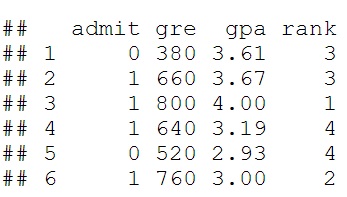
Example1

A researcher is interested in how variables, such as GRE (Graduate Record Exam scores), GPA (grade point average) and prestige of the undergraduate institution, effect admission into graduate school. The response variable, admit/don't admit, is a binary variable. This dataset has a binary response (outcome, dependent) variable called **admit**. There are three predictor variables: **gre**, **gpa** and **rank**. We will treat the variables **gre** and **gpa** as continuous. The variable **rank** takes on the values 1 through 4. Institutions with a rank of 1 have the highest prestige, while those with a rank of 4 have the lowest.



**Using the logit model**

The code below estimates a logistic regression model using the glm (generalized linear model) function. First, we convert rank to a factor to indicate that rank should be treated as a categorical variable.

**install.packages("packagename")**

**library**(aod)

**library**(ggplot2)

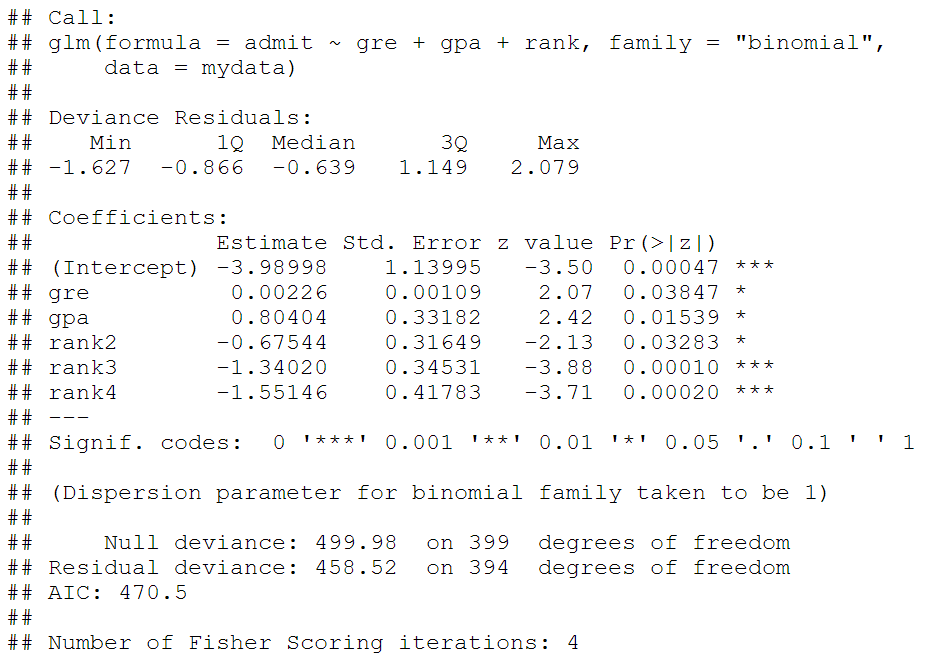
**library**(Rcpp)

mydata$rank <- **factor**(mydata$rank)

mylogit <- **glm**(admit ~ gre + gpa + rank, data = mydata, family = "binomial")

Since we gave our model a name (mylogit), R will not produce any output from our regression. In order to get the results we use the summary command:

summary(mylogit)

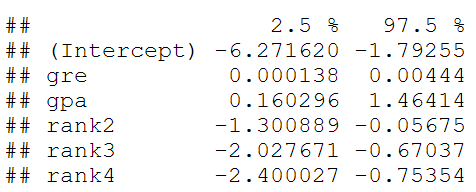


* In the output above, the first thing we see is the call, this is R reminding us what the model we ran was, what options we specified, etc.
* Next we see the deviance residuals, which are a measure of model fit. This part of output shows the distribution of the deviance residuals for individual cases used in the model. Below we discuss how to use summaries of the deviance statistic to assess model fit.
* The next part of the output shows the coefficients, their standard errors, the z-statistic (sometimes called a Wald z-statistic), and the associated p-values. Both **gre** and **gpa** are statistically significant, as are the three terms for **rank**. The logistic regression coefficients give the change in the log odds of the outcome for a one unit increase in the predictor variable.
* For every one unit change in **gre**, the log odds of admission (versus non-admission) increases by **0.002**.
* For a one unit increase in**gpa**, the log odds of being admitted to graduate school increases by **0.804**.
* The indicator variables for **rank** have a slightly different interpretation. For example, having attended an undergraduate institution with **rank** of 2, versus an institution with a **rank** of 1, changes the log odds of admission by **-0.675**.

