MODULE 6

LOGISTIC REGRESSION

Chapter Objectives:

1.

Contents

6.1 Logit Transformation **148**
6.2 Logistic Regression **151**
6.3 Logistic Regression in R **152**

LOGISTIC REGRESSION models are used when a researcher is investigating the relationship between a binary categorical response variable and a quantitative explanatory variable.[1] Typically, logistic regression is used to predict the probability of membership in one level of the response variable given a particular value of the explanatory variable. In some instances, the reverse will be solved for, such that one finds the value of the explanatory variable where a certain probability of the response variable would occur. Binary[2] logistic regression would be used in each of the following situations:

1. Predict the probability that a species of bat is of a certain subspecies based on the size of the canine tooth.
2. Predict the probability that a household will accept an offer to install state-subsidized solar panels given the household's income.
3. Predict the probability that a beetle will die when exposed to a certain concentration of a chemical pollutant.
4. Predict the probability of mortality for a patient given a certain "score" on a medical exam.
5. Identify the concentration of a chemical that will result in a 50% mortality rate for an animal.

Δ Binary Logistic Regression: A linear model where a binary response variable is examined with a quantitative explanatory variable.

6.1 Logit Transformation

The binary response variable in a logistic regression is treated as an indicator variable, where a "success" is coded as a "1" and a "failure" is coded as a "0."[3] Fitting a linear regression to this response variable plotted against the quantitative explanatory variable immediately exposes two problems (Figure 6.1). First, the linearity (and homoscedasticity) assumptions of linear regression are not met. Second, predicted probabilities from this model can be less than 0 and greater than 1, even within the domain of the explanatory variable. Clearly, a linear regression cannot be used to model this type of data.

Logistic regression is focused on the probability of "success" (p_i) at a given value of the quantitative explanatory variable (x_i). These probabilities are often calculated within "windows" of the x_i, as seldom are there many response observations at each given explanatory observation. For example, the probability of "success" may be calculated for the data shown in Figure 6.1 within "windows" that are 2.5 units wide on the x-axis (Figure 6.2). From this, it is seen that a model for the probability of "success" is clearly non-linear.

The odds of an event occurring is the probability of that event occurring divided by the probability of that event not occurring. Thus, the odds of "success" is computed with

$$odds_i = \frac{p_i}{1 - p_i}$$

Thinking in terms of odds takes some practice. For example, an odds of 5 means that the probability of a "success" is five times more likely than the probability of a "failure," whereas an odds of 0.2 means that the probability of a "failure" is five times (i.e., $\frac{1}{0.2} = 5$) more likely than the probability of a "success."

[1] Strictly, a logistic regression can be used with a categorical explanatory variable or multiple explanatory variables of mixed type. These notes will focus on the simplest situation where there is only one quantitative explanatory variable.
[2] This qualifying statement is needed as not all logistic regressions have a response variable with only two levels.
[3] Indicator variables were discussed in great detail in Section 5.1.

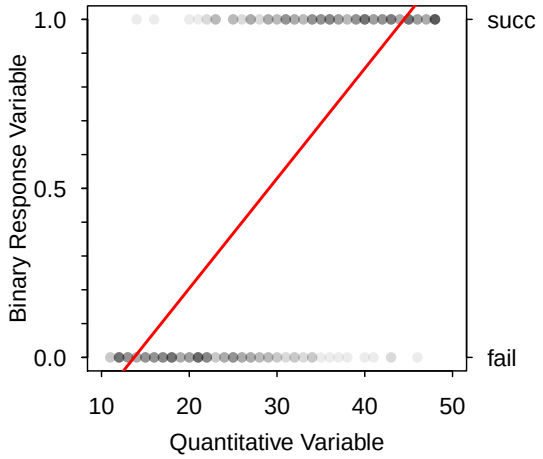

Figure 6.1. Plot of the binary response variable, as an indicator variable, versus a quantitative explanatory variable with the best-fit linear regression line super-imposed. Note that darker points have more individuals over-plotted at that coordinate.

