GLM + Survival Analysis Coursework

Pre-processing

We use the code given in start.R to load the data into a dataframe called "titanic.df".

```
load("titanic.rdata")

n<-nrow(titanic.df)
n

head(titanic.df)

summary(titanic.df)

##

titanic.df$Pclass<-factor(titanic.df$Pclass)
titanic.df$Sex<-factor(titanic.df$Sex)
titanic.df$Age<-factor(titanic.df$Age)
titanic.df$Embarked<-factor(titanic.df$Embarked)
```

Q1

The response variable is given by

$$y_i = \begin{cases} 0 & \text{if passenger } i \text{ does not survive} \\ 1 & \text{if passenger } i \text{ survives} \end{cases}$$

As we are looking at the response variables on a case-by-case basis (i.e. $n_i = 1$ using the notation from the lecture notes), we have

$$y_i \sim Bernoulli(\pi_i)$$

where π_i is the probability of passenger i surviving.

With the multi-level categorical variables, we use dummy variables. For example, with the Pclass predictor, instead of having one predictor which may take values $1, 2$ or 3, we use a binary variable for 2nd class passengers and another for 3rd class passengers. 1st class passengers are then represented when both of the binary variables are equal to 0.

The logistic regression model is given by

$$log\left(\frac{\pi_i}{1-\pi_i}\right) = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \beta_3 x_{3i} + \beta_4 x_{4i} + \beta_5 x_{5i} + \beta_6 x_{6i} + \beta_7 x_{7i} + \beta_8 x_{8i}$$

where β_0 is the intercept term,
x_{1i} is the binary indicator value for passenger i being in second class,
x_{2i} indicates passenger i being in third class,
x_{3i} indicates passenger i being male,
x_{4i} indicates passenger i being an adult,
x_{5i} indicates passenger i being a senior,
x_{6i} is the integer value for the number of adults and/or children on board for passenger i,
x_{7i} indicates passenger i embarking at Queenstown,
and x_{8i} indicates passenger i embarking at Southampton.

The corresponding β_j are the coefficients for each of the predictors.

The following code is used to fit our model.

```
# fit the model
model = glm(Survived ~ Pclass + Sex + Age + Parch + Embarked,
              data = titanic.df, family = binomial)

# the glm() function automatically creates dummy variables for
# the multi-level categorical variables

# print the summary
summary(model)
```

We get the following output from the summary(model) call.

```
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   3.5307     0.4009   8.806  < 2e-16 ***
Pclass2      -0.9303     0.2775  -3.352 0.000802 ***
Pclass3      -2.0729     0.2649  -7.827 5.01e-15 ***
Sexmale      -2.5566     0.2154 -11.868  < 2e-16 ***
Age2         -0.7980     0.2592  -3.079 0.002077 **
Age3         -1.8475     0.6687  -2.763 0.005732 **
Parch        -0.1520     0.1148  -1.324 0.185661
EmbarkedQ    -0.8867     0.5729  -1.548 0.121647
EmbarkedS    -0.5003     0.2655  -1.884 0.059533 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

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

    Null deviance: 960.90  on 711  degrees of freedom
Residual deviance: 652.31  on 703  degrees of freedom
AIC: 670.31

Number of Fisher Scoring iterations: 4
```

A low p-value for a predictor, given in the $\Pr(>|Z|)$ column of the coefficients table, implies that that predictor is significant. That is, we have sufficient evidence to reject the null hypothesis that the coefficient is zero.

The intercept, Pclass and Sex predictors are significant at the 0.001 level. Age is significant at the 0.01 level. The remaining predictors are less significant and would be candidates for removal from the model.

Q2

We fit the requested model using the following code.

```
# define the model
model2 = glm(Survived ~ Pclass + Sex + Age, data = titanic.df,
                                           family = binomial)

# perform analysis of deviance between our earlier full model and
# our new reduced model
anova(model2, model, test = ’Chisq’)
```

The anova command yielded the following output.

```
Analysis of Deviance Table

Model 1: Survived ~ Pclass + Sex + Age
Model 2: Survived ~ Pclass + Sex + Age + Parch + Embarked
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1       706     658.40
2       703     652.31  3   6.0839   0.1076
```

A p-value of $0.1076 > 0.05$ (or whatever significance level we choose) suggests that there is insufficient evidence to reject the null hypothesis (which states that removing Parch and Embarked does not significantly worsen model fit).

Note that we could have calculated this p-value manually, using the following line of code.

```
1-pchisq(658.40 - 652.31, 3)
```

We can use a similar methodology to see whether we can reduce the model any further. First, look at the summary of the model using Pclass, Sex and Age.

```
# see if we can further reduce this model

# check the summary of this model
summary(model2)
```

We get the following output from the summary call.

```
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   3.0672     0.3376   9.084  < 2e-16 ***
Pclass2      -1.0673     0.2646  -4.034 5.48e-05 ***
Pclass3      -2.2334     0.2551  -8.754  < 2e-16 ***
Sexmale      -2.4809     0.2057 -12.061  < 2e-16 ***
Age2         -0.7608     0.2495  -3.050  0.00229 **
Age3         -1.8734     0.6612  -2.833  0.00461 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 960.9  on 711  degrees of freedom
Residual deviance: 658.4  on 706  degrees of freedom
AIC: 670.4

Number of Fisher Scoring iterations: 4
```

