal to $z_j = -(1/\mu_j)(y_j/n_j - \mu_j) + log(\mu_j)$ and iterate the computation of least squares estimates $\widehat{\beta}$.

Note that the CEB data contains grouped "counts" computed as $y_j = \mu_j n_j$ where the μ_j values are rounded. And, as a result, the y_j counts are not integers. This does not affect our "by hand" calculation of $\widehat{\beta}$ whatsoever because we never use the full Poisson PMF in our computations; the glm function in R, on the other hand, will throw many warnings if the y_j values are not rounded, apparently because it uses the PMF via dpois "under the hood". The only differences between our fitted $\widehat{\beta}$ and glm's are due to rounding y_j's.

```
n <- nrow(ceb)
group.sizes <- ceb$n
Y <- ceb$y
# IRLS - factor coding
# initialize with mu = Y/group.sizes
options(contrasts = c('contr.treatment', 'contr.treatment'))
X <- model.matrix(y~dur+res+educ, data = ceb)
mu <- Y/group.sizes
XB <- log(mu)
W <- diag(as.numeric(mu))
z <- -(1/mu)*(Y/group.sizes-mu) + XB
beta <- solve(t(X)%*%W%*%X)%*%t(X)%*%W%*%z
tol <- 0.0001
difference <- 1
maxiter <- 100
iter <- 1
while((difference > tol) & (iter < maxiter)){
  XB <- X%*%beta
  mu <- exp(XB)
  W <- diag(as.numeric(group.sizes*mu))
  z <- (Y/diag(W) - rep(1,n)) + XB
  beta.old <- beta
  beta <- solve(t(X)%*%W%*%X)%*%t(X)%*%W%*%z
```

```
  difference <- max(abs(beta - beta.old))
  iter<-iter+1
}
beta
```

```
##                    [,1]
## (Intercept)  0.05695417
## dur10-14     1.37053208
## dur15-19     1.61423104
## dur20-24     1.78548879
## dur25-29     1.97679469
## dur5-9       0.99765038
## resSuva     -0.15121728
## resurban    -0.03895822
## educnone    -0.02308034
## educsec+    -0.33266474
## educupper   -0.12474575
```

```
## the glm function can be used with offset equal to logarithm of the group sizes
my.glm <- glm(round(y)~dur+res+educ, family = poisson(link = 'log'), data = ceb, offse
summary(my.glm)
```

```
##
## Call:
## glm(formula = round(y) ~ dur + res + educ, family = poisson(link = "log"),
##     data = ceb, offset = log(n))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2960  -0.6641   0.0725   0.6336   3.6782
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.05754    0.04803   1.198    0.231
## dur10-14     1.36940    0.05107  26.815  < 2e-16 ***
## dur15-19     1.61376    0.05119  31.522  < 2e-16 ***
## dur20-24     1.78491    0.05121  34.852  < 2e-16 ***
## dur25-29     1.97641    0.05003  39.501  < 2e-16 ***
## dur5-9       0.99693    0.05274  18.902  < 2e-16 ***
## resSuva     -0.15166    0.02833  -5.353 8.63e-08 ***
## resurban    -0.03924    0.02463  -1.594    0.111
## educnone    -0.02297    0.02266  -1.014    0.311
## educsec+    -0.33312    0.05390  -6.180 6.41e-10 ***
## educupper   -0.12425    0.03000  -4.142 3.44e-05 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 3731.852  on 69  degrees of freedom
## Residual deviance:   70.665  on 59  degrees of freedom
## AIC: 522.14
##
## Number of Fisher Scoring iterations: 4
```

3.4 Inference on GLMs

No doubt you noticed the glm function output produces standard errors, "z" values, and p-values for each fitted coefficient, just as you would find accompanying summarized lm output. But, what is the justification for these p-values?

Since $\hat{\beta}$ is an MLE, standard likelihood theory holds that $\hat{\beta} \dot{\sim} N_p(\beta, I^{-1}(\beta))$ for "large" n where $I^{-1}(\beta)$ is the Fisher information. As discussed above, the "observed information" (which is the Hessian) is equal to $-\phi^{-1}(X^\top W X)^{-1}$ where W is the weight matrix in the final iteration of IRLS, and $-\phi^{-1}(X^\top W X)^{-1}$ coincides with the Fisher Information when we replace $h(\mu_i)$ by $E(h(\mu_i)) = 1$ in the IRLS (then called Fisher scoring) updates. Therefore, $\hat{\beta} \dot{\sim} N_p(\beta, \phi^{-1}(X^\top W X)^{-1})$ for "large" n, but this is only useable if ϕ is known (which it is, and equals 1, for the Poisson and Binomial models). Otherwise, we replace ϕ by its MLE and use the corresponding Student's t distribution with $n-p$ degrees of freedom rather than the standard normal for inference on β_j.

The upshot is that we may base tests for, e.g., $H_0 : \beta_j = 0$, on Student's t with $n-p$ df; i.e.,

$$\text{Reject } H_0 : \beta_j = 0 \text{ if } \left| \frac{\hat{\beta}_j}{\sqrt{\hat{\phi}^{-1}(X^\top W X)^{-1}_{j,j}}} \right| > t_{1-\alpha/2, n-p}.$$

Multivariate Wald simultaneous $100(1-\alpha)\%$ confidence regions are given by the eliiptical contours:

$$\phi \text{ known:} \quad \{\beta : (\hat{\beta} - \beta)^\top \phi^{-1} (X^\top W X)^{-1} (\hat{\beta} - \beta) < \chi^2_{1-\alpha, p}\}$$

$$\phi \text{ unknown:} \quad \{\beta : (\hat{\beta} - \beta)^\top \hat{\phi}^{-1} (X^\top W X)^{-1} (\hat{\beta} - \beta) < F_{1-\alpha, p, n-p}\}$$

Moreover, an approximate 95% CI for the mean response $\mu = g^{-1}(x^\top \beta)$ for covariate vector x is given by the Delta method interval:

$$g^{-1}(x^\top\hat{\beta}) \pm t_{1-\alpha/2,n-p}\sqrt{\hat{\phi}^{-1}\left[\nabla_\beta g^{-1}(x^\top\beta)\right]^\top (X^\top W X)^{-1}\left[\nabla_\beta g^{-1}(x^\top\beta)\right]}.$$

In multiple linear regression (Gauss-Markov) models we use partial F tests (which are likelihood ratio tests) to test for significance of sets of covariates, i.e., $H_0 : \beta_j = \beta_{j+1} = \cdots = \beta_{j+\ell} = 0$. For GLMs, similar tests are available. For models where ϕ is known, we have

$$-2\{\ell(\hat{\beta}_{h_0}) - \ell(\hat{\beta})\} \overset{H_0}{\sim} F_{\ell,n-p}$$

where $\hat{\beta}_{h_0}$ is the MLE under the null hypothesis where ℓ coefficients are set equal to 0.

