Topics in Data Science

Lecture 9

Department of Mathematics and Statistics

PURDUE UNIVERSITY NORTHWEST

Classification

Classification is the process of finding or discovering a model or function which helps in separating the data into multiple categorical classes i.e. discrete values. In classification, data is categorized under different labels according to some parameters given in input and then the labels are predicted for the data. The derived mapping function could be demonstrated in the form of "IF-THEN" rules. The classification process deal with the problems where the data can be divided into binary or multiple discrete labels.
The Classification algorithm is a Supervised Learning technique that is used to identify the category of new observations on the basis of training data. In Classification, a program learns from the given dataset or observations and then classifies new observation into a number of classes or groups.
Linear Regression and Logistic Regression are the two famous Machine Learning Algorithms which come under supervised learning technique. Since both the algorithms are of supervised in nature hence these algorithms use labeled dataset to make the predictions. But the main difference between them is how they are being used. The Linear Regression is used for solving Regression problems whereas Logistic Regression is used for solving the Classification problems.

Logistic Regression

Logistic regression is a part of a category of statistical models called generalized linear models(GLM). The GLM is a unification of both linear and nonlinear regression models that also allows the incorporation of nonnormal response distributions. In GLM, the response variable distribution must be a member of exponential family which includes normal, poisson, binomial, exponential, gamma etc. This broad class of models includes ordinary regression and ANOVA, as well as multivariate statistics such as ANCOVA and loglinear regression.
Logistic regression allows one to predict a discrete outcome, such as group membership, from a set of variables that may be continuous, discrete, dichotomous, or a mix of any of these. Generally, the dependent or response variable is dichotomous, such as presence/absence or success/failure.
Discriminant analysis is also used to predict group membership with only two groups. However, discriminant analysis can only be used with continuous independent variables. Thus, in instances where the independent variables are a categorical, or a mix of continuous and categorical, logistic regression is preferred.

Logistic Regression

The dependent variable in logistic regression is usually dichotomous, that is, the dependent variable can take the value 1 with a probability of success π, or the value 0 with probability of failure $1 - \pi$. This type of variable is called a Bernoulli (or binary) variable. Although not as common, applications of logistic regression have also been extended to cases where the dependent variable is of more than two cases, known as multinomial or polytomous [Tabachnick and Fidell (1996) use the term polychotomous].
As mentioned previously, the independent or predictor variables in logistic regression can take any form. That is, logistic regression makes no assumption about the distribution of the independent variables. They do not have to be normally distributed, linearly related or of equal variance within each group.The relationship between the predictor and response variables is not a linear function in logistic regression, instead, the logistic regression function is used, which is the logit transformation of π:

Odds ratio

Odds is usually defined in statistics as the probability an event will occur divided by the probability that it will not occur. In other words, it's a ratio of successes (or wins) to losses (or failures). The OR represents the odds that an outcome will occur given a particular exposure, compared to the odds of the outcome occurring in the absence of that exposure.

```
> p<-c(0.003,0.1, 0.3, 0.5, 0.8, 0.9) #probability
> odds<-p/(1-p)
> logit<- log(odds) # Logit=log(odds)
> cbind(p, odds, logit)
> cbind(p, odds, logit)
          p         odds      logit
[1,] 0.003 0.003009027 -5.8061385
[2,] 0.100 0.111111111 -2.1972246
[3,] 0.300 0.428571429 -0.8472979
[4,] 0.500 1.000000000  0.0000000
[5,] 0.800 4.000000000  1.3862944
[6,] 0.900 9.000000000  2.1972246
```

Sigmoidal Response Functions for Binary Responses

The three response functions which can be used for modeling binary responses from zero-one coding are the following:

- Probit mean Response Function
- Logistic mean response function
- Complementary log log response function.

Note that all of these response functions have the following properties:

- Bounded between 0 and 1.
- They are sigmoidal means S-shaped
- Approaches asymptotically to 0 and 1.

Simple Logistic Regression

When the response variable is binary, taking 1 and 0 with probabilities π and $1-\pi$, respectively, y is a Bernoulli random variable with parameter $E(y) = \pi$. The simple logistic regression model is given by

$$\pi_i = \frac{e^{\beta_0+\beta_1 x_i}}{1+e^{\beta_0+\beta_1 x_i}}$$

We know that $E(y_i) = 1(\pi_i) + 0(1-\pi_i) = \pi_i \Longrightarrow \pi_i = E(y_i)$
Straightforward algebra yields

$$E(y_i) = \pi_i = [1+\exp(-\beta_0 - \beta_1 x_i)]^{-1}$$

The x observations are assumed to be known constants. If x observations are random, $E(y_i)$ is viewed as a conditional mean given the value of x_i.
The parameter π defines the mean of the distribution: $E(y_i) = \pi$. The logistic regression models the success probability as a function of the severity(x). That is, $\pi = \pi(x)$ so that for case i with severity x_i, $\pi_i = \pi(x_i)$.

Once the MLE b_0 and b_1 are found, we substitute these values in the response function

$$\pi_i = \frac{e^{\beta_0+\beta_1 x_i}}{1+e^{\beta_0+\beta_1 x_i}}$$

to obtain the response function

$$\widehat{\pi}_i = \frac{e^{b_0+b_1 x_i}}{1+e^{b_0+b_1 x_i}}$$

Hence, the logistic response function will be

$$\widehat{\pi} = \frac{e^{b_0+b_1 x}}{1+e^{b_0+b_1 x}}$$

Considering the logit transformation we can rewrite the response function as

$$\widehat{\pi}' = b_0 + b_1 x$$

where $\widehat{\pi}' = \ln\left(\frac{\widehat{\pi}}{1-\widehat{\pi}}\right)$.

Remark: The logarithm of the odds $\ln\left[\frac{\pi}{1-\pi}\right]$, is referred to as the log odds or the logit. It is clear that the logistic regression model assumes a linear model for the logit; that is,

$$\ln\left[\frac{\pi}{1-\pi}\right] = \beta_0 + \beta_1 x_1$$

What is b_1 in a logistic regression?

