SS9055B: Generalized Linear Models

Section 5: Inference for Generalized Linear Models

Objectives

The objectives of this lecture are to introduce methods of conducting hypothesis tests and computing confidence intervals for the regression coefficients of a GLM. By the end of this lecture you should be able to:

- derive Wald test statistics and confidence intervals compute them in `R`,
- compute and explain the likelihood ratio test statistic,
- define the deviance and explain its uses, and
- conduct likelihood ratio tests and compute likelihood ratio based confidence intervals in `R`.

Introduction

In the previous reading, we defined the structure of a GLM, computed the likelihood equations from which point estimates of the regression coefficients can be computed, constructed the Fisher information matrix which is needed to approximate sampling distribution of the regression coefficients, and derived the iteratively (re)weighted least squares algorithm from which the MLEs can be obtained. We will now consider the remaining points of inference including constructing confidence intervals and model comparison. Throughout this section we will assume that the dispersion parameter, ϕ, is both fixed and known.

Example

Once again, I will consider the linear regression model as our first example of the methods of generalized linear models. In particular, I will model the `mtcars` data that we analysed in the very first reading. As a reminder, the data contain observations of 11 variables on 32 cars that were extracted from motor trend magazine in 1974. We will focus on modeling the efficiency of the engine in miles per gallon (`mpg`) as a function of the engine displacement in cubic inches (`disp`), a measure of the engine's size. The data are plotted in Figure 1.

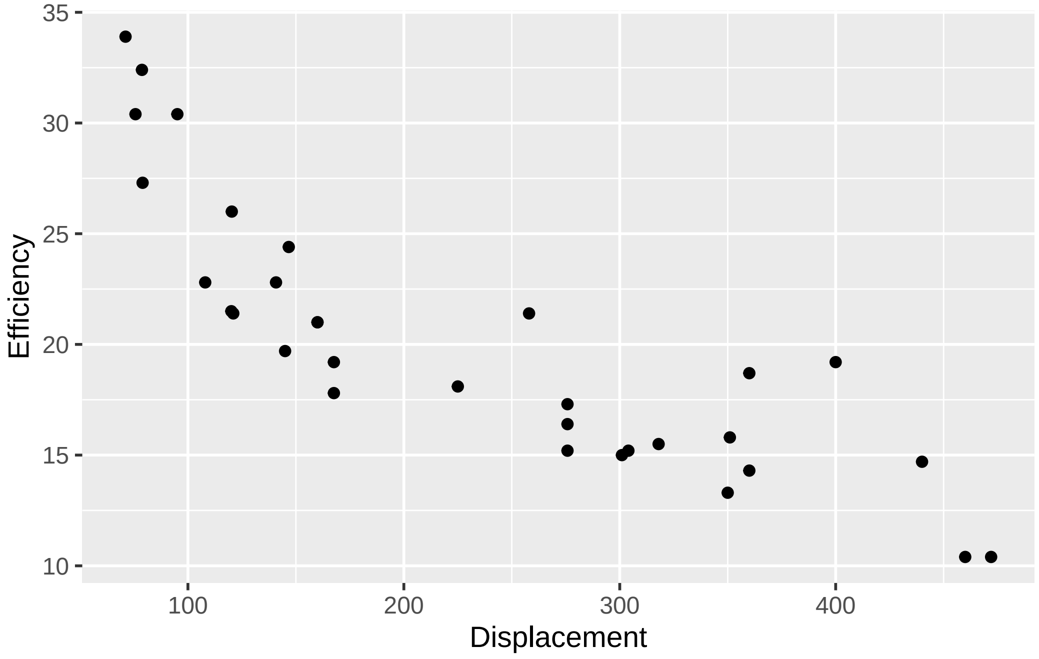

Figure 1: Efficiency of the engines of the 32 cars in the `mtcars` data set in miles per cubic inch as a function of the engine displacement in cubic inches.

Note that the relationship between the displacement and efficiency appears to be slightly curved and so we will work with the log of the efficiency as our response instead. I will also rescale the displacement to 100s of cubic inches in order that the results are on a reasonable scale. The following steps can be used to fit the linear regression model and produce basic results:

1. Fitting the model with `lm()`:

```r
mtcars <- mtcars |>
  mutate(disp2 = disp/100)

lm1 <- lm(log(mpg) ~ disp2,data=mtcars)
summary(lm1)
```

```
Call:
lm(formula = log(mpg) ~ disp2, data = mtcars)

Residuals:
     Min       1Q   Median       3Q      Max
-0.21183 -0.10837 -0.04732  0.08251  0.35546

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  3.44555    0.05429   63.47  < 2e-16 ***
disp2       -0.21152    0.02080  -10.17  3.1e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1435 on 30 degrees of freedom
Multiple R-squared:  0.7751,	Adjusted R-squared:  0.7676 
F-statistic: 103.4 on 1 and 30 DF,  p-value: 3.095e-11
```

