15.1 - Logistic Regression

https://onlinecourses.science.psu.edu/stat501

2018

Contents

1 Logistic Regression

1.1 Introduction

Logistic regression models a relationship between predictor variables and a categorical response variable. For example, we could use logistic regression to model the relationship between various measurements of a manufactured specimen (such as dimensions and chemical composition) to predict if a crack greater than 10 mils will occur (a binary variable: either yes or no). Logistic regression helps us estimate a probability of falling into a certain level of the categorical response given a set of predictors. We can choose from three types of logistic regression, depending on the nature of the categorical response variable:

Binary Logistic Regression:

Used when the response is binary (i.e., it has two possible outcomes). The cracking example given above would utilize binary logistic regression. Other examples of binary responses could include passing or failing a test, responding yes or no on a survey, and having high or low blood pressure.

Nominal Logistic Regression:

Used when there are three or more categories with no natural ordering to the levels. Examples of nominal responses could include departments at a business (e.g., marketing, sales, HR), type of search engine used (e.g., Google, Yahoo!, MSN), and color (black, red, blue, orange).

Ordinal Logistic Regression:

Used when there are three or more categories with a natural ordering to the levels, but the ranking of the levels do not necessarily mean the intervals between them are equal. Examples of ordinal responses could be how students rate the effectiveness of a college course on a scale of 1-5, levels of flavors for hot wings, and medical condition (e.g., good, stable, serious, critical).

Particular issues with logistic regression include nonnormal error terms, nonconstant error variance, and constraints on the response function (i.e., the response is bounded between 0 and 1). We will investigate ways of dealing with these in the binary logistic regression setting here. There is some discussion of the nominal and ordinal logistic regression settings in Section 15.2.

1.2 Binary Logistic Regression

The **multiple binary logistic regression model** is the following:

$$\begin{aligned}\pi &= \frac{\exp(\beta_0+\beta_1x_1+\cdots+\beta_{p-1}x_{p-1})}{1+\exp(\beta_0+\beta_1x_1+\cdots+\beta_{p-1}x_{p-1})}\\ &= \frac{\exp(\mathbf{X}\beta)}{1+\exp(\mathbf{X}\beta)}\\ &= \frac{1}{1+\exp(-\mathbf{X}\beta)},\end{aligned}$$

where here π denotes a probability and *not* the irrational number 3.14. . . .

- π is the probability that an observation is in a specified category of the binary Y variable, generally called the "success probability".
- Notice that the model describes the *probability of an event* happening as a function of X variables. For instance, it might provide estimates of the probability that an older person has heart disease.
- With the logistic model, estimates of π from equations like the one above will always be between 0 and 1. The reasons are:
  - The numerator $\exp(\beta_0+\beta_1x_1+\cdots+\beta_{p-1}x_{p-1})$ must be positive, because it is a power of a positive value (e).
  - The denominator of the model is (1 + numerator), so the answer will always be less than 1.
- With one X variable, the theoretical model for π has an elongated "S" shape (or sigmoidal shape) with asymptotes at 0 and 1, although in sample estimates we may not see this "S" shape if the range of the X variable is limited.

For a sample of size n, the likelihood for a binary logistic regression is given by:

$$\begin{aligned}L(\beta;\mathbf{y},\mathbf{X}) &= \prod_{i=1}^{n}\pi_i^{y_i}(1-\pi_i)^{1-y_i}\\ &= \prod_{i=1}^{n}\left(\frac{\exp(\mathbf{X}_i\beta)}{1+\exp(\mathbf{X}_i\beta)}\right)^{y_i}\left(\frac{1}{1+\exp(\mathbf{X}_i\beta)}\right)^{1-y_i}\end{aligned}$$

This yields the log likelihood:

$$\begin{aligned}\ell(\beta) &= \sum_{i=1}^{n}[y_i\log(\pi_i)+(1-y_i)\log(1-\pi_i)]\\ &= \sum_{i=1}^{n}[y_i\mathbf{X}_i\beta-\log(1+\exp(\mathbf{X}_i\beta))].\end{aligned}$$

Maximizing the likelihood (or log likelihood) has no closed-form solution, so a technique like iteratively reweighted least squares is used to find an estimate of the regression coefficients, $\hat{\beta}$.

To illustrate, consider data published on $n = 27$ leukemia patients. The data (`leukemia_remission.txt`) has a response variable of whether leukemia remission occurred `remiss`, which is given by a 1.

The predictor variables are cellularity of the marrow clot section `cell`, smear differential percentage of blasts `smear`, percentage of absolute marrow leukemia cell infiltrate `infil`, percentage labeling index of the bone marrow leukemia cells `li`, absolute number of blasts in the peripheral blood `blast`, and the highest temperature prior to start of treatment `temp`.

