BI476: Biostatistics - Case Studies

Lec06: Linear Models and Generalizations

Maoying,Wu
ricket.woo@gmail.com

Dept. of Bioinformatics & Biostatistics
Shanghai Jiao Tong University

Spring, 2018

Outline

1. General Likelihood Theory
2. Generalized Linear Models (GLMs)
3. General Linear Regression
4. Logistic Regression
5. Poisson Regression

Next Section ...

1. General Likelihood Theory
2. Generalized Linear Models (GLMs)
3. General Linear Regression
4. Logistic Regression
5. Poisson Regression

General Likelihood Theory

Consider a random variable $Y \sim f(Y;\theta)$ and n i.i.d. observations $y = \{y_1, \ldots, y_n\}$

- The **likelihood function**

$$L(y;\theta) = \prod_{i=1}^{n} f(y_i;\theta)$$

- The **log-likelihood function**

$$\ell(y;\theta) = \sum_{i=1}^{n} \log f(y_i;\theta)$$

- **Maximum likelihood estimator**

$$\widehat{\theta} = \arg\max_{\theta} L(y;\theta) = \arg\max_{\theta} \ell(y;\theta)$$

General Likelihood Theory

- **Fisher's score function**:

$$u(\theta) = \partial\ell(y;\theta)/\partial\theta$$

- $\hat{\theta}_{\text{MLE}}$ can be obtained by solving

$$u(\theta) = 0$$

- **Fisher's information matrix**

$$I(\theta) = Var[u(\theta)] = E[u(\theta)u^T(\theta)] = -E\left[\frac{\partial^2\ell(y;\theta)}{\partial\theta\partial\theta^T}\right]$$

- **Asymptotic normal distribution**:

$$\sqrt{n}(\hat{\theta} - \theta) \sim N(0_p, I^{-1}(\theta))$$

Newton-Raphson Method for MLE

Generally MLE has no closed-form solution, numerical methods needed.

- First-order Taylor's extension for $u(\theta)$:

$$u(\theta) \approx u(\theta^{(0)}) - I(\theta^{(0)})(\theta - \theta^{(0)})$$

Algorithm 1 Newton's method for MLE

```
Initialize with $\theta^{(0)}$
while $\|\theta^{(k)} - \theta^{(k-1)}\| > \epsilon$ do
    Solve for $\theta^{(k)}$
        $$I(\theta^{(k-1)})(\theta^{(k)} - \theta^{(k-1)}) = u(\theta^{(k-1)})$$
end while
```

Likelihood Theory: Hypothesis Testing

Wald Test

$$\sqrt{n}(\hat{\theta} - \theta) \sim N\left(0, I^{-1}(\theta)\right), \theta \in \mathbb{R}^p$$

- $H_0 : \theta = \theta_0$
- **Test statistic**:

$$W = (\hat{\theta} - \theta_0)^T I^{-1}(\hat{\theta})(\hat{\theta} - \theta_0)$$

- Asymptotically χ^2 distributed:

$$W \sim \chi^2_p$$

Likelihood Theory: Hypothesis Testing

Score Test

$$u(\theta) \sim N\left(0_p, I(\theta)\right), \theta \in \mathbb{R}^p$$

- $H_0 : \theta = \theta_0$
- **Test statistic**:
$$Q = u^T(\theta_0) I^{-1}(\theta_0) u(\theta_0)$$
- Asymptotically χ^2 distributed:
$$Q \sim \chi^2_p$$

Likelihood Theory: Hypothesis Testing

Likelihood Ratio Test (LRT) for Comparing Nested Models

Two models with p_1 and p_2 ($p1 > p2$) parameters:

$$\begin{aligned}\hat{\theta}_1 &= \arg\max_{\theta_1 \in \mathbb{R}^{p_1}} l(y;\theta_1) \\ \hat{\theta}_2 &= \arg\max_{\theta_2 \in \mathbb{R}^{p_2}} l(y;\theta_2)\end{aligned}$$

- Two models have no significant difference.
- **Test statistic**:

$$D = 2\left(l(y;\hat{\theta}_1) - l(y;\hat{\theta}_2)\right)$$

- Asymptotically χ^2 distributed:

$$D \sim \chi^2_p, p = p_1 - p_2$$

Next Section ...

