Math 523 Assignment 2

part a

```
library(ggplot2)
awards <- read.csv("awards.csv")
attach(awards)
fit<-glm(numawards~1+math,family=poisson(link=log))
```

The estimated parameters are:

```
coef(fit)
```

```
## (Intercept)        math
##  -5.3335321   0.0861656
```

```
summary(fit)
```

```
##
## Call:
## glm(formula = numawards ~ 1 + math, family = poisson(link = log))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1853  -0.9070  -0.6001   0.3246   2.9529
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -5.333532   0.591261  -9.021   <2e-16 ***
## math         0.086166   0.009679   8.902   <2e-16 ***
## ---
## 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: 204.02  on 198  degrees of freedom
## AIC: 384.08
##
## Number of Fisher Scoring iterations: 6
```

From the summary above, the p-value of the estimated coefficients for math is very small(<0.05), thus math is a significant parameter. And the 95% confidence interval for beta1 is (0.067,0.105), from below:

```
confint(fit,level=0.95)
```

```
## Waiting for profiling to be done...
```

```
##                   2.5 %    97.5 %
## (Intercept) -6.52038334 -4.200322
## math         0.06737466  0.105356
```

part b

```
fit1<-glm(numawards~1+as.factor(prog),family=poisson(link=log),x=TRUE)
```

Therea are three parameters in this model. The estimated parameters are:

```
coef(fit1)
```

```
##      (Intercept) as.factor(prog)2 as.factor(prog)3
##       -1.6094379        1.6094379        0.1823216
```

```
summary(fit1)
```

```
##
## Call:
## glm(formula = numawards ~ 1 + as.factor(prog), family = poisson(link = log),
##     x = TRUE)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4142  -0.6928  -0.6325   0.0000   3.3913
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -1.6094     0.3333  -4.828 1.38e-06 ***
## as.factor(prog)2   1.6094     0.3473   4.634 3.59e-06 ***
## as.factor(prog)3   0.1823     0.4410   0.413    0.679
## ---
## 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: 234.46  on 197  degrees of freedom
## AIC: 416.51
##
## Number of Fisher Scoring iterations: 6
```

And from the summary, we could conclude that the intercept and the factor predictor(prog=2) is significant since their p-values are very small.

```
I <- t(fit1$x)%*%diag(fit1$weights)%*%fit1$x
I.inv <- solve(I)
sd <- sqrt(diag(I.inv))
sd
```

```
##      (Intercept) as.factor(prog)2 as.factor(prog)3
##        0.3333333        0.3473254        0.4409585
```

```
#wald test
beta <- fit1$coefficients
p.val <- pchisq((beta/sd)^2,df=1,lower.tail=FALSE)
p.val
```

```
##      (Intercept) as.factor(prog)2 as.factor(prog)3
##     1.376940e-06     3.590060e-06     6.792649e-01
```

```
#likelihood ratio test
fit0<-glm(numawards~1,family=poisson(link=log),x=TRUE)
anova(fit1,fit0,test="Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: numawards ~ 1 + as.factor(prog)
## Model 2: numawards ~ 1
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1       197     234.46
## 2       199     287.67 -2  -53.212 2.787e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

By the wald test and the log-likelihood ratio test, the p-values for the parameter prog. are both very small in this two tests, thus we could conclude that prog. is a significant factor.

part c

```
m1<-glm(numawards~1+math+as.factor(prog),family=poisson(link=log))
summary(m1)
```

```
##
## Call:
## glm(formula = numawards ~ 1 + math + as.factor(prog), family = poisson(link = log))
##
## 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 ***
## math              0.07015    0.01060   6.619 3.63e-11 ***
## as.factor(prog)2  1.08386    0.35825   3.025  0.00248 **
## as.factor(prog)3  0.36981    0.44107   0.838  0.40179
## ---
## 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
```

```
m2<-glm(numawards~1+math*as.factor(prog),family=poisson(link=log))
summary(m2)
```