We can use the confint function to obtain confidence intervals for the coefficient estimates. Note that for logistic models, confidence intervals are based on the profiled log-likelihood function. We can also get CIs based on just the standard errors by using the default method.

**## CIs using profiled log-likelihood**

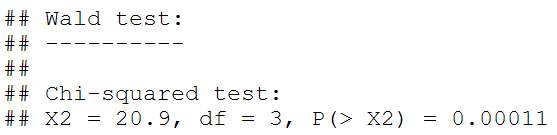
**confint(mylogit)**



**Test for an overall effect of rank using the wald.test function of the aod library.**

We use the wald.test function. b supplies the coefficients, while Sigma supplies the variance covariance matrix of the error terms, finallyTerms tells R which terms in the model are to be tested, in this case, terms 4, 5, and 6, are the three terms for the levels of rank.

**wald.test(b = coef(mylogit), Sigma = vcov(mylogit), Terms = 4:6)**

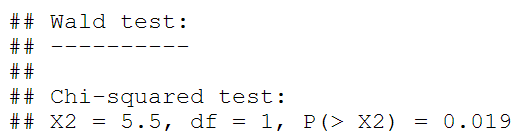


The chi-squared test statistic of 20.9, with three degrees of freedom is associated with a p-value of 0.00011 indicating that the overall effect of rank is statistically significant.

test additional hypotheses about the differences in the coefficients for the different levels of rank. Below we test that the coefficient for rank=2 is equal to the coefficient for rank=3. The first line of code below creates a vector l that defines the test we want to perform. In this case, we want to test the difference (subtraction) of the terms for rank=2 and rank=3 (i.e., the 4th and 5th terms in the model). To contrast these two terms, we multiply one of them by 1, and the other by -1. The other terms in the model are not involved in the test, so they are multiplied by 0. The second line of code below uses L=l to tell R that we wish to base the test on the vector l (rather than using the Terms option as we did above).

**l <- cbind(0,0,0,1,-1,0)**

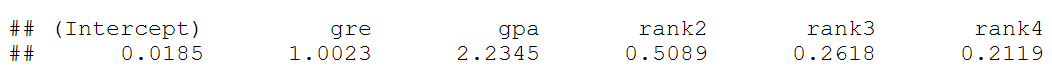
**wald.test(b = coef(mylogit), Sigma = vcov(mylogit), L = l)**



The chi-squared test statistic of 5.5 with 1 degree of freedom is associated with a p-value of 0.019, indicating that the difference between the coefficient forrank=2 and the coefficient for rank=3 is statistically significant.

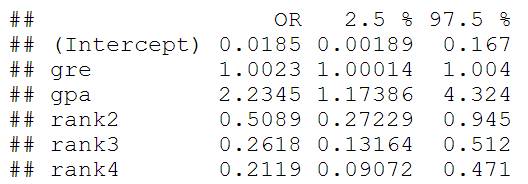
**## odds ratios only**

**exp(coef(mylogit))**



*## odds ratios and 95% CI*

**exp(cbind(OR = coef(mylogit), confint(mylogit)))**



Now we can say that for a one unit increase in **gpa**, the odds of being admitted to graduate school (versus not being admitted) increase by a factor of 2.23.