1. General Likelihood Theory
2. Generalized Linear Models (GLMs)
3. General Linear Regression
4. Logistic Regression
5. Poisson Regression

Generalized Linear Models (GLMs)

- Extension to linear models.
- The response Y follows a conditional exponential-**family** distribution.
- No assumption on the predictors, X
- Y and X are connected through a linear prediction model:

$$\eta = X\beta$$

  - $\eta = g(\mu)$, where $g(\cdot)$ is a **link function** of the conditional expected value of Y given X.
  - $\mu = E(Y|X)$ is the expected value of Y given X.

```
glm(formula = Y ~ X, family=gaussian(link="identity"))
```

Three Components of GLMs

Exponential Family Distributions

An exponential family is a family of distributions with pdf of the form:

$$f(Y;\theta,\phi) = \exp\left(\frac{Y\theta - b(\theta)}{a(\phi)} + c(\phi, Y)\right) g(Y)$$

where

- $g(Y)$ does not depend on θ or ϕ;
- θ: canonical parameter; ϕ: dispersion parameter;
- $a(\cdot), b(\cdot), c(\cdot)$ are known functions.

Example (Gaussian Distribution)

$$Y \sim \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(Y-\mu)^2}{2\sigma^2}\right)$$

- $\theta = \mu; b(\theta) = \mu^2/2$
- $\phi = \sigma; a(\phi) = \sigma^2$
- $c(\phi, Y) = -Y^2/(2\sigma^2) - \log(2\pi)$

Three Components of GLMs

Exponential Families: Properties

- Canonical statistic $t(y) = y$;
- Canonical parameter $\theta^* = \theta / a(\phi)$
- The cumulant generating function

$$\kappa\{\theta^*\} = \kappa\{\theta/a(\phi)\} = b(\theta)/a(\phi)$$

- The expected value of Y:

$$E(Y) = \frac{\partial}{\partial\theta^*}\kappa\{\theta/a(\phi)\} = a(\phi)\frac{\partial}{\partial\theta}\kappa\{\theta/a(\phi)\} = b'(\theta)$$

- The variance of Y:

$$V(Y) = \frac{\partial^2}{\partial\theta^{*2}}\kappa\{\theta/a(\phi)\} = a(\phi)^2\frac{\partial^2}{\partial\theta^2}\kappa\{\theta/a(\phi)\} = b''(\theta)a(\phi)$$

- If $Y_i \overset{\text{i.i.d}}{\sim} f(Y_i;\theta,\phi)$ for $i = 1,\ldots,n$, then the joint distribution of Y_i's also follows an exponential-family distribution. And $\sum Y_i$ is the **sufficient and complete statistics** for the canonical parameter.

Three Components of GLMs

Linear Predictors

$$\eta = X\beta = \beta_0 + X_1\beta_1 + \cdots + X_p\beta_p$$

is the dot product between the design matrix ($X = (\mathbf{1}, X_1, \ldots, X_p)$) and the coefficient ($\beta = (\beta_0, \ldots, \beta_p)^T$).

Three Components of GLMs

Link Function

$$E(Y) = \mu$$
$$g(\mu) = \eta = \beta_0 + X_1\beta_1 + \cdots + X_p\beta_p$$

- $\eta = g(\mu)$ must be 1-1 mapping.
- Usually $g(\cdot)$ is chosen to be a simple, continuously differentiable function.
- If possible, $g(\cdot)$ should be chosen based on data.