In the case of a linear regression the estimated regression coefficient b_1 represent the slope of the fitted line but in case of the logistic regression it represents the the logarithm of the ratio of odds of two consecutive observations. This means

$$b_1 = \log\left(\frac{odds_2}{odds_1}\right)$$

where $\log(odds_1)$ is the logarithm of the estimated odds when $x = x_j$ and $\log(odds_2)$ is the logarithm of the estimated odds when $x = x_{j+1}$
Hence on taking the antilog in the above expression we have the estimated ratio of odds, called the odds ratio denoted by $\widehat{OR}$ is equal to $\exp(b_1)$.
Therefore

$$\widehat{OR} = \frac{odds_2}{odds_1} = \exp(b_1)$$

For example, a regression coefficient $b_1 = -0.2$ with $\exp(b_1) = 0.82$ indicates that a change from x to $x + 1$ reduces the odds of occurrence by the multiplicative factor 0.82; it reduces the odds of occurrence by 18%. A value of $b_1 = 0$ and $\exp(b_1) = 1$ a change in the explanatory variable has no effect on the odds of occurrence.

Example

The board of directors of professionals association conducted a random sample survey of 30 members to assess the effects of several possible amount os dues increase. The table below is the result of the survey. X denotes the dollar increase in annual dues posited in the survey interview and $Y = 1$ if the interviewee indicated the membership will not be renewed at the amount of the dues increase and 0 denotes if the membership will be renewed.

y	x	y	x	y	x
0.0	30.0	1.0	30.0	0.0	30.0
0.0	31.0	0.0	32.0	0.0	33.0
1.0	34.0	0.0	35.0	0.0	35.0
1.0	35.0	1.0	36.0	0.0	37.0
0.0	38.0	1.0	39.0	0.0	40.0
1.0	40.0	1.0	40.0	0.0	41.0
1.0	42.0	1.0	43.0	1.0	44.0
0.0	45.0	1.0	45.0	1.0	45.0
0.0	46.0	1.0	47.0	1.0	48.0
0.0	49.0	1.0	50.0	1.0	50.0

Assuming logistic regression is appropriate we have the following maximum likelihood parameter estimates: $\hat{\beta}_0 = -4.8075$ and $\hat{\beta}_1 = 0.1251$. Hence, the fitted response function is given by

$$\hat{\pi} = [1 + \exp(4.8075 - 0.1251 x_i)]^{-1}$$

Example

```
> data=read.table("C://CS5900/membership.txt", header=T)
> y<-data$y
> x<-data$x
> model=glm(y~x,family=binomial(logit))
> model
Call:  glm(formula = y ~ x, family = binomial(logit))
Coefficients:
(Intercept)            x
    -4.8075       0.1251
Degrees of Freedom: 29 Total (i.e. Null);  28 Residual
Null Deviance:      41.46
Residual Deviance: 37.46        AIC: 41.46
```

We can display the fitted model using R code below

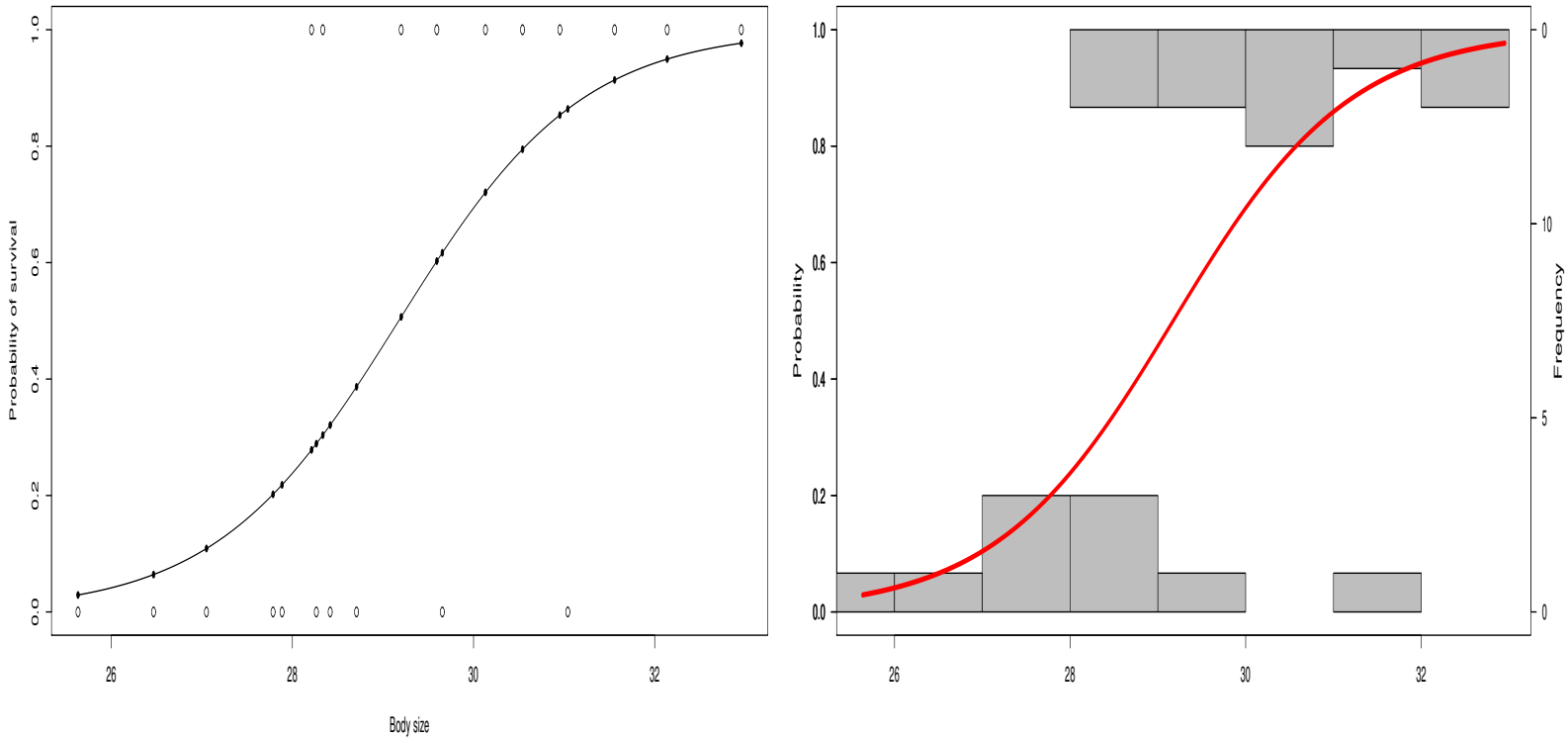
```
> plot(x,y)
> curve(predict(model,data.frame(x=x),type="resp"),add=TRUE)
> points(x,fitted(model),pch=20)
```

