```
> glm_probs_def_2 <- predict(
   glm_fit_def_2,data.frame(student=c('Yes','No')), type = "response")
> glm_probs_def_2
         1          2
0.04313859 0.02919501
```

Multiple Logistic Regression

We now consider the problem of predicting a binary response using multiple predictors.

We can generalize $\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X$ as follows:

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X_1 + ... + \beta_p X_p \qquad (*)$$

where $X = (X_1, ..., X_p)$ are p predictors.

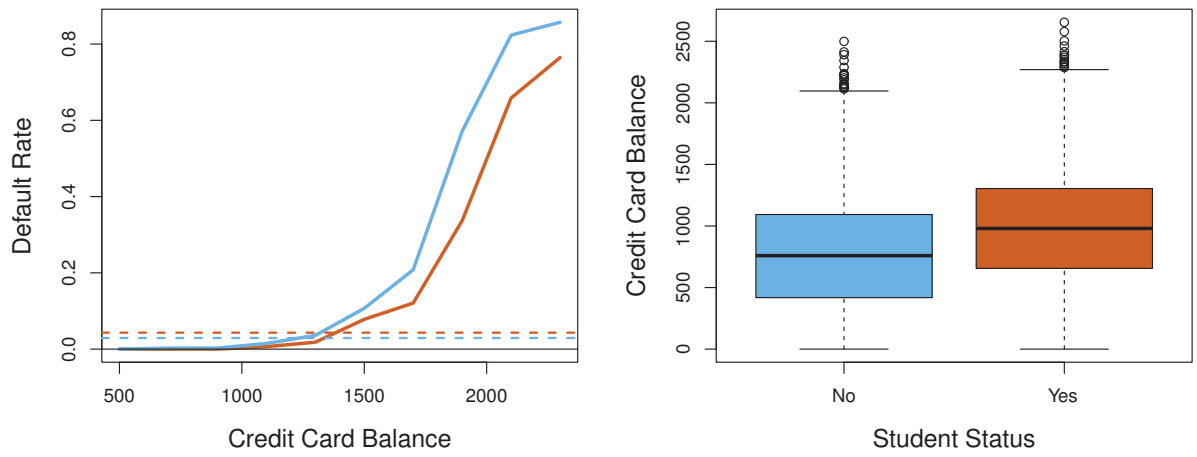
(*) can be rewritten as

$$p(X) = \frac{e^{\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p}}{1 + e^{\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p}}$$

Just like before, we use the maximum likelihood method to estimate $\beta_0, \beta_1, ..., \beta_p$.

For the Default data, we use glm() function to find the estimated coefficients of logistic regression model that predicts the probability of default using balance, income, and student status.

```
> glm_fit_def_3 <- glm(default ~ . , data = Default, family = binomial)
> summary(glm_fit_def_3)

Call:
glm(formula = default ~ ., family = binomial, data = Default)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.4691  -0.1418  -0.0557  -0.0203   3.7383

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.087e+01  4.923e-01 -22.080  < 2e-16 ***
studentYes  -6.468e-01  2.363e-01  -2.738  0.00619 **
balance      5.737e-03  2.319e-04  24.738  < 2e-16 ***
income       3.033e-06  8.203e-06   0.370  0.71152
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2920.6  on 9999  degrees of freedom
Residual deviance: 1571.5  on 9996  degrees of freedom
AIC: 1579.5

Number of Fisher Scoring iterations: 8
```



Default data. Left: Default rates are shown for students (orange) and non-students (blue). The solid lines display default rate as a function of balance, while the horizontal broken lines display the overall default rates. Right: Boxplots of balance for students (orange) and non-students (blue) are shown.

students tend to have higher balances than non-students, so their marginal default rate is higher than for non-students.
But for each level of balance, students default less than non-students.

The predict() function can be used to predict the probability that an individual defaults. For example, a student with a credit card balance of $1,500 and an income of $40, 000 has an estimated probability of default of 0.058 while a non-student with the same balance and income has an estimated probability of default of 0.105.

```
> glm_probs_def_3 <- predict(glm_fit_def_3,data.frame(student=c('Yes','No'),
    balance=1500, income=40000), type = "response")
> glm_probs_def_3
```

```
         1          2
0.05788194 0.10499192
```

If no data set is supplied to the predict() function, then the probabilities are computed for the training data that was used to fit the logistic regression model.

```
> glm_probs_def_4 <- predict(glm_fit_def_3, type = "response")
> glm_probs_def_4[1:10]
           1            2            3            4            5
1.428724e-03 1.122204e-03 9.812272e-03 4.415893e-04 1.935506e-03
           6            7            8            9           10
1.989518e-03 2.333767e-03 1.086718e-03 1.638333e-02 2.080617e-05

> contrasts(Default$default)
    Yes
No    0
Yes   1


> dim(Default)
[1] 10000     4
> glm_pred_def <- rep("No", 10000)
> glm_pred_def[glm_probs_def_4 > .5] = "Yes"
> table(glm_pred_def, Default$default)

glm_pred_def   No  Yes
         No  9627  228
         Yes   40  105

> mean(glm_pred_def==Default$default)
[1] 0.9732

> mean(glm_pred_def!=Default$default)
[1] 0.0268
```

training err. rate $= \frac{228+40}{10000} = 0.268$

← training err. rate

The diagonal elements of the confusion matrix indicate correct predictions, while the off-diagonals represent incorrect predictions.