Prediction

We will start by calculating the predicted probability of admission at each value of rank, holding **gre** and **gpa** at their means. First we create and view the data frame.

newdata1 <- **with**(mydata,

**data.frame**(gre = **mean**(gre), gpa = **mean**(gpa), rank = **factor**(1:4)))

*## view data frame*

newdata1

## gre gpa rank

## 1 588 3.39 1

## 2 588 3.39 2

## 3 588 3.39 3

## 4 588 3.39 4

The **newdata1$rankP**tells R that we want to create a new variable in the dataset (data frame) **newdata1** called **rankP**, the rest of the command tells R that the values of **rankP** should be predictions made using the**predict( )** function.

newdata1$rankP <- **predict**(mylogit, newdata = newdata1, type = "response")

newdata1

## gre gpa rank rankP

## 1 588 3.39 1 0.517

## 2 588 3.39 2 0.352

## 3 588 3.39 3 0.219

## 4 588 3.39 4 0.185

In the above output we see that the predicted probability of being accepted into a graduate program is 0.52 for students from the highest prestige undergraduate institutions (**rank**=1), and 0.18 for students from the lowest ranked institutions (**rank**=4), holding **gre** and **gpa** at their means.

We may also wish to see measures of how well our model fits. This can be particularly useful when comparing competing models. The output produced by**summary(mylogit)** included indices of fit (shown below the coefficients), including the null and deviance residuals and the AIC. One measure of model fit is the significance of the overall model. This test asks whether the model with predictors fits significantly better than a model with just an intercept (i.e., a null model). The test statistic is the difference between the residual deviance for the model with predictors and the null model. The test statistic is distributed chi-squared with degrees of freedom equal to the differences in degrees of freedom between the current and the null model (i.e., the number of predictor variables in the model).

To find the difference in deviance for the two models (i.e., the test statistic) we can use the command:

**with**(mylogit, null.deviance - deviance)

## [1] 41.5

The degrees of freedom for the difference between the two models is equal to the number of predictor variables in the mode, and can be obtained using:

**with**(mylogit, df.null - df.residual)

## [1] 5

Finally, the p-value can be obtained using:

**with**(mylogit, **pchisq**(null.deviance - deviance, df.null - df.residual, lower.tail = FALSE))

## [1] 7.58e-08

The chi-square of 41.46 with 5 degrees of freedom and an associated p-value of less than 0.001 tells us that our model as a whole fits significantly better than an empty model. This is sometimes called a likelihood ratio test (the deviance residual is -2\*log likelihood). To see the model's log likelihood, we type:

**logLik**(mylogit)

## 'log Lik.' -229 (df=6)

example 2

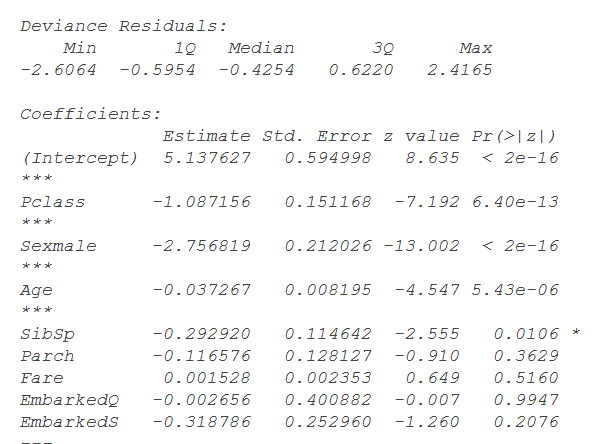
dataset

The dataset (training) is a collection of data about some of the passengers (889 to be precise), and the goal of the competition is to predict the survival (either 1 if the passenger survived or 0 if they did not) based on some features such as the class of service, the sex, the age etc. As you can see, we are going to use both categorical and continuous variables.

model fitting

model <- glm(Survived ~.,family=binomial(link='logit'),data=train)

summary(model)