Example- Ecological Data

The survival response and body size conducted in a ecological study

```
> data
   bodysize survive
1  25.63320       0
2  26.46724       0
3  27.05107       0
4  27.78609       0
5  27.88550       0
6  28.21167       1
7  28.26452       0
8  28.33589       1
9  28.41717       0
10 28.70792       0
11 29.20096       1
12 29.59447       1
13 29.65526       0
14 30.13143       1
15 30.54080       1
16 30.95309       1
17 31.04085       0
18 31.55717       1
19 32.13806       1
20 32.95669       1
> bodysize=data$bodysize
> survive=data$survive
> plot(bodysize,survive,xlab="Body size",ylab="Probability of survival")
> g=glm(survive~bodysize,family=binomial)
> curve(predict(g,data.frame(bodysize=x),type="resp"),add=TRUE)
> points(bodysize,fitted(g),pch=20)
```

Example- Ecological Data

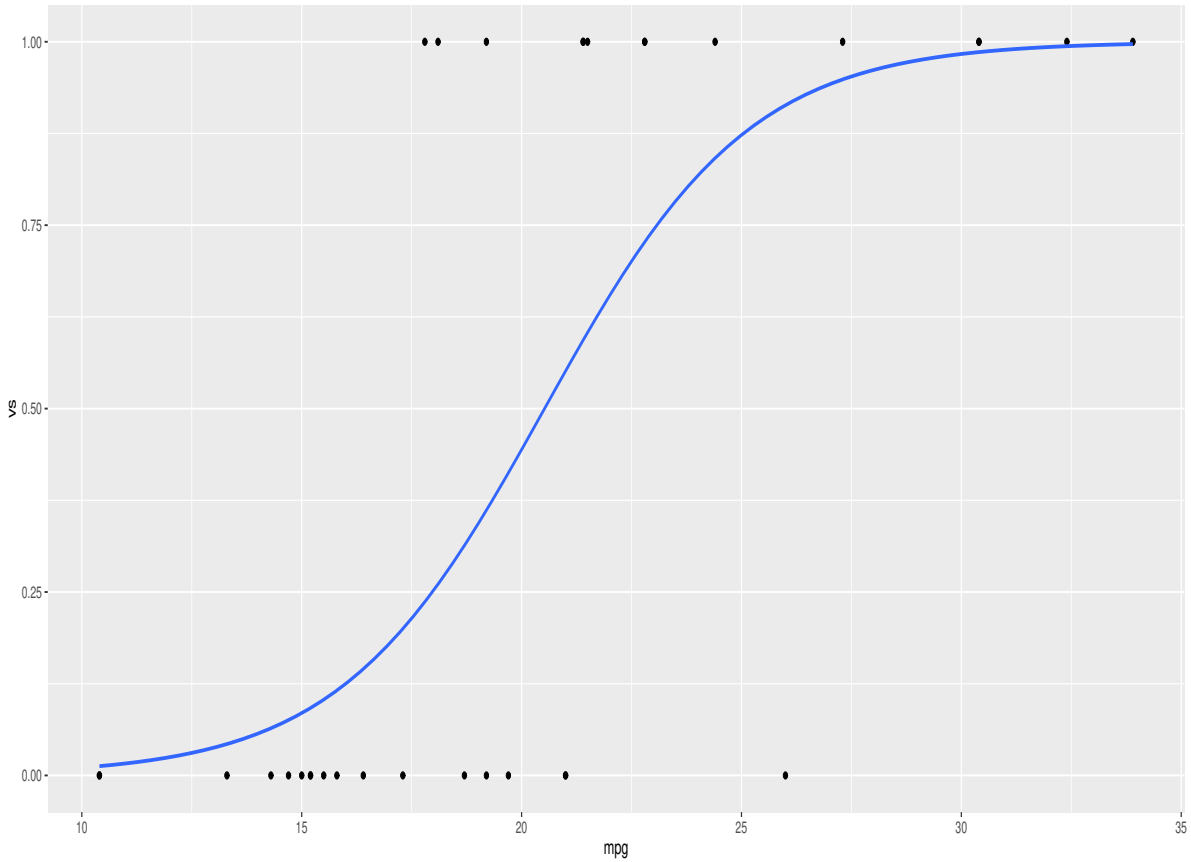


We can draw the logistic curve along with the histogram (displayed in the right above) with R code

```
>library(popbio)
>logi.hist.plot(bodysize,survive,boxp=FALSE,type="hist",col="gray")
```

Logistic model-ggplot

```
library(tidyverse)
data(mtcars)
logmodel<- glm(vs ~ mpg, data=mtcars, family=binomial(link="logit"))
ggplot(mtcars, aes(x=mpg, y=vs)) + geom_point()+
stat_smooth(method="glm", method.args=list(family="binomial"), se=TRUE)
```



Example-Prostate Cancer

Treatment for prostate cancer changes depending on whether or not the lymphatic nodes surrounding the protest are affected. In order to avoid an invasive investigation procedure (opening the abdominal cavity), a certain number of variables are considered as explanatory variables for the binary variable y: if y=0 the cancer has not reached the lymphatic system and if y=1 the cancer has reached the lymphatic system. A sample of 53 observations is available in the Brightspace.
The variables include
age: Age of the patient at the time of diagnosis
acid: Serum acid phosphate level
Xray: X-ray analysis, 0=negative, 1=positive
size: Tumor size
grade: State of the tumor as determined by biopsy, 0=medium, 1=serious
log.acid: Logarithm of the acidity level

```
> prostate=read.table("C:\\Users\\aryalg\\Data Sets\\Prostate.txt", header=T)
> head(data,5)
  age acid Xray size grade Y   log.acid
1  66 0.48    0    0     0 0 -0.7339692
2  68 0.56    0    0     0 0 -0.5798185
3  66 0.50    0    0     0 0 -0.6931472
4  56 0.52    0    0     0 0 -0.6539265
5  58 0.50    0    0     0 0 -0.6931472
```

Example-Prostate Cancer

Fitting logistic regression with different regressor variable

```
> model1=glm(Y~log.acid,data=prostate,family=binomial)
> summary(model1)
Call:
glm(formula = Y ~ log.acid, family = binomial, data = prostate)
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.9802  -0.9095  -0.7266   1.1951   1.7302
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)    0.404      0.509   0.794   0.4274
log.acid       2.245      1.040   2.159   0.0309 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
(Dispersion parameter for binomial family taken to be 1)
    Null deviance: 70.252  on 52  degrees of freedom
Residual deviance: 64.813  on 51  degrees of freedom
AIC: 68.813
```

