Chapter 8

Generalized Linear Models

`In God We Trust; All Others Must Bring Data`
– *Willian Edwards Deming*

The Multiple Regression model we explored in the last chapter assumed that the error terms were independently and identically distributed as Normal. Generalized linear models ("GLM") extend the traditional multiple regression model to include error terms following different distributions. R provides the function `glm()` to estimate generalized linear models in similar ways as the previously studied `lm()` function. The only additional information required by the `glm()` command is the $family$ argument that specifies the distribution of the error term.

We will begin our exploration of GLMs with most popular one - Logit Model or Logistic Regression for a binary response variable.

8.1 Logistic Regression

Logistic regression is used to model dichotomous outcome variables - variables that take binary values - True/False, Yes/No, 1/0 etc. Before we run a regression, the binary response variable needs to be transformed to a continuous variable of wide range. The standard approach is to calculate log odds - log of odds ratio. In the logit model the log odds of the outcome variable is modeled as the response to the linear combination of the predictor variables. Log-odds have the recognizable curve as shown in the Figure 8.1.

For illustrating logistic regression, we will use a dataset of 400 graduate school applications.

```
admit.data <- read.csv("binary.csv")
names(admit.data)
## [1] "admit" "gre"   "gpa"   "rank"
```

The dataset consists four columns - the outcome variable *admit* is a binary variable. The predictor variables are *gre* (the GRE score), *gpa* (the undergrad GPA) and *rank* (a categorical variable from 1 to 4 indicating the status or prestige of the institution, with 1 denoting the highest prestige).

We will first convert *rank* as a factor and then run a logistic regression.

```r
probs <- seq(0, 1, 0.01)
odds = probs / (1 - probs); logodds = log(odds)
plot(probs, logodds)
```

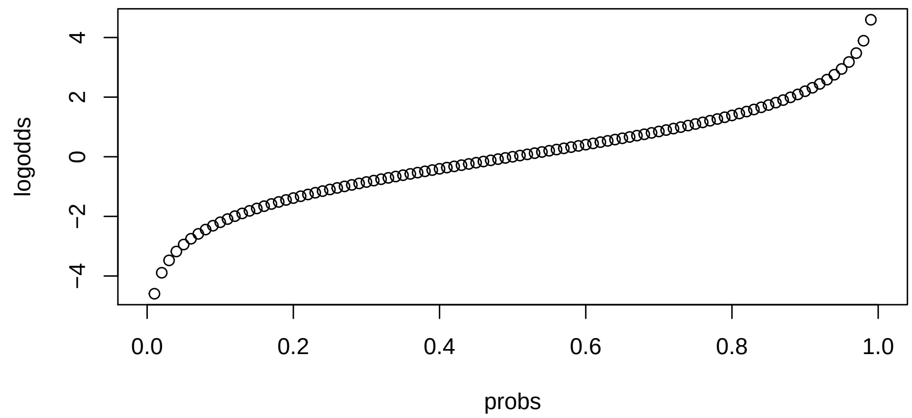

Figure 8.1: Plot of Log of Odds Ratio

```r
admit.data$rank <- factor(admit.data$rank)
logit.model <- glm(admit ~ gre + gpa + rank, data = admit.data,
                   family = "binomial")
summary(logit.model)
##
## Call:
## glm(formula = admit ~ gre + gpa + rank, family = "binomial",
##     data = admit.data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6268  -0.8662  -0.6388   1.1490   2.0790
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.989979   1.139951  -3.500 0.000465 ***
## gre          0.002264   0.001094   2.070 0.038465 *
## gpa          0.804038   0.331819   2.423 0.015388 *
## rank2       -0.675443   0.316490  -2.134 0.032829 *
## rank3       -1.340204   0.345306  -3.881 0.000104 ***
## rank4       -1.551464   0.417832  -3.713 0.000205 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 499.98  on 399  degrees of freedom
## Residual deviance: 458.52  on 394  degrees of freedom
## AIC: 470.52
##
## Number of Fisher Scoring iterations: 4
```

While interpreting the output, keep in mind that the response variable modeled here is not the variable *admit* but the log-odds of *admit*. The output of the model is similar to linear regression - we see residuals, coefficients and p-value - all with similar interpretations as before. We see that both *gre* and *gpa* are statistically significant (p-value < 0.05). The *rank* factor has been converted into three dummy variables - all statistically significant.

Looking at the coefficients, we can see that for 1 unit change in GRE score, log odds of admission increases by 0.002. Similarly, for 1 unit change in GPA, log odds of admission increases by 0.804. Attending an institution of rank 2 vs an institution of rank 1 reduces the log odds of admission by 0.6754.