```
##
## Call:
## glm(formula = numawards ~ 1 + math * as.factor(prog), family = poisson(link = log))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2295  -0.7958  -0.5298   0.2528   2.6826
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -3.86179    2.49317  -1.549    0.121
## math                   0.04400    0.04721   0.932    0.351
## as.factor(prog)2      -0.44107    2.60299  -0.169    0.865
## as.factor(prog)3      -0.84473    2.86990  -0.294    0.768
## math:as.factor(prog)2  0.02841    0.04870   0.583    0.560
## math:as.factor(prog)3  0.02290    0.05421   0.422    0.673
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 287.67  on 199  degrees of freedom
## Residual deviance: 189.10  on 194  degrees of freedom
## AIC: 377.16
##
## Number of Fisher Scoring iterations: 6
```

The model without interaction has four parameters, and the model with interaction has 6 parameters.

For the model without interaction, both the score of math and the programs students enrolled has a positve relationship with the number of awards. If the score of math increases two times, the expected number of awards will increase 0.006 times, and if the student is enrlled in an academic program, the expected number of awards will increase by 1.08.

For the model with interaction, the score of math is positively related with the number of awards while the program students enrolled in is negatively related with the number of awards. And if we consider these two parameter together, they will have a posite effect on the number of awards students obtain.

part d

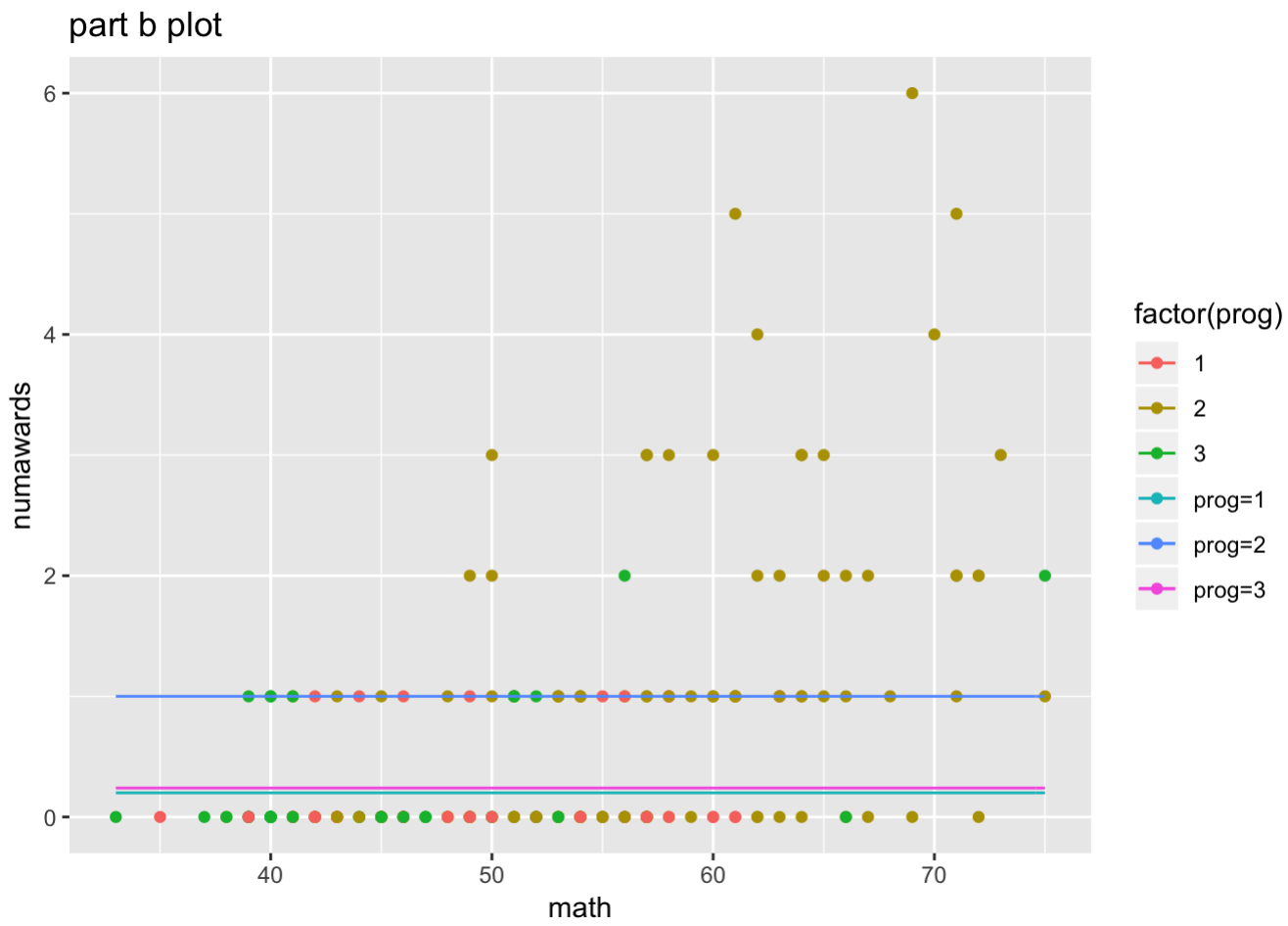
plot for part a：