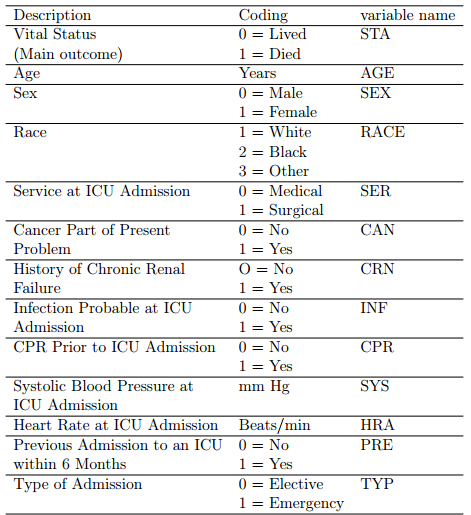


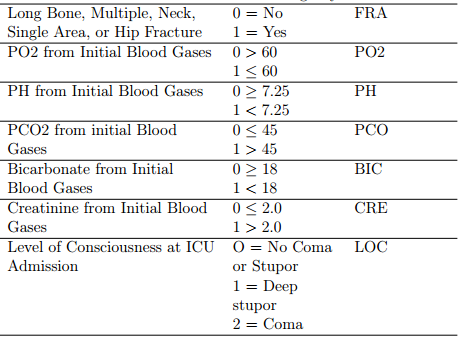
interpret

Now we can analyze the fitting and interpret what the model is telling us.  
First of all, we can see that SibSp, Fare and Embarked are not statistically significant. As for the statistically significant variables, sex has the lowest p-value suggesting a strong association of the sex of the passenger with the probability of having survived. The negative coefficient for this predictor suggests that all other variables being equal, the male passenger is less likely to have survived. Remember that in the logit model the response variable is log odds: ln(odds) = ln(p/(1-p)) = a\*x1 + b\*x2 + … + z\*xn. Since male is a dummy variable, being male reduces the log odds by 2.75 while a unit increase in age reduces the log odds by 0.037.

Logistic regression with dummy or indicator variables

there are 20 variables in this data set, which we describe in the table below





The main outcome is vital status, alive or dead, coded as 0/1 respectively, under the

variable name sta. For this illustrative example, we will investigate the effect of the

dichotomous variables sex, ser, and loc.

# read the data into R

> icu.dat <- read.table(file="g:\\icudat.txt", header = T)

# Take a quick look at the data

> summary(icu.dat)

# Create the subset of variables we need

> icu1.dat <- data.frame(sta=icu.dat$sta, loc=icu.dat$loc,8

sex=icu.dat$sex, ser=icu.dat$ser)

# Look at reduced data set

> summary(icu1.dat)

# Notice that loc, sex, and ser need to be made into factor variables

icu1.dat <- data.frame(sta=icu.dat$sta, loc=as.factor(icu.dat$loc),

sex=as.factor(icu.dat$sex), ser=as.factor(icu.dat$ser))

# Look at reduced data set again, this time with factor variables

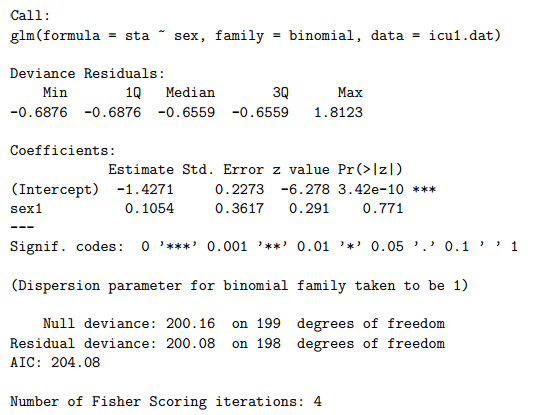
> summary(icu1.dat)

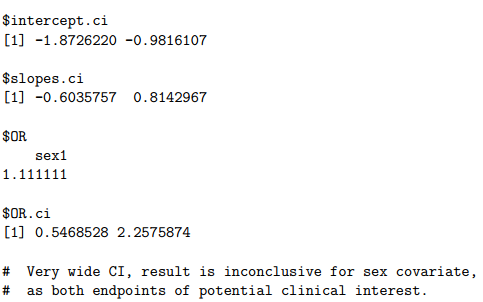
# Simple logistic regression for each variable:

> output <- glm(sta ~ sex, data=icu1.dat, family=binomial)

> logistic.regression.or.ci(output)

$regression.table





similarly

> output <- glm(sta ~ ser, data=icu1.dat, family=binomial)

> logistic.regression.or.ci(output)

$regression.table

> output <- glm(sta ~ loc, data=icu1.dat, family=binomial)

> logistic.regression.or.ci(output)