Example (Gaussian Distribution)

Since $b(\theta) = \mu^2/2 = \theta^2/2$, $E(Y) = b'(\theta) = \theta = \mu$ The canonical link function is

$$\eta = E(Y) = \mu$$

Iteratively Reweighted Least Square (IRWLS)

MLE Solution for GLMs

- $Y_i \sim \exp\left(\frac{y_i\theta_i - b(\theta_i)}{a(\phi)} + c(\phi, y_i)\right)$
- $\theta_i = \theta(\mu_i) = \theta(h(\eta_i)) = \theta(h(x_i^T\beta)), h = g^{-1}$
- The log-likelihood

$$\begin{array}{rcl} l(\beta) & = & \sum_{i=1}^{n} l_i(\beta) \\ l_i(\beta) & = & \frac{1}{a(\phi)}\left[y_i\theta(h(x_i^T\beta)) - b(\theta(h(x_i^T\beta)))\right] \end{array}$$

- β can be obtained through Newton-Raphson's method with Fisher scoring

Score function

$$u_r(\beta) = \frac{\partial l(\beta)}{\partial \beta_r} = \sum_{i=1}^{n} u_{ir} = \sum_{i=1}^{n} \frac{\partial l_i(\beta)}{\partial \beta_r}$$

We can Compute u_{ir}:

$$\frac{\partial l_i(\beta)}{\partial \beta_r} = \frac{\partial l_i}{\partial \theta_i} \frac{\partial \theta_i}{\partial \mu_i} \frac{\partial \mu_i}{\partial \eta_i} \frac{\partial \eta_i}{\partial \beta_r}$$

where

- $\frac{\partial l_i}{\partial \theta_i} = \frac{y_i - b'(\theta_i)}{a(\phi)} = \frac{y_i - \mu_i}{a(\phi)}$
- $\frac{\partial \theta_i}{\partial \mu_i} = \left(\frac{\partial \mu_i}{\partial \theta_i}\right)^{-1} = v_i^{-1}$ since
  - $\mu_i = b'(\theta_i)$
  - $\frac{\partial \mu_i}{\partial \theta_i} = b''(\theta_i) = Var(\mu_i) = v_i$
- $\frac{\partial \eta_i}{\partial \beta_r} = x_{ir}$

$$\frac{\partial l_i}{\partial \beta_r} = \frac{1}{a(\phi)} x_{ir} w_i \frac{\partial \eta_i}{\partial \mu_i} (y_i - \mu_i)$$

where $w_i = v_i^{-1} (\partial \eta_i / \partial \mu_i)^{-2}$

Score Function

$$\frac{\partial l}{\partial \beta} = \frac{1}{a(\phi)} X^T W \left[\frac{\partial \eta}{\partial \mu} (Y - \mu) \right]$$

where $W = \mathrm{diag}(w_i)$.

Fisher's Information Matrix

Similarly, we can get

$$A = \frac{1}{a(\phi)} X^T W X$$

Newton-Raphson Method for Fisher Scoring

$$X^T W(\beta^{(t-1)})X(\beta^{(t)} - \beta^{(t-1)}) = X^T W(\beta^{(t-1)}) \left[\frac{\partial \eta}{\partial \mu}(Y - \mu(\beta^{(t-1)}))\right]$$

Then we can get:

$$\begin{aligned} X^T W(\beta^{(t-1)})X\beta^{(t)} &= X^T W(\beta^{(t-1)}) \left\{X\beta^{(t-1)} \left[\tfrac{\partial \eta}{\partial \mu}(Y - \mu(\beta^{(t-1)}))\right]\right\} \\ &= X^T W(\beta^{(t-1)}) z(\beta^{(t-1)}) \end{aligned}$$

That is the reason why it is called **Iteratively Reweighted Least Squares**.

A Clinical Trial on Diastolic Blood Pressure (DBP)

Here we present a data set of diastolic blood pressure measured in small clinical trials in hypertension from the mid-to-late 1960s and for approximately a decade thereafter. Diastolic blood pressure (DBP) was measured (mmHg) in the supine position at baseline (i.e., “DBP1”) before randomization and monthly thereafter up to 4 months as indicated by `DBP2`, `DBP3`, `DBP4` and `DBP5`. Patients’ age and sex were recorded at baseline and represent potential covariates.