There are several methods to estimate ϕ when unknown. Pearson's method observes that

$$\phi^{-1}X^2 \dot{\sim} \chi^2_{n-p} \quad \text{where} \quad X^2 := \sum_{i=1}^n \frac{(Y_i - \hat{\mu}_i)^2}{\phi V(\hat{\mu}_i)}$$

if the model fits the data adequately. Hence, $\hat{\phi}_P = X^2/(n-p)$ ought to be a good estimate of ϕ. For certain data sets, such as Poisson data with low counts, the Pearson estimate may behave badly, and a modified version (Fletcher's estimator) is preferred:

$$\hat{\phi}_F = \frac{\hat{\phi}_P}{1-\overline{s}}, \quad \text{where}$$
$$\overline{s} := n^{-1}\sum_{i=1}^n V'(\hat{\mu}_i)\frac{(y_i - \hat{\mu}_i)}{V(\hat{\mu}_i)}.$$

The *Deviance difference* for models A and B where $A \subset B$ is given by $D_A - D_B = -2\{\ell(\hat{\beta}_A) - \ell(\hat{\beta}_B)\}\phi$. The scaled deviance difference is

$$D^*_A - D^*_B = -2\{\ell(\hat{\beta}_A) - \ell(\hat{\beta}_B)\} \dot{\sim} \chi^2_\ell$$

where the difference in number of fitted parameters is ℓ. Despite the notation, the scaled deviance difference does depend on ϕ, whereas the deviance difference does not. Two alternative tests make use of the scaled deviance to compare nested GLMs. The first is analogous to the partial F test:

$$F = \frac{(D^*_A - D^*_B)/\ell}{D^*_B/(n-p)} \dot{\sim} F_{\ell,n-p}$$

but the approximation to the F distribution is very rough. Alternatively, we can replace the scale parameter in the scaled deviance by its Pearson (or Fletcher) estimator and obtain

$$\hat{D}^*_A - \hat{D}^*_B \dot{\sim} \chi^2_\ell,$$

where the "hat" on the scaled deviances indicates their dependence on $\hat{\phi}$.

3.4.1 Inference and prediction for the CEB data using Poisson regression

Next, we'll demonstrate computations of confidence intervals and tests for significance of sets of covariates within the Poisson regression model for the CEB data. We can either do these computations "by hand" using the results of our IRLS or we can use built-in R functions like glm and confint.

For Wald-type confidence intervals for regression coefficients we require the inverse Hessian (the estimate of the inverse Fisher information, equal when Fisher scoring is used). We compute this from the final iteration of IRLS (assuming the algorithm converged). For most GLMs we will need to estimate the scale parameter ϕ and we include the Pearson and Fletcher estimates below—but in the Poisson model $\phi = 1$. Note both estimates are close to 1. The p-values included in the summarized glm output imply the intercept is not significantly different from zero, but that several other coefficients are different from zero, including, e.g., β_1, β_2, and β_3. Our Wald-type CIs and p-values agree with the glm p-values; for example, our 95% CI for β_0 computed "by hand" is (-0.0372 0.1511), suggesting the intercept is not significantly different from zero, and our 95% CI for β_1 is (1.2704, 1.4706) with a p-value for the test of $\beta_1 = 0$ that is indistinguishable from 0.

We computed the deviance difference (which is minus twice the difference in log-likelihood) between the intercept-only model and the full model. The deviance difference is about 3661 on ten degrees of freedom (the difference in number of fitted coefficients between the models). If the intercept only model fits, then this deviance difference should be comparable to a Chi-squared r.v. with 10 degrees of freedom, but the corresponding p-value is basically zero, supporting the claim that the full model fits much better than the intercept-only model. Compare our deviance difference calculation to the output of glm: glm includes the null deviance and residual deviance, the difference of these two gives the deviance difference statistic used to compare the intercept-only and full models. It is about 3661, agreeing almost exactly with our "by hand" calculation.

```
Hessian <- t(X)%*%W%*%X
inv.Hessian <- solve(Hessian)
p <- length(beta)

# Just for illustration, phi = 1 for Poisson model
Pearson.X2 <- sum(((Y - group.sizes * mu)^2) / (group.sizes * mu))
Pearson.phi <- Pearson.X2 / (n-p)
s.bar <- mean((Y - group.sizes * mu) / (group.sizes * mu))
Fletcher.phi <- Pearson.phi/(1-s.bar)
Pearson.phi
```

```
## [1] 1.211949
```

```
Fletcher.phi
```

```
## [1] 1.194283
```

```
# CIs for the first 4 regression coefficients
#   If phi were unknown, it's estimate would appear in the estimated standard error of
#   estimated coefficient
# beta[1] + qt(c(0.025,0.975),n-p)*sqrt((1/Pearson.phi)*inv.Hessian[1,1])
beta[1] + qnorm(c(0.025,0.975))*sqrt(inv.Hessian[1,1])
```

```
## [1] -0.03721264  0.15112097
```

```
beta[2] + qnorm(c(0.025,0.975))*sqrt(inv.Hessian[2,2])
```

```
## [1] 1.270425 1.470639
```

```
2*(1-pnorm(abs(beta[2]/sqrt(inv.Hessian[2,2]))))
```

```
## [1] 0
```

```
beta[3] + qnorm(c(0.025,0.975))*sqrt(inv.Hessian[3,3])
```

```
## [1] 1.513870 1.714592
```

```
beta[4] + qnorm(c(0.025,0.975))*sqrt(inv.Hessian[4,4])
```

```
## [1] 1.685091 1.885886
```

```
# The R function confint can also be used with GLMs to provide confidence intervals fo

confint(my.glm, 'dur10-14')
```

```
## Waiting for profiling to be done...
```

```
##    2.5 %   97.5 %
## 1.270141 1.470370
```

```
# "Model F Test" - testing that all (non-intercept) coefficients equal zero
# for Poisson model, since phi is known, we have a LRT equivalent to a partial F test
# based on deviance difference
Ybar <- sum(Y)/sum(group.sizes)
D <- -2*(sum(Y*log(group.sizes*Ybar))-sum(group.sizes*Ybar) - sum(Y*log(group.sizes*mu))+sum(grou
Ybar
```

```
## [1] 3.960497
```

```
D
```

```
## [1] 3660.872
```

```
1-pchisq(D,p-1)
```

```
## [1] 0
```

```
# with rounded Ys and using glm output
Yr <- round(Y)
mu.glm <- exp(X%*%matrix(my.glm$coefficients,p,1))
Ybar <- sum(Yr)/sum(group.sizes)
D <- -2*(sum(Yr*log(group.sizes*Ybar))-sum(group.sizes*Ybar) - sum(Yr*log(group.sizes*mu.glm))+su
Ybar
```

```
## [1] 3.960403
```

```
D
```

```
## [1] 3661.186
```

```
1-pchisq(D,p-1)
```

```
## [1] 0
```

3.5 Model Checking/Diagnostics

It is essential statistical practice to check whether the model adequately fits the data. If the model fits poorly, then inferences/predictions garnered from the model are suspect. In multiple linear regression we assess model fit by analyzing residuals. When the multiple linear regression model fits, the residuals should

be approximately standard normal. Lack of fit manifests in residuals that are skewed or heavy-tailed, contain outliers, or tend to increase in variability with one or more covariates and/or the predicted responses.

Model-checking for GLMs can be done in essentially the same manner—the key is to find a quantity that reasonably fills the role of residuals in multiple linear regression. For GLMs, there are two choices, the Pearson residuals and the deviance residuals.