2. Constructing confidence intervals for the regression coefficients:

```r
confint(lm1)
```

```
                 2.5 %     97.5 %
(Intercept)  3.3346736  3.5564224
disp2       -0.2540076 -0.1690416
```

3. Constructing the analysis of variance table:

```r
anova(lm1)
```

```
Analysis of Variance Table

Response: log(mpg)
          Df  Sum Sq Mean Sq F value    Pr(>F)    
disp2      1 2.13058 2.13058   103.4 3.095e-11 ***
Residuals 30 0.61816 0.02061                      
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

4. Comparing to intercept only model with AIC:

```r
lm0 <- lm(log(mpg) ~ 1, data=mtcars)
AIC(lm0,lm1)
```

```
    df       AIC
lm0  2  16.26504
lm1  3 -29.48383
```

The results indicate that there is an important association between the displacement of an engine and the log efficiency. This is shown by the t-statistic of -10.2 on 30 degrees of freedom which gives a p-value $< .001$. Note that the F-statistic provides an equivalent test in this case because there is only one predictor in the model. The F-value is 103.4 on 1 and 30 degrees of freedom which again gives the p-value of $< .001$. Alternatively, the difference in AIC for the model including displacement and the intercept only model is 45.8. All of these provide very strong evidence that the log of efficiency is associated with size of the engine. The estimated slope is $-.21$ with 95% CI (-.25,-.17). Transforming this back to the natural scale, we would predict that the relative efficiency of two engines differing in displacement by 100 cubic inches is $e^{-.21} = .809\%$ (95% CI=.776,.845). I.e., we lose about 19% of the efficiency for every 100 cubic inch increase in displacement.

We will now repeat the analysis treating the linear regression model as a GLM and refitting the model in `R`. The main function for fitting GLM in `R`, which we will use over and over again for the rest of this semester, is `glm()` (no surprises there). The format of the function is similar to `lm()`, except that it requires a further argument specifying the random component (the distribution of the data about the mean) and, optionally, a link function. Each family has a default link function. This is usually the canonical link function and will be used if you don't specify otherwise. The default for the normal (Gaussian) family is the identity link, the default for the binomial is logit, and the default for the Poisson is log. Since we are assuming that the dispersion parameter is known I will set $\sigma = .144$, the estimate produced by `lm()`. If we don't do this then `R` is smart enough to use exact inference for the Gaussian model instead of approximate inference based on the approximate distribution of the MLE and will exactly reproduce the output from `lm()`. I don't want `R` to do this because I want to show the differences between the small sample approach implemented in `lm()` and the asymptotic inference presented by `glm()`. Note that the dispersion parameter does not need to be set when fitting the model, since the MLEs do not depend on its value. Instead, we need to be set its value when we construct the summary. The dispersion parameter for a Gaussian GLM is actually the variance, not the standard deviation, and so we need to use the value $.144^2$:

```r
glm1 <- glm(log(mpg) ~ disp2,data=mtcars,family=gaussian())
summary(glm1, dispersion=.144^2)
```

```
Call:
glm(formula = log(mpg) ~ disp2, family = gaussian(), data = mtcars)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
```

```
(Intercept)  3.44555    0.05446   63.27   <2e-16 ***
disp2       -0.21152    0.02087  -10.14   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for gaussian family taken to be 0.020736)

    Null deviance: 2.74874  on 31  degrees of freedom
Residual deviance: 0.61816  on 30  degrees of freedom
AIC: -29.484