The Age predictor is still less significant than the others. Removing it may improve the model.

```
# build a model without the Age predictor
model_noage = glm(Survived ~ Pclass + Sex, data = titanic.df,
                                            family = binomial)

# perform another analysis of deviance
anova(model_noage, model2, test = ’Chisq’)
```

We get the following output.

```
Analysis of Deviance Table

Model 1: Survived ~ Pclass + Sex
Model 2: Survived ~ Pclass + Sex + Age
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1       708     672.06
2       706     658.40  2   13.662  0.00108 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

The small p-value of 0.00108 implies that we reject the null hypothesis (which states that including Age in the model does not provide a significantly better fit). i.e. we should keep Age in the model.

Q3

The following code creates the pred.surv variable, then uses it to create a confusion matrix. It then calculates the usual metrics which come with confusion matrices.

```
# using our model, predict the probabilities for each data point
predicted_probabilities = predict(model2, type = "response")

# create new binary variable pred.surv based on predicted
# probabilities
pred.surv = ifelse(predicted_probabilities > 0.5, 1, 0)

# create the table / confusion matrix
confusion_matrix = table(titanic.df$Survived, pred.surv)

# calculate the proportion of correctly classified sample cases
# correct predictions appear on the diagonals
# divide this by the total number of predictions made
accuracy = sum(diag(confusion_matrix)) / sum(confusion_matrix)

# output
print(confusion_matrix)
print(paste("The accuracy of the model:", accuracy))

# we could further analyse the confusion matrix by
# considering other metrics, such as precision, recall and
# F1-score

# calculate precision
precision = confusion_matrix[2, 2] / sum(confusion_matrix[, 2])

# calculate recall
recall = confusion_matrix[2, 2] / sum(confusion_matrix[2, ])

# calculate F1-score
f1_score = 2 * (precision * recall) / (precision + recall)

# output
print(paste("Precision:", precision))
print(paste("Recall:", recall))
print(paste("F1-score:", f1_score))
```

The following are some of the outputs from the above code.

```
pred.surv
      0   1
  0 359  65
  1  90 198

"The accuracy of the model: 0.782303370786517"
"Precision: 0.752851711026616"
"Recall: 0.6875"
"F1-score: 0.718693284936479"
```

The proportion of sample cases correctly classified by the model is given by the accuracy. The model also tends to predict cases where the passengers did not survive more often than the cases where they did survive.

Q4(i)

We calculate the odds using the coefficient values given by the summary function, then using some basic calculations.

```
# for an adult female travelling 2nd class, we have
# Pclass2 = Age2 = 1
# Pclass3 = Sexmale = Age3 = 0
# read the output of summary(model2) to obtain the coefficient
# values
summary(model2)

# the log odds are then given by
log_odds = 3.0672 - 1.0673 - 0.7608

# odds are given by exponentiating
odds = exp(log_odds)
# observe the value
odds
```

We get the odds of survival as 3.452505. This means that the probability of an adult female travelling 2nd class surviving is 3.452505 the probability of them not surviving.

Q4(ii)

The odds for the two cases are calculated in the same way as in Q4(i). The odds ratio is then given by dividing the two.

```
# for an adult female travelling first class,
# Age2 = 1
# Pclass2 = Pclass3 = Sexmale = Age3 = 0

# again, read the output of summary(model2) to obtain the
# coefficients
log_odds_1 = 3.0672 - 0.7608
odds_1 = exp(log_odds_1)

# for a senior male travelling third class,
# Pclass3 = Sexmale = Age3 = 1
# Pclass2 = Age2 = 0

log_odds_2 = 3.0672 - 2.2334 - 2.4809 - 1.8734
odds_2 = exp(log_odds_2)

# calculate the odds ratio
odds_ratio = odds_1 / odds_2

# output
odds_ratio
```

We get the odds ratio as 339.3052. This implies that adult females travelling first class were 339.3052 times more likely to survive, compared to a senior male travelling third class.

Q5

We use the odds calculated in Q4(i) to calculate the corresponding probability of survival.

```
# we already know the odds of survival for an adult female
# travelling second class from Q4(i), given by the variable ’odds’
odds

# after rearranging, we can get the probability of survival by the
# formula prob = odds/(1+odds)
prob = odds/(1+odds)
# output
prob
```

We get the probability of survival as 0.7754073.

We can then use this to calculate the standard error, using the fact that for $y \sim Bernoulli(p)$, the variance is $p(1-p)$. We then proceed to calculate the confidence interval.

```
# we can calculate the standard error for this observation by
```

```
se_prob = sqrt(prob * (1-prob))
# the variance for a Bernoulli random variable is p(1-p)

# the critical value for N(0,1) at the 95% confidence level is
# 1.96

# the confidence interval bounds are therefore
lower = prob - 1.96 * se_prob
upper = prob + 1.96 * se_prob

# convert back to probabilities
lower_prob = 1/(1+exp(-lower))
upper_prob = 1/(1+exp(-upper))

confidence_interval = c(lower_prob, upper_prob)

# output
confidence_interval
```

We get the 95% confidence interval as [0.4893697, 0.8310858].