Hence, the fitted model is

$$\log\left(\frac{\pi(x)}{1-\pi(x)}\right) = 0.404 + 2.245x$$

Equivalently,

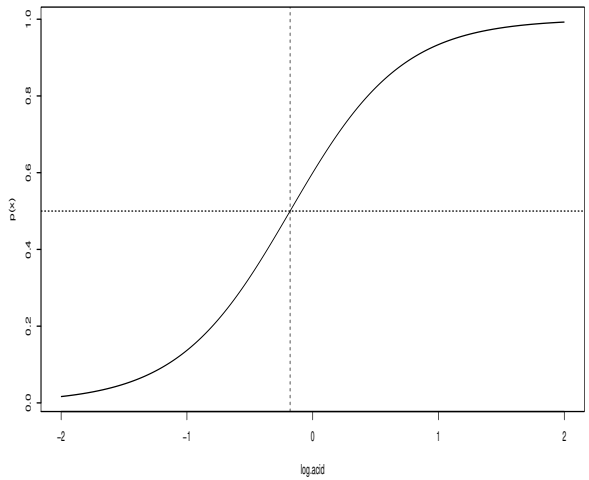
$$\pi(x) = \frac{\exp(0.404 + 2.245x)}{1 + \exp(0.404 + 2.245x)}.$$

Note that the p-value for testing the hypothesis $\beta_1 = 0$ Vs $\beta_1 \neq 0$ is 0.0309. Therefore the variable log.acid is retained in the model.

Example-Prostate Cancer

Fitting logistic regression with different regressor variable

```
> model1=glm(Y~log.acid,data=prostate,family=binomial)
> beta<-coef(model1)
> x<-seq(-2,2,0.01)
> y<- exp(beta[1]+beta[2]*x)/(1+exp(beta[1]+beta[2]*x))
> plot(x,y, type="l", xlab="log.acid", ylab="p(x)")
> abline(h=0.5, lty=2)
> xlim<--beta[1]/beta[2]
> abline(v=xlim, lty=2)
```



Observe that

$$\hat{p}(x) = \begin{cases} \leq 0.5 & \text{for} \quad x \leq -0.18 \\ > 0.5 & \text{for} \quad x > -0.18 \end{cases}$$

Performance of Logistic Regression Model

To evaluate the performance of a logistic regression model, we can consider a few metrics.

- AIC (Akaike Information Criteria): The analogous metric of adjusted R-squared in logistic regression is AIC. AIC is the measure of fit which penalizes model for the number of model coefficients. Therefore, we always prefer model with minimum AIC value.
- Null Deviance and Residual Deviance: Null Deviance indicates the response predicted by a model with nothing but an intercept. Lower the value, better the model. Residual deviance indicates the response predicted by a model on adding independent variables. Lower the value, better the model.
- Confusion Matrix: It is nothing but a tabular representation of Actual vs Predicted values. This helps us to find the accuracy of the model and avoid overfitting.
- We can calculate the accuracy of our model by

$$\frac{\text{True Positives} + \text{True Negatives}}{\text{True Positives} + \text{True Negatives} + \text{False Positives} + \text{False Negatives}}$$

Performance of Logistic Regression Model

- We can calculate the accuracy of our model by

$$\frac{\text{True Positives} + \text{True Negatives}}{\text{True Positives}+\text{True Negatives}+\text{False Positives} +\text{False Negatives}}$$

- From the confusion matrix we can calculate the Sensitivity and Specificity

$$\text{Sensitivity} = \frac{\text{True Poistives}}{\text{True Positives}+\text{False Negatives}}$$

and

$$\text{Specificity} = \frac{\text{True Negatives}}{\text{True Negatives} + \text{False Positives}}$$

Confusion Matrix

```
>data=read.table("C:\\Users\\aryalg\\Prostate.txt", header=T)
>attach(data)
>model1=glm(Y~log.acid,data=data,family=binomial)
>summary(model1)
>p=predict(model1, data = data, type = "response")
>library(caret)
>pp <- ifelse(p > 0.5 , 1, 0)
>confusionMatrix(data =factor(pp), reference=factor(Y),positive = "1")

          Reference
Prediction  0  1
         0 29 15
         1  4  5
               Accuracy : 0.6415
                 95% CI : (0.498, 0.7686)
    No Information Rate : 0.6226
    P-Value [Acc > NIR] : 0.44829
                  Kappa : 0.1444
    Mcnemar's Test P-Value : 0.02178
            Sensitivity : 0.25000
            Specificity : 0.87879
         Pos Pred Value : 0.55556
         Neg Pred Value : 0.65909
             Prevalence : 0.37736
         Detection Rate : 0.09434
   Detection Prevalence : 0.16981
      Balanced Accuracy : 0.56439

       'Positive' Class : 1
```

Interpreting the Confusion Matrix

The confusion matrix displays the following information:

- **True Positives (TP):** 5, the number of correctly predicted serious (class 1)
- **True Negatives (TN):** 29, the number of correctly predicted no-serious (class 0)
- **False Negatives (FN):** 15, the number of serious incorrectly predicted as non-serious (class 1)
- **False Positives (FP):** 4, the number of non-serious incorrectly predicted as serious (class 0)
- **Accuracy:** 64.15%, the proportion of correct predictions (both true positives and true negatives) among the total number of cases. (34/53).
- No Information Rate (NIR): 0.6226, the accuracy that could be obtained by always predicting the majority class (class 0 in this case). (33/53)
- **Kappa:** 0.1444, a metric that considers both the true positive rate and the false positive rate, providing a more balanced assessment of the model's performance. Kappa ranges from -1 to 1, with 0 indicating no better than random chance, and 1 indicating perfect agreement between predictions and true values.
- **Mcnemar's Test P-Value:** 0.02178, the p-value for a statistical test comparing the number of false positives and false negatives. A large p-value (typically greater than 0.05) indicates that there is no significant difference between the number of false positives and false negatives.

