### Step 1: Load the Data

For this example, we'll use the **Default** dataset from the ISLR package. We can use the following code to load and view a summary of the dataset:

```
#load dataset
data <- ISLR::Default

#view summary of dataset
summary(data)
```

```
 default    student       balance           income     
 No :9667   No :7056   Min.   :   0.0   Min.   :  772  
 Yes: 333   Yes:2944   1st Qu.: 481.7   1st Qu.:21340  
                       Median : 823.6   Median :34553  
                       Mean   : 835.4   Mean   :33517  
                       3rd Qu.:1166.3   3rd Qu.:43808  
                       Max.   :2654.3   Max.   :73554  
```

```
#find total observations in dataset
nrow(data)
```

```
[1] 10000
```

### Step 2: Create Training and Test Samples

Next, we'll split the dataset into a training set to *train* the model on and a testing set to *test* the model on.

```
#make this example reproducible
set.seed(1)

#Use 70% of dataset as training set and remaining 30% as testing
set
sample <- sample(c(TRUE, FALSE), nrow(data), replace=TRUE,
prob=c(0.7,0.3))
train <- data[sample, ]
```

```
test <- data[!sample, ]
```

### Step 3: Fit the Logistic Regression Model

Next, we'll use the **glm** (general linear model) function and specify family="binomial" so that R fits a logistic regression model to the dataset:

```
#fit logistic regression model
model <- glm(default~student+balance+income, family="binomial",
data=train)

#disable scientific notation for model summary
options(scipen=999)

#view model summary
summary(model)

Call:
glm(formula = default ~ student + balance + income, family =
"binomial",
    data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.5586  -0.1353  -0.0519  -0.0177   3.7973

Coefficients:
                  Estimate    Std. Error z value             Pr(>|z|)
(Intercept) -11.478101194   0.623409555 -18.412 <0.0000000000000002
***
studentYes   -0.493292438   0.285735949  -1.726               0.0843
.
balance       0.005988059   0.000293765  20.384 <0.0000000000000002
***
income        0.000007857   0.000009965   0.788               0.4304
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2021.1  on 6963  degrees of freedom
Residual deviance: 1065.4  on 6960  degrees of freedom
AIC: 1073.4

Number of Fisher Scoring iterations: 8
```

### Step 4: Use the Model to Make Predictions

Once we've fit the logistic regression model, we can then use it to make predictions about whether or not an individual will default based on their student status, balance, and income:

```
#define two individuals
new <- data.frame(balance = 1400, income = 2000, student = c("Yes",
"No"))

#predict probability of defaulting
predict(model, new, type="response")
```

```
         1          2
0.02732106 0.04397747
```

### Step 5: Model Diagnostics

Lastly, we can analyze how well our model performs on the test dataset.

By default, any individual in the test dataset with a probability of default greater than 0.5 will be predicted to default. However, we can find the optimal probability to use to maximize the accuracy of our model by using the **optimalCutoff()** function from the InformationValue package:

```
library(InformationValue)
```

```
#convert defaults from "Yes" and "No" to 1's and 0's
test$default <- ifelse(test$default=="Yes", 1, 0)

#find optimal cutoff probability to use to maximize accuracy
optimal <- optimalCutoff(test$default, predicted)[1]
optimal
```

```
[1] 0.5451712
```

**Step 6: Plotting**

Lastly, we can plot the ROC (Receiver Operating Characteristic) Curve which displays the percentage of true positives predicted by the model as the prediction probability cutoff is lowered from 1 to 0. The higher the AUC (area under the curve), the more accurately our model is able to predict outcomes:

```
#plot the ROC curve
plotROC(test$default, predicted)
```