The following gives the estimated logistic regression equation and associated significance tests from R:

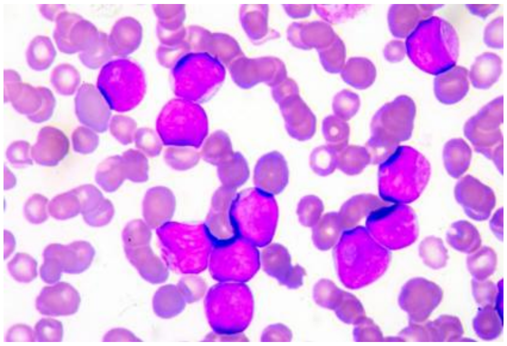


Figure 1: Leukemia cells.

```
> remission <- read.table("leukemia_remission.txt",sep="",skip = 1,header=T,row.names = 1)
> lrmodel.full <- glm(remiss ~ cell + smear + infil + li + blast + temp,
+                              family = binomial(link = "logit"), data = remission)
> summary(lrmodel.full)
```

```
Call:
glm(formula = remiss ~ cell + smear + infil + li + blast + temp,
    family = binomial(link = "logit"), data = remission)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.95165  -0.66491  -0.04372   0.74304   1.67069

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  58.0385    71.2364   0.815   0.4152
cell         24.6615    47.8377   0.516   0.6062
smear        19.2936    57.9500   0.333   0.7392
infil       -19.6013    61.6815  -0.318   0.7507
li            3.8960     2.3371   1.667   0.0955 .
blast         0.1511     2.2786   0.066   0.9471
temp        -87.4339    67.5735  -1.294   0.1957
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 34.372  on 26  degrees of freedom
Residual deviance: 21.751  on 20  degrees of freedom
AIC: 35.751

Number of Fisher Scoring iterations: 8
```

1.3 Wald Test

The **Wald test** is the test of significance for individual regression coefficients in logistic regression (recall that we use *t*-tests in linear regression). For maximum likelihood estimates, the ratio

$$Z = \frac{\hat{\beta}_i}{\text{s.e.}(\hat{\beta}_i)}$$

can be used to test $H_0 : \beta_i = 0$. The standard normal curve is used to determine the *p*-value of the test. Furthermore, confidence intervals can be constructed as

$$\hat{\beta}_i \pm z_{1-\alpha/2}\,\text{s.e.}(\hat{\beta}_i).$$

```
> confint(lrmodel.full)
```

```
                   2.5 %     97.5 %
(Intercept)  -70.9683777 222.202990
cell         -27.7332544 138.404531
smear        -60.4544868 152.174139
infil       -159.7565104  67.536927
li             0.1944541   9.526820
blast         -4.5238625   4.715064
temp        -244.7720744  24.913187
```

Estimates of the regression coefficients, $\hat{\beta}$, are given in the R output table in the column labeled “Estimate”. This table also gives coefficient *p*-values based on Wald tests. The index of the bone marrow leukemia cells `li` has the smallest *p*-value and so appears to be closest to a significant predictor of remission occurring. After looking at various subsets of the data,

```
> (lrmodel.step <- step(lrmodel.full))
```

```
Start:  AIC=35.75
remiss ~ cell + smear + infil + li + blast + temp

        Df Deviance    AIC
- blast  1   21.755 33.755
- infil  1   21.857 33.857
- smear  1   21.869 33.869
- cell   1   22.071 34.071
<none>       21.751 35.751
- temp   1   23.877 35.877
- li     1   26.095 38.095

Step:  AIC=33.76
remiss ~ cell + smear + infil + li + temp

        Df Deviance    AIC
- infil  1   21.858 31.858
- smear  1   21.869 31.869
- cell   1   22.073 32.073
<none>       21.755 33.755
- temp   1   24.198 34.199
- li     1   30.216 40.216

Step:  AIC=31.86
remiss ~ cell + smear + li + temp
```

```
        Df Deviance    AIC
- smear  1   21.953 29.953
<none>       21.858 31.858
- temp   1   24.292 32.292
- cell   1   24.477 32.477
- li     1   30.434 38.434

Step:  AIC=29.95
remiss ~ cell + li + temp

       Df Deviance    AIC
<none>      21.953 29.953
- temp  1   24.341 30.341
- cell  1   24.648 30.648
- li    1   30.829 36.829


Call:  glm(formula = remiss ~ cell + li + temp, family = binomial(link = "logit"), 
    data = remission)

Coefficients:
(Intercept)         cell           li         temp  
     67.634        9.652        3.867      -82.074  

Degrees of Freedom: 26 Total (i.e. Null);  23 Residual
Null Deviance:      34.37 
Residual Deviance: 21.95    AIC: 29.95
```

```
> summary(lrmodel.step)
```

```

Call:
glm(formula = remiss ~ cell + li + temp, family = binomial(link = "logit"), 
    data = remission)

Deviance Residuals: 
     Min        1Q    Median        3Q       Max  
-2.02043  -0.66313  -0.08323   0.81282   1.65887  

Coefficients:
            Estimate Std. Error z value Pr(>|z|)  
(Intercept)   67.634     56.888   1.189   0.2345  
cell           9.652      7.751   1.245   0.2130  
li             3.867      1.778   2.175   0.0297 *
temp         -82.074     61.712  -1.330   0.1835  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 34.372  on 26  degrees of freedom
Residual deviance: 21.953  on 23  degrees of freedom
AIC: 29.953
```

```
Number of Fisher Scoring iterations: 7
```