Number of Fisher Scoring iterations: 2
```

The output looks similar to the output from the linear regression model in many ways, but there are some subtle differences.

Model Comparison

Wald Tests

The first difference is that the labels for the columns containing the test statistic and corresponding p-value have changed from `t value` and `Pr(>|t|)` to `z value` and `Pr(>|z|)` This is because inference is now based on the Wald tests which depend on the asymptotic normal distribution of the MLEs rather than the exact t-distribution which is corrected for having to estimate σ^2. This has also changed the entries in the table. The z-statistics are slightly closer to zero, and p-value for testing the significance of the slope decreased (though it went from very highly important to even more important).

Likelihood Ratio Tests

Next we will look at likelihood ratio tests for GLM. However, we are going to have to do a little more work first to understand the output in `R`. Suppose that we have fit a generalized linear model that follows the form given in the previous section of the notes. Suppose that the model includes p predictor variables and we wish to test the hypothesis that k of these values are equal to some fixed values. As in Section 2 of the reading on Maximum Likelihood Inference, we will divide $\boldsymbol{\beta}$ into two parts $\boldsymbol{\beta} = (\boldsymbol{\beta}_0', \boldsymbol{\beta}_1)'$ such that the null and alternative hypotheses are written as

$$H_0 : \boldsymbol{\beta}_0 = \mathbf{0} \text{ versus } H_a : \boldsymbol{\beta}_0 \neq \mathbf{0}.$$

The likelihood ratio statistic is:

$$\Lambda = \frac{\max_{\boldsymbol{\beta}_1} L(\mathbf{0}, \boldsymbol{\beta}_1, \phi|\boldsymbol{y})}{\max_{\boldsymbol{\beta}_0, \boldsymbol{\beta}_1} L(\boldsymbol{\beta}_0, \boldsymbol{\beta}_1, \phi|\boldsymbol{y})} = \frac{L(\boldsymbol{\beta}_0^*, \hat{\boldsymbol{\beta}}_1, \phi|\boldsymbol{y})}{L(\hat{\boldsymbol{\beta}}, \phi|\boldsymbol{y})}.$$

where $\hat{\boldsymbol{\beta}}$ is the (unrestricted) maximum likelihood estimator and $\hat{\boldsymbol{\beta}}_1$ is the value of $\boldsymbol{\beta}_1$ that maximizes the likelihood when $\boldsymbol{\beta}_0 = \mathbf{0}$. Under the usual regularity conditions (which are satisfied by any GLM) $-2\log\Lambda \xrightarrow{\mathcal{D}} \chi^2_k$ where the degrees of freedom is determined the difference in the dimension of the parameter space between the two hypotheses, k.

Now consider the form of the likelihood ratio test statistic for a GLM. From the previous notes we know that the density of Y_i has the form

$$f(y_i|\boldsymbol{\beta}, \phi) = \exp\left(\omega_i \left[\frac{y_i\theta_i - b(\theta_i)}{\phi}\right] + c(y_i, \phi)\right)$$

which implies that the likelihood is

$$L(\boldsymbol{\beta}, \phi|\boldsymbol{y}) = \exp\left[\sum_{i=1}^N \frac{\omega_i}{\phi}\left(y_i\theta_i - b(\theta_i)\right) + \sum_{i=1}^N c(y_i, \phi)\right].$$

If we let $\hat{\theta}_i$ and $\tilde{\theta}_i$ denote the estimates of the natural parameter for the i^{th} observation, θ_i, under the null and alternative hypotheses, respectively, then the likelihood ratio becomes

$$\begin{aligned}\Lambda &= \frac{\exp\left[\sum_{i=1}^N \frac{\omega_i}{\phi}\left(y_i\hat{\theta}_i - b(\hat{\theta}_i)\right) + \sum_{i=1}^N c(y_i, \phi)\right]}{\exp\left[\sum_{i=1}^N \frac{\omega_i}{\phi}\left(y_i\tilde{\theta}_i - b(\tilde{\theta}_i)\right) + \sum_{i=1}^N c(y_i, \phi)\right]} \\ &= \exp\left[\sum_{i=1}^N \frac{\omega_i}{\phi}\left(y_i\hat{\theta}_i - b(\hat{\theta}_i)\right) - \sum_{i=1}^N \frac{\omega_i}{\phi}\left(y_i\tilde{\theta}_i - b(\tilde{\theta}_i)\right)\right].\end{aligned}$$

Further

$$\begin{aligned}G^2 &= -2\left[\sum_{i=1}^N \frac{\omega_i}{\phi}(y_i\hat{\theta}_i - b(\hat{\theta}_i)) - \sum_{i=1}^N \frac{\omega_i}{\phi}(y_i\tilde{\theta}_i - b(\tilde{\theta}_i))\right] \\ &= 2\sum_{i=1}^N \frac{\omega_i}{\phi}[y_i(\tilde{\theta}_i - \hat{\theta}_i) - (b(\tilde{\theta}_i) - b(\hat{\theta}_i))].\end{aligned}$$

This may still look complicated, but consider what this means. For any model in the class of GLM all you need to conduct the likelihood ratio test are the values of the natural parameters for each observation under the null and alternative hypotheses. You can then plug these values into this formula and immediately compute the likelihood ratio test statistic.

The likelihood ratio test can be conducted in `R` with the function `anova()`, as in linear regression. However, the function can conduct several different tests for GLM (including the score test) and so we need to specify that we want the LRT. To test the significance of the slope in the example we can fit a new model including only the intercept term

```r
glm0 <- glm(log(mpg) ~ 1,data=mtcars,family=gaussian())
```

and compare the two models

```r
anova(glm0, glm1, dispersion = .144^2, test="LRT")
```

```
Analysis of Deviance Table

Model 1: log(mpg) ~ 1
Model 2: log(mpg) ~ disp2
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
1        31    2.74874
2        30    0.61816  1   2.1306 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Once again, we need to supply the value of the dispersion parameter that will be used in computing G^2[1] The p-value for the LRT testing the significance of displacement is very small and provides exactly the same conclusion as for the Wald test. Note that the equivalence is due to the fact that we are fitting a model with normal errors, and the two p-values will not always be equal. As we have mentioned before, the LRT is generally more powerful and so will produce smaller p-values on average.

