Chapter 5 Building Logistic Regression Models

5.1 Model selection
5.2 Model checking (Deviance, Residuals)
5.3 Watch out for “sparse” categorical data

Model Selection with Many Predictors

Example (Horseshoe Crabs)

Y = whether female crab has satellites (1 = yes, 0 = no).

Explanatory variables:

- Weight
- Width
- Color (ML, M, MD, D) w/ dummy vars c_1, c_2, c_3
- Spine condition (3 categories) w/ dummy vars s_1, s_2

Consider model for crabs:

$$\begin{aligned}\text{logit}(\mathrm{P}(Y=1)) &= \alpha + \beta_1 c_{+}\beta_2 c_2 + \beta_3 c_3 + \beta_4 s_1 + \beta_5 s_1 \\ &\quad + \beta_5 \text{weight} + \beta_7 \text{width}\end{aligned}$$

```
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -8.06501    3.92855  -2.053   0.0401 *
C2          -0.10290    0.78259  -0.131   0.8954
C3          -0.48886    0.85312  -0.573   0.5666
C4          -1.60867    0.93553  -1.720   0.0855 .
S2          -0.09598    0.70337  -0.136   0.8915
S3           0.40029    0.50270   0.796   0.4259
Weight       0.82578    0.70383   1.173   0.2407
Width        0.26313    0.19530   1.347   0.1779
---

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 225.76  on 172  degrees of freedom
Residual deviance: 185.20  on 165  degrees of freedom
AIC: 201.2
```

None of the terms is significant in the Wald test, but...

- Residual deviance: 185.20 is the deviance of the model fitted.

$$\begin{aligned}H_a:\ \text{logit}(\mathrm{P}(Y=1)) &= \alpha + \beta_1 c_{+}\beta_2 c_2 + \beta_3 c_3 + \beta_4 s_1 + \beta_5 s_1 \\ &\quad + \beta_5 \text{weight} + \beta_7 \text{width}\end{aligned}$$

- Null deviance: 225.76 is the deviance under the model:

$$H_0:\ \text{logit}(\mathrm{P}(Y=1)) = \alpha.$$

H_0 means $\beta_1 = \beta_2 = \cdots = \beta_7 = 0$ in the model under H_a, which means none of the predictor has an effect.

$$\begin{aligned}\text{LR statistic} &= -2(L_0 - L_1) = \text{diff. of deviances} \\ &= 225.76 - 185.20 = 40.56\end{aligned}$$

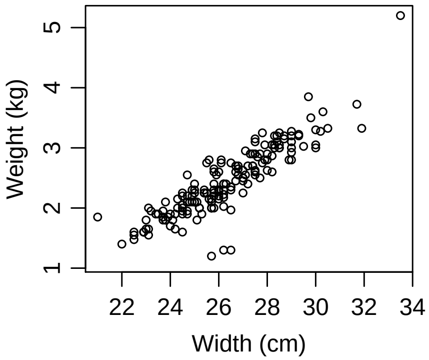
df = 7, P-value < 0.0001
Strong evidence saying **at least one predictor has an effect.**

But NONE of the terms is significant in the Wald test. Why?

Multicollinearity

Multicollinearity, which means “strong correlations among predictors”, causes troubles in linear models and GLMs.

E.g., Corr(weight,width) $= 0.89$



Recall β_i is partial effect of x_i on response controlling for other variables in model.
Sufficient to pick one of Weight and Width for a model.

Backward Elimination

1. Start with a complex model (e.g., including all predictors and interactions)
2. Drop “least significant” (i.e., largest P-value) variable among highest-order terms.
   - Cannot remove a main effect term w/o removing its higher-order interactions
   - Cannot remove a single dummy var. of a categorical predictor w/ > 2 categories
3. Refit model.
4. Continue until all variables left are “significant”

- Other automatic model selection procedures: forward selection, stepwise

Akaike Information Criterion (AIC)

Akaike information criterion (AIC) is a model selection criterion that selects the model minimizes

$$\text{AIC} = -2(\text{maximized log-likelihood}) + 2(\text{num. of parameters}).$$

- prefer simple models (few parameters) with good fit
- can be used to compare models that **neither is a special case of the other**, e.g., binomial models w/ diff. link functions

Example (Mouse Muscle Tension, Revisit)

We demonstrate the Backward Elimination procedure for the Mouse Muscle Tension data.

```
> mouse.muscle = read.table("mousemuscle.dat",header=T)
> mouse.muscle
     W M D tension.high tension.low
1 High 1 1            3           3
2 High 1 2           21          10
3 High 2 1           23          41
4 High 2 2           11          21
5  Low 1 1           22          45
6  Low 1 2           32          23
7  Low 2 1            4           6
8  Low 2 2           12          22

> attach(mouse.muscle)
> T = cbind(tension.high,tension.low) # response
> M = as.factor(M)                    # Muscle Type
> D = as.factor(D)                    # Drug
```

Backward elimination starts from the most complex model — 3-way interaction model, and then check significance of the highest order term — 3-way interactions.

```
> glm3 = glm(T ~ W*M*D, family=binomial)
> glm2 = glm(T ~ W*M + M*D + W*D, family=binomial)

> anova(glm2,glm3,test="Chisq")
Analysis of Deviance Table
Model 1: T ~ W * M + M * D + W * D
Model 2: T ~ W * M * D
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1         1      0.111
2         0      0.000  1    0.111    0.739
```

3-way interaction is not significant.