The primary objective in the analysis of this dataset is to test whether treatment A (new drug) may be effective in lowering DBP as compared to B (placebo) and to describe changes in DBP across the times at which it was measured.

Importing the dataset

```
dbp <- read.table("data/dbp.txt", header=T)
```

Next Section ...

1. General Likelihood Theory
2. Generalized Linear Models (GLMs)
3. General Linear Regression
4. Logistic Regression
5. Poisson Regression

Linear regression models

$$y = X\beta + \epsilon$$

- The quantitative “predictors”, X, contains the treatments in the clinical trials as well as other factors.
- X is the so-called design matrix with one column for treatment.
- y is the measured clinical endpoint (e.g., DBP)
- β is the vector of regression parameters
- $\epsilon \sim N(0, \sigma^2)$ is the error term
- Treatment can be incorporated into the regression using a *dummy variable*, X_1 as

$$X_1 = \begin{cases} 0 & TRT = A \\ 1 & TRT = B \end{cases}$$

- $X_2 = Age, X_3 = DBP1$

Estimation of the Linear Models

- Least-squares estimation of β by minimizing

$$\begin{aligned} \sum_{i=1}^{n} \epsilon_i^2 &= (y - X\beta)^T (y - X\beta) \\ &= y^T y - 2\beta X^T y + \beta^T X^T X \beta \end{aligned}$$

- Taking the partial derivative of the error sum of squares with respect to the component of β and setting to zero leads to

$$X^T X \beta = X^T \hat{\beta}$$

- When $X^T X$ is invertible

$$\hat{\beta} = (X^T X)^{-1} X^T y$$

Least-squares estimation

Predicted values

$\hat{y} = X\hat{\beta} = X(X^TX)^{-1}X^Ty = Hy$, where $H = X(X^TX)^{-1}X^T$ is called the *hat matrix*

Residuals for diagnostics

$\hat{\epsilon} = y - \hat{y} = y - X\hat{\beta} = (I - H)y$

Residual sum of squares

$RSS = \hat{\epsilon}^T\hat{\epsilon} = y^T(I - H)^T(I - H)y = y^T(I - H)y$

Unbiasedness

$\hat{\beta}$ is unbiased with variance $var(\hat{\beta}) = (X^TX)^{-1}\sigma^2$ if $var(\epsilon) = \sigma^2 I$

Variance estimate

$\hat{\sigma}^2 = \frac{\hat{\epsilon}^T\hat{\epsilon}}{n-p} = \frac{RSS}{n-p}$, where $n - p$ is the *degrees of freedom* of the model.

Coefficient of determination

$R^2 = \frac{(\hat{y}-y)^T(\hat{y}-y)}{(y-\bar{y})^T(y-\bar{y})}$

Next Section ...

1. General Likelihood Theory
2. Generalized Linear Models (GLMs)
3. General Linear Regression
4. Logistic Regression
5. Poisson Regression

Clinical Trials for Beta Blockers

This is a multi-center clinical trials in 22 centers to evaluate the efficacy of beta-blockers in reducing the mortality after myocardial infarction.

- `Deaths` is the number of deaths and `Total` is the total number of patients enrolled at each clincal `Center`.
- `Treatment` represents whether patients at each center were randomzied to `Control` or `Treated` (Beta-blocker)
- The outcome (death or non-death) are binomial and are used to demonstrate the application of logistic regression as well as remedies for over-dispersion using a `quasi-likelihood` approach.