Saturated Model and Deviance

A particularly important model is called the saturated model. Given a set of observed responses, $Y_1, \dots, Y_N$, the saturated model is the model that sets the expected means equal to the observed values, $\hat{\mu}_i = y_i$. This is a hypothetical model (we will never actually fit it to data) and can be constructed in many different ways. For example, we could create a model with $n-1$ covariates such that $x_{ij} = 1(i = j)$, $i = 1, \dots, n-1$. However this is done, the model will have the same number of parameters as there data points and will fit the data as closely as possible within the class of selected models. In the example, we could introduce indicator variables for each of the students so that the model had 16 parameters, a separate mean for each student, and the fitted means were exactly equal to the observed means. This model is not very useful because it overfits the data. In the case of a linear regression model with normal errors the residuals would all be zero, and we would learn nothing about the relationship in the mean melting times of the hard and caramel candies. However, it represents the best

[1] This is not actually necessary and the ‘anova()‘ function will use an estimate of the dispersion parameter based on the more complex model if this is not specified. However, I like to specify it so I know exactly what is happening.

possible fit to the data and so we can use it as the basis for testing the fit of another, restricted model that might provide useful results. The model in which $\hat{\mu}_i = y_i$ for all $i = 1, \ldots, N$ is called the saturated model.

Following the derivation of the LRT in the previous section we will let $\tilde{\theta}_i$ and $\hat{\theta}_i$ represent the fitted values of the natural parameter from the saturated model and the model of interest. The quantity:

$$D(\boldsymbol{y}, \hat{\boldsymbol{\mu}}) = 2\sum_{i=1}^{N} \omega_i[y_i(\tilde{\theta}_i - \hat{\theta}_i) - (b(\tilde{\theta}_i) - b(\hat{\theta}_i))]$$

which is equal to ϕ times the value of G^2 for comparing the saturated model and the model of interest is called the deviance of the model and $D(\boldsymbol{y}, \hat{\boldsymbol{\mu}})/\phi$ is called the scaled deviance. Since we are assuming that the dispersion parameter is fixed, the two models differ only in terms of the means for each observation. The notation $D(\boldsymbol{y}, \hat{\boldsymbol{\mu}})$ is intended to indicate that we are comparing the saturated model, for which the vector of fitted means is $\boldsymbol{y}$, with another, reduced model, for which the vector of fitted means is $\hat{\boldsymbol{\mu}}$.

The deviance (or more properly, the scaled deviance) serves two important purposes. First, the deviance for a model can be used to assess the fit of the model. If we can fit the data perfectly with the saturated model then why would we settle for a model that is significantly worse? This question can be answered by comparing a model of interest with the saturated model. Rejecting the null hypothesis implies that the model of interest is significantly worse than the saturated model which is taken as evidence that the restricted model does not fit the data sufficiently well. On the other hand, if we can't reject the null hypothesis then this implies that the restricted model is as good, and simpler than, the saturated model. This is called the deviance goodness-of-fit test.

The second result is that we can rewrite the likelihood ratio test between two models in terms of their respective scaled deviances. Suppose that we have two nested models such that the predictors in one model are a subset of the predictors in the other [2]. Suppose that the vectors of fitted means are $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ respectively. Then the likelihood ratio test statistic for comparing these models is simply the difference in deviances

$$G^2 = \frac{D(\boldsymbol{y}, \hat{\boldsymbol{\mu}}_1)}{\phi} - \frac{D(\boldsymbol{y}, \hat{\boldsymbol{\mu}}_2)}{\phi}.$$

Once again, under the null hypothesis (i.e., that the simper model is correct) the likelihood ratio test statistic has an approximate chi square distribution with degrees of freedom equal to the difference in the number of parameters between the two models.

The reason that this is important is that the deviance for a model can be computed very quickly, and it is automatically produced in the summary for a GLM in `R`. Nested models can then be compared easily simply by subtracting the deviances to compute the test statistic,

[2] More rigorously, one model is said to be nested in another if the columns of the design matrix for the first model are linear combinations of the columns in the design matrix of the second model.

computing the difference in the number of parameters to obtain the degrees of freedom for the chi-square tests, and obtaining the associated chi-square tail probability to compute the p-value. The deviance in the summary produced by `R` is called the `Residual Deviance`. By default, `R` also provides the deviance of the model with only an intercept, called the `Null Deviance`.

In the previous output, you can see that the `Residual Deviance` for the model including displacement is .6182. Still assuming that $\phi = \sigma^2$ is known and equal to .114, the scaled deviance is $.6812/.114^2 = 32.90$. The difference in the number of parameters in the saturated model and the model including displacement is 32-2=30. Hence, the p-value for the deviance goodness-of-fit test is $p = P(\chi^2_{30} > 32.90) = .327$. Based on this test there is no reason to reject the fit of the model including displacement. It would still be wise to look at residuals, but this test provides no reason to doubt the fit of the simple linear regression model.