An alternative way to check significance:

```
> drop1(glm3, test="Chisq")
Single term deletions
Model:
T ~ W * M * D
       Df Deviance    AIC   LRT Pr(>Chi)
<none>       0.000 46.117
W:M:D   1    0.111 44.228 0.111    0.739
```

Only 3-way interaction is shown in the output of drop1 because drop1 drops one term at a time, other lower-order terms (W,M,D,W*M,M*D,W*D) cannot be dropped if 3-way interaction is in the model.

After eliminating the insignificant 3-way interaction, we consider the model with all 2-way interactions.

```
> glm2 = glm(T ~ W*M + M*D + W*D, family=binomial)
> drop1(glm2,test="Chisq")
Single term deletions

Model:
T ~ W * M + M * D + W * D
       Df Deviance    AIC     LRT Pr(>Chi)
<none>     0.11100 44.228
W:M     1  1.05827 43.175 0.94727   0.3304
M:D     1  2.80985 44.926 2.69886   0.1004
W:D     1  0.11952 42.236 0.00852   0.9264
```

Among the highest order terms (2-way interaction), W:D has the largest P-value and hance is least significant, so W:D is eliminated.

After eliminating (W:D), we fit the model

$$W * M + M * D = W + M + D + W * M + M * D$$

```
> glm2a = glm(T ~ W*M + M*D, family=binomial)

> drop1(glm2a, test="Chisq")
Single term deletions

Model:
T ~ W * M + M * D
       Df Deviance    AIC    LRT Pr(>Chi)
<none>      0.1195 42.236
W:M     1   1.0596 41.176 0.9401  0.33225
M:D     1   4.6644 44.781 4.5449  0.03302 *
```

This time, W:M is eliminated for it has the largest P-value among two-way interaction terms.

After eliminating W:M, we fit the model W + M*D
Note W is still in the model as we eliminate W:M from the model W*M + M*D.

```
> # glm2b = glm(T ~ M*D, family=binomial)           # not this one!
> glm2b = glm(T ~ W + M*D, family=binomial)
> drop1(glm2b, test="Chisq")
Single term deletions

Model:
T ~ W + M * D
       Df Deviance    AIC    LRT Pr(>Chi)
<none>      1.0596 41.176
W       1   1.5289 39.646 0.4693  0.49332
M:D     1   5.3106 43.427 4.2510  0.03923 *
```

Though W is of lower order than M:D, but W is not a component of M:D. The model is still hierarchical if we drop W and keep M:D.

Then we check model $M * D$, as M:D is significant. We cannot eliminate further or the model is not hierarchical.

```
> glm2c = glm(T ~ M*D, family=binomial)
> drop1(glm2c, test="Chisq")
Single term deletions

Model:
T ~ M * D
       Df Deviance    AIC   LRT Pr(>Chi)
<none>      1.5289 39.646
M:D     1   7.6979 43.814 6.169    0.013 *
```

The model selected by the backward elimination procedure is $M * D$.

This model also has the smallest AIC value, 39.646, among all models considered.

Backward Elimination in R

R function step() can do the backward elimination procedure we've just done automatically.

```
> step(glm3, test="Chisq")
Start:  AIC=46.12
T ~ W * M * D
        Df Deviance    AIC   LRT Pr(>Chi)
- W:M:D  1    0.111 44.228 0.111    0.739
<none>        0.000 46.117

Step:  AIC=44.23
T ~ W + M + D + W:M + W:D + M:D
       Df Deviance    AIC     LRT Pr(>Chi)
- W:D   1  0.11952 42.236 0.00852   0.9264
- W:M   1  1.05827 43.175 0.94727   0.3304
<none>     0.11100 44.228
- M:D   1  2.80985 44.926 2.69886   0.1004

Step:  AIC=42.24
T ~ W + M + D + W:M + M:D
       Df Deviance    AIC    LRT Pr(>Chi)
- W:M   1   1.0596 41.176 0.9401  0.33225
<none>      0.1195 42.236
- M:D   1   4.6644 44.781 4.5449  0.03302 *
```

```
Step:  AIC=41.18
T ~ W + M + D + M:D
       Df Deviance    AIC    LRT Pr(>Chi)
- W     1   1.5289 39.646 0.4693  0.49332
<none>      1.0596 41.176
- M:D   1   5.3106 43.427 4.2510  0.03923 *

Step:  AIC=39.65
T ~ M + D + M:D
       Df Deviance    AIC   LRT Pr(>Chi)
<none>      1.5289 39.646
- M:D   1   7.6979 43.814 6.169    0.013 *

Call:  glm(formula = T ~ M + D + M:D, family = binomial)

Coefficients:
(Intercept)           M2           D2        M2:D2
   -0.65233      0.09801      1.12611     -1.19750

Degrees of Freedom: 7 Total (i.e. Null);  4 Residual
Null Deviance:      19.02
Residual Deviance: 1.529        AIC: 39.65
```

Forward Selection in R

The R function `step()` can also do forward selection, which starts with a model with only an intercept (`~1`), and one most significant variable is added at each step, until none of remaining variables are "significant" when added to the model.