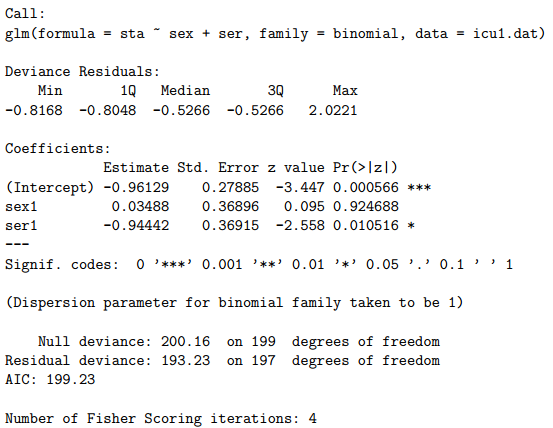
$regression.table

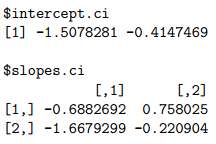
# Multivariate logistic model for sex and ser

> output <- glm(sta ~ sex + ser, data=icu1.dat, family=binomial)

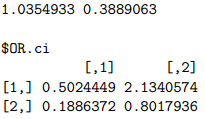
> logistic.regression.or.ci(output)

$regression.table









# Almost identical results as for univariate model

# As we suspected, no confounding.

# Overall conclusion: Service at ICU Admission (ser)

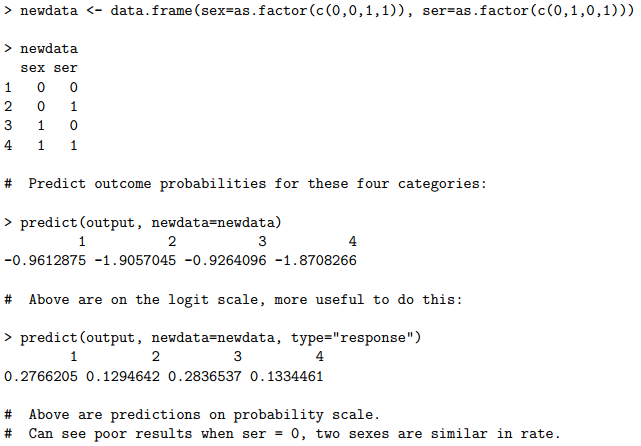
# seems to be associated with a strong effect, the

# surgical unit being associated with fewer deaths

# (OR = 0.39, 95% CI = (.19, 0.80). Nothing much

# can be said about sex or loc, results are inconclusive.

# Can predict some death rates depending on sex/ser categories:

# Create a data set for all possible category combinations:

Logistic regression example with many variables

# Read in full data set

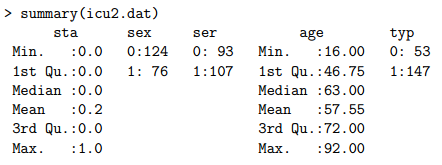
icu.dat <- read.table(file="g:\\icudat.txt", header = T)

# Create the reduced data set we will use:

> icu2.dat <- data.frame(sta=icu.dat$sta, sex=as.factor(icu.dat$sex),

ser=as.factor(icu.dat$ser), age = icu.dat$age, typ=as.factor(icu.dat$typ))

# Check descriptive statistics

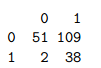


# Quick look at correlations

# Check effect of age and typ, two variables we did

# not look at before

> table(icu2.dat$sta, icu2.dat$typ)



# Looks like a large effect: those presenting at

# emergency (typ=1) have a much higher death rate.

# Let’s also look at a table between ser and typ:

> table(icu2.dat$ser, icu2.dat$typ)