Next we can compare the initial model and the intercept only model. The deviance of the intercept only model is 2.749. Subtracting the deviances and dividing by $\phi = \sigma^2 = .144^2$ to obtain the difference in the scaled deviances gives the test statistic value of $2.131/.144^2 = 103$[3]. The difference in the number of parameters is one, and so the p-value is $p = P(\chi^2_1 > 103) < .001$. This is very strong evidence that there is an effect of displacement on the efficiency of an engine. Of course, this was obvious from the plot and, in fact, we didn't even have to do this test separately because it simply reproduces the z-test for `disp2` given in the table of coefficients. However, exactly the same procedure can be applied to compare models differing by more than one degree of freedom (i.e., multiple continuous covariates or categorical predictors) in which case the test is not reported within the table of coefficients.

AIC

An alternative (and in my opinion, the better alterantive) to hypothesis testing is to compare models via the AIC or some other model selection criterion. As mentioned in the reading on Linear Regression, the AIC is not restricted to nested models and can be used to compare any pair of models. Moreover, it is very easy to compare the AIC for GLM. Suppose that we have two models with AIC values AIC_1 and AIC_2 and recall that all of the information for comparing these two models is contained in the difference, $\text{AIC}_1 - \text{AIC}_1$. The actual values of AIC_1 and AIC_2 are not important. If we let l_1 and l_2 denote the values of the log-likelihood for the two models and k_1 and k_2 then number of parameters then

$$\begin{aligned}\text{AIC}_1 - \text{AIC}_2 &= (-2L_1 + 2k_1) - (-2L_2 + 2k_2)\\ &= -2(L_1 - L_2) + 2(k_1 - k_2).\end{aligned}$$

[3] Note that the output from 'anova()' reports the difference in the deviances, not the difference in the scaled deviances which is needed to compute the LRT. However, the p-value is correct.

But, minus twice the difference in the log-likelihood is equal to the difference in the scaled deviance, exactly as in the calculation of the likelihood ratio test statistic, G^2. Hence

$$\text{AIC}_1 - \text{AIC}_2 = \frac{D(\boldsymbol{y}, \hat{\boldsymbol{\mu}}_1)}{\phi} - \frac{D(\boldsymbol{y}, \hat{\boldsymbol{\mu}}_2)}{\phi} + 2(k_1 - k_2).$$

In the example, the model including displacement has a deviance of .618 and 2 parameters (ignoring the dispersion/variance which we have assumed to be fixed and known). The intercept only model has a deviance of 2.749 (since this is the null model) and only 1 parameter. Hence, the difference in AIC is

$$\frac{.618}{.144^2} - \frac{2.749}{.144^2} + 2(2-1) = -100.768.$$

Once again, this provides very strong evidence to support the model that includes the displacement.

Note that in this case the difference AIC I have computed does not match the difference in the AIC values that are provided by the `summary` function or reported by the `AIC` function. The reason is that there is a bug in these functions. The AIC computed for the intercept only model does not make use of the fixed dispersion parameter, even if this is supplied as an argument. If you run the commands `summary(glm0)` and `summary(glm0, dispersion = .144^2)` then you will notice that the standard error of the estimated intercept, the value of the test statistic, and the reported value of the dispersion parameter all change, but the value of the AIC remains this same. The reason is that the AIC is actually computed when the model is fitted using the residual variance to estimate dispersion and is not adjusted for the value of the new dispersion parameter that is provided as an argument. This is incorrect. Thankfully, we will not have to worry about this for the logistic regression and Poisson log-linear models because the dispersion parameter for these models is always equal to 1 (for now at least).

Confidence Intervals

Wald Type Confidence Intervals

I don't know of a function to compute Wald type confidence intervals for the regression coefficients directly, but these can be computed quickly and easily with simple computations. All we need to do to construct a $(1-\alpha)100\%$ confidence interval is to add and subtract $z_{\alpha/2}$ times the standard error from the estimates. The 95% confidence intervals for the parameters are

```
## Extract standard errors
se <- glm1.summ$coeff[,"Std. Error"]

## Compute Wald type CI
glm1$coeff + 1.96*outer(se,c(-1,1))
```

```
                  [,1]       [,2]
(Intercept)  3.3388031  3.5522929
disp2       -0.2524253 -0.1706239
```

Using the approximate, Wald type inference we are 95% certain that the slope relating the log of the efficiency to the displacement of an engine lies in (-.25,-17). This confidence interval is slightly narrower than the interval provided by `lm()`, again because we are assuming that σ^2 is known and using the asymptotic normal sampling distribution in place of the true t sampling distribution which has heavier tails.

Likelihood Ratio Based Confidence Intervals

Unfortunately, the simple formula for the LRT statistic makes it no easier to compute confidence intervals based on the LRT statistic. This still has to be done numerically. However, `R` provides a function, `confint()`, that will do the numeric computations for us:

```
confint(glm1)
```

```
                 2.5 %     97.5 %
(Intercept)  3.3391421  3.5519539
disp2       -0.2522955 -0.1707538
```

The results indicate that a 95% confidence interval for the slope, β_1, is $(-.252, -.171)$ which is exactly the same as the Wald type interval. Once again, this occurs because we are working with a simple model with Gaussian errors and known dispersion and will not happen in general.