Pearson residuals are defined $e_i^P = \frac{Y_i - \hat{\mu}_i}{\sqrt{V(\mu_i)}}$, or, sometimes, $e_i^P = \frac{Y_i - \hat{\mu}_i}{\sqrt{\hat{\phi} V(\mu_i)}}$. The first definition results in quantities that should be approximately zero-mean normal random variates with variance ϕ whereas the second definition provides standard normal quantities. Some practitioners prefer the deviance residuals to the Pearson residuals as the latter are often observed to be asymmetric and, hence, not as "normal" as expected. The deviance is equal to

$$\text{Deviance} = -2\phi\{\ell(\hat{\beta}) - \sup \ell\}$$

where $\ell(\hat{\beta})$ is the loglikelihood evaluated at the MLEs and $\sup \ell$ is the loglikelihood with $\mu_i = y_i$, i.e., a totally saturated model. Multiplying by ϕ removes the dependence of the loglikelihood on the scale parameter. The deviance can be written as a sum of terms, say, Deviance $= \sum_{i=1}^{n} d_i$, and each observation's contribution d_i to the deviance is used to define the deviance residuals as follows:

$$e_i^D = \text{sign}(y_i - \mu_i)\sqrt{d_i}.$$

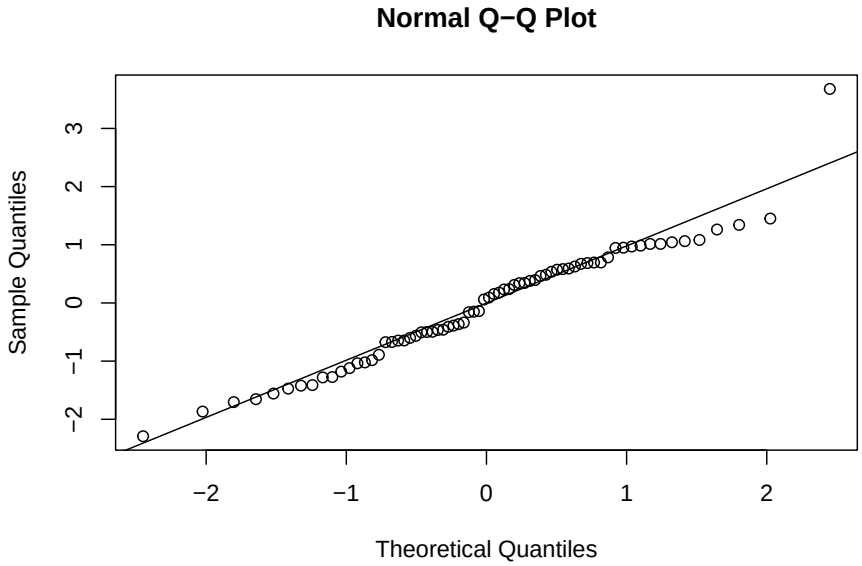
3.5.1 Residual analysis for CEB data

Using either the deviance residuals or the Pearson residuals shows a few important things. First, the residuals are approximately standard normal, with the exception of one "outlier", observation 17. Second, if we sort the observations by fitted mean response $\hat{\mu}_i$ from least to greatest, we see absolutely no trend up or down, or any pattern at all, in the residuals. This implies we have correctly modeled the mean-variance relationship, and that we have also correctly modeled the mean as a linear function of the covariates.

We obtain similar plots by simply running "plot(glm object)", but there are some default differences. Plotting a glm object will provide plots of "residuals" versus "fitted values", but in fact, the labels on these plots are slightly misleading. The residuals vs. fitted values plots uses either Pearson or deviance residuals (cannot tell for sure from the plot or the documentation) versus the logarithm of the fitted responses $\log(\hat{y}_j) = \log(n_j \hat{\mu}_j)$.

```
dev <- -2*((Y*log(group.sizes*mu))-(group.sizes*mu) - ((Y*log(Y))-(Y)))
deviance.resids <- ifelse((Y-group.sizes*mu) < 0,-1,1)*sqrt(dev)
pearson.resids <- (Y - group.sizes*mu)/sqrt(group.sizes*mu)
```

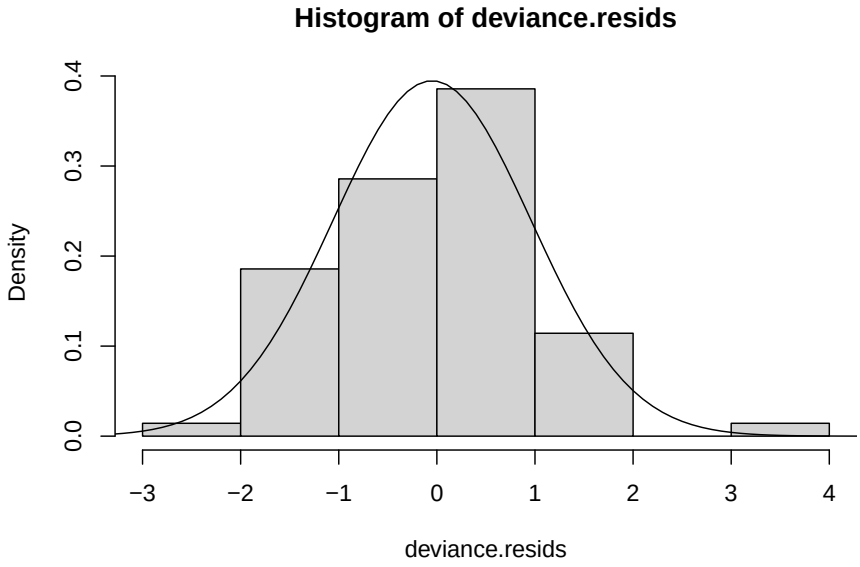
```
# residual plots using deviance residuals
qqnorm(deviance.resids)
qqline(deviance.resids)
```



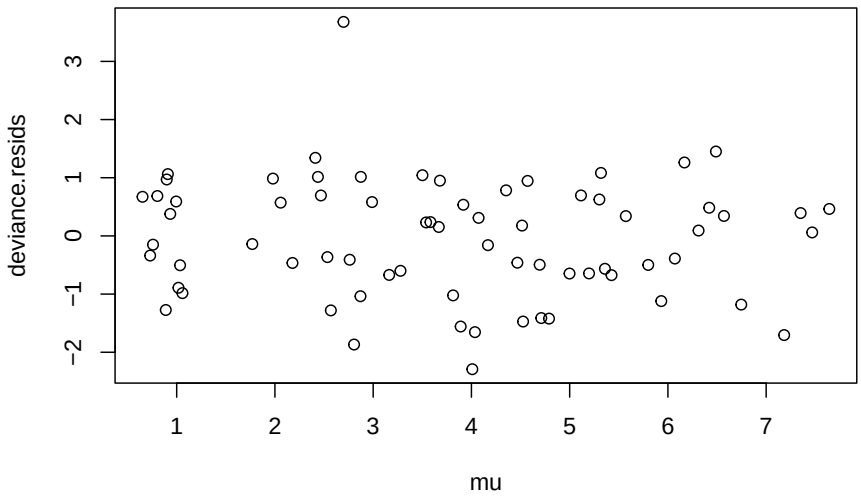

```
shapiro.test(deviance.resids)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  deviance.resids
## W = 0.96065, p-value = 0.02719
```

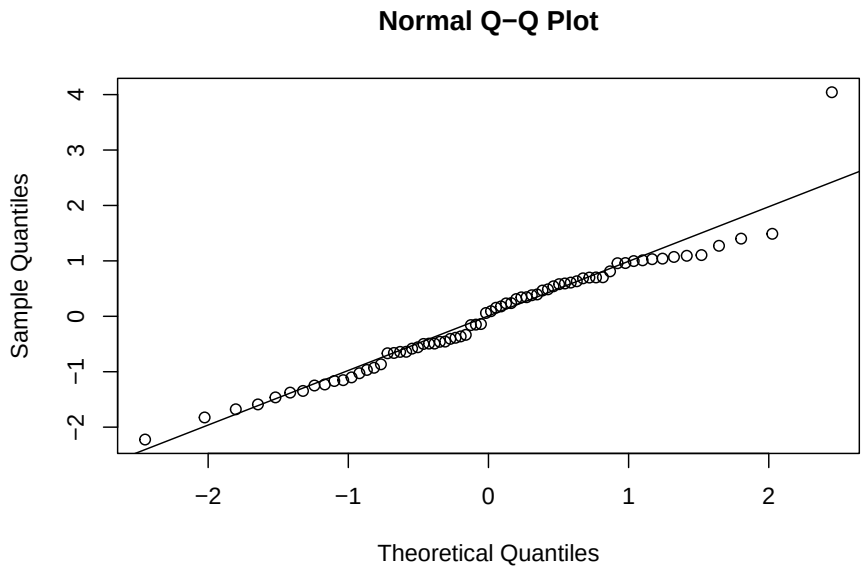
```
hist(deviance.resids, freq = FALSE)
dev.norm <- function(x) dnorm(x,mean(deviance.resids), sd(deviance.resids))
curve(dev.norm, -5,5, add = TRUE)
```




```
plot(mu, deviance.resids)
```