To run forward selection, you'll need to specify the "scope" of the search.

```
> step(glm(T ~1, family=binomial), scope=~W*M*D, direction="forward", test="Chisq")
Start:  AIC=51.14
T ~ 1
       Df Deviance    AIC    LRT Pr(>Chi)
+ D     1   12.460 46.577 6.5586  0.01044 *
+ M     1   13.579 47.695 5.4405  0.01967 *
<none>      19.019 51.136
+ W     1   18.957 53.073 0.0626  0.80251

Step:  AIC=46.58
T ~ D
       Df Deviance    AIC    LRT Pr(>Chi)
+ M     1   7.6979 43.814 4.7627  0.02908 *
<none>     12.4605 46.577
+ W     1  12.2889 48.406 0.1716  0.67871
```

Forward Selection in R (Cont'd)

```
Step:  AIC=43.81
T ~ D + M
       Df Deviance    AIC    LRT Pr(>Chi)
+ M:D   1   1.5289 39.646 6.1690   0.0130 *
+ W     1   5.3106 43.427 2.3872   0.1223
<none>      7.6979 43.814

Step:  AIC=39.65
T ~ D + M + D:M
       Df Deviance    AIC     LRT Pr(>Chi)
<none>      1.5289 39.646
+ W     1   1.0596 41.176 0.46928   0.4933

Call:  glm(formula = T ~ D + M + D:M, family = binomial)

Coefficients:
(Intercept)           D2           M2        D2:M2
   -0.65233      1.12611      0.09801     -1.19750

Degrees of Freedom: 7 Total (i.e. Null);  4 Residual
Null Deviance:      19.02
Residual Deviance: 1.529        AIC: 39.65
```

Both backward elimination and forward selection choose the model $M + D + M * D$.

$$\text{logit}(\pi_{ijk}) = \alpha + \beta_i^M + \beta_j^D + \beta_{ij}^{MD}$$

```
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.65233    0.24664  -2.645 0.008174 **
M2           0.09801    0.34518   0.284 0.776445
D2           1.12611    0.33167   3.395 0.000686 ***
M2:D2       -1.19750    0.48482  -2.470 0.013512 *
```

The fitted coefficients are

$$\begin{aligned} \widehat{\alpha} = -0.652, \quad \widehat{\beta}_1^M &= 0, \quad \widehat{\beta}_2^M = 0.098, \quad \widehat{\beta}_{11}^{MD} = 0, \quad \widehat{\beta}_{12}^{MD} = 0, \\ \widehat{\beta}_1^D &= 0, \quad \widehat{\beta}_2^D = 1.126, \quad \widehat{\beta}_{21}^{MD} = 0, \quad \widehat{\beta}_{22}^{MD} = -1.198. \end{aligned}$$

For Type 1 muscle, the odds of lowering muscle tension for Drug 2 is estimated to be $e^{\widehat{\beta}_2^D} = e^{1.126} \approx 3.0$ times the odds for Drug 1. For Type 2 muscle, the odds ratio is only

$$e^{\widehat{\beta}_2^D + \widehat{\beta}_{22}^{MD}} = e^{1.126 - 1.198} \approx 0.93.$$

5.1.1 How Many Predictors Can You Use?

- One published simulation study suggests > 10 outcomes of each type (S or F) per "predictor" (count dummy variables for factors).
  Example: $n = 1000$, $(Y = 1)$30 times, $(Y = 0)$ 970 times
  Model should contain $\leq \frac{30}{10} = 3$ predictors.
  Example: $n = 173$ crabs, $(Y = 1)$ 111 crabs, $(Y = 0)$ 62 crabs
  Use $\leq \frac{62}{10} \approx 6$ predictors.
- Can further check fit with residuals for grouped data, influence measures, cross validation.

5.2 MODEL CHECKING

- 5.2.1 Likelihood-Ratio Model Comparison Tests
  - introduced in the handouts for Chapter 4 already
- 5.2.2 Goodness of Fit and the Deviance
- 5.2.4 Residuals for Logit Models

Goodness of Fit and the Deviance

Binomial response data are usually of the following form:

	condition of the trials (explanatory variables)				number of trials	number of success
Condition 1	x_{11}	x_{12}	$\dots$	x_{1k}	n_1	y_1
Condition 2	x_{21}	x_{22}	$\dots$	x_{2k}	n_2	y_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$
Condition N	x_{N1}	x_{N2}	$\dots$	x_{Nk}	n_N	y_N

where $y_1, y_2, \dots, y_N$ are independent and

$$y_i \sim \text{Binomial}(n_i, \pi(\mathbf{x}_i)).$$

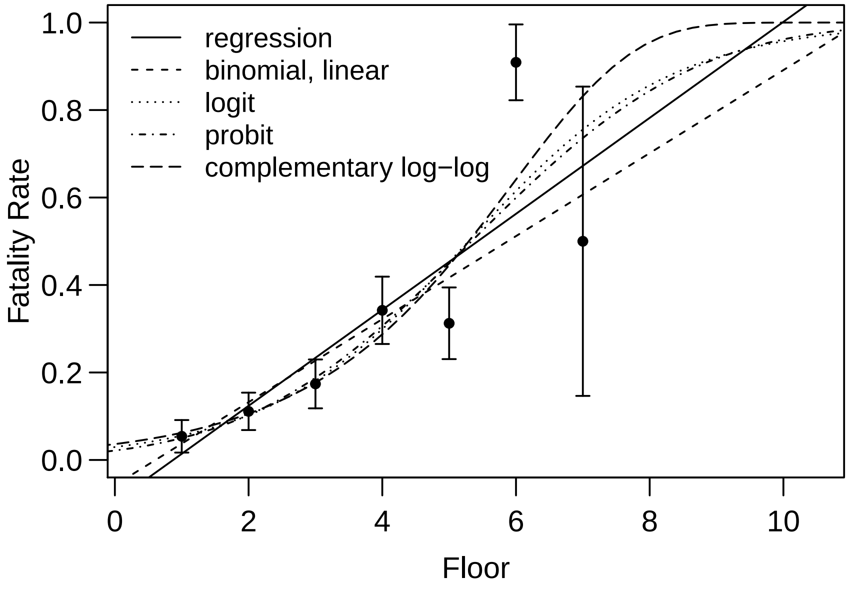
where $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{ik})$.