```
d<-split(awards,awards$prog)
p1<-ggplot(awards,aes(x=math,y=numawards)) + geom_point(aes(color = factor(prog))) +
  ggtitle("part a plot")
f1<- function(x) exp(fit$coefficients[1]+fit$coefficients[2]*x)
p1 + stat_function(fun = f1, colour = "red")
```

plot for part b:

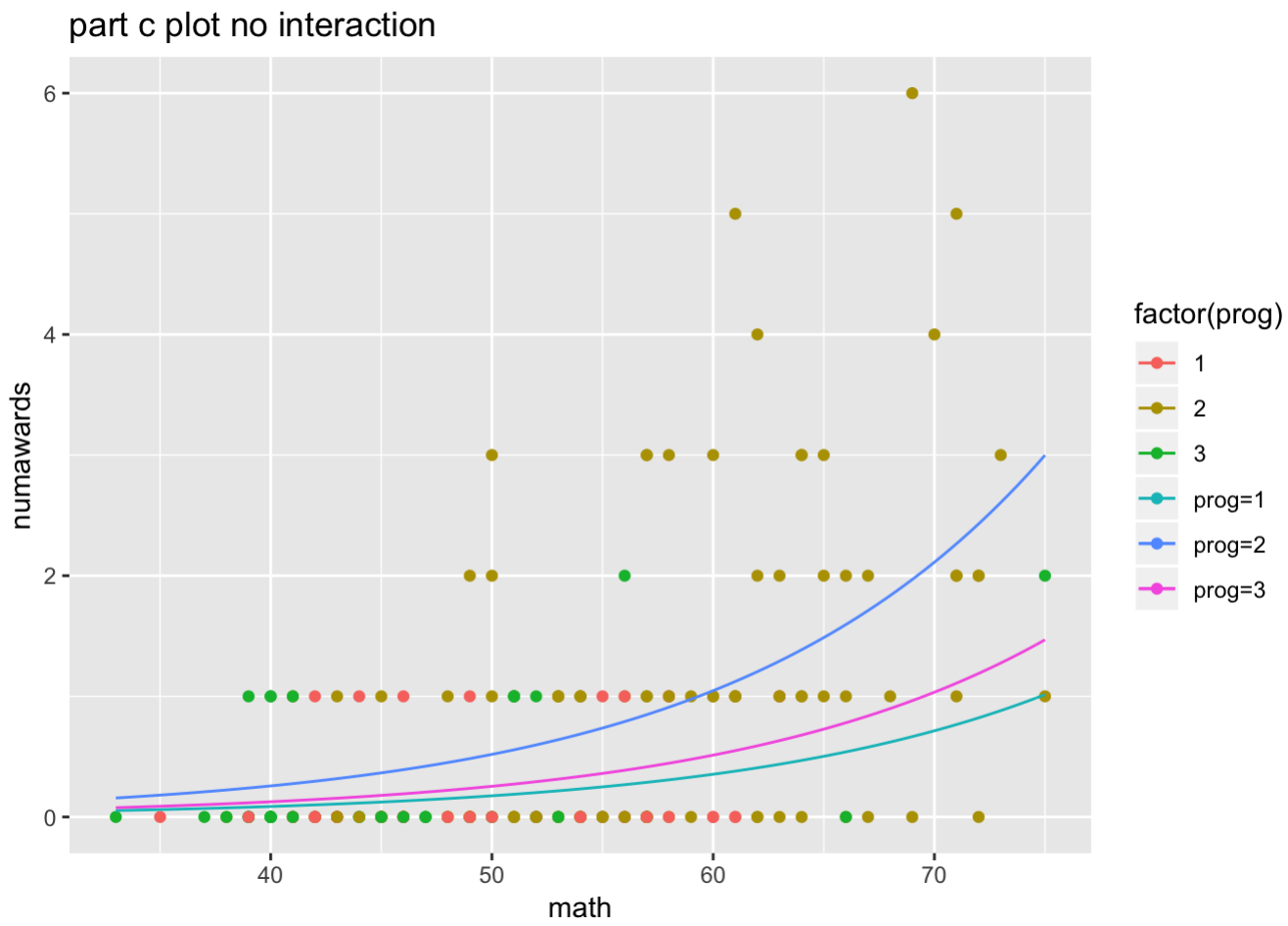
```
p2<-ggplot(awards,aes(x=math,y=numawards)) + geom_point(aes(color = factor(prog))) +
  ggtitle("part b plot")