we find that a good model is one which only includes the labeling index as a predictor:

```
> lrmodel.reduced <- glm(remiss ~ li, family = binomial(link = "logit"), data = remission)
> summary(lrmodel.reduced)
```

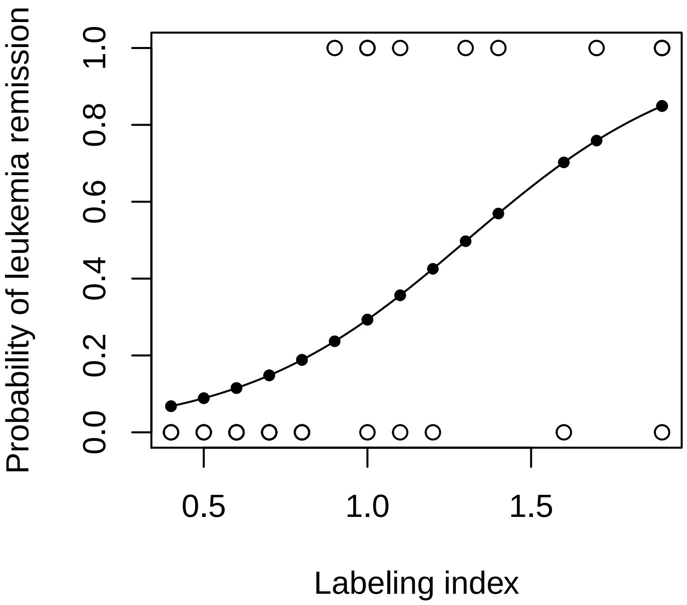
```

Call:
glm(formula = remiss ~ li, family = binomial(link = "logit"),
    data = remission)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.9448  -0.6465  -0.4947   0.6571   1.6971

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   -3.777      1.379  -2.740  0.00615 **
li             2.897      1.187   2.441  0.01464 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 34.372  on 26  degrees of freedom
Residual deviance: 26.073  on 25  degrees of freedom
AIC: 30.073

Number of Fisher Scoring iterations: 4
```

Regression equation:

$$P(1) = \exp(Y')/(1 + \exp(Y'))$$
$$Y' = -3.78 + 2.90 \cdot \texttt{li}$$

Since we only have a single predictor in this model we can create a Binary Fitted Line Plot to visualize the sigmoidal shape of the fitted logistic regression curve:

```
> attach(remission)
> plot(li,remiss,xlab="Labeling index",ylab="Probability of leukemia remission")
> curve(predict(lrmodel.reduced,data.frame(li=x),type="resp"),lty=1.5,add=TRUE)
> points(li,fitted(lrmodel.reduced),pch=20) # opcional
```



1.4 Odds, Log Odds, and Odds Ratio

There are algebraically equivalent ways to write the logistic regression model:

The first is

$$\frac{\pi}{1-\pi} = \exp(\beta_0 + \beta_1 x_1 + \cdots + \beta_{p-1} x_{p-1})$$

which is an equation that describes the odds of being in the current category of interest. By definition, the odds for an event is $P/(1-P)$ such that P is the probability of the event. For example, if you are at the racetrack and there is a 80% chance that a certain horse will win the race, then his odds are $0.80/(1-0.80) = 4$, or $4:1$.

The second is

$$\log\left(\frac{\pi}{1-\pi}\right) = \beta_0 + \beta_1 x_1 + \cdots + \beta_{p-1} x_{p-1},$$

which states that the (natural) logarithm of the odds is a linear function of the X variables (and is often called the **log odds**). This is also referred to as the **logit transformation** of the probability of success, π.