Our model correctly predicted that 105 individuals would default and that 9627 would not, for a total of 105+9627=9732 correct predictions.

```
> Default_test<- Default[9001:10000, ]
> glm_fit_def_5 <- glm(default ~ . , data = Default, family = binomial,
   subset=1:9000)
> glm_probs_def_5 <- predict(glm_fit_def_5, Default_test, type = "response")
> glm_pred_def_1 <- rep("No", 1000)
> glm_pred_def_1[glm_probs_def_5 > .5] = "Yes"
> table(glm_pred_def_1, Default_test$default)

glm_pred_def_1  No Yes
           No  958  27
           Yes   6   9
> mean(glm_pred_def_1==Default_test$default)
[1] 0.967
> mean(glm_pred_def_1!=Default_test$default)
[1] 0.033
```

← test error rate

Here we used first 9000 observations to train our model and tested its performance by predicting the default on last 1000 observations. In this case, logistic regression correctly predicted the default 96.7% of the time.

Multinomial Logistic Regression

So far we have discussed logistic regression with two classes, $K = 2$. We sometimes wish to classify a response variable that has more than two classes.

- It is possible to extend the two-class logistic regression approach to the setting of $K > 2$ classes. This extension is sometimes known as *multinomial logistic regression.*
- To do this, we first select a single class to serve as the baseline; without loss of generality, we select the Kth class for this role.
- Then the model has the form

$$\Pr(Y = k|X) = \frac{e^{\beta_{k0}+\beta_{k1}X_1+...+\beta_{kp}X_p}}{1 + \sum_{l=1}^{K-1} e^{\beta_{l0}+\beta_{l1}X_1+...+\beta_{lp}X_p}}, \quad k = 1, \ldots, K-1$$

and

$$\Pr(Y = K|X) = \frac{1}{1 + \sum_{l=1}^{K-1} e^{\beta_{l0}+\beta_{l1}X_1+...+\beta_{lp}X_p}}$$

we can show that for $k=1, \cdots, K-1$,

$$\log\left(\frac{\Pr(Y=k|X)}{\Pr(Y=K|X)}\right) = \beta_{k0} + \beta_{k1}X_1 + \cdots + \beta_{kp}X_p$$

This log odds between any pair of classes is linear in the features.

- An alternative coding for multinomial logistic regression is known as the softmax coding.
- In the softmax coding, rather than selecting a baseline class, we treat all K classes symmetrically, and assume that for $k = 1, ..., K$,

$$\Pr(Y = k|X) = \frac{e^{\beta_{k0}+\beta_{k1}X_1+...+\beta_{kp}X_p}}{\sum_{l=1}^{K} e^{\beta_{l0}+\beta_{l1}X_1+...+\beta_{lp}X_p}}$$

The log odds ratio between the kth and k'th classes equals

$$\log\left(\frac{Pr(Y=k|X)}{Pr(Y=k'|X)}\right) = (\beta_{k0}-\beta_{k'0}) + (\beta_{k1}-\beta_{k'1})X_1 + \cdots + (\beta_{kp}-\beta_{k'p})X_p$$

You may use multinom( ) function which is part of nnet library to fit a multinomial logistic regression in R.

Discriminant Analysis

Here the approach is to model the distribution of X in each of the classes separately, and then use Bayes' theorem to flip things around and obtain $\Pr(Y|X)$.

When we use normal (Gaussian) distributions for each class, this leads to linear or quadratic discriminant analysis.

- Suppose that we wish to classify an observation into one of K classes, where $K \geq 2$. In other words, the qualitative response variable Y can take on K possible distinct and unordered values.
- Let $\pi_k = \Pr(Y = k)$ represent the overall, marginal, or prior probability that a randomly chosen observation comes from the kth class.
- Let $f_k(x) = \Pr(X = x|Y = k)$ denote the density for X in class k. Here we will use normal densities for these, separately in each class.

- Then Bayes' theorem states that

$$\Pr(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$$

- In general, estimating π_k is easy if we have a random sample from the population: we simply compute the fraction of the training observations that belong to the kth class.

- However, estimating the density function $f_k(x)$ is much more challenging. As we will see, to estimate $f_k(x)$, we will typically have to make some simplifying assumptions.

Linear Discriminant Analysis for $p = 1$

- We assume that $f_k(x)$ is normal or Gaussian. In the one-dimensional setting, the normal density takes the form

$$f_k(x) = \frac{1}{\sqrt{2\pi}\sigma_k} \exp\left(-\frac{1}{2\sigma_k^2}(x - \mu_k)^2\right)$$

  where μ_k and σ_k^2 are the mean and variance parameters for the kth class. We will also assume that all the $\sigma_k = \sigma$ are the same.

- Plugging this into Bayes formula, we get a rather complex expression for $p_k(x) = \Pr(Y = k|X = x)$:

$$p_k(x) = \frac{\pi_k \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(x - \mu_k)^2\right)}{\sum_{l=1}^{K} \pi_l \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(x - \mu_l)^2\right)}$$

- To classify at the value $X = x$, we need to see which of the $p_k(x)$ is largest.

- Taking logs, and discarding terms that do not depend on k, we see that this is equivalent to assigning x to the class with the largest discriminant score:

$$\delta_k(x) = x \cdot \frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log(\pi_k)$$