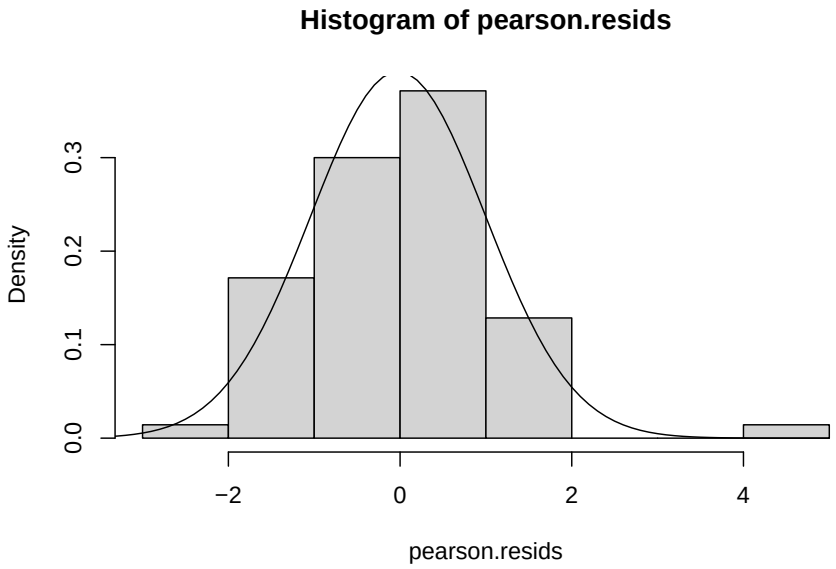
```
# residual plots using pearson residuals
qqnorm(pearson.resids)
qqline(pearson.resids)
```



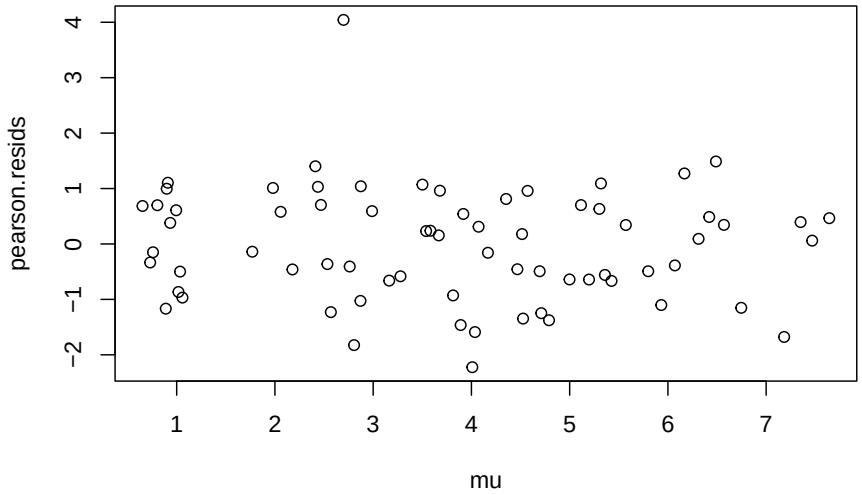

```
shapiro.test(pearson.resids)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  pearson.resids
## W = 0.94988, p-value = 0.007147
```

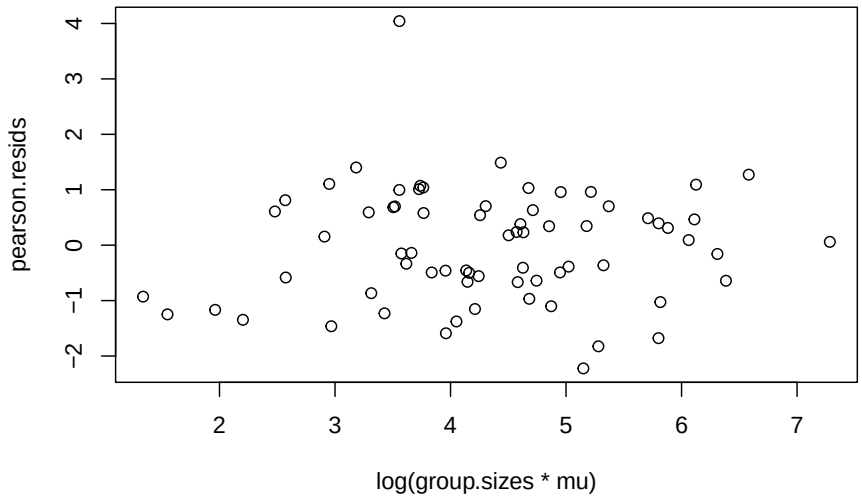
```
hist(pearson.resids, freq = FALSE)
pearson.norm <- function(x) dnorm(x,mean(pearson.resids), sd(pearson.resids))
curve(pearson.norm, -5,5, add = TRUE)
```