The **odds ratio** (which we will write as θ) between the odds for two sets of predictors (say $\mathbf{X}_{(1)}$ and $\mathbf{X}_{(2)}$) is given by

$$\theta = \frac{(\pi/(1-\pi))|_{\mathbf{X}=\mathbf{X}_{(1)}}}{(\pi/(1-\pi))|_{\mathbf{X}=\mathbf{X}_{(2)}}}$$

For binary logistic regression, the odds of success are:

$$\frac{\pi}{1-\pi} = \exp(\mathbf{X}\beta).$$

By plugging this into the formula for θ above and setting $\mathbf{X}_{(1)}$ equal to$\mathbf{X}_{(2)}$ except in one position (i.e., only one predictor differs by one unit), we can determine the relationship between that predictor and the response. The odds ratio can be any nonnegative number. An odds ratio of 1 serves as the baseline for comparison and indicates there is no association between the response and predictor. If the odds ratio is greater than 1, then the odds of success are higher for higher levels of a continuous predictor (or for the indicated level of a factor). In particular, the odds increase multiplicatively by $\exp(\beta_j)$ for every one-unit increase in $\mathbf{X}_j$. If the odds ratio is less than 1, then the odds of success are less for higher levels of a continuous predictor (or for the indicated level of a factor). Values farther from 1 represent stronger degrees of association.

For example, when there is just a single predictor, X, the odds of success are:

$$\frac{\pi}{1-\pi} = \exp(\beta_0 + \beta_1 X)$$

If we increase X by one unit, the odds ratio is

$$\theta = \frac{\exp(\beta_0 + \beta_1(X+1))}{\exp(\beta_0 + \beta_1 X)} = \exp(\beta_1).$$

The odds ratio for `li` is calculated as $\exp(2.89726)$.

```
> exp(summary(lrmodel.reduced)$coefficients["li",1])
```

```
[1] 18.12449
```

The 95% confidence interval is calculated as $exp(2.89726 \pm z_{0.975} * 1.19)$, where $z_{0.975} = 1.960$ is the 97.5^{th} percentile from the standard normal distribution.

```
> exp(summary(lrmodel.reduced)$coefficients["li",1] +
+      qnorm(c(0.025,0.5,0.975)) * summary(lrmodel.reduced)$coefficients["li",2])
```

```
[1]   1.770284  18.124486 185.561725
```

The interpretation of the odds ratio is that for every increase of 1 unit in `li`, the estimated odds of leukemia reoccurring are multiplied by 18.1245. However, since the `li` appears to fall between 0 and 2, it may make more sense to say that for every .1 unit increase in `li`, the estimated odds of remission are multiplied by $\exp(2.89726 \times 0.1) = 1.336$. Then

- At $\texttt{li} = 0.9$, the estimated odds of leukemia reoccurring is $\exp(-3.77714 + 2.89726 * 0.9) = 0.310$.
- At $\texttt{li} = 0.8$, the estimated odds of leukemia reoccurring is $\exp(-3.77714 + 2.89726 * 0.8) = 0.232$.
- The resulting odds ratio is $\frac{0.310}{0.232} = 1.336$, which is the ratio of the odds of remission when $\texttt{li} = 0.9$ compared to the odds when $\texttt{li} = 0.8$.

Notice that $1.336 \times 0.232 = 0.310$, which demonstrates the multiplicative effect by $\exp(0.1\hat{\beta}_1)$ on the odds.

1.5 Likelihood Ratio (or Deviance) Test

The **likelihood ratio** test is used to test the null hypothesis that any subset of the β's is equal to 0. Suppose we test that $r < p$ of the β's are equal to 0. Then the likelihood ratio test statistic is given by:

$$\Lambda^* = -2(\ell(\hat{\beta}^{(0)}) - \ell(\hat{\beta})),$$

where $\ell(\hat{\beta})$ is the log likelihood of the fitted (full) model and $\ell(\hat{\beta}^{(0)})$ is the log likelihood of the (reduced) model specified by the null hypothesis evaluated at the maximum likelihood estimate of that reduced model. This test statistic has a χ^2 distribution with $p - r$ degrees of freedom. R presents results for this test in terms of "deviance", which is defined as $-2\times$ log-likelihood. The notation used for the test statistic is typically $G^2 =$ deviance (reduced) – deviance (full).

This test procedure is analagous to the general linear F test procedure for multiple linear regression. However, note that when testing a single coefficient, the Wald test and likelihood ratio test will *not* in general give identical results.