Residuals for GLM

Suppose that we fit a model, compare it with the saturated model, and find out that our smaller model does not fit the data adequately. The obvious question to ask is why the model doesn't fit. I.e., for which observations are the observed and expected values inconsistent. One way to answer this question is to compute residuals, much as we do for simple linear regression models. However, there is an additional challenge because the observations in most generalized linear models are not homoscedastic. The variance depends on the values of the covariates. This means that we would expect the residuals to be larger for some values of the covariate, and so we have to account for the difference in the variance before we can conclude that a specific residual is large.

Standardized Residuals

Recall that the variance for Y_i in a generalized linear model is

$$\text{Var}(Y_i) = \frac{w_i}{\phi} b''(\theta_i)$$

and that θ_i is itself a one-to-one function of the mean, μ_i. If we ignore the weights (this often seems to happen in the literature) then we can write $\text{Var}(Y_i) = v(\mu_i)$ for some function $v(\cdot)$ that depends on the cumulant function and the dispersion parameter. That is, the framework of the generalized linear model induces a relationship between the mean and the variance. In the binomial model with Y_i representing the proportion of successes $v(\mu_i) = \mu_i(1-\mu_i)/n_i$. For the Poisson model $v(\mu_i) = \mu_i$. The function $v(\cdot)$ is called the variance function.

If we don't account for the differences in variances when we look at the residuals then we might focus on residuals that are large by chance and miss some important errors where the variance is small. One way to account for this is to standardize residuals. That is, we compute:

$$r_i = \frac{y_i - \hat{\mu}_i}{SE(y_i - \hat{\mu}_i)}.$$

To compute this we need to compute the standard error of the raw residual. Full details are provided in Section 4.5.7 of **?** and I won't reproduce them here. The final result is that the standardized residual for the i^{th} observation is

$$r_i = \frac{y_i - \hat{\mu}_i}{\sqrt{1 - \hat{h}_i}}$$

where h_i is the i^{th} diagonal element of $H = W^{1/2}X(X'W^{-1}X)'W^{1/2}$ with W as defined before. This is the generalization of the hat matrix in linear regression and $\hat{h}_i$ is the generalization of the leverage that can be used to assess whether an observation has the potential to be an influential point.

There are individual functions in `R` to compute the fitted values, residuals, leverage, and Cook's distance for a fitted GLM. However, there is a very useful function called `augment()` from the package `broom` that acts as a wrapper for these function. Calling `augment()` will compute all of these values and add them as columns back into the original data set:

```r
## Compute fitted values, leverage, and standardized residuals
mtcars_aug <- augment(glm1)
```

The new data set contains the response and predictor variables along with columns `.fitted`, `.resid`, `.hat`, `.sigma`, and `.std.resid` providing the fitted value, residual (but with a twist), leverage, estimated residual standard deviation, Cook's distance, and standardized residual for each observation.

Figures 2 displays the standardized residuals for the model including displacement as a predictor of the log efficiency plotted versus the fitted values. If the model fits the data then the plot should share the same features as a residual plot for linear regression. The residuals should form a band of constant width with no visible trends, and 95% of the values should lie between -2 and 2. In this case, there is one value outside of this range which suggests that this is an outlier. There also seems to be a trend in the residuals (curving down and then up) which suggests that the log transformation may not have accounted for the non-linearity.

Figure 3 displays the leverage for the same model. The plot is not very informative since when there is only one covariate it will always show a parabolic shape. However, the plot of Cook's distance in Figure 4 shows some reason for concern. The value of Cook's distance for the point with a fitted value of 2.6, which was the point with the large residual, is much higher than that of the other values and suggests that it may be having undue influence on the analysis. We should probably take a closer look at this point.