Sensitivity, Specificity, and Other Metrics

- **Sensitivity (Recall or True Positive Rate)** : 0.2500, the proportion of actual positive cases (serious cases) that were correctly identified by the model.
- **Specificity:** 0.87879, the proportion of actual negative cases (non-serious) that were correctly identified by the model.
- **Positive Predictive Value (PPV):** 0.55556, the proportion of positive predictions (predicted serious cases) that were actually positive (truely serious) (5/9).
- **Negative Predictive Value (NPV):** 0.65909, the proportion of negative predictions (predicted non-serious) that were actually negative (true non-serious).(29/44)
- **Prevalence:** 0.37736, the proportion of the true positive cases (serious cases) in the dataset. (20/53)
- **Detection Rate:** 0.09434, the proportion of true positive cases that were correctly detected by the model. (5/53)
- **Detection Prevalence:** 0.16981, the proportion of cases predicted as positive (serious) by the model. (9/53)
- **Balanced Accuracy:** 0.56439, the average of sensitivity and specificity, providing a balanced assessment of the model's performance across both classes.

ROC curve

Extending the two-by-two table of positive and negative, rather than selecting a single cutoff, we can examine the full range of cutoff values from 0 to 1. For each possible cutoff value, we can form a two-by-two table. Plotting the pairs of sensitivity and specificities (or, more often, sensitivity versus (1- specificity) on a scatter plot provides an ROC (Receiver Operating Characteristic) curve. The area under this curve (AUC of the ROC) provides an overall measure of fit of the model. The area under curve (AUC), referred to as index of accuracy(A) or concordance index, is a perfect performance metric for ROC curve. Higher the area under curve, better the prediction power of the model

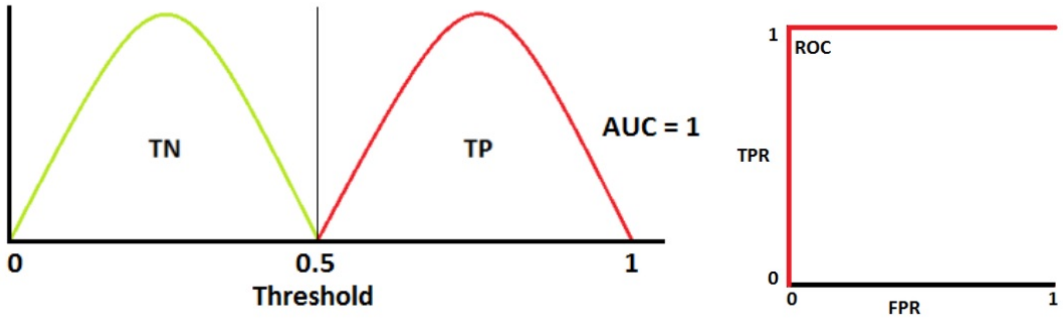
In particular, the AUC provides the probability that a randomly selected pair of subjects, one truly positive, and one truly negative, will be correctly ordered by the test.

An AUC of 1.0 indicates a perfect classifier, while an AUC of 0.5 suggests that the classifier is no better than random chance.

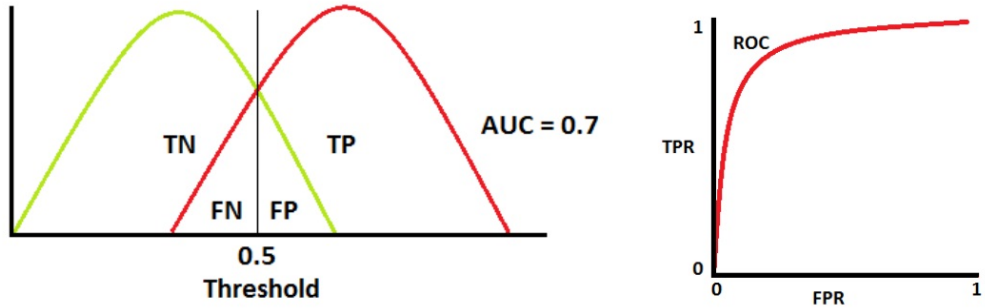
We can use `ROCR` package or `pROC` packages in R to draw the ROC curve.

ROC curve

When model is 100% accurate then AUC Score is 1. The curve will look like as below where red distribution curve is of the positive class and the green distribution curve is of the negative class. There is not overlapping between 2 classes. The ROC Curve and AUC looks like below where whole area comes under Area Under Curve