To illustrate, the relevant R input and output from the leukemia example is:

```
> lrmodel.null <- glm(remiss ~ 1, family = binomial(link = "logit"), data = remission)
> model.reduced <- lrmodel.null  # constant only
> model.full <- lrmodel.reduced  # only 'li' as predictor
> anova(model.reduced, model.full, test = "Chisq")
```

Resid. Df	Resid. Dev	Df	Deviance	Pr(>Chi)
26	34.37177	NA	NA	NA
25	26.07296	1	8.298801	0.0039671

```
> model.reduced$deviance-model.full$deviance
```

```
[1] 8.298801
```

Since there is only a single predictor for this example, this table simply provides information on the likelihood ratio test for `li` model (p-value of 0.004), which is similar but not identical to the earlier Wald test result (p-value of 0.015). The Deviance Table includes the following:

- The null (reduced) model in this case has no predictors, so the fitted probabilities are simply the sample proportion of successes, $9/27 = 0.333333$. The log-likelihood for the null model is $\ell(\hat{\beta}^{(0)}) = -17.1859$, so the deviance for the null model is $-2 \times -17.1859 = 34.372$, which is shown in the row 1 in the Deviance Table.
- The log-likelihood for the fitted (full) model is $\ell(\hat{\beta}) = -13.0365$, so the deviance for the fitted model is $-2 \times -13.0365 = 26.073$, which is shown in the row 2 in the Deviance Table.
- The likelihood ratio test statistic is therefore $\Lambda^* = -2(-17.1859 - (-13.0365)) = 8.299$, which is the same as $G^2 = 34.372 - 26.073 = 8.299$.
- The p-value comes from a χ^2 distribution with $2 - 1 = 1$ degrees of freedom.

When using the likelihood ratio (or deviance) test for more than one regression coefficient, we can first fit the "full" model to find deviance (full), which is shown in the row 2 in the resulting full model Deviance Table. Then fit the "reduced" model (corresponding to the model that results if the null hypothesis is true) to find deviance (reduced), which is shown in the row 2 in the resulting reduced model Deviance Table. For example, the relevant Deviance Tables for the Disease Outbreak example on pages 581-582 of Applied Linear Regression Models (4th ed) by Kutner et al are:

```
> DiseaseOutbreak <- read.csv("DiseaseOutbreak.txt", row.names=1, sep="")
> DiseaseOutbreak$socio <- factor(DiseaseOutbreak$socio, labels = c("Upper","Middle","Lower"))
> DiseaseOutbreak$sector <- factor(DiseaseOutbreak$sector)
```

Full model:

```
> model.full <- glm(disease ~ age + socio + sector + age:socio + age:sector + socio:sector,
+                                   family = binomial(link = "logit"), data = DiseaseOutbreak)
> model.null <- glm(disease ~ 1, family = binomial(link = "logit"), data = DiseaseOutbreak)
> anova(model.null,model.full,test = "Chisq")
```

Resid. Df	Resid. Dev	Df	Deviance	Pr(>Chi)
97	122.31761	NA	NA	NA
88	93.99587	9	28.32173	0.0008426

Reduced model:

```
> model.reduced <- glm(disease ~ age + socio + sector,
+                      family = binomial(link = "logit"), data = DiseaseOutbreak)
> anova(model.null,model.reduced,test="Chisq")
```

Resid. Df	Resid. Dev	Df	Deviance	Pr(>Chi)
97	122.3176	NA	NA	NA
93	101.0542	4	21.26346	0.0002808

Here the full model includes three single-factor predictor terms and three two-factor interaction terms, while the reduced model excludes the interaction terms. The test statistic for testing the interaction terms is $G^2 = 101.054 - 93.996 = 7.058$, which is compared to a chi-square distribution with $10 - 5 = 5$ degrees of freedom to find the p-value > 0.05 (meaning the interaction terms are not significant).

```
> anova(model.reduced,model.full,test="Chisq")
```

Resid. Df	Resid. Dev	Df	Deviance	Pr(>Chi)
93	101.05415	NA	NA	NA
88	93.99587	5	7.058277	0.2163422

Alternatively, select the corresponding predictor terms last in the full model and request R to output Sequential (Type I) Deviances. Then add the corresponding Sequential Deviances in the resulting Deviance Table to calculate G^2. For example, the relevant Deviance Table for the Disease Outbreak example is:

```
> anova(model.full,test="Chisq")
```

	Df	Deviance	Resid. Df	Resid. Dev	Pr(>Chi)
NULL	NA	NA	97	122.31761	NA
age	1	7.404971	96	114.91264	0.0065044
socio	2	3.410362	94	111.50227	0.1817395
sector	1	10.448124	93	101.05415	0.0012277
age:socio	2	5.584876	91	95.46927	0.0612716
age:sector	1	1.120189	90	94.34908	0.2898777
socio:sector	2	0.353211	88	93.99587	0.8381104

The test statistic for testing the interaction terms is $G^2 = 5.5849 + 1.1202 + 0.3532 = 7.0583$, the same as in the first calculation.