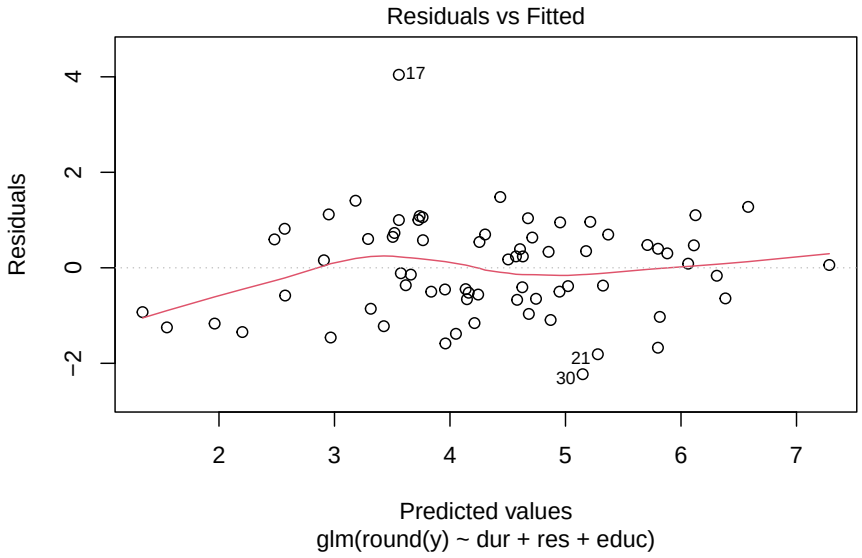

```
plot(mu, pearson.resids)
```

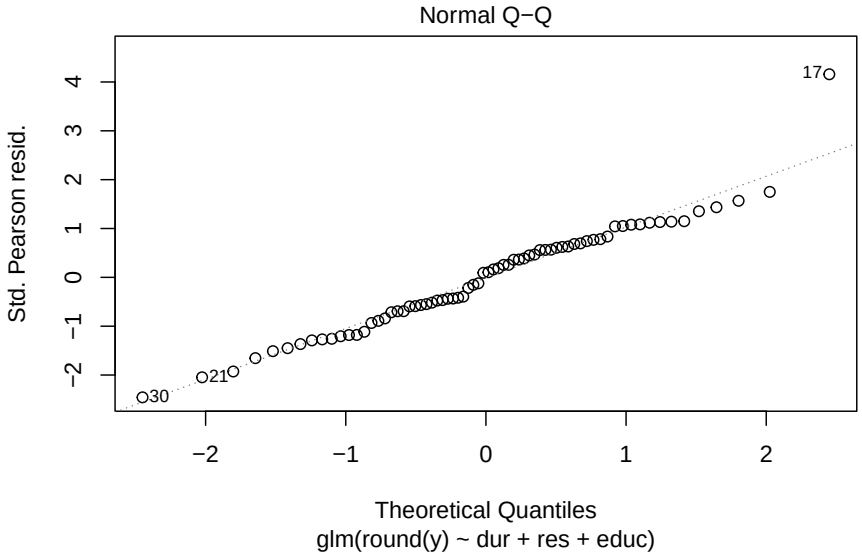


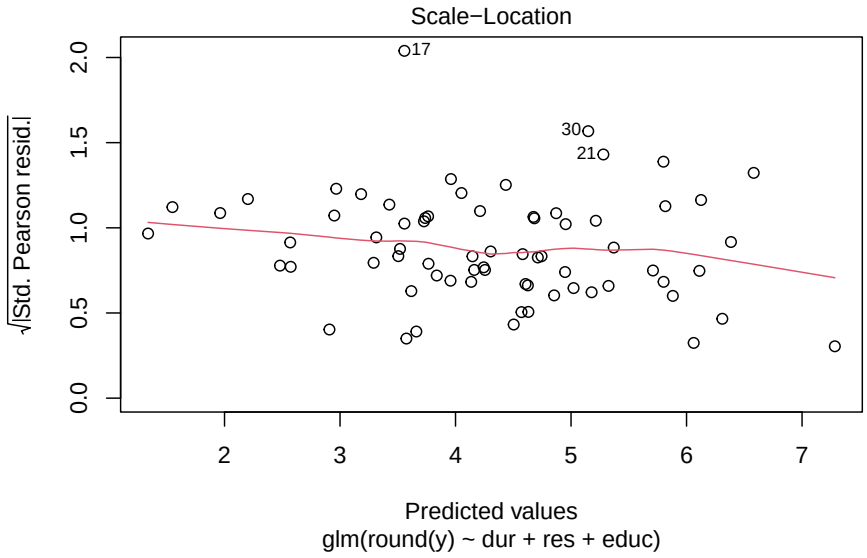
```
# plots are using what look like Pearson (or maybe deviance) residuals versus predicted values u
plot(log(group.sizes*mu), pearson.resids)
```

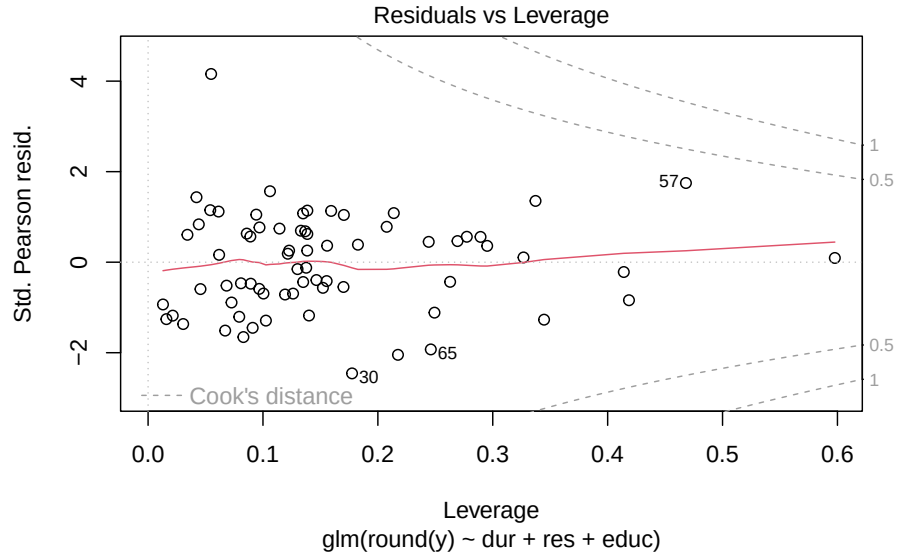



```
plot(my.glm)
```


Residuals vs Fitted
Residuals
17
21
30
Predicted values
glm(round(y) ~ dur + res + educ)


Normal Q−Q
Std. Pearson resid.
17
21
30
Theoretical Quantiles
glm(round(y) ~ dur + res + educ)


Scale−Location
√|Std. Pearson resid.|
17
30
21
0.0
0.5
1.0
1.5
2.0
2
3
4
5
6
7
Predicted values
glm(round(y) ~ dur + res + educ)


Residuals vs Leverage
Std. Pearson resid.
57
65
30
1
0.5
0.5
1
Cook's distance
−2
0
2
4
0.0
0.1
0.2
0.3
0.4
0.5
0.6
Leverage
glm(round(y) ~ dur + res + educ)

3.6 Outlier analysis using Cook's distance

Outliers are observations corresponding to large residuals. They may occur due to chance, or, more likely, indicate a lack of model fit for a specific observation. The lack of fit may be due to something innocuous like a mistake made in recording data; or, it may be that the observation in question is very different from the others in the sample, and does not follow the same response-covariate relationship.

Outliers are not a problem, however, unless their inclusion causes the fitted model to be substantially different than had they been excluded. Therefore, we do not necessarily care that a particular residual is large, but that it is influential on the model fit.

In multiple linear regression we may measure this influence by Cook's distance of the data point, which is related to both the magnitude of its residual and its leverage, as measured by the hat (or influence) matrix. For GLMs we can define the Cook's distance in the same manner: the Cook's distance of data point k is given by

$$C_k = \frac{1}{(p+1)} \sum_{i=1}^{n} \frac{(\hat{\mu}_i^{[k]} - \hat{\mu}_i)^2}{\hat{\phi} V(\hat{\mu}_i)}; \text{ or}$$

$$C_k = \frac{(e_k^P)^2}{\hat{\phi}(p+1)} \frac{h_k}{(1-h_k)^2}$$

where $H = W^{1/2} X (X^\top W X)^{-1} X^\top W^{1/2}$ is the hat matrix and h_k is its k^{th} diagonal entry. Large Cook's distance implies the model predictions change substantially when the data point in question is removed.

3.6.1 Outlier analysis for the CEB data

Typically data points with Cook's distance > 1 are considered highly influential and the case for their exclusion is considerable. In this case, our apparent outlier is not influential, nor is it even the most influential observation sampled.

Below we have computed Cook's distance "by hand" and by using the corresponding built-in R function; the only (very slight) difference between the two seems to be caused by the fact the built-in function uses the glm object, which we fitted after rounding the responses, whereas our "by hand" calculation uses Pearson residuals based on the original (unrounded) responses.

```
ceb2 <- ceb[-17,]
my.glm2 <- glm(round(y)~dur+res+educ, family = poisson(link = 'log'), data = ceb2, offs

W2 <- sqrt(W)
h <- W2%*%X%*%solve(t(X)%*%W%*%X)%*%t(X)%*%W2
```

```
((pearson.resids[17]^2)/p)*(h[17,17]/((1-h[17,17])^2))
```

```
## [1] 0.0912897
```

```
cd <- cooks.distance(my.glm)

cd[17]
```

```
##         17
## 0.09123744
```

```
sort(cd)
```

```
##           44           12           16           43           58           26
## 0.0001578983 0.0002173653 0.0003181272 0.0004391595 0.0004872775 0.0008276467
##           31           48           69            5           40           37
## 0.0009604834 0.0010416260 0.0011600877 0.0011813567 0.0015364693 0.0017374273
##           63           15           59           64            8           18
## 0.0017913034 0.0020157684 0.0022231406 0.0023307859 0.0024333875 0.0027257717
##            1           29           54           56           34           62
## 0.0027795217 0.0028483882 0.0029171612 0.0029318194 0.0030296972 0.0030322783
##           55           36           27           33            9           60
## 0.0033869538 0.0034063422 0.0048780994 0.0049440311 0.0052402581 0.0053485413
##           47            6           24            3           11           23
## 0.0055965679 0.0056512722 0.0056712755 0.0057553959 0.0059235574 0.0061112712
##           50           41           14           49            4            2
## 0.0062842199 0.0063288779 0.0064608723 0.0067079644 0.0067366794 0.0068678049
##           66           25           13            7           61           71
## 0.0072889100 0.0074308461 0.0082351514 0.0104259702 0.0110105154 0.0114006236
##           70           42           52           20           28           32
## 0.0115384460 0.0145603375 0.0149361729 0.0164709118 0.0173145140 0.0189913643
##           51           38           53           19           39           67
## 0.0191422888 0.0203317679 0.0205784895 0.0220776790 0.0225077753 0.0265192825
##           35           10           45           22           46           17
## 0.0290870153 0.0375227510 0.0462157923 0.0771290592 0.0847041708 0.0912374357
##           21           65           30           57
## 0.1058278940 0.1102296604 0.1184297114 0.2448544534
```