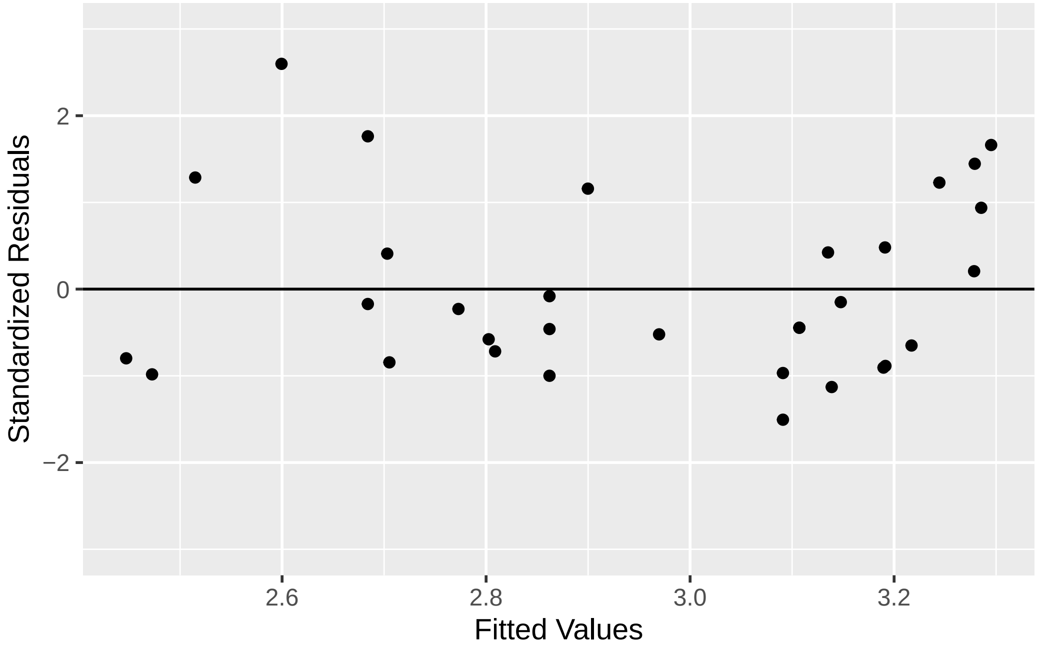

Figure 2: Standardized residuals versus the fitted values for the model of the log of efficiency versus displacement.

Deviance Residuals

The twist with the variable `.resid` computed by augment is that these are not the raw residuals (i.e., the difference between the observed and fitted values). Instead, these residuals take one of two forms depending on the arguments to `augment()`.

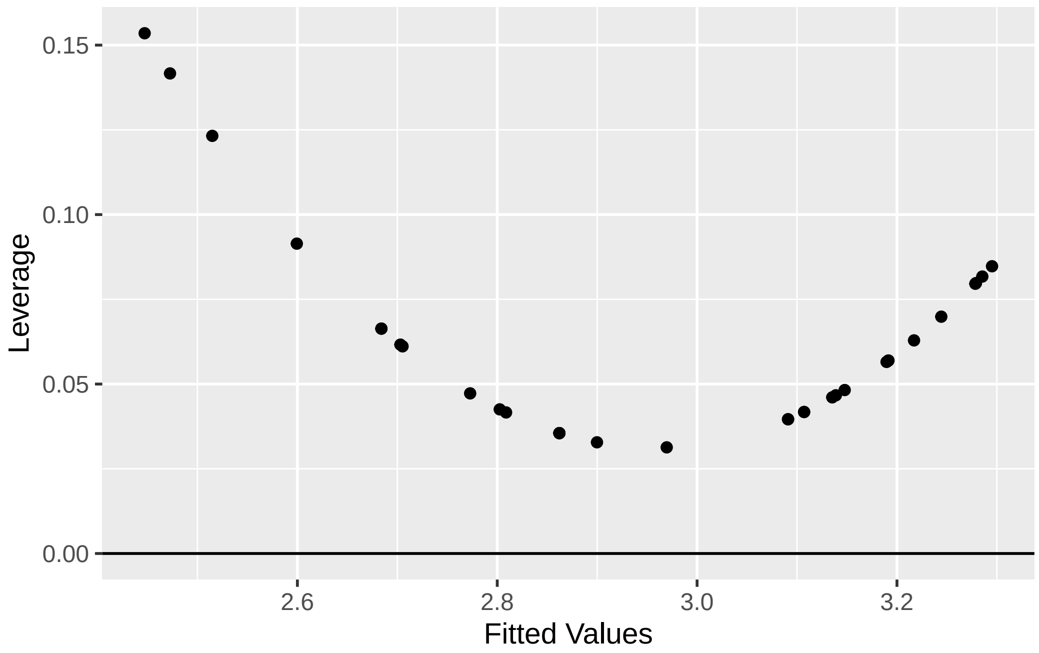

Figure 3: Leverage values versus displacement for the model of the log of efficiency versus displacement.