E.g., the data of fatal falls in handouts for Chapter 3 are of this form.

floor level x	total falls n_x	fatal falls y_x
1	37	2
2	54	6
3	46	8
4	38	13
5	32	10
6	11	10
7	2	1

Back to the Example of Fatal Falls

Which model fits the data the best?



Likelihood Revisit

A way to choose models is to compare their max. (log-)likelihoods.

$$\text{likelihood}: \prod\nolimits_i [\widehat{\pi}(\mathbf{x}_i)]^{y_i}[1-\widehat{\pi}(\mathbf{x}_i)]^{n_i - y_i}$$

$$\text{log-likelihood}: \sum\nolimits_i \{y_i \log \widehat{\pi}(\mathbf{x}_i) + (n_i - y_i)\log[1-\widehat{\pi}(\mathbf{x}_i)]\}$$

where $\widehat{\pi}(x)$ is the model fitted probabilities. E.g., for a probit model with a single predictor x

$$\widehat{\pi}(x) = \Phi(\widehat{\alpha} + \widehat{\beta}x).$$

Maximized log-likelihoods of four models of the fatal falls data:

Model	*Maximized Log-Likelihood*
linear	-102.4135
logit	-101.1594
probit	-101.2476
complementary log-log	-101.0744

The complementary log-log model has the largest log-likelihood. Is it the best?

Upper Bound of Maximized (Log-)Likelihood

Regardless of the functional form of $\pi(\mathbf{x}_i)$, the likelihood and log-likelihood must be of the form

$$\text{likelihood}: \prod\nolimits_i [\pi(\mathbf{x}_i)]^{y_i}[1-\pi(\mathbf{x}_i)]^{n_i-y_i}$$

$$\text{log-likelihood}: \sum\nolimits_i \{y_i \log \pi(\mathbf{x}_i) + (n_i-y_i)\log[1-\pi(\mathbf{x}_i)]\}$$

Since $y_i \log \pi(\mathbf{x}_i) + (n_i-y_i)\log[1-\pi(\mathbf{x}_i)]$ is the log-likelihood for a single observation $y_i \sim \text{binomial}(n_i, \pi(\mathbf{x}_i))$, which reaches its max when $\pi(\mathbf{x}_i)$ equals its MLE y_i/n_i, we know

$$y_i \log \widehat{\pi}(\mathbf{x}_i) + (n_i-y_i)\log[1-\widehat{\pi}(\mathbf{x}_i) \le y_i \log\left(\frac{y_i}{n_i}\right) + (n_i-y_i)\log\left(\frac{n_i-y_i}{n_i}\right).$$

So

the maximized log-likelihood of **any** model

$$= \sum\nolimits_i \{y_i \log \widehat{\pi}(\mathbf{x}_i) + (n_i-y_i)\log[1-\widehat{\pi}(\mathbf{x}_i)]\}$$

$$\le \sum\nolimits_i \left\{ y_i \log\left(\frac{y_i}{n_i}\right) + (n_i-y_i)\log\left(\frac{n_i-y_i}{n_i}\right)\right\}$$

floor level x	total falls n_x	fatal falls y_x
1	37	2
2	54	6
3	46	8
4	38	13
5	32	10
6	11	10
7	2	1

For the data of fatal falls, this upper bound for the maximized log-likelihood is

$$\begin{aligned}
&2\log\left(\frac{2}{37}\right) + (37-2)\log\left(\frac{37-2}{37}\right)\\
&+6\log\left(\frac{6}{54}\right) + (54-6)\log\left(\frac{54-6}{54}\right)\\
&+\cdots\\
&+1\log\left(\frac{1}{2}\right) + (2-1)\log\left(\frac{2-1}{2}\right)\\
&= -96.89521
\end{aligned}$$

Model	*Maximized Log-Likelihood*
linear	-102.4135
logit	-101.1594
probit	-101.2476
complementary log-log	-101.0744
upper bound	-96.8952

Deviance

The deviance of a model is 2 times the diff. of its maximized log-likelihood and the upper bound.

$$\begin{aligned}
\text{Deviance} &= -2(\text{max. log-likelihood} - \text{upper bound})\\
&= -2\Bigg(\sum\nolimits_i \{y_i \log \widehat{\pi}(\mathbf{x}_i) + (n_i-y_i)\log[1-\widehat{\pi}(\mathbf{x}_i)]\}\\
&\qquad - \sum\nolimits_i \left\{ y_i \log\left(\frac{y_i}{n_i}\right) + (n_i-y_i)\log\left(\frac{n_i-y_i}{n_i}\right)\right\}\Bigg)\\
&= 2\sum\nolimits_i \left\{ y_i \log\left(\frac{y_i}{n_i\widehat{\pi}(\mathbf{x}_i)}\right) + (n_i-y_i)\log\left(\frac{n_i-y_i}{n_i(1-\widehat{\pi}(\mathbf{x}_i))}\right)\right\}\\
&= 2\sum_i (\text{observed})\log\left(\frac{\text{observed}}{\text{fitted}}\right)\\
&= G^2
\end{aligned}$$