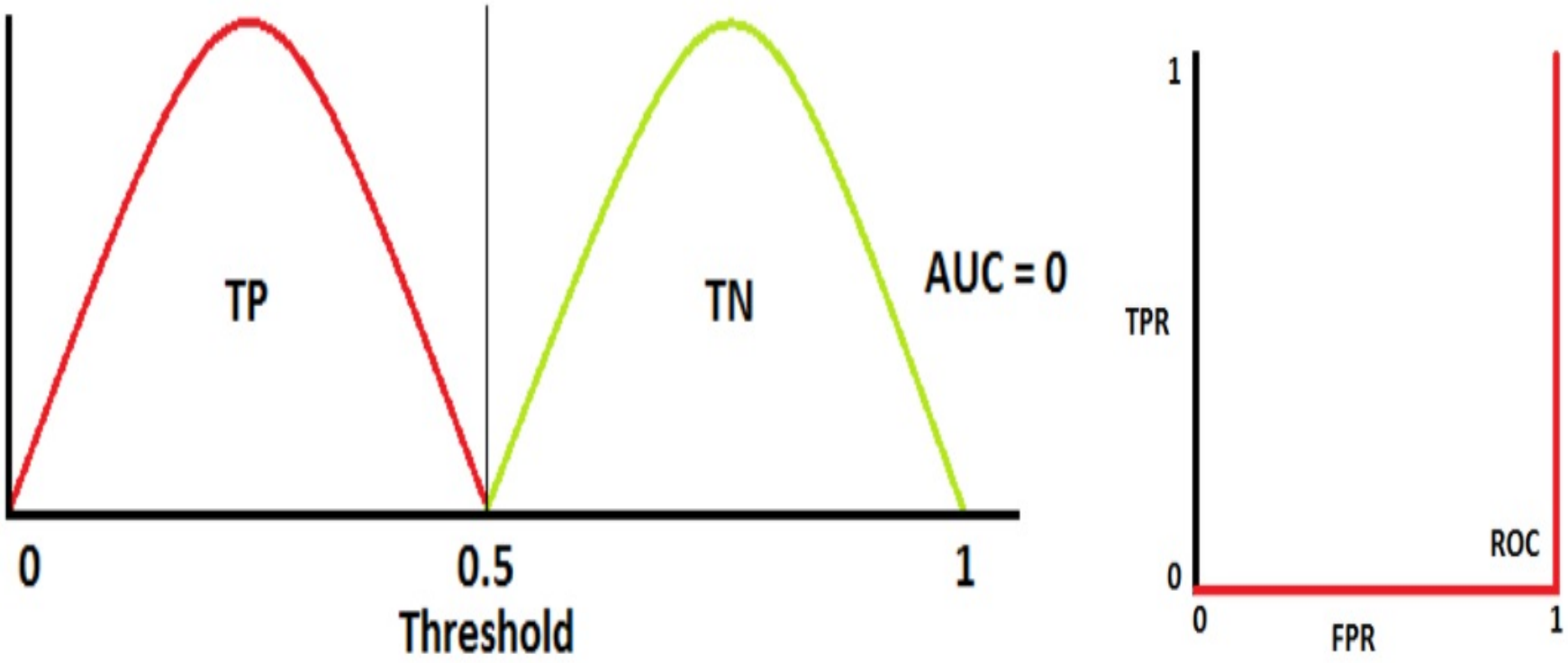


If there is some overlapping is there between 2 classes. some points are wrongly identified so FP(False Positive) class and FN(False Negative) classes overlapping. The ROC and AUC curve is shown below



ROC curve

When AUC is approximately 0, the model is actually reciprocating the classes. It means the model is predicting a negative class as a positive class and vice versa.

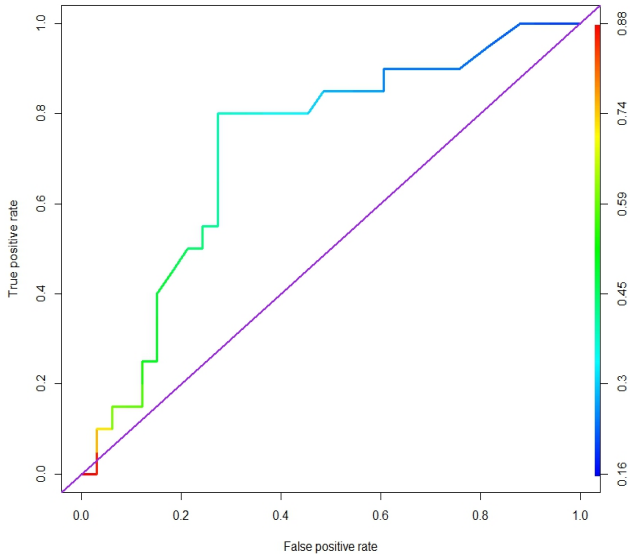


ROC curve-Example

```
> data=read.table("C:\\Users\\aryalg\\Prostate.txt", header=T)
> attach(data)
> head(data)
> model1=glm(Y~log.acid,data=data,family=binomial)
> p=predict(model1, data = data, type = "response")
> library(pROC)
> roc_obj <- roc(data$Y, p)
> plot(roc_obj)
> auc_value <- auc(roc_obj)
> auc_value
Area under the curve: 0.725
```

ROC curve

```
> library(ROCR)
> data=read.table("C:\\Users\\aryalg\\Prostate.txt", header=T)
> attach(data)
> model1=glm(Y~log.acid,data=data,family=binomial)
> p=predict(model1, data = data, type = "response")
> O <- prediction(p, data$Y)
> prf_model<- performance(O, measure = "tpr", x.measure = "fpr")
> plot(prf_model, colorize = TRUE, lwd=3)
> abline(0,1, lwd="2", col="purple")
```



Multiple Logistic Regression

The simple logistic regression model

$$\pi_i = [1 + \exp(-\beta_0 - \beta_1 x_i)]^{-1}$$

can be extended to more than one predictor variable. The multiple logistic regression model for $x_1, x_2, ..., x_{p-1}$ predictor variables can be expressed as

$$E(y) = \frac{exp(\mathbf{x}'\boldsymbol{\beta})}{1 + exp(\mathbf{x}'\boldsymbol{\beta})}$$

equivalently

$$E(y) = \left[1 + \exp(-\mathbf{x}'\boldsymbol{\beta})\right]^{-1}$$

where,

$$\boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{p-1} \end{pmatrix}, \mathbf{x} = \begin{pmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_{p-1} \end{pmatrix}, \mathbf{x}_i = \begin{pmatrix} 1 \\ x_{i1} \\ x_{i2} \\ \vdots \\ x_{i,p-1} \end{pmatrix}$$

The fitted logistic response function and the fitted values can be expressed as

$$\widehat{\pi} = \frac{\exp(\mathbf{x}'\mathbf{b})}{1 + exp(\mathbf{x}'\mathbf{b})} = [1 + \exp(-\mathbf{x}'\mathbf{b})]^{-1}$$

and

$$\widehat{\pi}_i = \frac{\exp(\mathbf{x}_i'\mathbf{b})}{1 + \exp(\mathbf{x}_i'\mathbf{b})} = [1 + \exp(-\mathbf{x}_i'\mathbf{b})]^{-1}$$

Example-Market Data

A marketing research firm was investigating whether a family will purchase a new car during next year using logistic regression. A random sample of 33 suburban families was selected. Data on annual family income(x_1 in thousand dollars) and the current age of the oldest family automobile (x_2, in years) was collected. A follow up interview conducted 12 months later was to determine whether the family actually purchased a new car $y = 1$ or didn't purchase a new car $y = 0$ during the year. The data is as below:

y	x_1	x_2	y	x_1	x_2
0.0	32.0	3.0	0.0	45.0	2.0
1.0	60.0	2.0	0.0	53.0	1.0
0.0	25.0	4.0	1.0	68.0	1.0
1.0	82.0	2.0	1.0	38.0	5.0
0.0	67.0	2.0	1.0	92.0	2.0
1.0	72.0	3.0	0.0	21.0	5.0
0.0	26.0	3.0	1.0	40.0	4.0
0.0	33.0	3.0	0.0	45.0	1.0
1.0	61.0	2.0	0.0	16.0	3.0
1.0	18.0	4.0	0.0	22.0	6.0
0.0	27.0	3.0	1.0	35.0	3.0
1.0	40.0	3.0	0.0	10.0	4.0
0.0	24.0	3.0	1.0	15.0	4.0
0.0	23.0	3.0	0.0	19.0	5.0
1.0	22.0	2.0	0.0	61.0	2.0
0.0	21.0	3.0	1.0	32.0	5.0
0.0	17.0	1.0			

Example- Market Data

Considering that a multiple logistic regression model fits the data we have the maximum likelihood estimates of the parameters β_0, β_1 and β_2 are

$$\begin{aligned} b_0 &= \hat{\beta}_0 = -4.7393 \\ b_1 &= \hat{\beta}_1 = 0.0677 \\ b_2 &= \hat{\beta}_2 = 0.5986 \end{aligned}$$