1.6 Goodness-of-Fit Tests

Overall performance of the fitted model can be measured by several different goodness-of-fit tests. Two tests that require replicated data (multiple observations with the same values for all the predictors) are the **Pearson chi-square goodness-of-fit test** and the **deviance goodness-of-fit test** (analagous to the multiple linear regression lack-of-fit F-test). Both of these tests have statistics that are approximately chi-square distributed with $c - p$ degrees of freedom, where c is the number of distinct combinations of the predictor variables. When a test is rejected, there is a statistically significant lack of fit. Otherwise, there is no evidence of lack of fit.

By contrast, the **Hosmer-Lemeshow goodness-of-fit test** is useful for unreplicated datasets or for datasets that contain just a few replicated observations. For this test the observations are grouped based on their estimated probabilities. The resulting test statistic is approximately chi-square distributed with $c - 2$ degrees of freedom, where c is the number of groups (generally chosen to be between 5 and 10, depending on the sample size).

To illustrate, the relevant R input and output from the leukemia example is:

```
> library(ResourceSelection)
> hoslem.test(remission$remiss, fitted(lrmodel.reduced), g=9)
```

```

	Hosmer and Lemeshow goodness of fit (GOF) test
```

```
data:  remission$remiss, fitted(lrmodel.reduced)
X-squared = 7.3293, df = 7, p-value = 0.3954
```

Since there is no replicated data for this example, the deviance and Pearson goodness-of-fit tests are invalid. However, the Hosmer-Lemeshow test does not require replicated data so we can interpret its high p-value as indicating no evidence of lack-of-fit.

1.7 R^2

The calculation of R^2 used in linear regression does not extend directly to logistic regression. One version of R^2 used in logistic regression is defined as

$$R^2 = \frac{\ell(\hat{\beta}_0) - \ell(\hat{\beta})}{\ell(\hat{\beta}_0) - \ell_S(\beta)},$$

where $\ell(\hat{\beta}_0)$ is the log likelihood of the model when only the intercept is included and $\ell_S(\beta)$ is the log likelihood of the saturated model (i.e., where a model is fit perfectly to the data). This R^2 does go from 0 to 1 with 1 being a perfect fit. With unreplicated data, $\ell_S(\beta) = 0$, so the formula simplifies to:

$$R^2 = \frac{\ell(\hat{\beta}_0) - \ell(\hat{\beta})}{\ell(\hat{\beta}_0)} = 1 - \frac{\ell(\hat{\beta})}{\ell(\hat{\beta}_0)}.$$

To illustrate, the relevant R input and output from the leukemia example is:

```
> library(pscl)
> pR2(lrmodel.reduced) # look for "McFadden"
```

```
        llh     llhNull          G2    McFadden        r2ML        r2CU 
-13.0364823 -17.1858825   8.2988006   0.2414424   0.2646164   0.3675138 
```

Recall from above that $\ell(\hat{\beta}) = -13.0365$ and $\ell(\hat{\beta}_0) = -17.1859$, so:

$$R^2 = 1 - \frac{-13.0365}{-17.1859} = 0.2414.$$

Note that we can obtain the same result by simply using deviances instead of log-likelihoods since the -2 factor cancels out:

$$R^2 = 1 - \frac{26.073}{34.372} = 0.2414.$$

```
> (R2 <- 1-lrmodel.reduced$deviance/lrmodel.null$deviance)
```

```
[1] 0.2414424
```

1.8 Residuals

1.8.1 Raw Residuals

The **raw residual** is the difference between the actual response and the estimated probability from the model. The formula for the raw residual is

$$r_i = y_i - \hat{\pi}_i.$$

1.8.2 Pearson Residuals

The **Pearson residual** corrects for the unequal variance in the raw residuals by dividing by the standard deviation. The formula for the Pearson residuals is