#when prog=1
f_b1 <- function(x) exp(fit1$coefficients[1])
#when prog=2
f_b2 <- function(x) exp(fit1$coefficients[1]+fit1$coefficients[2])
#when prog=3
f_b3 <- function(x) exp(fit1$coefficients[1]+fit1$coefficients[3])

p2 + stat_function(fun = f_b1, aes(colour = "prog=1")) +
  stat_function(fun = f_b2, aes(colour ="prog=2")) +
  stat_function(fun = f_b3, aes(colour = "prog=3"))
```

```
## Warning in data.frame(x = xseq, y = do.call(fun, c(list(quote(x_trans)), :
## row names were found from a short variable and have been discarded

## Warning in data.frame(x = xseq, y = do.call(fun, c(list(quote(x_trans)), :
## row names were found from a short variable and have been discarded

## Warning in data.frame(x = xseq, y = do.call(fun, c(list(quote(x_trans)), :
## row names were found from a short variable and have been discarded
```



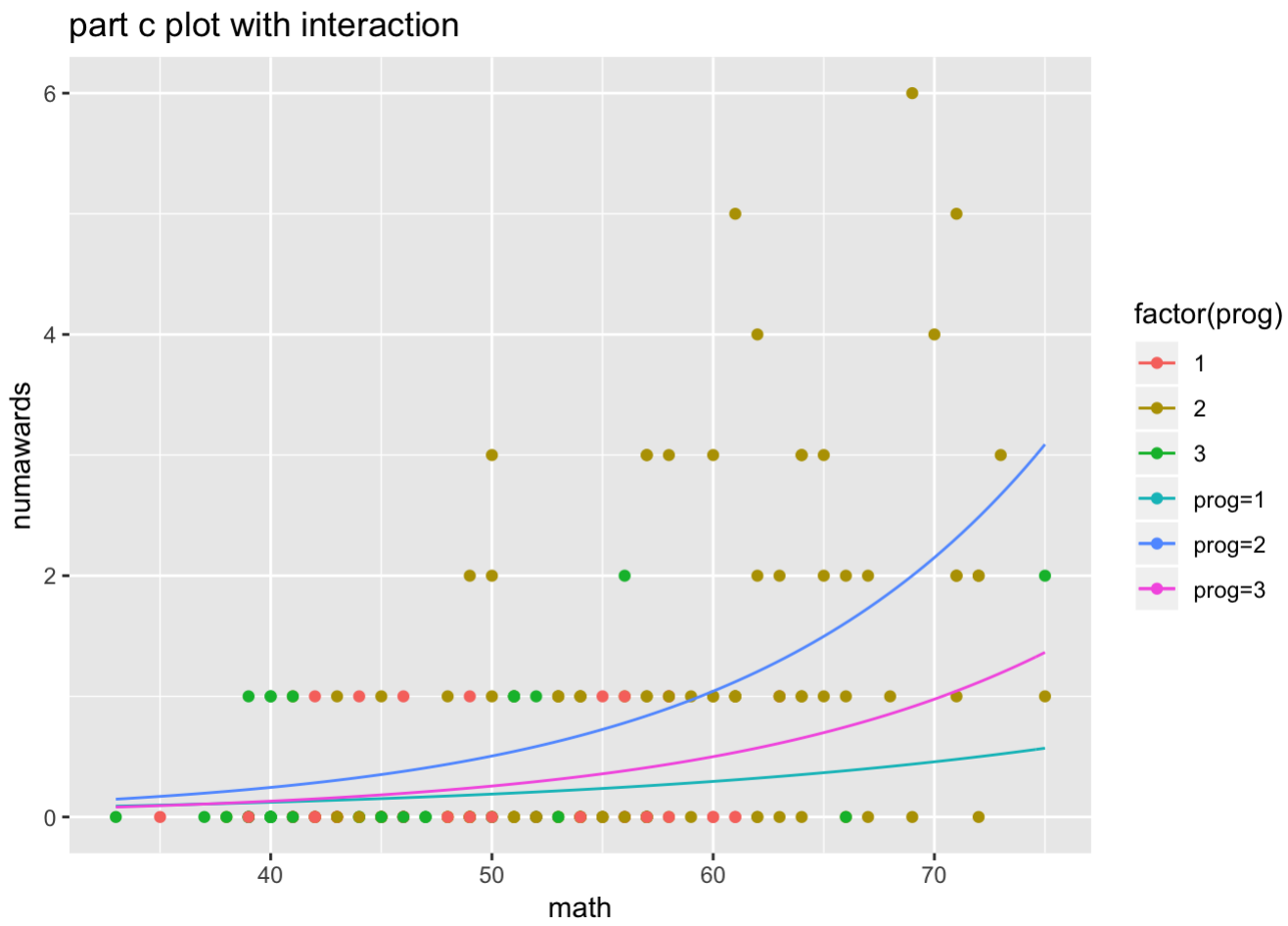
plot for part c First we plot the model with no interaction:

```
p3_1<-ggplot(awards,aes(x=math,y=numawards)) + geom_point(aes(color = factor(prog))) +
  ggtitle("part c plot no interaction")
#prog=1
f_c1 <- function(x) exp(m1$coefficients[1]+m1$coefficients[2]*x)
#prog=2
f_c2 <- function(x) exp(m1$coefficients[1]+m1$coefficients[2]*x+m1$coefficients[3])
#prog=3
f_c3 <- function(x) exp(m1$coefficients[1]+m1$coefficients[2]*x+m1$coefficients[4])
p3_1 + stat_function(fun = f_c1, aes(colour = "prog=1")) +
  stat_function(fun = f_c2, aes(colour ="prog=2")) +
  stat_function(fun = f_c3, aes(colour = "prog=3"))
```



Now plot the model with interaction below:

```
p3_2<-ggplot(awards,aes(x=math,y=numawards)) + geom_point(aes(color = factor(prog))) +
  ggtitle("part c plot with interaction")

#prog=1
f_c11 <- function(x) exp(m2$coefficients[1]+m2$coefficients[2]*x)
#prog=2
f_c22 <- function(x) exp(m2$coefficients[1]+m2$coefficients[2]*x+m2$coefficients[3]+m2$coefficie
nts[5]*x)
#prog=3
f_c33 <- function(x) exp(m2$coefficients[1]+m2$coefficients[2]*x+m2$coefficients[4]+m2$coefficie
nts[6]*x)

p3_2 + stat_function(fun = f_c11, aes(colour = "prog=1")) +
  stat_function(fun = f_c22, aes(colour ="prog=2")) +
  stat_function(fun = f_c33, aes(colour = "prog=3"))
```



From the four plot above, we can see that the model in part c fits the best, and the model with interaction is slightly better than that without interaction.

part e

```
deviance(fit)
```

```
## [1] 204.0213
```

```
deviance(fit1)
```

```
## [1] 234.46
```

```
deviance(m1)
```

```
## [1] 189.4496
```

```
deviance(m2)
```

```
## [1] 189.1016
```

we will want to minimize the deviance, therefore, the model in part c with interaction would be the best choice since it has the smallest deviance among all models.