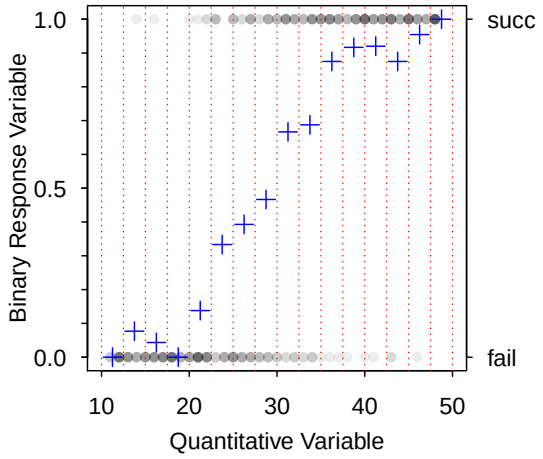

Figure 6.2. Plot of the binary response variable, as an indicator variable, versus a quantitative explanatory variable with the vertical lines representing the "windows" in which the probability of "success" (blue pluses) were calculated. Note that darker points have more individuals over-plotted at that coordinate.

Furthermore, it is instructive to note that an odds of 1 means that the probability of a "success" and a "failure" are equally likely. Finally, note that odds are bounded below by 0 (i.e., negative odds are impossible) but are not bounded above (i.e., odds can increase to positive infinity). A plot of the odds for the same data shown in Figure 6.3 illustrates these characteristics of odds.

While the plot of the odds of "success" versus the quantitative explanatory variable is not linear, it does have the characteristic shape of an exponential function. Exponential functions are "linearized" by transforming the response variable with natural logarithms (see Chapter 4). The natural log of the odds is called the "logit" transformation. Thus,

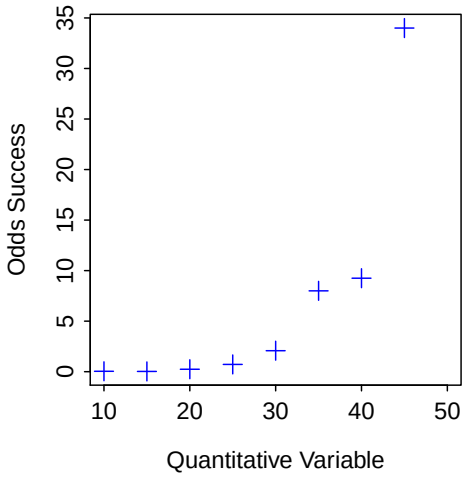

Figure 6.3. Plot of the odds of a "success" at various "windows" of the quantitative explanatory variable.

$$logit(p_i) = log\left(\frac{p_i}{1-p_i}\right)$$

and the plot of $logit(p_i)$ versus the explanatory variable will be linear (Figure 6.4).

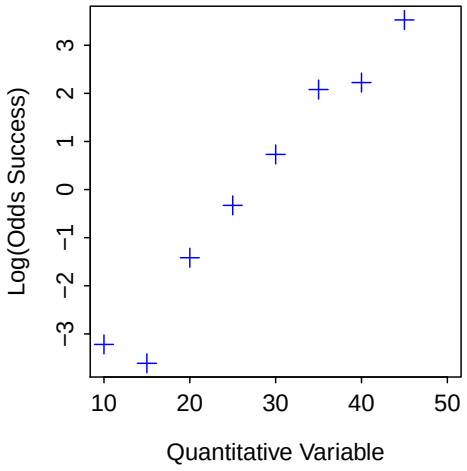

Figure 6.4. Plot of the logit transformed probability of a "success" (i.e., log odds of a "success") at various "windows" of the quantitative explanatory variable.

6.2 Logistic Regression

6.2.1 The Model

The logit transformation is a common transformation for "linearizing" the relationship between the probability of "success" and the quantitative explanatory variable. The logit transformation is the basis for a logistic regression, such that the logistic regression model is

$$log\left(\frac{p_i}{1-p_i}\right) = \alpha + \beta_1 x_i \tag{6.2.1}$$

where α is the "intercept" parameter and β_1 is the "slope" parameter.