$$p_i = \frac{r_i}{\sqrt{\hat{\pi}_i(1-\hat{\pi}_i)}}$$

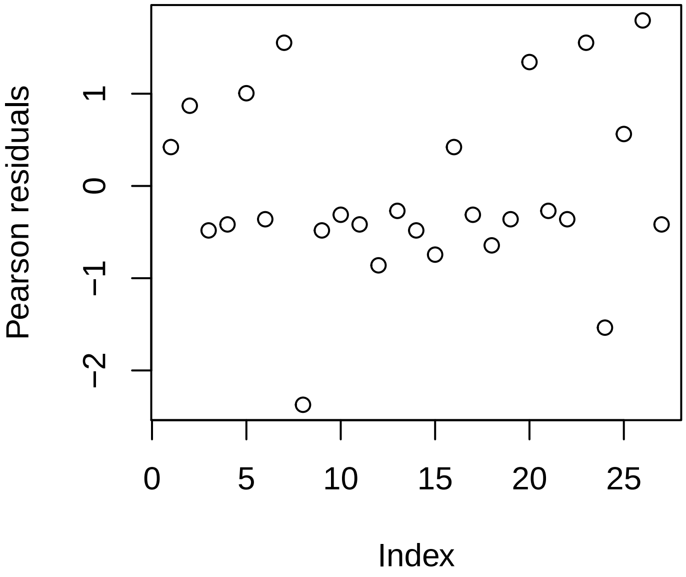
1.8.3 Deviance Residuals

Deviance residuals are also popular because the sum of squares of these residuals is the deviance statistic. The formula for the deviance residual is

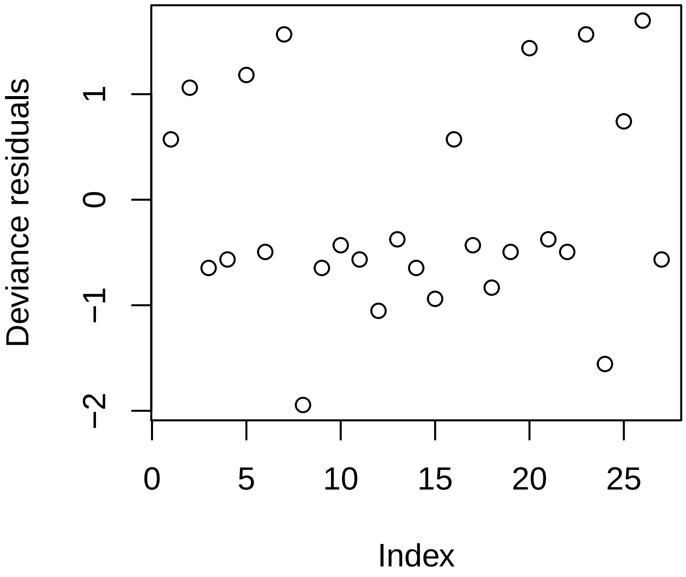
$$d_i = \pm\sqrt{2\left[y_i \log\left(\frac{y_i}{\hat{\pi}_i}\right) + (1-y_i)\log\left(\frac{1-y_i}{1-\hat{\pi}_i}\right)\right]}.$$

Here are the plots of the Pearson residuals and deviance residuals for the leukemia example.

```
> plot(residuals(lrmodel.reduced, type = "pearson"), ylab="Pearson residuals")
```



```
> plot(residuals(lrmodel.reduced, type = "deviance"), ylab="Deviance residuals")
```



There are no alarming patterns in these plots to suggest a major problem with the model.

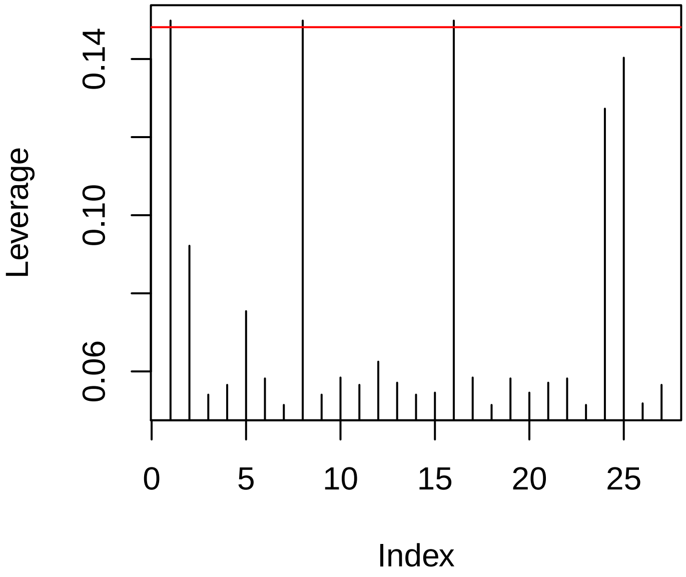
1.9 Hat Values

The hat matrix serves a similar purpose as in the case of linear regression – to measure the influence of each observation on the overall fit of the model – but the interpretation is not as clear due to its more complicated form. The hat values (leverages) are given by

$$h_{ii} = \hat{\pi}_i(1 - \hat{\pi}_i)\mathbf{x}_i'(\mathbf{X}'\mathbf{W}\mathbf{X})\mathbf{x}_i$$

where $\mathbf{W}$ is an $n \times n$ diagonal matrix with the values of $\hat{\pi}_i(1 - \hat{\pi}_i)$ for $i = 1, \ldots, n$ on the diagonal. As before, we should investigate any observations with $h_{ii} > 3p/n$ or, failing this, any observations with $h_{ii} > 2p/n$ *and very isolated*.

```
> library(boot)
> leverage <- glm.diag(lrmodel.reduced)$h
> plot(leverage, type="h", ylab="Leverage")
> abline(h=2*2/length(leverage),col="red")
```