Importing the data

```
betablocker <- read.table("data/betablocker.txt", header=TRUE)
head(betablocker)
```

```
##   Deaths Total Center Treatment
## 1      3    39      1   Control
## 2     14   116      2   Control
## 3     11    93      3   Control
## 4    127  1520      4   Control
## 5     27   365      5   Control
## 6      6    52      6   Control
```

Logistic regression for binary/binomial outcomes

- When the response is Bernoulli (e.g., death/alive or cured/uncured) or binomial (e.g. number of deaths among a fixed total number of patients)
- Linear regression method may yield biased estimates of the covariate parameters.
- $Y_i \sim \text{Bin}(n_i, p_i)$ can be written as

$$P(Y_i = y_i) = \binom{n_i}{y_i} p_i^{y_i} (1 - p_i)^{n_i - y_i}$$

- In the generalized linear model (GLM) framework, a **linear predictor** is used to model the linear relationship and a **link function** to link the linear predictor to the binomial probability p_i
  - Linear predictor

$$\eta_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_q x_{iq} = X_i \beta$$

  - The most commonly used link function for binomial response is the **logit** function

$$\eta = \log\left(\frac{p}{1 - p}\right)$$

  - Other link functions: **probit** and **complementary log-log**

MLE for Binomial Data

- The likelihood function

$$L(\beta|y) = \prod_{i=1}^{r} \binom{n_i}{y_i} p_i^{y_i} (1-p_i)^{n_i - y_i}$$

- The log-likelihood function

$$\ell(\beta|y) = \sum_{i=1}^{r} \left[\binom{n_i}{y_i} + y_i \log(p_i) + (n_i - y_i)\log(1-p_i) \right]$$

- For **logit** link function, $\eta = \log(\frac{p}{1-p})$, then $p = \frac{e^\eta}{1+e^\eta}$. The loglikelihood function becomes

$$\begin{aligned} \ell(\beta|y) &= \textstyle\sum_{i=1}^{r} \left[\log \binom{n_i}{y_i} + y_i \log(p_i) + (n_i - y_i)\log(1-p_i) \right] \\ &= \textstyle\sum_{i=1}^{r} \left[y_i \eta_i - n_i \log(1+e^{\eta_i}) + \log \binom{n_i}{y_i} \right] \end{aligned}$$

- This can be solved using numerical iterative approach.

Fitting a logistic regression model

```
# fit a logistic regression using glm
beta.glm <- glm(cbind(Deaths, Total-Deaths) ~ Treatment + Center,
                family=binomial(link="logit"), data=betablocker)
# print the model fitting
summary(beta.glm)
```

```
##
## Call:
## glm(formula = cbind(Deaths, Total - Deaths) ~ Treatment + Center,
##     family = binomial(link = "logit"), data = betablocker)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -5.0927  -1.4314  -0.2882   1.7262   5.5478
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -2.107580   0.058503 -36.025  < 2e-16 ***
## TreatmentTreated -0.256580   0.049429  -5.191 2.09e-07 ***
## Center          -0.008361   0.004516  -1.851   0.0641 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

Over-dispersion issues

- **Over-dispersion** is a common phenomenon in GLM including logistic and Poisson regression.
- Over-dispersion is evident if the deviance from the fitted model is too large.
- In the MLE theory, redidual deviance is asymptotically distributed as χ^2 with appropriate degrees of freedom.
- For χ^2 distributed deviance, its value should be close to its degrees of freedom.
  - When the deviance is greater than the df, **over-dispersion** occurs.
  - When the deviance is smaller than the df, **under-dispersion** occurs.

Estimate and Adjust the Dispersion parameter

Two-stage approach

- Estimate the dispersion parameter using Pearson's χ^2 as

$$\hat{\phi} = \frac{\chi^2}{n - p}$$

- Use the estimated dispersion parameter to adjust the model fit for further statistical inference.

Estimating and fitting the dispersion parameter

```
est.dp <- sum(resid(beta.glm, type="pearson")^2)/beta.glm$df.res
summary(beta.glm, dispersion=est.dp)
```

