Homework 8

Statistical Computing, STAT 3675Q

Anthony Anderson

General Instructions

- Answer the questions by inserting R code and necessary comments. Your output must contain the R code (do not use the `echo=FALSE option`).
- After you complete the assignment, save it under the file name `LastName-FirstName-HW8.pdf`
- Then submit the compiled PDF file through HuskyCT by April 23, 2019, at 11:59 PM.

Question 1 [20 points]

Data

The file `binary.csv` contains the following variables:

- `admit`: 1: admit; 0: no admit
- `gre`: GRE(Graduate Record Exam) scores
- `gpa`: GPA scores
- `rank`: Prestige of undergraduate institution. 1 indicates the highest prestige. 4 indicates the lowest prestige

a. Read in the data and display a summary of the data set.

```
binary <-read.csv("binary.csv")
summary(binary)
```

```
##      admit             gre             gpa             rank
##  Min.   :0.0000   Min.   :220.0   Min.   :2.260   Min.   :1.000
##  1st Qu.:0.0000   1st Qu.:520.0   1st Qu.:3.130   1st Qu.:2.000
##  Median :0.0000   Median :580.0   Median :3.395   Median :2.000
##  Mean   :0.3175   Mean   :587.7   Mean   :3.390   Mean   :2.485
##  3rd Qu.:1.0000   3rd Qu.:660.0   3rd Qu.:3.670   3rd Qu.:3.000
##  Max.   :1.0000   Max.   :800.0   Max.   :4.000   Max.   :4.000
```

b. Convert the variable `admit` to a factor. Label 0 as `No` and 1 as `Yes`. Show the total number of counts in each category (No/Yes).

```
binary$admit <- factor(binary$admit, labels=c("No","Yes"))
table(binary$admit)
```

```
##
##  No Yes
## 273 127
```

c. Convert the variable `rank` to a factor. Show the total number of counts in each category.

```
binary$rank <- as.factor(binary$rank)
table(binary$rank)
```

```
##
##   1   2   3   4
##  61 151 121  67
```

d. Perform a logistic regression, using `admit` as the response variable. Use the other 3 variables as predictors.

```
log.fit<-glm(admit~gre+gpa+rank,data=binary, family=binomial())
summary(log.fit)
```

```
##
## Call:
## glm(formula = admit ~ gre + gpa + rank, family = binomial(),
##     data = binary)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6268  -0.8662  -0.6388   1.1490   2.0790
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.989979   1.139951  -3.500 0.000465 ***
## gre          0.002264   0.001094   2.070 0.038465 *
## gpa          0.804038   0.331819   2.423 0.015388 *
## rank2       -0.675443   0.316490  -2.134 0.032829 *
## rank3       -1.340204   0.345306  -3.881 0.000104 ***
## rank4       -1.551464   0.417832  -3.713 0.000205 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 499.98  on 399  degrees of freedom
## Residual deviance: 458.52  on 394  degrees of freedom
## AIC: 470.52
##
## Number of Fisher Scoring iterations: 4
```

e. Can you draw conclusions about how well the model fits the data based on the output in part d.? If yes, what's your conclusion? If no, what else can you do to determine how well the model fits the data?

```
qchisq(.95,df=3)
```

```
## [1] 7.814728
```

The drop in deviance between the null model and the full model is $499.98 - 458.52 = 41.46$
Comparing this to a chi-square with 3 degrees of freedom, it is much larger, so we would infer that the model with the 3 predictors does a significantly better job at predicting than the null model. The output from the summary does not have any F tests, t-tests, or R^2 values, so we infer performance between different models (the empty null model and the full model in this case) by comparing their difference in deviances to appropriate chi-squared statistics.

f. Write down the fitted model and interpret the meaning of each parameter.

$$g(\mu) = \log\frac{\pi}{1-\pi} = -3.99 + .0022(\text{gre}) + .804(\text{GPA}) - .675(\text{rank2}) - 1.34(\text{rank3}) - 1.55(\text{rank4})$$

These coeff. and model are in terms of log-odds. So to make interpretation easier:

```
exp(coef(log.fit))
```

```
## (Intercept)         gre         gpa       rank2       rank3       rank4
##   0.0185001   1.0022670   2.2345448   0.5089310   0.2617923   0.2119375
```