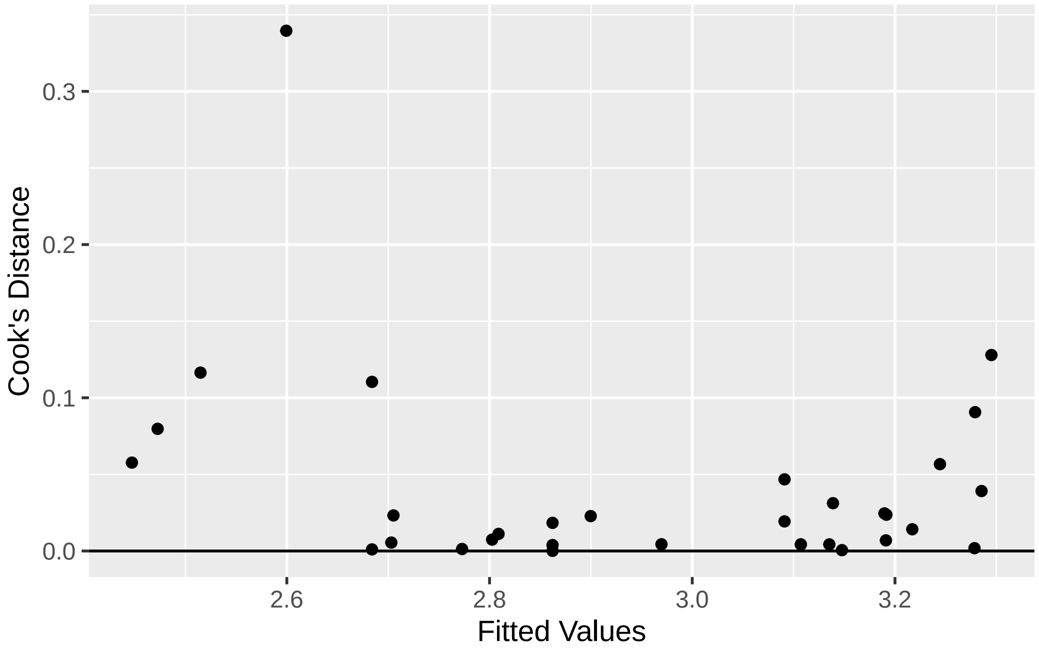

Figure 4: Leverage values versus displacement for the model of the log of efficiency versus displacement.

The standardized residuals are generally the best choice and can easily be computed for standard generalized linear models. However, there are some alternatives that are commonly used as well. One alternative way to construct residuals is to consider the individual contributions to the goodness-of-fit test statistics. Suppose that the deviance test statistic is large enough that the fit of the model is rejected. The test statistic is the sum of individual contributions, so this means that at least one of the individual contributions must be large. We can identify unusual observations by looking for observations with large contributions to the goodness of fit test statistic.

Let $\tilde{\theta}_i$ and $\hat{\theta}_i$ denote the fitted values of θ_i under the saturated model and the reduced model respectively. Consider that:

$$D(\boldsymbol{y}, \hat{\boldsymbol{\mu}}) = 2\sum_{i=1}^{N} \omega_i(y_i(\tilde{\theta}_i - \hat{\theta}_i) - (b(\tilde{\theta}_i) - b(\hat{\theta}_i))) = \sum_{i=1}^{N} d_i$$

where

$$d_i = 2\omega_i(y_i(\tilde{\theta}_i - \hat{\theta}_i) - (b(\tilde{\theta}_i) - b(\hat{\theta}_i))).$$

These values are always positive, so we have to account for the sign of the observed and expected values. The signed values:

$$\text{sign}(y_i - \hat{\mu}_i)\sqrt{d_i}, \quad i = 1, \ldots, N$$

are called the deviance residuals. By default, the values in `.resid` represent the deviance residuals. You can also set this by setting the argument `type.residual = "deviance"` in the call to `augment()`.

Pearson Residuals

Another test for assessing the goodness-of-fit of a model can be derived via the score test. For a generalized linear model the score test for testing goodness of fit leads to the test statistic

$$X^2 = \sum_{i=1}^{N} \frac{(y_i - \hat{\mu}_i)^2}{\widehat{\text{Var}(Y_i)}}.$$

In the special case of a Poison model for which $\mu_i = \text{Var}(Y_i) = \lambda_i$ the statistic becomes

$$X^2 = \sum_{i=1}^{N} \frac{(y_i - \hat{\lambda}_i)^2}{\hat{\lambda}_i} = \sum \frac{(\text{observed} - \text{fitted})^2}{\text{fitted}}$$

which is the Pearson chi-square statistic for testing goodness-of-fit of a model to counts in a contingency table. Hence, X^2 is called the generalized Pearson test statistic.

Residuals can be constructed from the generalized Pearson test statistic by looking at the signed squared roots of contributions for each observation. The resulting residual is

$$e_i = \frac{y_i - \hat{\mu}_i}{\sqrt{\widehat{\mathrm{Var}(Y_i)}}}.$$

These residuals are computed by `augment()` if `type.residual = "pearson"`.

Fitted Values

This will sound very odd at first, but there are actually multiple different fitted values for most GLM. Suppose that we have a generalized linear model with the link function $g(\cdot)$ such that

$$g(\mu_i) = \eta_i$$

where μ_i is the mean for observation i and η_i is the linear predictor. We can either compute the fitted value for the mean, $\hat{\mu}_i$ or we can compute the fitted value for linear predictor, $\hat{\eta}_i$. In `R`, these are referred to as the fitted values on the scales of the `response` and the `link`, respectively and the type of fitted value computed by `augment()` is determined by the argument `type.predict`. If `type.predict = "response"` then `augment()` computes the fitted value of the mean and if `type.predict = "link"` then `augment()` computes the fitted value of the linear predictor.

In most cases, it is easier to conduct inference on the scale of the linear predictor (i.e., setting `type.predict = "link"`) and then to transform the results by applying $g^{-1}(\cdot)$ to obtain inference on the scale of the mean. The reason for this is that the linear predictor is usually unbounded, whereas the mean is often restricted. This means that we can, e.g., construct Wald-type confidence intervals for the linear predictor and then transform them to the scale of the mean without having to worry about whether they might fall outside of the parameter space. This is generally the strategy that I will adopt through the rest of the course.