For the logistic model of the fatal falls data,

floor level	observed fatal count	fitted fatal count	observed live count	fitted live count
1	2	2.06	35	34.94
2	6	5.52	48	48.48
3	8	8.31	38	37.69
4	13	11.36	25	26.64
5	10	14.47	22	17.53
6	10	6.76	1	4.24
7	1	1.51	1	0.49

$$\begin{aligned}
\text{Deviance} = 2\Bigg[&2\log\left(\frac{2}{2.06}\right) + 35\log\left(\frac{35}{34.94}\right)\\
&+6\log\left(\frac{6}{5.52}\right) + 48\log\left(\frac{48}{48.48}\right)\\
&+\ldots\\
&+1\log\left(\frac{1}{1.51}\right) + 1\log\left(\frac{1}{0.49}\right)\Bigg] \approx 8.5283
\end{aligned}$$

```
> ff = read.table("falls.dat",h=T)
> ff.logit = glm(cbind(fatal,live) ~ floor,
                  family = binomial(link="logit"),data=ff)
> summary(ff.logit)
Call:
glm(formula = cbind(fatal, live) ~ floor, family = binomial(link = "logit"),
    data = ff)

Deviance Residuals:
       1         2         3         4         5         6         7
-0.04171   0.21116  -0.11936   0.57263  -1.61351   2.22062  -0.77799

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -3.4920     0.5009  -6.971 3.14e-12 ***
floor         0.6600     0.1253   5.267 1.38e-07 ***
---
(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 42.0319  on 6  degrees of freedom
Residual deviance:  8.5283  on 5  degrees of freedom
AIC: 33.451

Number of Fisher Scoring iterations: 4
```

The Saturated Model

The upper bound for maximized log-likelihoods itself is also the maximized likelihood for a model — the **saturated model**.

The *saturated model* is the most complex model possible for the data, which has a separate parameter $\pi_i = \pi(\mathbf{x}_i)$ for each (n_i, y_i) and fits the data perfectly that

$$\widehat{\pi}_i = \frac{y_i}{n_i}.$$

Example (Fatal Falls). The saturate model has a separate parameter π_i for each floor level $i = 1, 2, 3 \ldots, 7$.

The Saturated Model

- number of parameters in the saturated model = number of observations in data
- If the number of parameters in a model is the same as the number of observations, then this model is usually the saturated model.

Example (Mouse Muscle Tension). The saturate model is the 3-way interaction model, for it has 8 parameters, same as the number of observations.

- Deviance for the saturated model = 0

```
> mouse.muscle = read.table("mousemuscle.dat",header=T)
> mouse.muscle
     W M D tension.high tension.low
1 High 1 1            3           3
2 High 1 2           21          10
3 High 2 1           23          41
4 High 2 2           11          21
5  Low 1 1           22          45
6  Low 1 2           32          23
7  Low 2 1            4           6
8  Low 2 2           12          22
```

```
> glm3 = glm(cbind(tension.high,tension.low) ~ W*M*D,
+               family=binomial, data=mouse.muscle)
> summary(glm3)

Call:
glm(formula = cbind(tension.high, tension.low) ~ W * M * D, family = binomial,
    data = mouse.muscle)

Deviance Residuals:
[1]  0  0  0  0  0  0  0  0

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.9743     3.4157  -0.285    0.775
WLow         -2.3438     3.8528  -0.608    0.543
M             0.2324     1.7956   0.129    0.897
D             1.5524     1.8611   0.834    0.404
WLow:M        1.3243     2.3163   0.572    0.568
WLow:D        0.7400     2.1398   0.346    0.729
M:D          -0.8105     1.0103  -0.802    0.422
WLow:M:D     -0.4360     1.3071  -0.334    0.739

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1.9019e+01  on 7  degrees of freedom
Residual deviance: 1.1324e-14  on 0  degrees of freedom
AIC: 46.117

Number of Fisher Scoring iterations: 3
```

Goodness of Fit and the Deviance

For a model M of interest, let L_M denote the its maximized log-likelihood. As the upper bound for maximized log-likelihoods itself is the maximized log-likelihood for the saturated model L_S, the **deviance** of the model M equals

$$\text{Deviance} = -2[L_M - (\text{upper bound})] = -2(L_M - L_S),$$

which is the likelihood ratio test statistic comparing

$$H_0 : \text{Model } M \quad \text{v.s.} \quad H_a : \text{saturated model.}$$

Deviance has an approx. **chi-squared** distribution w/

$$\begin{aligned}\text{df} &= (\#\text{ of parameters in saturated model}) \\ &\qquad - (\#\text{ of parameters in Model } M) \\ &= (\#\text{ of observations}) - (\#\text{ of parameters in Model } M)\end{aligned}$$

However, this approx. is good only when all observations (n_i, y_i) have large n_i.

Goodness of Fit and the Deviance

- Large deviance indicates lack of fit
- Small deviance means the model fits nearly as good as the best possible model

Goodness of Fit test for the four models of fatal falls data:

Model	*Deviance*	d.f.	*P*-value
linear (identity)	11.04	5	0.0507
probit	8.70	5	0.1214
logit	8.53	5	0.1294
complementary log-log	8.36	5	0.1376

Goodness-of-fit tests shows the 3 binomial models w/ logit, probit, complementary log-log link fit the data nearly as good as each other, and their fits are a bit better than the model w/ identity link.

