# Lab Notes

*Chapter 9*

*OpenIntro Biostatistics*

## Overview

1. Simple Logistic Regression
    - *OI Biostat* Section 9.xx
2. Multiple Logistic Regression
    - *OI Biostat* Section 9.xx

Lab 1 inroduces simple logistic regression, a model for the association of a binary response variable with a single predictor variable.

Lab 2 discusses multiple logistic regression, an extension of simple logistic regression that allows for several predictors. The use of the Akaike Information Criterion as a metric for model selection is also discussed.

# Lab 1: Simple Logistic Regression

### Fitting a Logistic Regression Model

The **glm()** function is used to fit logistic regression models. It has the following generic structure:

```
glm(y ~ x, data, family = binomial(link = "logit"))
```

where the first argument specifies the variables used in the model; in this example, the model regresses a response variable y against an explanatory variable x. The second argument is used only when the dataframe name is not already specified in the first argument. Running the function creates an *object* (of class 'lm' and 'glm') that contains several components, such as the model coefficients. The model coefficients are directly displayed upon running glm(), while other components can be accessed through either the $ notation or specific functions like summary(). The argument family = binomial(link = "logit") is specific to logistic regression; the texttt{glm()} function is capable of running families of general linear models that are not discussed in this course.

The following example shows fitting a linear model that predicts the estimated log odds of death before discharge from resting heart rate, using data from icu.

```
#load the data
library(aplore3)
data("icu")

#fitting logistic model
glm(sta ~ hra, data = icu, family = binomial(link = "logit"))
```

```
##
## Call:  glm(formula = sta ~ hra, family = binomial(link = "logit"), data = icu)
##
## Coefficients:
## (Intercept)          hra
##   -1.679129     0.002941
##
## Degrees of Freedom: 199 Total (i.e. Null);  198 Residual
## Null Deviance:       200.2
## Residual Deviance: 200   AIC: 204
```

To fit a linear model that predicts the estimated log odds of survival to discharge from resting heart rate, it is necessary to relevel the factor sta such that a 1 corresponds to individuals who survived to discharge. This can be accomplished with **factor()** and **rev()**. The rev() function reverses elements. In the example below, applying rev) to a vector {1, 2, 3} produces a vector {3, 2, 1}.

```
#check levels
levels(icu$sta)
```

```
## [1] "Lived" "Died"
```

```
#relevel survival
icu$sta = factor(icu$sta, levels = rev(levels(icu$sta)))
```

```
#check levels
levels(icu$sta)
```

```
## [1] "Died"  "Lived"
```

```
#example of using rev()
a = c(1, 2, 3)
rev(a)
```

```
## [1] 3 2 1
```

The following example shows outputting the model summary, selectively outputting model coefficients from the model fit, and extracting the numeric value of a coefficient.

```
#name the model
model.hra = glm(sta ~ hra, data = icu, family = binomial(link = "logit"))

#model summary
summary(model.hra)
```

```
##
## Call:
## glm(formula = sta ~ hra, family = binomial(link = "logit"), data = icu)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8524   0.6339   0.6579   0.6784   0.7533
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.679129   0.679863   2.470   0.0135 *
## hra         -0.002941   0.006552  -0.449   0.6535
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 200.16  on 199  degrees of freedom
## Residual deviance: 199.96  on 198  degrees of freedom
## AIC: 203.96
##
## Number of Fisher Scoring iterations: 4
```

```
#model summary of coefficients
summary(model.hra)$coef
```

```
##                 Estimate  Std. Error    z value   Pr(>|z|)
## (Intercept)  1.679128937 0.679862734  2.4698058 0.01351864
## hra         -0.002941381 0.006552235 -0.4489127 0.65349464
```

```r
#extract value of slope coefficient
coef(model.hra)[2]
```

```
##          hra
## -0.002941381
```

As in linear regression, the `predict()` function can be used to evaluate the regression equation for specific values of a predictor variable. The following example shows predicting the estimated log odds of survival to discharge for an individual with resting heart rate of 98 bpm.

```r
predict(model.hra, newdata = data.frame(hra = 98))
```

```
##        1
## 1.390874
```

# Lab 2: Multiple Logistic Regression

### Working with Several Predictors

The **glm()** function is used to fit linear models. It has the following generic structure:

```
glm(y ~ x1 + x2, data, family = binomial(link = "logit"))
```

where the first argument specifies the variables used in the model; in this example, the model regresses a response variable y against two explanatory variables x1 and x2. Additional predictor variables can be added to the model formula with the + symbol, and an interaction between two variables is specified with the * symbol.

The following example shows fitting a linear model that predicts the estimated log odds of survival to discharge from age and gender, and a linear model that predicts the estimated log odds of survival to discharge from age, gender, and their interaction.

```
#fitting model with age and gender
glm(sta ~ age + gender, data = icu, family = binomial(link = "logit"))
```

```
##
## Call:  glm(formula = sta ~ age + gender, family = binomial(link = "logit"),
##     data = icu)
##
## Coefficients:
##  (Intercept)           age  genderFemale
##      3.05669      -0.02758       0.01131
##
## Degrees of Freedom: 199 Total (i.e. Null);  197 Residual
## Null Deviance:       200.2
## Residual Deviance: 192.3     AIC: 198.3
```

```
#fitting model with age, gender, and an interaction term
glm(sta ~ age*gender, data = icu, family = binomial(link = "logit"))
```

```
##
## Call:  glm(formula = sta ~ age * gender, family = binomial(link = "logit"),
##     data = icu)
##
## Coefficients:
##      (Intercept)               age      genderFemale  age:genderFemale
##        3.0762954        -0.0279007        -0.0388512         0.0007774
##
## Degrees of Freedom: 199 Total (i.e. Null);  196 Residual
## Null Deviance:       200.2
## Residual Deviance: 192.3     AIC: 200.3
```

### Calculating AIC

The AIC of a logistic model can be extracted from `summary()` or computed via the **`AIC()`** function.

The following example shows how to output the AIC from the model predicting estimated odds of survival to discharge from resting heart rate.

```r
#use summary()$aic
summary(model.hra)$aic
```

```
## [1] 203.9604
```

```r
#use AIC()
AIC(model.hra)
```

```
## [1] 203.9604
```

### Collapsing Factor Levels

The `factor()` function can also be used to collapse levels of a factor.

The following example shows the re-defining of the levels of `loc`; the variable initially has three levels (`Nothing`, `Stupor`, and `Coma`). The levels `Stupor` and `Coma` can be combined into a single level `Unconscious`, while the level `Nothing` is renamed `Conscious`.

```r
#view levels of loc
levels(icu$loc)
```

```
## [1] "Nothing" "Stupor"  "Coma"
```

```r
#create the loc.binary variable
icu$loc.binary = icu$loc

#redefine the factor levels of loc.binary
levels(icu$loc.binary) = list("Conscious" = "Nothing",
                              "Unconscious" = c("Stupor", "Coma"))

#view levels of loc.binary
levels(icu$loc.binary)
```

```
## [1] "Conscious"   "Unconscious"
```

```r
#compare tables
table(icu$loc); table(icu$loc.binary)
```

```
##
## Nothing  Stupor    Coma
##     185       5      10
```

```
##
##   Conscious Unconscious
##         185          15
```