For every unit increase of their respective predictor, the odds of being admitted will go up by a factor of their coefficient. So if we increased gre by one unit, the odds of being admitted will go up by a factor of

roughly 1 (so no change). For GPA, it will go up by a factor of 2.34. If the university has a prestige rank of 2, admission odds go up by a factor of 1/2 (so actually the odds go down), and the same for rank3, which goes "up" by a factor of about 1/4 . This data is coded so that rank 1 is the reference level.

g. Create a new data set as the following: replicate the mean of `gre` 4 times; replicate the mean of `gpa` 4 times; `rank` goes from 1 to 4. Use your new data to predict the outcome for each rank. Interpret your results.

```
gre<- rep(mean(binary$gre),4)
gpa<-rep(mean(binary$gpa),4)
rank <- seq(1:4)

toy.set <-data.frame(gre,gpa,rank)
toy.set$rank<-factor(toy.set$rank)

toy.set$prob<- predict(log.fit,newdata=toy.set,type="response")
toy.set$prob
```

```
## [1] 0.5166016 0.3522846 0.2186120 0.1846684
```

As "rank" goes from 1 to 4, that is from more prestigious to less, the likelihood of being admitted goes down from 52% to 18%

Question 2 [20 points]

Data

The file `poisson.csv` contains the following variables:

- `id`: student id
- `num_awards`: number of awards earned by students at one high school
- `prog`: type of program in which the student was enrolled. 1 = "General", 2 = "Academic" and 3 = "Vocational"
- `math`: score on their final exam in math

a. Read in the data and display a summary of the data.

```
pois.data <- read.csv("poisson.csv")
summary(pois.data)
```

```
##        id           num_awards        prog            math
##  Min.   :  1.00   Min.   :0.00   Min.   :1.000   Min.   :33.00
##  1st Qu.: 50.75   1st Qu.:0.00   1st Qu.:2.000   1st Qu.:45.00
##  Median :100.50   Median :0.00   Median :2.000   Median :52.00
##  Mean   :100.50   Mean   :0.63   Mean   :2.025   Mean   :52.65
##  3rd Qu.:150.25   3rd Qu.:1.00   3rd Qu.:2.250   3rd Qu.:59.00
##  Max.   :200.00   Max.   :6.00   Max.   :3.000   Max.   :75.00
```

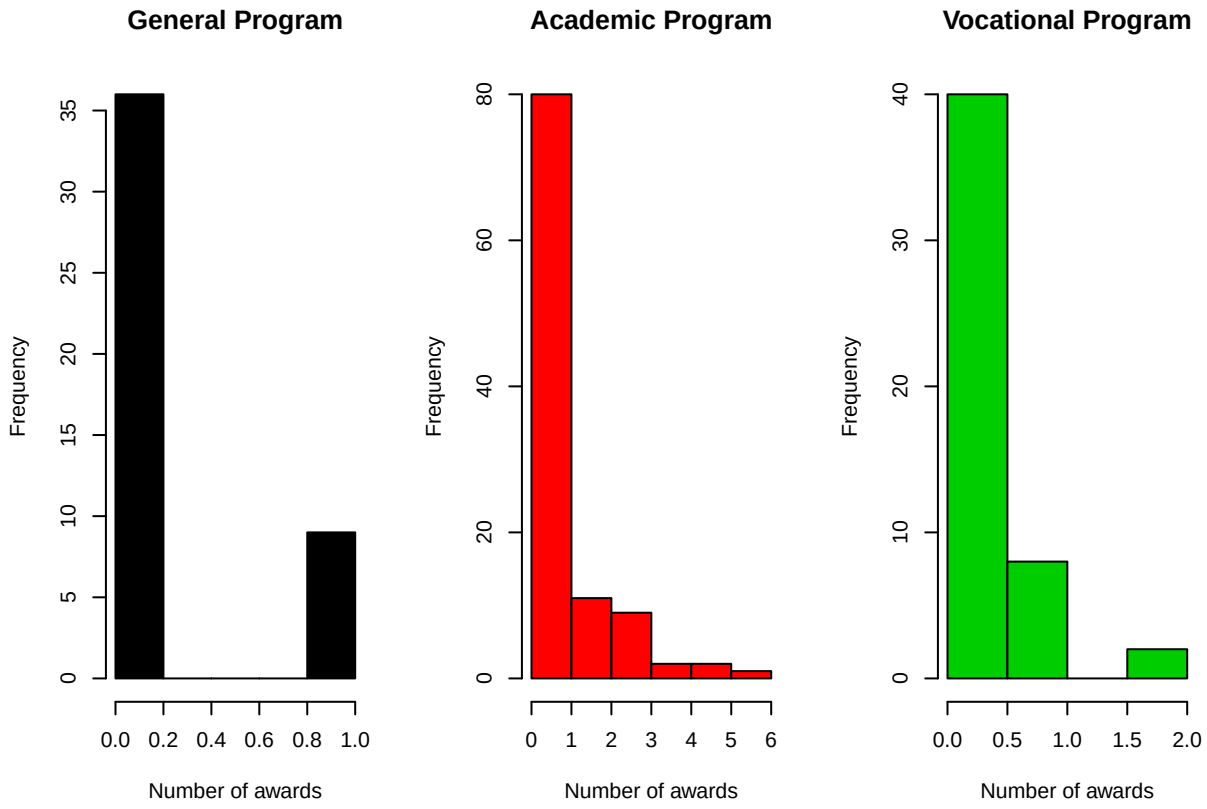
b. Convert the variable `prog` to be a factor. Label 1,2,3 as "General", "Academic" and "Vocational" respectively. Show the total number of counts in each category.

```
pois.data$prog.f <- factor(pois.data$prog, labels=c("General","Academic","Vocational"))
table(pois.data$prog.f)
```

```
##
##    General   Academic Vocational
##         45        105         50
```