Example (Mouse Muscle Tension)

For the mouse muscle tension data, the saturated model is the 3-way interaction model, the Goodness of fit test of a model is simply comparing the model with the 3-way interaction model.

```
> glm3 = glm(cbind(tension.high,tension.low) ~ W*M*D,
                  family=binomial, data=mouse.muscle)
> glm2 = glm(cbind(tension.high,tension.low) ~ M*D,
                  family=binomial, data=mouse.muscle)
> anova(glm2, glm3,test="Chisq")
Analysis of Deviance Table

Model 1: cbind(tension.high, tension.low) ~ M * D
Model 2: cbind(tension.high, tension.low) ~ W * M * D
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1         4     1.5289
2         0     0.0000  4   1.5289   0.8215
```

Goodness-of Fit Based on Pearson's Chi-Squared

One can also use Pearson's Chi-Squared statistic

$$X^2 = \sum_i \left\{ \frac{(y_i - n_i\pi(\mathbf{x}_i))^2}{n_i\widehat{\pi}(\mathbf{x}_i)} + \frac{[n_i - y_i - n_i(1-\widehat{\pi}(\mathbf{x}_i))]^2}{n_i(1-\widehat{\pi}(\mathbf{x}_i))} \right\}$$
$$= \sum \frac{(\text{observed} - \text{fitted})^2}{\text{fitted}}$$

to do goodness-of-fit test comparing

$$H_0 : \text{Model } M \quad \text{v.s.} \quad H_a : \text{saturated model.}$$

X^2 is different from Deviance but it has an approx. **chi-squared** distribution w/ same d.f. as Deviance.

Like deviance, the approx. for X^2 is good only when all observations (n_i, y_i) have large n_i.

Grouped Data v.s. Ungrouped Data

Although the ML estimates of parameters are the same for grouped or ungrouped data, the deviances are different.

For ungrouped data, $n_i = 1$ for all i and $y_i = 0$ or 1, so

$$L_S = \sum_i \left\{ y_i \log\left(\frac{y_i}{n_i}\right) + (n_i - y_i)\log\left(\frac{n_i - y_i}{n_i}\right) \right\}$$
$$= \sum_i \{ y_i \log(y_i) + (1-y_i)\log(1-y_i) \} = 0$$

and hence

$$\text{Deviance} = -2(L_M - L_S) = -2L_M.$$

Grouped Data v.s. Ungrouped Data

```
> ff = read.table("falls.dat", header=T)           # Grouped data
> ff.ug = read.table("fallsUG.dat", header=T)      # Ungrouped DATA

> ff.logit = glm(cbind(fatal,live) ~ floor, family=binomial, data=ff)
> ffug.logit =glm((outcome == "fatal")~floor,family=binomial, data=ff.ug)

> ff.logit$coef
(Intercept)       floor
 -3.4920438   0.6600324
> ffug.logit$coef             # same estimated coefficients
(Intercept)       floor
 -3.4920437   0.6600324

> ff.logit$deviance
[1] 8.52832
> ffug.logit$deviance         # different deviances
[1] 202.3187

> ff.logit$df.residual        # different df for deviances
[1] 5
> ffug.logit$df.residual
[1] 218
```

Grouped Data, Ungrouped Data, Continuous Predictors

- Only deviance computed based on grouped data can be used to do goodness of fit test. Deviances computed based on ungrouped data do not have approx. chi-squared dist..
- Continuous predictors usually have too many levels (e.g., Width in horseshoe crabs data) that deviances of models w/ such predictors do not have approx. chi-squared dist if the number of observations at each levels are too small.
- Even though deviances may not have approx. chi-squared dist., the difference of deviances of two models is often approx. Chi-squared.

  One can safely use the diff. of deviances to do likelihood ratio test for model comparison no matter what.

Summary for Deviance

For a Model M of interest

$$\begin{aligned}
\text{Deviance} &= -2(L_M - L_S)\\
&= 2\sum\nolimits_i \left\{ y_i \log\left(\frac{y_i}{n_i\widehat{\pi}(\mathbf{x}_i)}\right) + (n_i - y_i)\log\left(\frac{n_i - y_i}{n_i(1-\widehat{\pi}(\mathbf{x}_i))}\right)\right\}\\
&= 2\sum_i (\text{observed}) \log\left(\frac{\text{observed}}{\text{fitted}}\right)\\
&= G^2
\end{aligned}$$

where

$$\begin{aligned}
L_M &= \text{max. log-likelihood for Model } M\\
L_S &= \text{max. log-likelihood for the saturated model}\\
&= \text{the upper bound for max. log-likelihood of ANY model}
\end{aligned}$$

Deviance can be used to do goodness-of-fit test.

Residuals for Binomial Response Models

not limited to logistic models

When goodness-of-fit test suggests a GLM fits poorly, residuals can highlight where the fit is poor.

$$\text{Pearson Residual } e_i = \frac{y_i - n_i\widehat{\pi}_i}{\sqrt{n_i\widehat{\pi}_i(1-\widehat{\pi}_i)}}$$

$$\text{Standardized (Pearson) Residual } r_i = \frac{e_i}{\sqrt{1-h_i}}$$

- h_i = *leverage* of the observation i (details are skipped). The greater an observation's leverage, the greater its influence on the model fit.
- Note $\sum_i e_i^2 = X^2$ (Pearson chi-square)
- When model holds and $n_i\widehat{\pi}_i$ are large, e_i is approx. $N(0,\nu)$ but $\nu < 1$, r_i is approx. $N(0,1)$. $|r_i| > 2$ or 3 means lack of fit.
- Useful for grouped data only.