Chapter 4

Linear Mixed Models

4.1 ANOVA with random factors

The simplest linear mixed models are used to analyze linear models for one-way ANOVA, randomized complete block designs, and two-way ANOVA. *Mixed* models as opposed to *fixed* models (the linear models you have heretofore studied) are needed when factors have levels that are **random**. Random levels occur whenever the units making up those levels behave like random samples from a population. Two examples are given below. And, we discuss how to perform ANOVA-like tests for factors that are random rather than fixed. The upshot is that (at least for balanced experiments/datasets) the tests for fixed effects are identical to those for random effects, only the interpretation is different (and importantly so!).

4.1.1 Strength of metallic bonds

The dataset below, called "Bonds", contains responses for 21 samples of metals, 7 each for iron, nickel, and copper, that quantify the strength of metallic bonds. One sample from each metal was extracted from each of 7 ingots. We expect ingots to act like blocks—differences in ingots account for a substantial amount of variability in the responses, but the precise block effects are of no inferential/scientific inference. We only include the blocks in order to reduce the residual variance after accounting for block variance. A randomized controlled block design describes how this data was collected, but, if we repeated the experiment, the blocks (ingots) would be completely different. That is, the blocks are not *fixed* but *random*. Rather than estimating block effects that would surely change experiment to experiment, we should focus on estimating the amount of variability explained by the blocks, which should remain about

the same experiment to experiment. This suggests a different model than used to analyze RCBD experiments with fixed blocks.

The usual linear model for fixed blocks is

$$y_{ijk} = \mu + \alpha_i + \beta_j + \epsilon_{ij},$$

where y_{ij} is the response for treatment (metal) i in block (ingot) j; the α_i's are the metal (treatment) effects; the β_j's are the ingot (block) effects; and, $\epsilon_{ij} \overset{iid}{\sim} N(0, \sigma^2)$ are the random residuals.

The above linear model is the wrong model for this data because the block effects (and, hence, also the interaction effects) are meaningless outside of the given data set; these are not population-level parameters because the blocks are random rather than fixed. The appropriate model (given normality and independence of random residuals is reasonable) is the following *mixed effects model*:

$$y_{ij} = \mu + \alpha_i + \beta_j + \epsilon_{ij}, \tag{4.1}$$

where $\beta_j \overset{iid}{\sim} N(0, \sigma_b^2)$ and, independently, $\epsilon_{ij} \overset{iid}{\sim} N(0, \sigma^2)$.

For *balanced* experiments (the number of replicates is equal across each combination of factor levels) we can test for block and treatment effects by comparing nested/aggregated models. Let $\overline{Y}_{i\cdot}$ denote the mean response for metal i averaged over ingots. We can write down the following aggregated model from (4.1) as

$$\overline{y}_{i\cdot} = \mu + \alpha_i + \epsilon_i, \tag{4.2}$$

where $\epsilon_i = \frac{1}{J}\sum_{j=1}^{j} \epsilon_{ij}$. Then, ϵ_j has variance $\sigma_b^2 + \sigma^2/J$. The F statistic

$$F = \frac{J \cdot MSE_{agg}}{MSE_{full}}$$

where MSE_{agg} and MSE_{full} are the mean squared errors from the models in (4.2) and (4.1) can be used to test the hypothesis $H_0 : \sigma_b^2 = 0$.

4.1.2 Machine Productivity

The dataset given below contains the results of a designed experiment to evaluate worker productivity using 3 different industrial machines. The goal is to determine which machine is most productive while controlling for natural variation in worker productivity. The observed workers represent a random sample from a population of workers (blocks), analogous to the ingots in the previous example. The difference between the two examples (besides the context) is that the machine treatments are replicated wihtin workers, so that there are three

observations of a productivity score fore each worker for each type of machine. This means that we can fit a model with *interaction terms* capable of capturing changes in machine effects on productivity between different workers (if those changes are present):

$$y_{ijk} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \epsilon_{ijk}, \tag{4.3}$$

where k denotes the k^{th} replicate within machine i and worker j; and where $(\alpha\beta)_{ij}$ denote the machine-worker interaction effects.

Let $\overline{Y}_{ij\cdot}$ be the mean response averaging over replicates for the treatment i and block j combination. Then,

$$\begin{aligned} V(\overline{Y}_{ij\cdot}) &= V\left(K^{-1}\sum_{k=1}^{K} Y_{ijk}\right) \\ &= \frac{1}{K^2} V\left(\sum_{k=1}^{K}\{\mu + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \epsilon_{ijk}\}\right) \\ &= \frac{1}{K^2} V\left(K\mu + K\alpha_i + K\beta_j + K(\alpha\beta)_{ij} + \sum_{k=1}^{K}\epsilon_{ijk}\right) \\ &= \sigma_b^2 + \sigma_{ab}^2 + K^{-1}\sigma^2. \end{aligned}$$

If we rewrite the model for the cell mean responses as

$$\overline{y}_{ij\cdot} = \mu + \alpha_i + \beta_j + \epsilon_{ij}, \tag{4.4}$$

then the aggregated error term follows $\epsilon_{ij} \overset{iid}{\sim} N(0, \sigma_{ab}^2 + \sigma^2/K)$. The residual mean square (or called the mean squared error) of model (4.3) (let's call it MSE_{full}) has mean σ^2 with $n - p_1$ degrees of freedom where n is the sample size and p is the number of coefficients in the fitted model (p_1 equals the number of crossed factor levels, the number of blocks times the number of treatments). The residual mean square for the aggregated model in (4.4) (let's call it MSE_{agg}) has mean $\sigma_{ab}^2 + \sigma^2/K$ with $n/K - p_2$ degrees of freedom where p_2 is the number of treatments plus the number of blocks minus 1. An unbiased estimate of σ_{ab}^2 is given by $MSE_{\text{agg}} - \frac{1}{K} MSE_{\text{full}}$. Consider testing the null hypothesis $H_0 : \sigma_{ab}^2 = 0$. The statistic

$$F := \frac{K \cdot MSE_{agg}}{MSE_{full}} \overset{H_0}{\sim} F_{n/K-p_2, n-p_1},$$

that is, under the null hypothesis. The test that rejects H_0 for $F > F_{1-\alpha, n/K-p_2, n-p_1}$ is exactly equivalent to the partial F test between the full model and the full model without the interaction terms (the additive model).