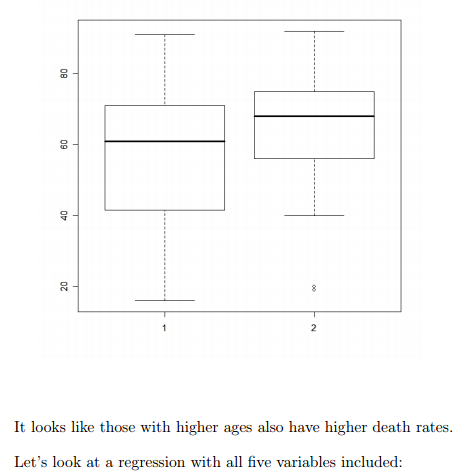


# Looks like there could be some confounding here,

# these are strongly related.

# Check the association between age and the outcome, sta

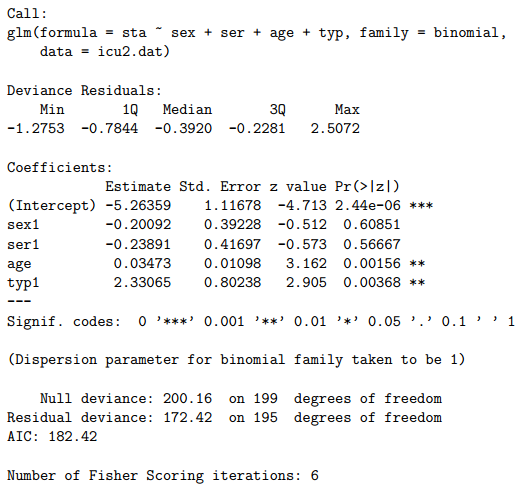
> boxplot(list(icu2.dat$age[icu2.dat$sta==0], icu2.dat$age[icu2.dat$sta==1]))

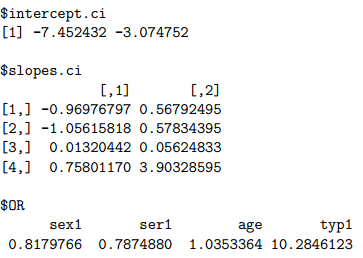


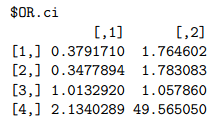
> output <- glm(sta ~ sex + ser + age + typ, data=icu2.dat, family=binomial)

> logistic.regression.or.ci(output)

$regression.table







As expected, age has a strong effect, with an odds ratio of 1.035 per year, or 1.03510 =

1.41 per decade (95% CI per year of (1.013, 1.058), so (1.138, 1.757) per decade). Typ

also has a very strong effect, with a CI of at least 2.

There does indeed seem to be some confounding between ser and typ, as the coefficient

estimate for ser has changed drastically from when typ was not in the model. In fact,

ser no longer looks “important”, it has been “replaced” by typ. Because of the high

correlation between ser and typ, it is difficult to separate out the effects of these two

variables.

Logistic regression with interaction terms

investigate an interaction term between sex and ser

# Create the variable that will be used in the interaction:

# Create a blank vector to store new variable

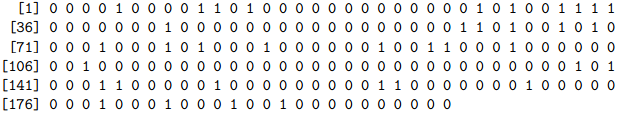
> ser.sex <- rep(0, length(icu1.dat$ser))

# Change value to 1 when both ser and sex are one

> for (i in 1:length(icu1.dat$ser)) {if (icu1.dat$ser[i] == 1

& icu1.dat$sex[i] == 1) ser.sex[i] <- 1 }

# Check new variable

> ser.sex

# Add new variable to data.frame,

# since all 0/1, do not bother with factors here

> icu3.dat <- data.frame(sta = icu1.dat$sta,

ser = icu1.dat$ser, sex = icu1.dat$sex, ser.sex = ser.sex)

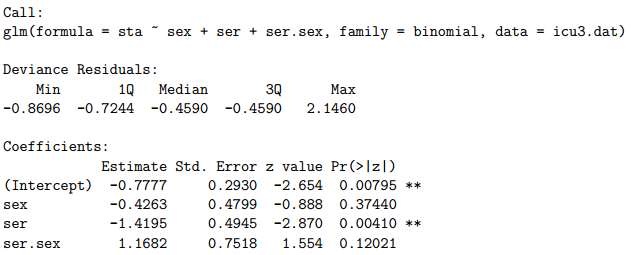
> summary(icu3.dat)

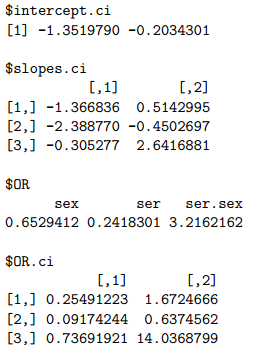
> Estimate the logistic regression model with interaction:

> output <- glm(sta ~ sex + ser + ser.sex, data=icu3.dat, family=binomial)

> logistic.regression.or.ci(output)

$regression.table





# Looking at the confidence interval, the result is inconclusive,

# but we can certainly not rule out an interaction term, OR = 3.22.

# Possibly some confounding as well, so cannot conclude any strong result.