Therefore the estimated regression model is

$$\hat{\pi} = [1 + exp(4.7393 - 0.0677x_1 - 0.5986x_2)]^{-1}$$

```
> data=read.table("C://CS 59000//market.txt", header=T)
> y<-data$y
> x1<-data$x1
> x2<-data$x2
> x1
 [1] 32 45 60 53 25 68 82 38 67 92 72 21 26 40 33 45 61 16 18 22 27 35 40 10 24  15 23 19 22 61 21 32 17
> x2
 [1] 3 2 2 1 4 1 2 5 2 2 3 5 3 4 3 1 2 3 4 6 3 3 3 4 3 4 3 5 2 2 3 5 1
> y
 [1] 0 0 1 0 0 1 1 1 0 1 1 0 0 1 0 0 1 0 1 0 0 1 1 0 0 1 0 0 1 0 0 1 0
> model=glm(y~x1+x2, family=binomial(logit))
> summary(model)
Call:
glm(formula = y ~ x1 + x2, family = binomial(logit))
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.73931    2.10195  -2.255   0.0242 *
x1           0.06773    0.02806   2.414   0.0158 *
x2           0.59863    0.39007   1.535   0.1249
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Example-Prostate Cancer (Data Provided in the Brightspace)

Treatment for prostate cancer changes depending on whether or not the lymphatic nodes surrounding the protest are affected. In order to avoid an invasive investigation procedure (opening the abdominal cavity), a certain number of variables are considered as explanatory variables for the binary variable y: if y=0 the cancer has not reached the lymphatic system and if y=1 the cancer has reached the lymphatic system. A sample of 53 observations is available in Brightspace
The variables include
age: Age of the patient at the time of diagnosis
acid: Serum acid phosphate level
Xray: X-ray analysis, 0=negative, 1=positive
size: Tumor size
grade: State of the tumor as determined by biopsy, 0=medium, 1=serious
log.acid: Logarithm of the acidity level

```
> data=read.table("C:\\Users\\aryalg\\Data Sets\\Prostate.txt", header=T)
> head(data,5)
  age acid Xray size grade Y   log.acid
1  66 0.48    0    0     0 0 -0.7339692
2  68 0.56    0    0     0 0 -0.5798185
3  66 0.50    0    0     0 0 -0.6931472
4  56 0.52    0    0     0 0 -0.6539265
5  58 0.50    0    0     0 0 -0.6931472
```

Example-Prostate Cancer

Logistic regression with all explanatory variables can be expressed as

```
> model3=glm(Y~.,data=prostate,family=binomial)
> summary(model3)

Call:
glm(formula = Y ~ ., family = binomial, data = prostate)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0960  -0.6102  -0.2863   0.4834   2.2000

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) 10.08672    7.83450   1.287   0.1979
age         -0.04289    0.06166  -0.696   0.4867
acid        -8.48006    7.63305  -1.111   0.2666
Xray         2.06673    0.85469   2.418   0.0156 *
size         1.38415    0.79546   1.740   0.0819 .
grade        0.85376    0.81247   1.051   0.2933
log.acid     9.60912    6.21652   1.546   0.1222
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 70.252  on 52  degrees of freedom
Residual deviance: 44.768  on 46  degrees of freedom
AIC: 58.768

Number of Fisher Scoring iterations: 5
```

Choosing the Model

The anova function can be used to compare two models using the deviance statistic to test the nullity of the coefficients of the bigger model.
For example in the prostate cancer data

```
> prostate=read.table("C:\\Users\\aryalg\\Data Sets\\Prostate.txt", header=T)
> model1=glm(Y~log.acid,data=prostate,family=binomial)
> model3=glm(Y~.,data=prostate,family=binomial)
> anova(model1,model3, test="Chisq")
Analysis of Deviance Table

Model 1: Y ~ log.acid
Model 2: Y ~ age + acid + Xray + size + grade + log.acid
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1        51     64.813
2        46     44.768  5   20.044 0.001226 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

As the p-value of the test is 0.001226, the null hypothesis is rejected. Thus the model 1 is rejected and at least one of the supplementary variable in model 3 (full model) is considered to be relevant. Like in the multiple linear regression model we can use `step` function to choose the best model.

```
> choose<-step(model, direction="backward")
```

Making a Prediction

Once a best logistic regression has been selected it can be used for prediction using the function `predict`.

```
> prostate=read.table("C:\\Users\\aryalg\\Data Sets\\Prostate.txt", header=T)
```

```
> model=glm(Y~., data=prostate,family="binomial")
> bestmodel=step(model, direction="backward")
> xnew<-matrix(c(61,0.60,1,0,1,-0.51),nrow=1)
> colnames(xnew)<-c("age","acid","Xray","size","grade","log.acid")
> xnew=as.data.frame(xnew)
> predict(bestmodel,xnew, type="response")
        1
0.4835626
```

Since the predicted probability is less than 0.5 we therefore predict $\hat{Y} = 0$, that is to say the cancer has not reached the lymphatic system.

Differences Between Linear and Logistic Regression

Linear Regression	**Logistic Regression**
Linear regression is used to predict the continuous dependent variable using a given set of independent variables.	Logistic regression is used to predict the categorical dependent variable using a given set of independent variables.
Linear regression is used for solving regression problem.	It is used for solving classification problems.
In this we predict the value of continuous variables In this we find best fit line.	In this we predict values of categorical variables In this we find S-Curve.
Least square estimation method is used for estimation of accuracy.	Maximum likelihood estimation method is used for Estimation of accuracy.
The output must be continuous value, such as price, age, etc.	Output must be categorical value such as 0 or 1, Yes or no, etc.
`> lm(y~x)`	`> glm(y~x, family=binomial)`

Softmax Regression

Softmax regression is also called as multinomial logistic regression and it is a generalization of logistic regression. Softmax regression is used to model categorical dependent variables and the categories must not have any order (or rank). For dependent variables which have order, we need to use ordinal logistic regression which we will discuss some other time. For example, rather than classifying emails as spam or ham we may want to classify as spam, work related email and personal email. Similarly we may classify the likelihood of admitting of high school students in one of the program: academic, general and vocational etc. Softmax function falls under the family of exponential family distribution. In softmax regression, we replace the sigmoid function from the logistic regression by the so-called softmax function. This function takes a vector of n real numbers as input and normalizes the vector into a distribution of n probabilities. That is, the function transforms all the n components from any real values to value in the interval (0,1).