Below we use R to compute ANOVA tables for the full model, full model without interaction, and the aggregated model. The F test statistic for the aggregated model is about 46.13 on 10 and 36 degrees of freedom, which exactly matches the partial F test between the full and additive models.

```r
library(nlme)

# aggregated model
Mach.agg <- aggregate(score~Machine*Worker, data = Machines, FUN=mean)

m2 <- lm(score~Machine+Worker, data = Mach.agg)

anova(m2)
```

```
## Analysis of Variance Table
##
## Response: score
##           Df Sum Sq Mean Sq F value    Pr(>F)
## Machine    2 585.09 292.544 20.5761 0.0002855 ***
## Worker     5 413.97  82.793  5.8232 0.0089495 **
## Residuals 10 142.18  14.218
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# full model with interaction
m0 <- lm(score~Machine*Worker, data = Machines)

anova(m0)
```

```
## Analysis of Variance Table
##
## Response: score
##                Df  Sum Sq Mean Sq F value    Pr(>F)
## Machine         2 1755.26  877.63  949.17 < 2.2e-16 ***
## Worker          5 1241.89  248.38  268.63 < 2.2e-16 ***
## Machine:Worker 10  426.53   42.65   46.13 < 2.2e-16 ***
## Residuals      36   33.29    0.92
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
(142.18*3/10)/(33.29/36)
```

```
## [1] 46.12628
```

```r
1-pf((142.18*3/10)/(33.29/36), 10, 36)
```

```
## [1] 0
```

```
# additive model (no interaction)
m1 <- lm(score~Machine+Worker, data = Machines)

anova(m1)
```

```
## Analysis of Variance Table
##
## Response: score
##           Df  Sum Sq Mean Sq F value    Pr(>F)
## Machine    2 1755.26  877.63  87.798 < 2.2e-16 ***
## Worker     5 1241.89  248.38  24.848 4.867e-12 ***
## Residuals 46  459.82   10.00
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(m1,m0)
```

```
## Analysis of Variance Table
##
## Model 1: score ~ Machine + Worker
## Model 2: score ~ Machine * Worker
##   Res.Df    RSS Df Sum of Sq     F    Pr(>F)
## 1     46 459.82
## 2     36  33.29 10    426.53 46.13 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

4.2 General Mixed Model Parameter Estimation

For experiments comparing responses between factors the ANOVA-type analyses above are sufficient. But, for more general models with random effects, e.g., those including continuous covariates, general-purpose methods are needed. The general mixed effects model may be written

$$Y = X\beta + Z\alpha + \epsilon$$

where Y is an $n \times 1$ response, X is an $n \times p$ design matrix for fixed (non-random) effects; Z is and $n \times a$ matrix for random effects; β is a $p \times 1$ non-random coefficient vector; $\alpha \sim N_a(0, \psi_\theta)$ is an $a \times 1$ multivariate normal random coefficient vector with mean 0 and covariance matrix ψ_θ indexed by a parameter θ; and $\epsilon \sim N_n(0, \Lambda_\theta)$ is a multivariate normal random residual vector with

covariance matrix Λ_θ. An alternative way of writing the model (quite succintly) is

$$Y \sim N_n(X\beta, Z\psi_\theta Z^\top + \Lambda_\theta).$$

4.2.1 Maximum Likelihood

If you are familiar with least squares (and weighted least squares) you may recognize the MLE of β is given by the weighted least squares solution

$$\hat{\beta}_{WLS} = (X^\top W^{-1} X)^{-1} X^\top W^{-1} Y$$

where $W^{-1} = (Z\psi_\theta Z^\top + \Lambda_\theta)^{-1}$. The WLS estimator is fine in theory, but hard to compute in practice because it requires inverting an $n \times n$ matrix. For large n this is computationally challenging—$O(n^3)$ computations—unless the matrix has some special "sparse" structure. It's not just the number of computations (and time) needed to compute an $n \times n$ matrix inverse that is problematic. For large matrices, algorithms for computing the inverse may not maintain adequate numerical precision, and, therefore, the resulting matrix may not truly equal the inverse desired. These errors in floating-point arithmetic will compound for large matrix inversion problems and lead to an estimate $\widehat{\beta}_{WLS}$ that is not actually equal to the true weighted least squares estimate, due to computational errors.

Typically, the direct WLS approach to maximum likelihood estimation is replaced by an iterative procedure in order to avoid the computation of an inverse of a large matrix. The WLS solution above is based on the joint density (likelihood) of the observations Y. Instead, consider the joint density of (Y, α) parameterized by (β, θ). This joint density is conveniently expressed by the product of conditional and marginal densities $f(y|\alpha; \beta, \theta) \cdot f(\alpha; \beta, \theta)$ given by

$$f(y|\alpha; \beta, \theta) = (2\pi)^{-2/n} |\Lambda_\theta|^{-1/2} \exp\{-\tfrac{1}{2}\|y - X\beta - Z\alpha\|^2_{\lambda_\theta^{-1}}\},$$

where $\|a\|_B := a^\top B a$, and

$$f(\alpha; \beta, \theta) = (2\pi)^{-p/2} |\psi_\theta|^{-1/2} \exp\{-\tfrac{1}{2}\alpha^\top \psi_\theta^{-1} \alpha\}.$$

Now, the joint density $f(y, \alpha; \beta, \theta)$ cannot be used directly as a likelihood, i.e., $L(\beta, \theta | Y, \alpha)$ because α is an unobserved random variable. Instead, we could average over α values (integrate out α) to obtain the marginal density $f(y; \beta, \theta)$ and use this density with $Y = y$ fixed at the observed values as a likelihood for (β, θ). To compute the this marginal density we begin with a tranformation and a Taylor expansion:

$$\begin{aligned} f(y; \beta, \theta) &= \int f(y, \alpha; \beta, \theta) d\alpha = \int \exp\{\log f(y\alpha; \beta, \theta)\} d\alpha \\ &= \int \exp\left\{ \log f(y, \hat{\alpha}; \beta, \theta) + \tfrac{1}{2}(\alpha - \hat{\alpha})^\top \frac{\partial^2 \log f(y, \alpha; \beta, \theta)}{\partial \alpha \partial \alpha^\top} (\alpha - \hat{\alpha}) \right\} d\alpha, \end{aligned}$$

where $\hat{\alpha}$ is the optimal *prediction* of α maximizing $f(y, \alpha; \beta, \theta)$ for a fixed (β, θ). This maximizer has an explicit form: $\hat{\alpha} = (Z^\top \Lambda_\theta^{-1} Z + \psi_\theta^{-1})^{-1} Z^\top \Lambda_\theta^{-1}(Y - X\beta)$. The second line above contains a Taylor expansion of $\log f(y, \alpha; \beta, \theta)$ around $\hat{\alpha}$ which is exact, i.e., the higher-order terms are zero due to the Gaussian form of the densities. Pulling out constant terms from the integration we have