```
##
## Call:
## glm(formula = cbind(Deaths, Total - Deaths) ~ Treatment + Center,
##     family = binomial(link = "logit"), data = betablocker)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -5.0927  -1.4314  -0.2882   1.7262   5.5478
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -2.107580   0.164234 -12.833   <2e-16 ***
## TreatmentTreated -0.256580   0.138763  -1.849   0.0645 .
## Center           -0.008361   0.012679  -0.659   0.5096
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 7.880919)
##
##     Null deviance: 332.99  on 43  degrees of freedom
## Residual deviance: [illegible]  on 41  degrees of freedom
```

Over-dispersion: Quasi-likelihood approach

Combined approach.

- Permits estimating the dispersion parameter along with the model parameters simultaneously - without assuming an error distribution.
- This approach requires only the mean and variance function to be specified.

Quasi-likelihood for binomial data

```
# fit a quasi-likelihood for binomial data
beta.glm2 <- glm(cbind(Deaths, Total-Deaths)~Treatment + Center,
        family=quasibinomial, data=betablocker)
# print the model fitting
summary(beta.glm2)
```

```
##
## Call:
## glm(formula = cbind(Deaths, Total - Deaths) ~ Treatment + Center,
##     family = quasibinomial, data = betablocker)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -5.0927  -1.4314  -0.2882   1.7262   5.5478
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -2.107580   0.164235 -12.833 6.03e-16 ***
## TreatmentTreated -0.256580   0.138764  -1.849   0.0717 .
## Center          -0.008361   0.012679  -0.659   0.5133
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

Next Section ...

1. General Likelihood Theory
2. Generalized Linear Models (GLMs)
3. General Linear Regression
4. Logistic Regression
5. Poisson Regression

A Clinical Trial on Familial Andenomatous Polyposis

A placebo-controlled clinical trial of a non-sterioidal anti- inflammatory drug (NSAID) in treating FAP.

A planned interim analysis of the number of polyps to reveal significant evidence favoring NSAID treatment to warrant termination of the trial.

- `number`: Number of colonic polyps after 12 months of treatment
- `age`: Age at the baseline for patient.
- `treat`: Treatment assignment allocation, “drug” or “placebo”

Importing the data

```
polyps <- read.table("data/polyps.txt", header=TRUE)
head(polyps)
```

```
##   number treat age
## 1      2  drug  16
## 2     17  drug  22
## 3      1  drug  23
## 4     25  drug  17
## 5      3  drug  23
## 6     33  drug  23
```

```
str(polyps)
```

```
## 'data.frame': 20 obs. of  3 variables:
##  $ number: int  2 17 1 25 3 33 3 1 4 63 ...
##  $ treat : Factor w/ 2 levels "drug","placebo": 1 1 1 1 1 1 1 1 1 2 ...
##  $ age   : int  16 22 23 17 23 23 23 22 42 20 ...
```

Poisson regression for clinical outcome with counts

If Y is Poisson ditributed with mean $\lambda \geq 0$,

$$P(Y = y) = \frac{e^{\mu} \mu^{y}}{y!}$$

For a Poisson random variable Y, $E(Y) = Var(Y) = \mu$.

Suppose $Y_i \sim Pois(\mu)$ represents count data and that we want to model Y_i as a function of a set of clinical covariates of $(x_1, \ldots, x_q)$.

A link function is needed to link μ_i to $(x_1, \ldots, x_q)$ with the linear predictor $\eta_i = X_i\beta$. The natural canonical log link function can be written as:

$$\log(\mu_i) = \eta_i = X_i\beta$$

Poisson regression

The log-likelihood function is

$$\begin{aligned} \ell\beta &= \textstyle\sum_{i=1}^n \log P(Y=y) = \sum_{i=1}^n \log\left[\frac{e^{\mu_i}\mu_i^{y_i}}{y_i!}\right] \\ &= \textstyle\sum_{i=1}^n \left[y_i x_i^T\beta - \exp(x_i^T\beta) - \log(y_i!)\right] \end{aligned}$$

Differetiating with respect to β gives the MLE for $\hat{\beta}$ as the solution to

$$\sum_{i=1}^n \left[y_i - \exp(x_i^T\hat{\beta})\right] = 0$$

There is no closed-form analytic solution. Numerical search algorithms are needed.