Deviance Residuals for Binomial Response Models

not limited to logistic models

The **deviance residual** is defined as

$$d_i = \text{sign}(y_i - \widehat{\mu}_i)\sqrt{2\left[y_i \log\left(\frac{y_i}{\widehat{\mu}_i}\right) + (n_i - y_i)\log\left(\frac{n_i - y_i}{n_i - \widehat{\mu}_i}\right)\right]}$$

where $\widehat{\mu}_i = n_i\widehat{\pi}(\mathbf{x}_i)$.

Standardized deviance residual $= \dfrac{d_i}{\sqrt{1-h_i}}$ where h_i is leverage.

- Observe that Deviance $= \sum_i d_i^2$.
- When model holds and $n_i\widehat{\pi}_i$ large, d_i approx. $N(0,\nu)$ but $\nu < 1$, should use standardized d_i
- Useful for grouped data only.

Example (Berkeley Graduate Admissions)

	Men			Women		
Dept	Number Admitted	Number Rejected	Percent Admitted	Number Admitted	Number Rejected	Percent Admitted
A	512	313	62%	89	19	82%
B	353	207	63%	17	8	68%
C	120	205	37%	202	391	34%
D	138	279	33%	131	244	35%
E	53	138	28%	94	299	24%
F	22	351	6%	24	317	7%

```
> UCB = read.table("UCBadmissions.dat",h=T)
> UCB
   Gender Dept Admitted Rejected
1    Male    A      512      313
2    Male    B      353      207
3    Male    C      120      205
4    Male    D      138      279
5    Male    E       53      138
6    Male    F       22      351
7  Female    A       89       19
8  Female    B       17        8
9  Female    C      202      391
10 Female    D      131      244
11 Female    E       94      299
12 Female    F       24      317
```

Let's first fit a model with only the main effects of Department and Gender, but no interactions.

```
> UCB.fit1 = glm(cbind(Admitted,Rejected) ~ Dept + Gender,
         family=binomial, data=UCB)
> summary(UCB.fit1)
Deviance Residuals:
      1        2        3        4        5        6        7        8
-1.2487  -0.0560   1.2533   0.0826   1.2205  -0.2076   3.7189   0.2706
      9       10       11       12
-0.9243  -0.0858  -0.8509   0.2052

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.68192    0.09911   6.880 5.97e-12 ***
DeptB       -0.04340    0.10984  -0.395    0.693
DeptC       -1.26260    0.10663 -11.841  < 2e-16 ***
DeptD       -1.29461    0.10582 -12.234  < 2e-16 ***
DeptE       -1.73931    0.12611 -13.792  < 2e-16 ***
DeptF       -3.30648    0.16998 -19.452  < 2e-16 ***
GenderMale  -0.09987    0.08085  -1.235    0.217
---
(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 877.056  on 11  degrees of freedom
Residual deviance:  20.204  on  5  degrees of freedom
AIC: 103.14

Number of Fisher Scoring iterations: 4
```

LRT indicates strong Dept effect, but little Gender effect (P-value ≈ 0.22). $\Rightarrow$ little evidence of gender bias in UCB graduate admissions.

```
> drop1(UCB.fit1, test="Chisq")
Single term deletions

Model:
cbind(Admitted, Rejected) ~ Dept + Gender
       Df Deviance    AIC    LRT Pr(>Chi)
<none>       20.20 103.14
Dept    5   783.61 856.55 763.40   <2e-16 ***
Gender  1    21.74 102.68   1.53   0.2159
```

However, ...

However, goodness of fit test shows the main effect model fits poorly. The Deviance $= 20.204$ can be obtained from the summary output, or from the commands below

```
> UCB.fit1$deviance
[1] 20.20428
```

The P-value for goodness of fit test ≈ 0.00114 is computed as follows.

```
> pchisq(20.204, df=5, lower.tail=F)
[1] 0.001144215
```

Apparently there is gender×dept interaction
(because the saturated model is the two-way interaction model).

R function `residuals()` gives deviance residuals by default, and Pearson residuals with option `type="pearson"`.

```
> residuals(UCB.fit1)                 # deviance residuals
          1           2           3           4           5           6 
-1.24867404 -0.05600850  1.25333751  0.08256736  1.22051370 -0.20756402 
          7           8           9          10          11          12 
 3.71892028  0.27060804 -0.92433979 -0.08577122 -0.85093316  0.20517793 

> residuals(UCB.fit1, type="pearson") # Pearson residuals
          1           2           3           4           5           6 
-1.25380765 -0.05602052  1.26287232  0.08260773  1.24151319 -0.20620096 
          7           8           9          10          11          12 
 3.51866744  0.26895159 -0.92077831 -0.08573167 -0.84403319  0.20648081 
```

By default, R function `rstandard()` gives standardized deviance residuals.

```
> rstandard(UCB.fit1)
         1          2          3          4          5          6 
-4.0107986 -0.2796622  1.8666312  0.1411928  1.6058628 -0.3046444 
         7          8          9         10         11         12 
 4.2564872  0.2814450 -1.8881065 -0.1413270 -1.6468462  0.3007342 
```