Given a logistic regression model, we can conduct further significance tests using the `wald test` in the package `aod`. For example, we can test whether the *rank* factors taken together are statistically significant or not. As the output below shows, the three factors taken together have a statistically significant impact.

```
library(aod)
wald.test(b = coef(logit.model), Sigma = vcov(logit.model),
          Terms = 4:6)
## Wald test:
## ----------
##
## Chi-squared test:
## X2 = 20.9, df = 3, P(> X2) = 0.00011
```

Similarly, we can test whether it makes a difference whether the students comes from an institution of rank 3 of rank 4. While both these ranks had a statistically significant coefficients in the logit model, the magnitude of the coefficients show little difference - so its a relevant question to ask. As we can see from the output below, the effects of rank 3 and rank4 are in fact not statistically distinguishable as far as their impact on log odds of admission.

```
coefflist <- cbind(0,0,0,0,1,-1)
wald.test(b = coef(logit.model), Sigma = vcov(logit.model),
          L = coefflist)
## Wald test:
## ----------
##
```

```
## Chi-squared test:
## X2 = 0.29, df = 1, P(> X2) = 0.59
```

As log odds are not intuitive to interpret, we can calculate coefficients as odds ratios. We can then use our model to predict probability of admission for a given set of input values.

```
exp(coef(logit.model))
## (Intercept)         gre         gpa       rank2       rank3       rank4
##   0.0185001   1.0022670   2.2345448   0.5089310   0.2617923   0.2119375
admitnew <- data.frame(gre = 700, gpa = 3.9, rank = 1)
admitnew$rank = factor(admitnew$rank)
predict(logit.model, newdata = admitnew, type = "response")
##         1
## 0.6749952
```

Now we can say that a unit change in GPA improves the odds of being admitted by a factor of 2.23.

8.2 Probit Model

A probit model is also suitable for binary response variables. In the probit model, the inverse standard normal distribution of the probability is modeled as a linear combination of the predictors. Probit transformation has a similar shape as logistic transformation as can be seen in Figure: 8.2.

We can use the same `glm()` command to run a probit model. We need to specify *link* as *probit* to ensure that the model is run on a probit transformed outcome variable.

```
probit.model <- glm(admit ~ gre + gpa + rank, data = admit.data,
                    family = binomial(link="probit"))
summary(probit.model)
##
## Call:
## glm(formula = admit ~ gre + gpa + rank, family = binomial(link = "probit")
##     data = admit.data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6163  -0.8710  -0.6389   1.1560   2.1035
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.386836   0.673946  -3.542 0.000398 ***
## gre          0.001376   0.000650   2.116 0.034329 *
## gpa          0.477730   0.197197   2.423 0.015410 *
## rank2       -0.415399   0.194977  -2.131 0.033130 *
## rank3       -0.812138   0.208358  -3.898 9.71e-05 ***
## rank4       -0.935899   0.245272  -3.816 0.000136 ***
## ---
```

```r
library(VGAM)
probs <- seq(0, 1, 0.01)
probitdata <- probit(probs)
plot(probs, probitdata)
```

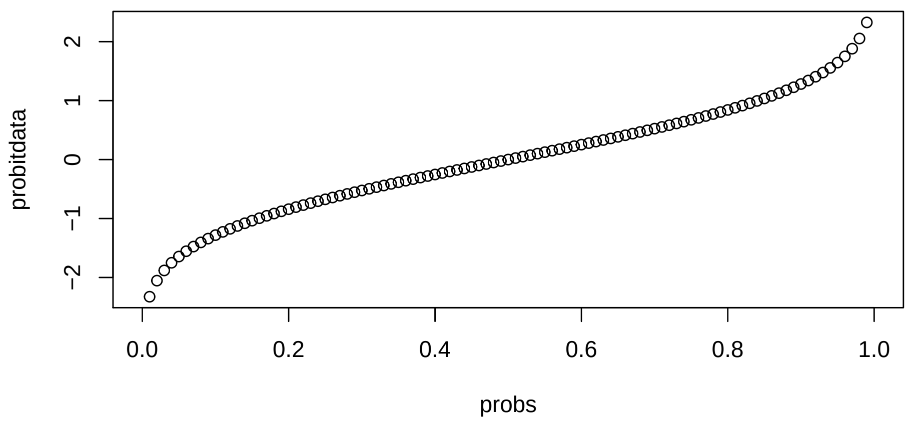

Figure 8.2: Plot of Probit Transformation

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 499.98  on 399  degrees of freedom
## Residual deviance: 458.41  on 394  degrees of freedom
## AIC: 470.41
##
## Number of Fisher Scoring iterations: 4
```

Usually, logit and probit models are pretty much interchangeable. We can demonstrate the fact by calculating predicted probability for the same values as for logit before. As we can see from the output below, the predicted probabilities are nearly identical.

```r
predict(probit.model, newdata = admitnew,  type = "response")
##         1
## 0.6697505
```

Much of our discussion of logistic regression is also applicable to probit models as well. The actual logit and probit transformations show some difference only very close to the edges (near probability either 0 or 1). This leads to minor, non-significant differences in model fit. However, logit models have the advantage of an easily interpretable result - log of odds ratio is easier to understand and interpret than the inverse of cumulative normal distribution.