Fitting a Poisson Model to the count data

```
# Poisson regression
m0.polyps <- glm(number ~ treat*age, polyps,
        family=poisson(link="log"))
# print the model fit
summary(m0.polyps)
```

```
##
## Call:
## glm(formula = number ~ treat * age, family = poisson(link = "log"),
##     data = polyps)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.2406  -3.0403  -0.0865   1.4392   5.8490
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)       3.261861   0.445995   7.314  2.6e-13 ***
## treatplacebo      1.257258   0.471626   2.666  0.00768 **
## age              -0.043033   0.019864  -2.166  0.03028 *
## treatplacebo:age  0.004631   0.020823   0.222  0.82402
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

Estimating and applying the dispersion parameter

```
est.dp <- sum(resid(m0.polyps, type="pearson")^2)/m0.polyps$df.res
summary(m0.polyps, dispersion=est.dp)
```

```
##
## Call:
## glm(formula = number ~ treat * age, family = poisson(link = "log"),
##     data = polyps)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.2406  -3.0403  -0.0865   1.4392   5.8490
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)      3.261861   1.504224   2.168   0.0301 *
## treatplacebo     1.257258   1.590673   0.790   0.4293
## age             -0.043033   0.066997  -0.642   0.5207
## treatplacebo:age 0.004631   0.070230   0.066   0.9474
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 11.37537)
##
```

Refitting Poisson model without interaction

```r
# Poisson regression
m1.polyps <- glm(number ~ treat + age, polyps,
        family=poisson(link="log"))
# print the model fit
summary(m1.polyps)
```

```
##
## Call:
## glm(formula = number ~ treat + age, family = poisson(link = "log"),
##     data = polyps)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.2212  -3.0536  -0.1802   1.4459   5.8301
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.169941   0.168210   18.84  < 2e-16 ***
## treatplacebo 1.359083   0.117643   11.55  < 2e-16 ***
## age         -0.038830   0.005955   -6.52 7.02e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
```

Quasi-Poisson Regression Model

```
# fit the quasi-Poisson
m2.polyps <- glm(number ~ treat + age, polyps,
                family=quasipoisson())
# print the model fit
summary(m2.polyps)
```

```
##
## Call:
## glm(formula = number ~ treat + age, family = quasipoisson(),
##     data = polyps)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.2212  -3.0536  -0.1802   1.4459   5.8301
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3.16994    0.55095   5.754 2.34e-05 ***
## treatplacebo    1.35908    0.38533   3.527  0.00259 **
## age            -0.03883    0.01951  -1.991  0.06284 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

Negative Binomial Approach

Another remedy to overdispersion is to use a more general distribution to relax the dependence of the mean and variance function, such as the negative binomial (NB) or gamma distribution to model the over- dispersion.

For a series of independent trials with $P(\text{success}) = p$, the random variable N for the number of trials until the k^{th} success is observed has a **negative-binomial** distribution with mdf:

$$P(N = n) = \binom{n-1}{k-1} p^k (1-p)^{n-k}$$

The NB is an extension of the Poisson distribution from a Bayesian perspective. It can be thought of as the Gamma-Poisson mixture. That is, NB is a Poisson(λ), where $\lambda \sim \Gamma(k, \frac{p}{1-p})$.

Mathematically, the NB is reparameterized for convenience in model fitting by defining $Y = N - k$ and $p = \frac{1}{1+\alpha}$ as

$$P(Y = y) = \binom{y+k-1}{k-1} \frac{\alpha^k}{(1+\alpha)^{y+k}}$$

Negative Binomial Approach

Therefore,

$$E(Y) = \mu = k\alpha$$
$$Var(Y) = k\alpha + k\alpha^2 = \mu + \mu^2/k$$

where an extra term $k\alpha^2$ can be used to model over-dispersion.