6.2.2 Intepreting the Slope Coefficient

The slope for any linear regression represents the average change in the response variable for a unit change in the explanatory variable. In logistic regression, this corresponds to the average (additive) change in the log odds of a "success" for a unit change in the explanatory variable, or

$$\begin{aligned}\beta_1 &= log(ODDS(Y|X+1)) - log(ODDS(Y|X+1)) \\ &= log\left(\frac{PR(Y|X+1)}{1-PR(Y|X+1)}\right) - log\left(\frac{PR(Y|X)}{1-PR(Y|X)}\right)\end{aligned}$$

where $ODDS(Y|X)$ generically represents the odds of and $PR(Y|X)$ generically represents the probability of Y (the response variable) for a given value of X (the explanatory variable). A change in the log odds is not readily interpretable. However, back-transforming the slope, i.e.,

$$e^{\beta_1} = \frac{ODDS(Y|X+1)}{ODDS(Y|X)} = \frac{\frac{PR(Y|X+1)}{1-PR(Y|X+1)}}{\frac{PR(Y|X)}{1-PR(Y|X)}}$$

results in a measure of the **multiplicative** change in the odds of a "success" for a unit change in the explanatory variable. For example, $\hat{\beta}_1$=0.2 would mean that the log odds of a "success" increases by 0.2, on average, for a one unit increase in X. The corresponding back-tranformed slope, $e^{0.2}$=1.22, indicates that the odds of a "success" are 1.22 **times** greater for a one unit increase in X. The back-transformed slope does not indicate what the odds are, only that the odds are 1.22 **times** greater with an increase of one unit in the explanatory variable.

Nearly all software will provide a "default" significance test for the slope. This significance test tests H_0 : $\beta_1 = 0$ versus H_A : $\beta_1 \neq 0$. Thus, this significance tests determines, with a one unit increase in the explanatory variable, if the (additive) change in log odds of "success" is zero or the (multiplicative) change in odds is one. The probability does not change if the (additive) change in log odds is zero or, equivalently, if the (multiplicative) change in odds is one. Thus, the default hypothesis test for the slope determines if the log odds or the odds of "success" is related to the explanatory variable.

6.2.3 Predictions

Inserting a given value of the explanatory variable into the fitted logistic regression model, i.e., into Equation (6.2.1), results in a predicted log odds of "success." Back-transforming this result gives a predicted odds of "success", i.e.,

$$\frac{p_i}{1-p_i} = e^{\alpha+\beta_1 x_i} \tag{6.2.2}$$

The predicted odds equation, Equation (6.2.2), can be algebraically solved for p_i to provide a formula for predicting the probability of "success", i.e.,

$$p_i = \frac{e^{\alpha+\beta_1 x_i}}{1+e^{\alpha+\beta_1 x_i}} \tag{6.2.3}$$

Equation (6.2.2) can also be algebraically solved for x_i to provide a formula for the value of X that corresponds to a given probability of "success", i.e.,

$$x_i = \frac{log\left(\frac{p_i}{1-p_i}\right) - \alpha}{\beta_1} \tag{6.2.4}$$

6.2.4 Variability Estimates

Standard errors or confidence intervals for the parameters of the logistic regression model can be computed with normal distribution theory. However, this theory does not hold for all logistic regression models. In addition, methods for computing confidence intervals for predicted probabilities or for predicted values of X for a particular probability are not well formed. Bootstrapping provides one method for producing confidence intervals for these parameters.

In bootstrapping, B (re)samples of the same size as the original sample are produced with replacement from the original sample. The logistic regression model is fit to each of these new (re)samples and the estimates of interest (i.e., slope, intercept, predicted probability, etc.) are computed each time. The estimates from each (re)sample are then aggregated and an approximate 95% confidence interval is computed as the values of the estimate that have 2.5% of all bootstrapped values smaller and greater (i.e., find the bounds that contain the most common 95% of bootstrapped values). Bootstrap confidence intervals are computer intensive, but provide an interval that does not depend on the shape of the underlying sampling distribution.

6.3 Logistic Regression in R

Bliss (1935) examined the mortality response of beetles to various concentrations of gaseous carbon disulphide (mg/liter). The concentration and whether or not the beetle survived the exposure to that concentraton was recorded for each beetle and stored in **Bliss.csv** (view, download, meta). The levels for the *outcome* variable will be coded as "0"s and "1"s in the order that they are listed with `levels()`. Thus, "dead" will be coded with "1"s and the proportion of "successes" will be the proportion of beetles that were dead at the end of the exposure to gaseous carbon disulphide.

```
> d <- read.csv("Bliss.csv")
> str(d)
'data.frame': 481 obs. of  2 variables:
 $ outcome: Factor w/ 2 levels "alive","dead": 2 2 2 2 2 2 1 1 1 1 ...
 $ conc   : num  49.1 49.1 49.1 49.1 49.1 49.1 49.1 49.1 49.1 49.1 ...
> levels(d$outcome)
[1] "alive" "dead"
```