Example

Consider the data related to the entering high-school students who make program choices among general programs, vocational programs and academic programs. Their choices might be modeled using their writing scores and social economic status. The data sets is available at `https://stats.idre.ucla.edu/stat/data/hsbdemo.csv`

```
> hsbdemo=read.csv("https://stats.idre.ucla.edu/stat/data/hsbdemo.csv")
> head(hsbdemo)
   id female    ses schtyp     prog read write math science socst       honors awards cid
1  45 female    low public vocation   34    35   41      29    26 not enrolled      0   1
2 108   male middle public  general   34    33   41      36    36 not enrolled      0   1
3  15   male   high public vocation   39    39   44      26    42 not enrolled      0   1
4  67   male    low public vocation   37    37   42      33    32 not enrolled      0   1
5 153   male middle public vocation   39    31   40      39    51 not enrolled      0   1
6  51 female   high public  general   42    36   42      31    39 not enrolled      0   1
> with(hsbdemo, table(ses,prog))
        prog
ses      academic general vocation
  high         42       9        7
  low          19      16       12
  middle       44      20       31
> with(hsbdemo, do.call(rbind,tapply(write,prog,function(x)c(M=mean(x), SD=sd(x)))))
                M       SD
academic 56.25714 7.943343
general  51.33333 9.397775
vocation 46.76000 9.318754
```

Example-Continued

```
> model=multinom(prog~ses+write, data=hsbdemo)
# weights:  15 (8 variable)
initial  value 219.722458
iter  10 value 179.983731
final  value 179.981726
converged
> summary(model)
Call:
multinom(formula = prog ~ ses + write, data = hsbdemo)
Coefficients:
          (Intercept)    seslow sesmiddle       write
general     1.689478 1.1628411 0.6295638 -0.05793086
vocation    4.235574 0.9827182 1.2740985 -0.11360389

Std. Errors:
          (Intercept)    seslow sesmiddle      write
general     1.226939 0.5142211 0.4650289 0.02141101
vocation    1.204690 0.5955688 0.5111119 0.02222000

Residual Deviance: 359.9635
AIC: 375.9635
```

Example-Contd

Note that the first row of the output compares prog=general with the prog=academic and the second row represents the comparision of prog=vocation and the prog=academic.
Let us declare the coefficients from the first row to be b_1 (b_{10} for the intercept, b_{11} for "seslow", b_{12} for "sesmiddle" and b_{13} for "write") and our coefficients for the second row to be b_2 (b_{20} for the intercept, b_{21} for "seslow", b_{22} for "sesmiddle" and b_{23} for "write"). Using these vlaues we can compute the log odds, which compares a given model to the baseline and informs us how a one unit change in an independent variable would change a dependent variable in the model compare to how it would change the same variable for the baseline. The equations can be written as

$$\ln\left(\frac{P(prog = general)}{P(prog = academic)}\right) = b_{10} + b_{11}(ses = 2) + b_{12}(ses = 3) + b_{13}(write)$$

$$\ln\left(\frac{P(prog = vocation)}{P(prog = academic)}\right) = b_{20} + b_{21}(ses = 2) + b_{22}(ses = 3) + b_{23}(write)$$

Example-Contd.

Based on the computer output and the equations, we can find out that a one-unit increase in the variable "write" is associated with the decrease in the log odds of being in "general program" versus "academic" program in the amount of 0.058.
Usually we refer "Relative Risk" in this kind of studies. The relative risk is the ratio of the probability of choosing any outcome other than baseline over that of the baseline category. To find the relative risk, we can exponent the coefficient from our model.

```
> exp(coef(model))
         (Intercept)   seslow sesmiddle     write
general     5.416653 3.199009  1.876792 0.9437152
vocation   69.101326 2.671709  3.575477 0.8926115
```

The relative risk ratio for one-unit increase in the variable "write" is 0.9437 for being in "general" program versus "academic" program.

Poisson Regression

Data is often collected in counts (e.g. the number of heads in 12 flips of a coin or the number of car thefts in a city during a year). Many discrete response variables have counts as possible outcomes. Binomial counts are the number of successes in a fixed number of trials, n. Poisson counts are the number of occurrences of some event in a certain interval of time (or space). While Binomial counts only take values between 0 and n, Poisson counts have no upper bound. We now consider a nonlinear regression model where the response outcomes are discrete counts that follow a Poisson distribution. Poisson regression provides a model that describes how the mean response, changes as a function of one or more explanatory variables. The logarithm of the mean can be modeled as a linear function of observed covariates. The result is a generalized linear model with Poisson response and link log. Poisson regression is a natural choice when the response variable is a small integer. The explanatory variables model the mean of the response variable.

Poisson Regression

We have n observations of a response random variable y, and $p > 1$ fixed explanatory variables, $x_1, x_2, \cdots, x_p$. We assume that for observation $i = 1, 2, \cdots, n$,

$$y_i \sim Poi(\mu_i)$$

where

$$\mu_i = \exp(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip})$$

Thus, our generalized linear model has a Poisson random component, a linear systematic component $\beta_0 + \beta_1 x_1 + \cdots \beta_p x_p$ and a log link

$$\log(\mu) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$$

A consequence of using the log link function with a linear predictor is that the explanatory variables affect the response mean multiplicatively. Consider a Poisson regression model with one explanatory variable: $\mu(x) = \exp(\beta_0 + \beta_1 x)$, where our notation emphasizes that μ changes as a function of x. When we increase the explanatory variable by c units, the result is
$\mu(x + c) = \exp(\beta_0 + \beta_1(x + c)) = \mu(x)\exp(c\beta_1)$. Thus, the ratio of the means at $x + c$ and at x is

$$\frac{\mu(x + c)}{\mu(x)} = \frac{\exp(\beta_0 + \beta_1(x + c))}{\exp(\beta_0 + \beta_1 x)} = \exp(c\beta_1).$$

Example

School administrators study the attendance behavior of high school juniors at two schools. Predictors of the number of days of absence include gender of the student and standardized test scores in math and language arts.
VARIABLES:
Daysabs: days absent
Math: standardized test scores
Langarts: standardized test scores
Male: gender (1=male, 0=female)

```
absence=read.csv("H://CS 59000//absence.csv", header=T)
head(absence)
tail(absence)
attach(absence)
model=glm(daysabs~math+langarts+male, family=poisson)
```