The log-likelihood function can be written as

$$\ell(\alpha, k) = \sum_{i=1}^{n} \left(y_i \log \frac{\alpha}{1+\alpha} - k \log(1+\alpha) + \sum_{j=0}^{y_j - 1} \log(j+k) - \log(y!) \right)$$

The *link function* to link mean response μ to a linear combination of the clinical covariates X is

$$\eta = X\beta = \log \frac{\alpha}{1+\alpha} = \log \frac{\mu}{\mu+k}$$

Fitting a negative-binomial regression model

```r
library(MASS)
# fit the negative binomial model
m3.polyps <- glm.nb(number ~ treat + age, polyps)
# print the model fit
summary(m3.polyps)
```

```
##
## Call:
## glm.nb(formula = number ~ treat + age, data = polyps, init.theta = 1.71
##     link = log)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.83270  -1.13898  -0.08851   0.33637   1.89785
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   3.15791    0.55753   5.664 1.48e-08 ***
## treatplacebo  1.36812    0.36903   3.707 0.000209 ***
## age          -0.03856    0.02095  -1.840 0.065751 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

Gamma-Poisson Model

Relationship of Exponential, Poisson and Gamma Distributions

Poisson and exponential distributions are very strongly related but they're fundamentally different because the Poisson is discrete (a count variable) and the exponential is continuous (a waiting time).

- If the time between a certain type of event is exponentially distributed with rate λ, then the number of events in a given time period of length t follows a Poisson distribution with parameter λt.
- The time to observe exacly n events is a sum of independent exponentially distributed random variables, and it follows a Gamma(λ, n) distribution (a.k.a **Erlang** distribution, to distinguish it from the general gamma distribution where n is allowed to be a non-integer).

Beta-Binomial Model - Priors

- If $X \sim$ Bernoulli(θ), then for $D = \{X_1, X_2, \ldots, X_n\}$, the likelihood

$$p(D\big|\theta) = \theta^{n_1}(1-\theta)^{n_0}$$

  where
  - $n_1 = \sum_{i=1}^{n} I(X_i = 1)$
  - $n_0 = \sum_{i=1}^{n} I(X_i = 0)$
- $n_1 \sim$ Bin(n, θ)
- For the sake of convenience, the prior for the parameter θ has the same form as the likelihood function (conjugate prior), then

$$p(\theta) = \theta^{\gamma_1}(1-\theta)^{\gamma_0}$$

- In a binomial model, the conjugate prior is a Beta distribution:

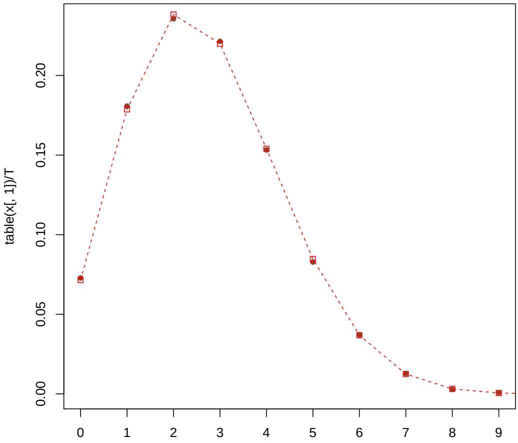
$$\text{Beta}(\theta|a, b) \propto \theta^{a-1}(1-\theta)^{b-1}$$

  where a, b are the hyperparameters.

Beta-Binomial Distribution

```r
R <- 10000
T <- 20000
x <- matrix(NA, R+T, 2)

n <- 10
alpha <- 7
beta <- 19

x[1,] <- c(1, .5)

for (i in 2:(R+T)) {
        x[i,2] <- rbeta(1, alpha, beta)
        x[i,1] <- rbinom(1, n, x[i,2])
}
x <- x[(R+1):(R+T), ]

plot(table(x[,1])/T, col="sienna4", type="p", pch=20)
betabinomialdensity <- function(x, n, alpha, beta)
        choose(n, x)*beta(alpha+x, beta+n-x)/beta(alpha,beta)
points(0:n, betabinomialdensity(0:n, n, alpha, beta),
        type="o", pch=22, lty=2, col="red")
```

Beta-Binomial Distribution


table(x[, 1])/T