6.3.1 Fitting the Logistic Regression Model

The logistic regression model is fit with the *general linear model* function, `glm()`, which takes a formula of the form `factor~quantitative` where `factor` represents the binomial response variable and `quantitative` represents the quantitative explanatory variable as the first argument and the data.frame that contains these variables in the `data=` argument. In addition, `family=binomial` must be included to force `glm()` to fit a logistic regression model. The results of `glm()` should be saved to an object. Submitting the saved `glm` object to `summary()` provides basic model summary results.

```
> glm1 <- glm(outcome~conc,data=d,family=binomial)
> summary(glm1)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -14.82300    1.28955  -11.49   <2e-16
conc          0.24942    0.02139   11.66   <2e-16

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 645.44  on 480  degrees of freedom
Residual deviance: 368.62  on 479  degrees of freedom
AIC: 372.62

Number of Fisher Scoring iterations: 5
```

From these results, the estimated intercept is -14.82 and the estimated slope is 0.249. In addition, it is apparent that there is a signifcant relationship between the log odds of death for the beetles and the concentration of the calcium disulphide (slope $p < 0.00005$).

The fit of the logistic regression model is seen using `fitPlot()` with the saved `glm()` object as the first argument. The number of "windows" for which proportions of successes are computed can be defined with `breaks=`, which can be set to a single number or a sequence of numbers that identify the endpoints of the "windows."[4] For example, the code below constructs a plot with "windows" that begin at 45 and step every 5 units to 80 (Figure 6.5).[5]

```
> par(mar=c(3.5,3.5,1,3), mgp=c(2,0.75,0),las=1,tcl=-0.2)
> fitPlot(glm1,breaks=seq(45,80,5),xlim=c(45,80),
         xlab="Concentration",ylab="Proportion Dead")
```

[4] This sequence is best created with `seq()` where the three argument are the starting, ending, and step value of the sequence.
[5] Note that the use of `par()` here is to make the plot look like other plots in the book and is likely not needed in general use.

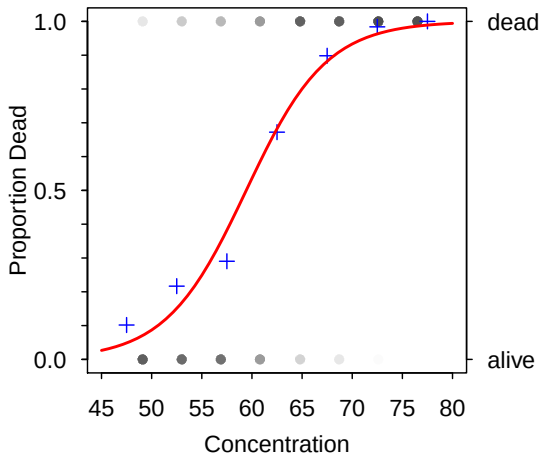

Figure 6.5. Plot of the binary outcome variable versus the concentration of calcium disulfide with the probability of death (blue pluses) calculated at every five units of concentration and the fitted logistic regression model superimposed (in red). Darker points have more individuals over-plotted at that coordinate.

Confidence intervals from normal theory are extracted from the saved `glm` object to `confint()`.

```
> confint(glm1)              # normal theory
Waiting for profiling to be done...
                  2.5 %      97.5 %
(Intercept) -17.4924161 -12.4235445
conc          0.2096978   0.2937584
```

Bootstrap samples are taken by sending the saved `glm` object to `bootCase()` and saving the results to an object.[6] The bootstrap samples are taken and the first six rows of results are shown below.

```
> bc1 <- bootCase(glm1)      # bootstrapping, be patient!
> head(bc1)
     (Intercept)      conc
[1,]   -14.48441 0.2438393
[2,]   -13.73752 0.2354945
[3,]   -15.30474 0.2607897
[4,]   -15.81387 0.2630108
[5,]   -16.28585 0.2721009
[6,]   -13.77281 0.2324942
```

The bootstrap confidence intervals are extracted from the `bootCase` object with `confint()`.

```
> confint(bc1)
                95% LCI     95% UCI
(Intercept) -17.727040 -12.7237878
conc          0.213892   0.2964944
```

[6] `bootCase()` is from the `car` package.