$$f(y; \beta, \theta) = f(y, \hat{\alpha}; \beta, \theta) \int \exp\left\{-\tfrac{1}{2}(\alpha - \hat{\alpha})^\top (Z^\top \lambda_\theta^{-1} Z + \psi_\theta^{-1})(\alpha - \hat{\alpha})\right\} d\alpha.$$

Using the fact the integrand is the kernel of a multivariate Gaussian density, we see the integral evaluates to $(2\pi)^{p/2}|Z^\top \Lambda_\theta^{-1} Z + \psi_\theta^{-1}|^{-1/2}$, and, hence

$$f(y; \beta, \theta) = f(y, \hat{\alpha}; \beta, \theta)(2\pi)^{p/2}|Z^\top \Lambda_\theta^{-1} Z + \psi_\theta^{-1}|^{-1/2}.$$

Now, using the marginal density of Y to define the loglikelihood we have

$$2\ell(\beta, \theta; y) = -\|y - X\beta - Z\hat{\alpha}\|_{\Lambda_\theta^{-1}} - \hat{\alpha}^\top \psi_\theta^{-1} \hat{\alpha} - \log|\Lambda_\theta| - \log|\psi_\theta| - \log|Z^\top \Lambda_\theta^{-1} Z + \psi_\theta^{-1}| - n \log 2\pi.$$

Given a fixed value of θ, the MLE $\hat{\beta}$ may be found by maximizing

$$-\|y - X\beta - Z\hat{\alpha}\|_{\Lambda_\theta^{-1}} - \hat{\alpha}^\top \psi_\theta^{-1} \hat{\alpha}$$

where, recall, $\hat{\alpha} = (Z^\top \Lambda_\theta^{-1} Z)^{-1} Z^\top \Lambda_\theta^{-1}(Y - X\beta)$. Plug in $\hat{\alpha}$ to see this becomes a WLS problem; hence, $\hat{\beta}$ is a WLS solution, and the weight matrix is given by $V = V_1 + V_2$, where (suppressing θ subscripts)

$$\begin{aligned} V_1 &= (I - Z(Z^\top \Lambda^{-1} Z)^{-1} Z^\top \Lambda^{-1})^\top \Lambda^{-1} (I - Z(Z^\top \Lambda^{-1} Z)^{-1} Z^\top \Lambda^{-1}) \\ &= \Lambda^{-1} - \Lambda^{-1} Z(Z^\top \Lambda^{-1} Z)^{-1} Z^\top \Lambda^{-1} \\ V_2 &= ((Z^\top \Lambda^{-1} Z)^{-1} Z^\top \Lambda^{-1})^\top \psi^{-1} ((Z^\top \Lambda^{-1} Z)^{-1} Z^\top \Lambda^{-1}) \\ &= \Lambda^{-1} Z(Z^\top \Lambda^{-1} Z)^{-1} \psi^{-1} (Z^\top \Lambda^{-1} Z)^{-1} Z^\top \Lambda^{-1} \end{aligned}$$

Computing the covariance of $\hat{\beta}$ still is computationally intensive. Computing the covariance of

$$(X^\top V X)^{-1} X^\top V Y$$

still requires inversion of the covariance of Y, which is the problematic component. Curiously, a Bayesian point of view provides a shortcut to the computation of the MLE covariance. Suppose we model (in a Bayesian sense) the parameter β with an improper constant prior, and the random component α with a multivariate normal prior with covariance ψ. Then, combining priors and multivariate normal likelihood we have the following posterior:

$$\log \Pi_n(\beta) = \text{const.} - \tfrac{1}{2}\left(y - X\beta - Z\alpha\right)^\top \Lambda^{-1} \left(y - X\beta - Z\alpha\right) - \tfrac{1}{2}\alpha^\top \psi^{-1} \alpha.$$

A nice (but otherwise hard to find) factorization of the posterior is due to Searle et al. (Variance Components, Section 9.2). Expand the exponent:

$$\begin{aligned} \log \Pi_n(\beta) = \text{const.} &- \tfrac{1}{2}\left(y - X\beta\right)^\top \Lambda^{-1} \left(y - X\beta\right) \\ &- \tfrac{1}{2}\alpha^\top (\psi^{-1} + Z^\top \Lambda^{-1} Z)\alpha + \alpha^\top Z^\top \Lambda^{-1}(y - X\beta) \end{aligned}$$

Let $A := \psi^{-1} + Z^\top \Lambda^{-1} Z$ and complete the square in α:

$$\begin{aligned}\log \Pi_n(\beta) = \text{const.} &- \tfrac{1}{2} (y - X\beta)^\top \Lambda^{-1} (y - X\beta) \\ &- \tfrac{1}{2}(\alpha - A^{-1} Z^\top \Lambda^{-1}(y - X\beta))^\top A(\alpha - A^{-1} Z^\top \Lambda^{-1}(y - X\beta)) \\ &+ \tfrac{1}{2}(y - X\beta)^\top \Lambda^{-1} Z A^{-1} Z^\top \Lambda^{-1}(y - X\beta).\end{aligned}$$

Combine the terms quadratic in $y - X\beta$. The resulting matrix $\Lambda^{-1} - \Lambda^{-1} Z A^{-1} Z^\top \Lambda^{-1}$ is equal to $(Z\psi Z^\top + \Lambda)^{-1}$, as can be seen by the following computation:

$$\begin{aligned}&(\Lambda^{-1} - \Lambda^{-1} Z A^{-1} Z^\top \Lambda^{-1})(Z\psi Z^\top + \Lambda) \\ &= \Lambda^{-1} Z\psi Z^\top + I - \Lambda^{-1} Z A^{-1} Z^\top \Lambda^{-1} Z\psi Z^\top - \Lambda^{-1} Z A^{-1} Z^\top \\ &= I + \Lambda^{-1} Z(\psi - A^{-1}(Z^\top \Lambda^{-1} Z\psi + I)) Z^\top \\ &= I + \Lambda^{-1} Z(\psi - \psi(Z^\top \Lambda^{-1} A\psi + I)^{-1}(Z^\top \Lambda^{-1} A\psi + I)) Z^\top \\ &= I\end{aligned}$$

Therefore, the posterior factorizes into

$$\begin{aligned}\log \Pi_n(\beta) = \text{const.} &- \tfrac{1}{2} (y - X\beta)^\top (Z\psi Z^\top + \Lambda)^{-1} (y - X\beta) \\ &- \tfrac{1}{2}(\alpha - A^{-1} Z^\top \Lambda^{-1}(y - X\beta))^\top A(\alpha - A^{-1} Z^\top \Lambda^{-1}(y - X\beta))\end{aligned}$$

Integrate over α to obtain the posterior for β. Marginally, α is multivariate normal; therefore, we obtain

$$\Pi_n(\beta) = (2\Pi)^{-n/2} |\Lambda|^{-1/2} |\psi|^{-1/2} |A|^{-1/2} \exp\left\{ -\tfrac{1}{2} (y - X\beta)^\top (Z\psi Z^\top + \Lambda)^{-1} (y - X\beta) \right\}.$$

It can be verified that $|Z\psi Z^\top + \Lambda| = |\Lambda||\psi||A|$ by using the following identity

$$|D^{-1} - CA^{-1}B| = |D||A^{-1}||A - BD^{-1}C|$$

where $Z\psi Z^\top + \Lambda = D^{-1} - CA^{-1}B = (\Lambda^{-1} - \Lambda^{-1} Z A^{-1} Z^\top \Lambda^{-1})^{-1}$; and, see, e.g., Appendix M equation 31 in Searle et al. (Variance Components). As a result, the posterior for β is multivariate normal with covariance $Z\psi Z^\top + \Lambda$.

Let $V := (Z\psi Z^\top + \Lambda)$. In the exponent, add and subtract $X\hat{\beta}$ in the quadratic to obtain

$$\begin{aligned}(y - X\beta)^\top V^{-1} (y - X\beta) = &\left(y - X\hat{\beta}\right)^\top V^{-1} (y - X\hat{\beta}) \\ &- 2\left(y - X\hat{\beta}\right)^\top V^{-1} (X\hat{\beta} - X\beta) \\ &+ \left(X\hat{\beta} - X\beta\right)^\top V^{-1} (X\hat{\beta} - X\beta).\end{aligned}$$

It is straightforward to show the middle term is zero. If we integrate out β (w.r.t. a multivariate normal density), then we obtain the following marginal posterior for the variance parameters:

$$\Pi_n(\theta, \Lambda) = \frac{(2\pi)^{-n/2}}{(2\pi)^{-p/2}} \frac{|V|^{-1/2}}{|X^\top V^{-1} X|^{-1/2}} \exp\left(-\tfrac{1}{2} \left(y - X\hat{\beta}\right)^\top V^{-1} (y - X\hat{\beta}) \right).$$