1.10 Studentized Residuals

We can also report Studentized versions of some of the earlier residuals. The **Studentized Pearson residuals** are given by

$$sp_i = \frac{p_i}{\sqrt{1 - h_{ii}}}$$

and the **Studentized deviance residuals** are given by

$$sd_i = \frac{d_i}{\sqrt{1 - h_{ii}}}$$

1.11 C and $\bar{C}$

C and $\bar{C}$ are extensions of Cook's distance for logistic regression. C measures the overall change in fitted logits due to deleting the ith observation for all points including the one deleted, while $\bar{C}$ excludes the deleted point. They are defined by:

$$C_i = \frac{p_i^2 h_{ii}}{p(1 - h_{ii})^2}$$

and

$$\bar{C}_i = \frac{p_i^2 h_{ii}}{p(1-h_{ii})}$$

```
> pr <- residuals(lrmodel.reduced, type = "pearson")
> head(pr^2 * leverage /(2*(1-leverage)^2))
```

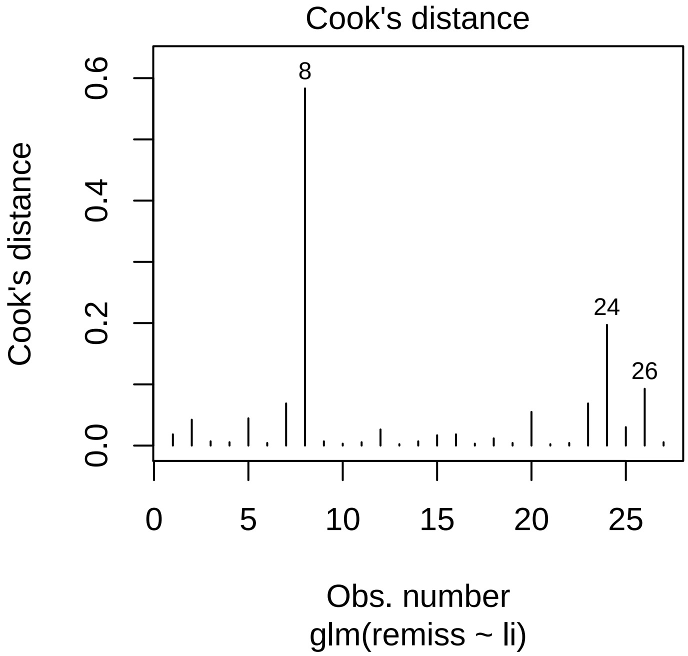
```
          1           2           3           4           5           6 
0.018419629 0.042306509 0.007019087 0.005525802 0.044593718 0.004271574 
```

```
> head(glm.diag(lrmodel.reduced)$cook)
```

```
          1           2           3           4           5           6 
0.018419629 0.042306509 0.007019087 0.005525802 0.044593718 0.004271574 
```

To illustrate, the relevant R input and output from the leukemia example is:

```
> plot(lrmodel.reduced, which=4)
```



```
> glmdiag <- glm.diag(lrmodel.reduced)
> glmdiag$res[8]   # jackknife deviance residual
```

```
        8 
-2.185013 
```

```
> glmdiag$rd[8]    # standardized deviance residual
```

```
        8 
-2.109289 
```

```
> glmdiag$rp[8]    # standardized Pearson residual
```

```
        8 
-2.572802 
```

```
> glmdiag$cook[8] # approximate Cook statistic
```

```
        8 
```

```
0.5833219
```

```
> glmdiag$h[8]    # leverage of the observation
```

```
[1] 0.1498395
```

The default residuals in this output are jackknife deviance residuals, so observation 8 has a deviance residual of -2.185, a standardized deviance residual of -2.11, a leverage (h) of 0.14984, and a Cook's distance (C) of 0.58.

1.12 DFDEV and DFCHI

DFDEV and **DFCHI** are statistics that measure the change in deviance and in Pearson's chi-square, respectively, that occurs when an observation is deleted from the data set. Large values of these statistics indicate observations that have not been fitted well. The formulas for these statistics are

$$\mathrm{DFDEV}_i = d_i^2 + \bar{C}_i$$

and

$$\mathrm{DFCHI}_i = \frac{\bar{C}_i}{h_{ii}}$$