Thus, the normal theory provides a 95% confidence interval for the slope from 0.210 to 0.294, whereas the bootstrap estimates are from 0.214 to 0.296. The confidence intervals for the slope from these two methods are not widely disparate in this example because the sampling distribution of the slope is approximately normal (Figure 6.6). The distributions of the parameters from the bootstrap samples (Figure 6.6) are constructed from the `bootCase()` object with `hist()`.

```
> hist(bc1)
```

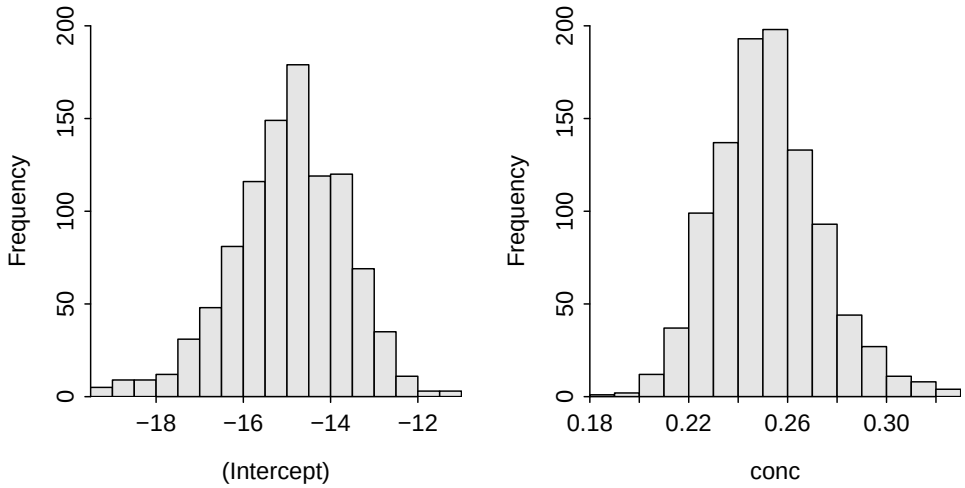

Figure 6.6. Histograms of the logistic regression parameter estimates from the bootstrapped samples.

6.3.2 Predictions

The predicted probability of death given a value of the concentration of calcium disulphide is found with `predict()`, similar to what was shown in previous chapters. The one difference is that if one wants to predict the probability, rather than the log odds, then `type="response"` must be included. For example, the predicted probability of death at a concentration of 70 mg/l is computed below.

```
> predict(glm1,data.frame(conc=70),type="response")
        1
0.9331487
```

Unfortunately, confidence intervals for this prediction can not be constructed with `predict()`. Bootstrap confidence intervals, however, can be constructed with a bit of work. First, one must create a small function that computes the predicted probability "by hand", i.e., code Equation (6.2.3) with

```
> predProb <- function(x,alpha,beta1) exp(alpha+beta1*x)/(1+exp(alpha+beta1*x))
```

A quick check of this function,

```
> predProb(70,coef(glm1)[[1]],coef(glm1)[[2]])
[1] 0.9331487
```

shows that it provides the same prediction as found with `predict()`. Now use `predProb()` to predict the probability given the parameter estimates for each bootstrap sample (by realizing that the intercepts are in the first and the slopes are in the second column of `bc1`). These predicted probabilities should be saved to an object that can be submitted to `quantile()` to compute the quantiles that represent the confidence bounds. These calculations are illustrated with

```
> p70 <- predProb(70,bc1[,1],bc1[,2])
> quantile(p70,c(0.025,0.975))
     2.5%     97.5%
0.9033577 0.9591189
```

Thus, one is 95% confident that the predicted probability of death for beetles exposed to 70 mg/l calcium disulphide is between 0.903 and 0.959.

A similar process is used to predict the concentration where 50%, for example, of the beetles will have died (in this case, Equation (6.2.4) must be coded as a function). For example,

```
> predX <- function(p,alpha,beta1) (log(p/(1-p))-alpha)/beta1
> x50 <- predX(0.5,bc1[,1],bc1[,2])
> quantile(x50,c(0.025,0.975))
    2.5%    97.5%
58.34560 60.45673
```

Thus, one is 95% confident that the predicted concentration where 50% of the beetles would be dead is between 58.3 and 60.5.