Logistic Regression Using R and the Titanic Data

Getting the Data

Download train.csv from http://www.kaggle.com/c/titanic-gettingStarted/data if you have not done so already and keep track of the folder where you save the fie.

Loading the Data

In RStudio, you will need to set the working directory to the folder containing *train.csv*. You can do this using the

```
setwd("~/...")
```

command where "..." needs to be replaced by the file path of the folder/directory you want to use. Alternatively, you can also set your working directory by selecting *Set Working Directory* under the *Session* tab of RStudio and then choosing the directory you need.

Once your working directory is set, use

```
all_data <- read.csv('train.csv')
```

to load *train.csv* into the data frame *all_data*.

Loading the R packages

The R packages used in this tutorial are: *caret, ggplot2* and *dplyr*. If you do not have these packages installed, use the

```
install.packages("package_name")
```

command in the R console to install them.
Once the packages are installed, use

```
library(dplyr); library(ggplot2); library(caret)
```

There might be a more efficient way of loading three packages at once - let me know if you know a way.

Getting the Data Ready

Let's look at what we have:

```
head(all_data)
```

Note that there is missing data in this data frame - it is represented by *NA*.

To find out how many passengers are kept track of in this data set, you could try:

```
nrow(all_data)
```

which gives us the number of rows in the data frame *all_data*.

To find the number of rows that have missing entries use

```
sum(is.na(all_data))
```

There are quite a few rows with missing entries! Type

```
?is.na
?sum
```

in the R Console window to find out more about these commands.

In this tutorial the logistic regression models we create will have *Survived* as the outcome variable, *Fare* as a continuous predictor and *Sex* as a categorical/dichotomous predictor. So, lets whittle down our data set to what we need

```
less_data <- select(all_data, PassengerId, Survived, Sex, Fare)
```

by creating a new data frame that has only 4 columns: *PassengerId, Survived, Sex* and *Fare*. Let's look at its first 10 rows:

```
head(less_data, 10)
```

No *NA*'s are to be seen in this chunk of the data. Let's see how many rows have *NA*'s in this smaller data set:

```
sum(is.na(less_data))
```

Oh, none! Fewer complications for us...

Although, we could use the *Sex* column with its *male* and *female* entries as is to create our logistic model, we will create a binary representation of this column, *0* will stand for *male* and *1* will stand for female. Here we go:

Step 1 Create a vector, *col*, of *1*'s of length equal to the number of rows of *less_data*:

```
col<- c(rep(1, nrow(less_data)))
```

Step 2 Replace the *1*'s with *0*'s in this new vector whenever the entry in the *Sex* column is *male*

```
col[less_data$Sex == "male"]= 0
```

Step 3 Add a new column called *Sex_Numeric* to the data frame *less_data*:

```
less_data$Sex_Numeric = col
```

There might be more elegant ways of doing this, but this will work for now.

Partitioning the Data

Once we create a model, we will want to test it. So we will partition our data (stored in the data frame *less_data*) into training data that we will use to create our model and test data on which we will test our model. 80% of the data (chosen at random) will go into the training data set, the remaining 20% will be used for testing. R does all this for us beautifully using the *createDataPartition()* function from the *caret* package:

```
inTrain <- createDataPartition(y=less_data$Survived, p=0.8, list = FALSE)

training_data <- less_data[inTrain,]

test_data <- less_data[-inTrain,]
```

Model 1: Continuous Predictor and Categorical Outcome Variable

Let's create our first model, where we will try to predict survival based on the fare paid to get on the ship.

```
logistic_model1 <- glm(Survived ~ Fare, data=training_data, family=binomial)

summary(logistic_model1)
```

Then let's test our model on *test_data*, by first adding *Prediction* column to our *test_data* data set:

```
test_data$Prediction <-predict(logistic_model1, test_data, type ="response")
```

Let's look at it:

```
select(test_data, Fare, Survived, Prediction)
```

Next, create yet another column, called *Success*, whose entries will be *TRUE* whenever the prediction worked, i.e. the probability of survival was within 0.5 of the actual survival value of *1* for survival and *0* for death, and *FALSE* otherwise:

```
test_data$Success<-(abs(test_d ata$Survived - test_data$Prediction) < 0.5 )
```

Let's see what we got now:

```
head(test_data)
```

So, how accurate is this model? To find out:

Step 1 Find the total number of successess:

```
sum(test_data$Success)
```

Step 2 Find the total number of test cases:

```
nrow(test_data)
```

Step 3 Find the accuracy:

```
sum(test_data$Success)/nrow(test_data)
```

Finally, let's look at a graph of the model:

```
ggplot(training_data, aes(x=Fare, y=Survived)) + geom_point(shape=1,
position=position_jitter(width=.05,height=.05)) + stat_smooth(method="glm", family="binomial", se=FALSE)
```

Pretty, isn't it?

Model 2: Categorical Predictor and Categorical Outcome Variable

For the second model, we will try to predict survival based on the sex of the passenger.

Here is the model:

```
logistic_model2 <- glm(Survived ~ Sex_Numeric, data = training_data, family = binomial)
```

To find out more about it type:

```
logistic_model2
```

and/or

```
summary(logistic_model2)
```

Befor we test our second model, we will need to redefine our test data, *test_data*, again, since we have mucked it up when we created our first model, so type in:

```
test_data <- less_data[-inTrain,]
```

Then everything goes on as before:

```
test_data$Prediction <-predict(logistic_model2, test_data, type ="response")

select(test_data, Fare, Survived, Prediction)

test_data$Success<-(abs(test_d ata$Survived - test_data$Prediction) < 0.5 )

head(test_data)

sum(test_data$Success)

nrow(test_data)

sum(test_data$Success)/nrow(test_data)

ggplot(training_data, aes(x=Sex_Numeric, y=Survived)) + geom_point(shape=1,
position=position_jitter(width=.05,height=.05)) + stat_smooth(method="glm", family="binomial", se=FALSE)

plot(jitter(training_dataSurvived, .2), jitter(training_dataSex_Numeric, .2)) curve(predict(logistic_model2,
data.frame(Sex_Numeric=x), type="response"), add=TRUE)
```

Model 3: Continuous and Categorical Predictors, Categorical Outcome Variable

For our third model, we will use both the fare paid and the sex of the passengers to predict survival.
After you set up the model:

```
logistic_model3 <- glm(Survived ~ Sex_Numeric + Fare, training_data, family = binomial)
```

The process is familiar:

```
summary(logistic_model3)

test_data <- less_data[-inTrain,]

test_data$Prediction <-predict(logistic_model3, test_data, type ="response")

head(test_data)

test_data$Success<-(abs(test_d ata$Survived - test_data$Prediction) < 0.5 )

head(test_data)

sum(test_data$Success)

nrow(test_data)

sum(test_data$Success)/nrow(test_data)
```

I did not come up with any graphs for this multivariable logistic regression model

Model 4 : Multiple Predictors with Interactions

You can always see if the model improves if you include interactions between the predictor variables.

Is this model better or worse than the previous ones:

```
logistic_model4 <- glm(Survived ~ Sex_Numeric + Fare + Sex_Numeric : Fare, training_data, family = binomial)
```