c. Produce side by side histograms for the three levels of the `prog` variable. On the `x` axis put the number of awards, and use a different color to fill the bars in each histogram.

```r
opar<-par(no.readonly = T)
par(mfrow=c(1,3))
hist(pois.data$num_awards[pois.data$prog==1], col=1, main="General Program", xlab="Number of awards")
hist(pois.data$num_awards[pois.data$prog==2], col=2, main="Academic Program", xlab="Number of awards")
hist(pois.data$num_awards[pois.data$prog==3], col=3, main="Vocational Program", xlab="Number of awards")
```



d. Conduct a Poisson regression. Let `num_awards` be the response and use the remaining two as predictors.

```r
pois.fit <- glm(num_awards~prog.f+math,data=pois.data, family=poisson())
summary(pois.fit)
```

```
##
## Call:
## glm(formula = num_awards ~ prog.f + math, family = poisson(),
##     data = pois.data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2043  -0.8436  -0.5106   0.2558   2.6796
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -5.24712    0.65845  -7.969 1.60e-15 ***
## prog.fAcademic   1.08386    0.35825   3.025  0.00248 **
## prog.fVocational 0.36981    0.44107   0.838  0.40179
## math             0.07015    0.01060   6.619 3.63e-11 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 287.67  on 199  degrees of freedom
## Residual deviance: 189.45  on 196  degrees of freedom
## AIC: 373.5
##
## Number of Fisher Scoring iterations: 6
```

e. Write down the fitted model and interpret the meaning of each coefficient.

$$g(\mu = \lambda) = -5.25 + 1.08(\text{Prog.fAcademic}) + .37(\text{Prog.fVocational}) + .07(\text{math})$$

This is the log fitted model, just as with the earlier logistic regression, exponentiating the coeff will make them easier to interpret, so:

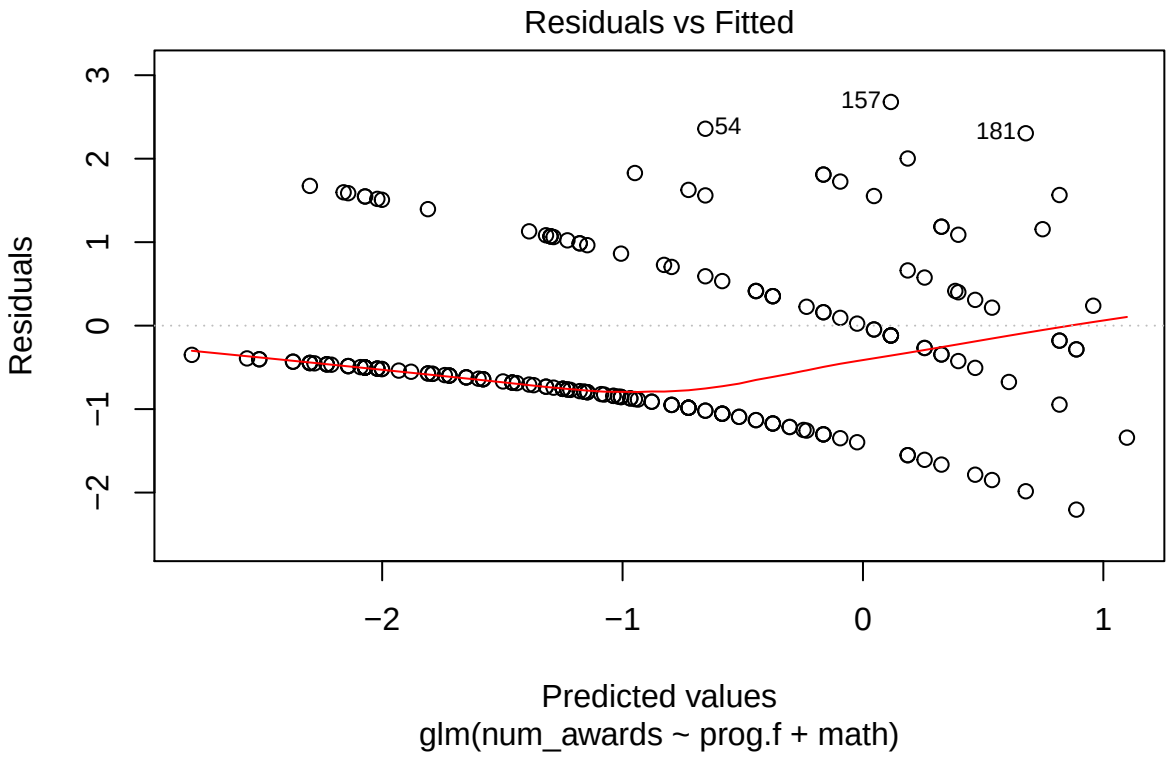
```
exp(coef(pois.fit))
```

```
##      (Intercept)   prog.fAcademic prog.fVocational             math
##       0.00526263       2.95606545       1.44745846       1.07267164
```

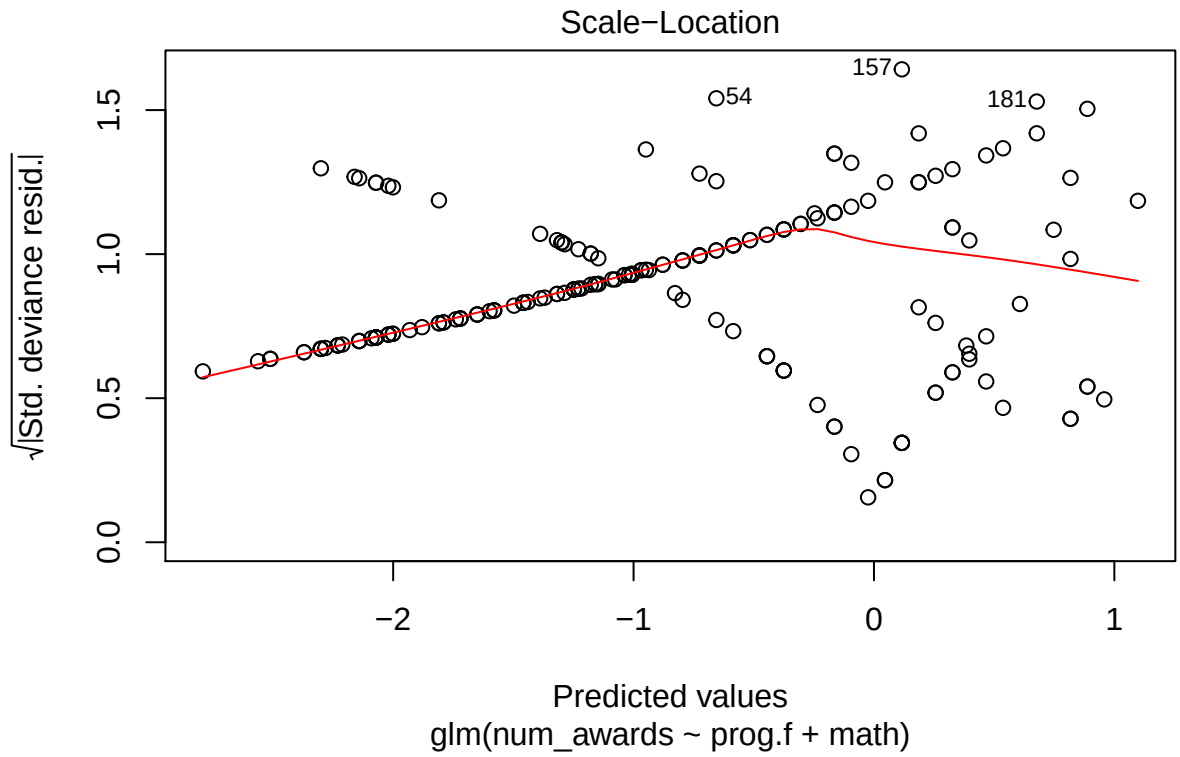
So for every unit increase of their respective predictors, the number of awards will change by a factor of the coefficient. So if a student is in the Academic program, they are predicted to have more awards by a factor of about 3, this is only 1.45 if the student is in the Vocational program. A unit increase in a students math score will result in the student winning more awards by a factor of 1.07 (so a very small increase). The "General" program is coded as the reference frame in this dataset.

f. Create and interpret the diagnostic plots for the analysis.