With option `type="pearson"`, `rstandard()` gives standardized Pearson residuals.

```
> rstandard(UCB.fit1, type="pearson")
         1          2          3          4          5          6 
-4.0272880 -0.2797222  1.8808316  0.1412619  1.6334924 -0.3026439 
         7          8          9         10         11         12 
 4.0272880  0.2797222 -1.8808316 -0.1412619 -1.6334924  0.3026439 
```

```
> pearson.res = round(residuals(UCB.fit1, type="pearson"),2)
> std.res = round(rstandard(UCB.fit1,type="pearson"), 2)
> cbind(UCB, pearson.res, std.res)
   Gender Dept Admitted Rejected pearson.res std.res
1    Male    A      512      313       -1.25   -4.03  <--
2    Male    B      353      207       -0.06   -0.28
3    Male    C      120      205        1.26    1.88
4    Male    D      138      279        0.08    0.14
5    Male    E       53      138        1.24    1.63
6    Male    F       22      351       -0.21   -0.30
7  Female    A       89       19        3.52    4.03  <--
8  Female    B       17        8        0.27    0.28
9  Female    C      202      391       -0.92   -1.88
10 Female    D      131      244       -0.09   -0.14
11 Female    E       94      299       -0.84   -1.63
12 Female    F       24      317        0.21    0.30
```

Standardized residuals suggest Dept. A as main source of lack of fit ($r_i = -4.03$ and 4.03), while Pearson residuals fail to catch the lack of fit of the first observation (Gender = Male, Dept = A).

Leaving out Dept. A, the model with Dept main effects and gender main effects fits well (Deviance $= 2.556$, df $= 4$, P-value ≈ 0.63.)

```
> UCB.fit2 = glm(cbind(Admitted,Rejected) ~ Dept + Gender,
                family=binomial, data=UCB, subset=(Dept != "A"))
> UCB.fit2$deviance
[1] 2.556429
> UCB.fit2$df.residual
[1] 4
> pchisq(2.556429, df=4, lower.tail=F)
[1] 0.6345606
```

Knowing the main effect model fits the data well when leaving out Dept. A, we can use it to do inference.

LRT shows gender effect is not significant (P-value $= 0.72$), meaning little evidence of gender bias in UCB graduate admissions in Dept. B, C, D, E, F.

```
> drop1(UCB.fit2, test="Chisq")
Single term deletions

Model:
cbind(Admitted, Rejected) ~ Dept + Gender
       Df Deviance    AIC    LRT Pr(>Chi)
<none>        2.56  71.79
Dept    4   500.85 562.08 498.29   <2e-16 ***
Gender  1     2.68  69.92   0.13   0.7236
```

In Dept. A, odds of admission for men are $\frac{512 \times 19}{313 \times 89} = 0.35$ times the odds for women.

Dept A	Admitted	Rejected
Male	512	313
Female	89	19

Conclusion:

- In Dept. A, women are more likely to be admitted
- In Dept. B-F, no significant diff. in admission rates of men and women.

However, if we ignore Dept, Gender effect is significant but in the opposite direction — odds of admission for men are $e^{0.61} = 1.84$ times the odds for women (95% CI for odds ratio is 1.625 to 2.087.) Men are more likely to be admitted. Why?

```
> UCB.fit3 = glm(cbind(Admitted,Rejected) ~ Gender,
                   family=binomial, data=UCB)
> UCB.fit3$coef
(Intercept)  GenderMale
 -0.8304864   0.6103524

> exp(confint(UCB.fit3))
                2.5 %    97.5 %
(Intercept) 0.3942898 0.4811371
GenderMale  1.6249557 2.0874993
```

- This is an example of Simpson's paradox.

Sparse Data

Caution: Parameter estimates in logistic regression can be infinite.

Example 1:

	S	F
$X = 1$	8	2
$X = 2$	10	0

Model:

$$\log\left(\frac{Pr(S)}{Pr(F)}\right) = \alpha + \beta x \qquad e^{\widehat{\beta}} = \text{odds-ratio} = \frac{8 \times 0}{2 \times 10} = 0$$

$$\widehat{\beta} = \text{log-odds-ratio} = -\infty$$

Empty cells in multi-way contingency table can cause infinite estimates.

Software may not realize this, and gives a finite estimate!

- Large `Number of Fisher Scoring iterations` is a warning sign
- Large values of SEs for coefficients are also warning signs

```
> S = c(8,10)
> F = c(2,0)
> X = c(1,2)
> glm1 = glm(cbind(S,F) ~ X, family = binomial)
> summary(glm1)
Call:
glm(formula = cbind(S, F) ~ X, family = binomial)

Deviance Residuals:
[1]  0  0

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   -22.35   54605.92       0        1
X              23.73   54605.92       0        1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2.9953e+00  on 1  degrees of freedom
Residual deviance: 2.4675e-10  on 0  degrees of freedom
AIC: 6.3947

Number of Fisher Scoring iterations: 22
```

Infinite estimates exist when x-values where $y = 1$ can be "separated" from x-values where $y = 0$.

Example 2:

```
> X = c(0,1,2,3,4,5,6,7)
> Y = c(0,0,0,0,1,1,1,1)
```

Model:

$$\text{logit}(Pr(Y = 1)) = \alpha + \beta x$$

What does the XY scatter plot look like?

```
> X = c(0,1,2,3,4,5,6,7)
> Y = c(0,0,0,0,1,1,1,1)
> glm2 = glm(Y ~ X, family = binomial)
Warning message:
glm.fit: fitted probabilities numerically 0 or 1 occurred
> summary(glm2)
Call:
glm(formula = Y ~ X, family = binomial)

Deviance Residuals:
       Min          1Q      Median          3Q         Max
-1.504e-05  -2.110e-08   0.000e+00   2.110e-08   1.504e-05

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   -160.3   285119.4  -0.001        1
X               45.8    80643.9   0.001        1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1.1090e+01  on 7  degrees of freedom
Residual deviance: 4.5253e-10  on 6  degrees of freedom
AIC: 4

Number of Fisher Scoring iterations: 25
```