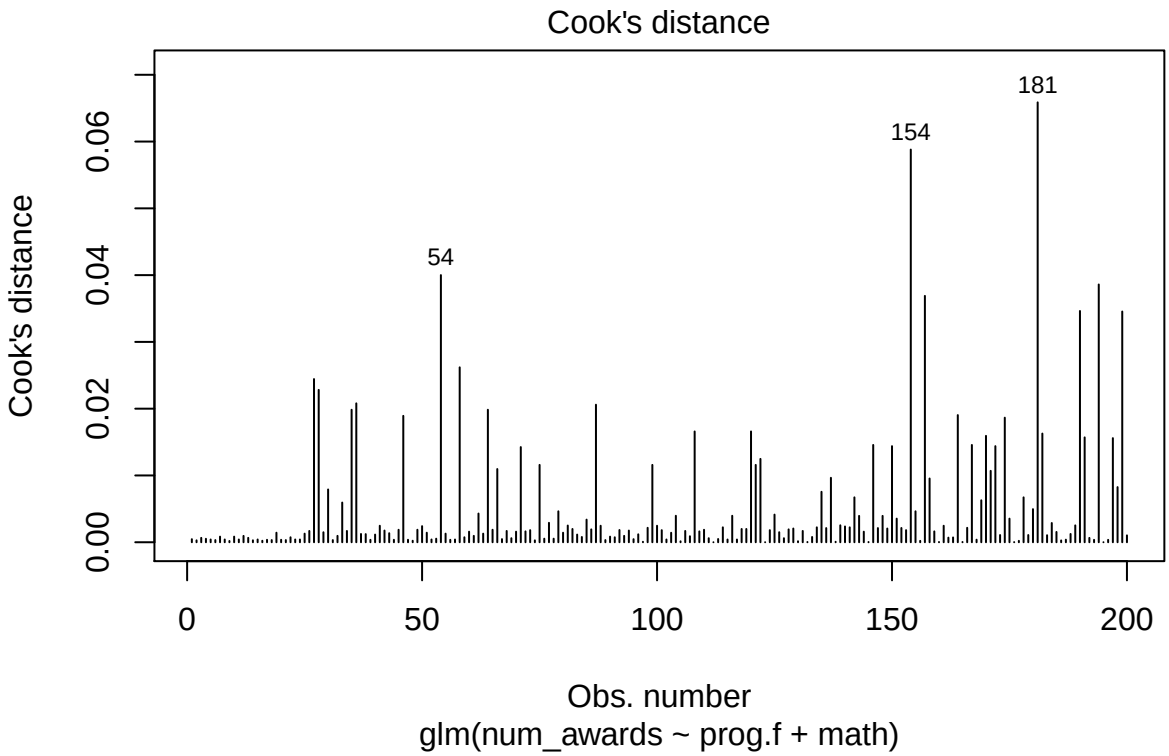
```
plot(pois.fit,which=1)
```




```
plot(pois.fit,which=3)
```




```
plot(pois.fit,which=4)
```



In the first graph, the residuals are roughly small and roughly 0 but as the number of awards increases, residuals spread out.
In the second graph, the residuals spread as predicted values go up, the "megaphone" effect, which is normal since this is a Poisson regression.
The third graph suggests observations 54, 154, and 181 are highly influential and require scrutiny.

g. Do you observe overdispersion?

```
library(qcc)
```

```
## Warning: package 'qcc' was built under R version 3.5.3
```

```
## Package 'qcc' version 2.7
```

```
## Type 'citation("qcc")' for citing this R package in publications.
```

```
qcc.overdispersion.test(pois.data$num_awards,type="poisson")
```

```
##
## Overdispersion test Obs.Var/Theor.Var Statistic    p-value
##        poisson data          1.759751  350.1905 1.9962e-10
```

The p-value is essentially 0, so we would infer that overdispersion is present.