MAST30027 Modern Applied, Statistics Assignment2

September 15,2019

Name: Tianyi Mo

Student ID: 875556

Tutorial time: Tue 2.15pm

Tutor: Qiuyi Li

1.Introduction

1.1 Background

This report is about evaluating chimpanzees prosocial tendency by analysing and fitting model to the data. The experiment has has two options, one is prosocial option, when human students participate the experiment, they nearly always choose the prosocial option when another student sits on the opposite side of the table. The question is whether a focal chimpanzee behaves similarly, choosing the prosocial option more often when another animal is present.

This report will analysis the dataset, visualize it, fit models to it and finally gives conclusion.

1.2 Data

There are four attributtes in the raw data. By looking through the dataset, it is found that there are 7 chimpanzees and every chimpanzee has the same number of instance if data (72 for each chimpanzee), therefore it is not tend to bias towards particular a chimpanzee. It should also be noted the the data is also balanced in condition and prosoc_left attribute, which means that each chimpanzee has 16 experiment on each combination of condition and prosoc left.

These are the 4 attributes in the raw dataset. actor (1 to 7) condition (0 or 1): prosoc left (0 or 1) pulled left (0 or 1)

```
# Load the dataset
dataset = read.delim("assign2.txt", header = TRUE, sep = " ")
```

2. Preprocessing

2.1 Create an New Attribute

Since the question is whether a focal chimpanzee choosing the prosocial option more often when another animal is present. Therefore, an new attribute "prosocial_action" is created, it means whether prosoc_left and pulled_left are same. If these two atributes are same, chimpanzee perform prosocially and prosocial_action has value TRUE and otherwise it has value FALSE.

```
dataset['prosocial_action'] = (dataset$prosoc_left == dataset$pulled_left)
#print out first10 rows in the dataset after adding new attribute prosocial_action
dataset[1:10,]
```

```
##    actor condition prosoc_left pulled_left prosocial_action
## 1      1         0           0           0             TRUE
## 2      1         0           0           1            FALSE
## 3      1         0           1           0            FALSE
## 4      1         0           0           0             TRUE
```

```
## 5       1         0           1           1             TRUE
## 6       1         0           1           1             TRUE
## 7       1         0           1           0            FALSE
## 8       1         0           1           0            FALSE
## 9       1         0           0           0             TRUE
## 10      1         0           0           0             TRUE
```

3.Visualization

3.1Percentage of prosocial action and other factors

These plots shows how does the percentage of pro_social action depend on condition, actor and prosoc_left.

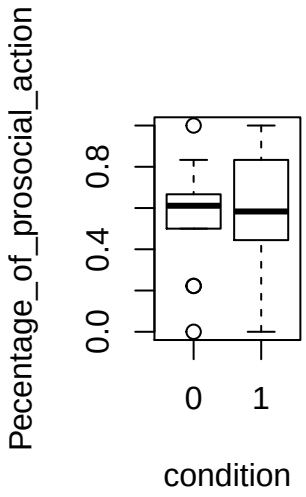
```
# select the prosocial_action = TRUE data only.
x = xtabs( ~ condition + prosoc_left+ actor, data = dataset[dataset$prosocial_action ==TRUE,])
x=data.frame(x)

# divid the Freq(count) by total(=18) and plot
x["Pecentage_of_prosocial_action"] = x$Freq/18
```

3.1.1 Percentage of prosocial action and Condition

```
#Mean of prosocial action and condition
prosocial = xtabs(prosocial_action ~ condition, data = dataset)
(prosocial_percentage = transform(prosocial, Freq=Freq/252))
```

```
##   condition      Freq
## 1         0 0.5555556
## 2         1 0.5793651
```

```
#Boxplot of prosocial action and condition
plot(Pecentage_of_prosocial_action~condition,x)
```



The first plot shows the relationship between condition and prosocial_action. The median decrease, and the variance increase. The mean percentage of choose prosocial increase from 0.5555556 to 0.5793651 in condition

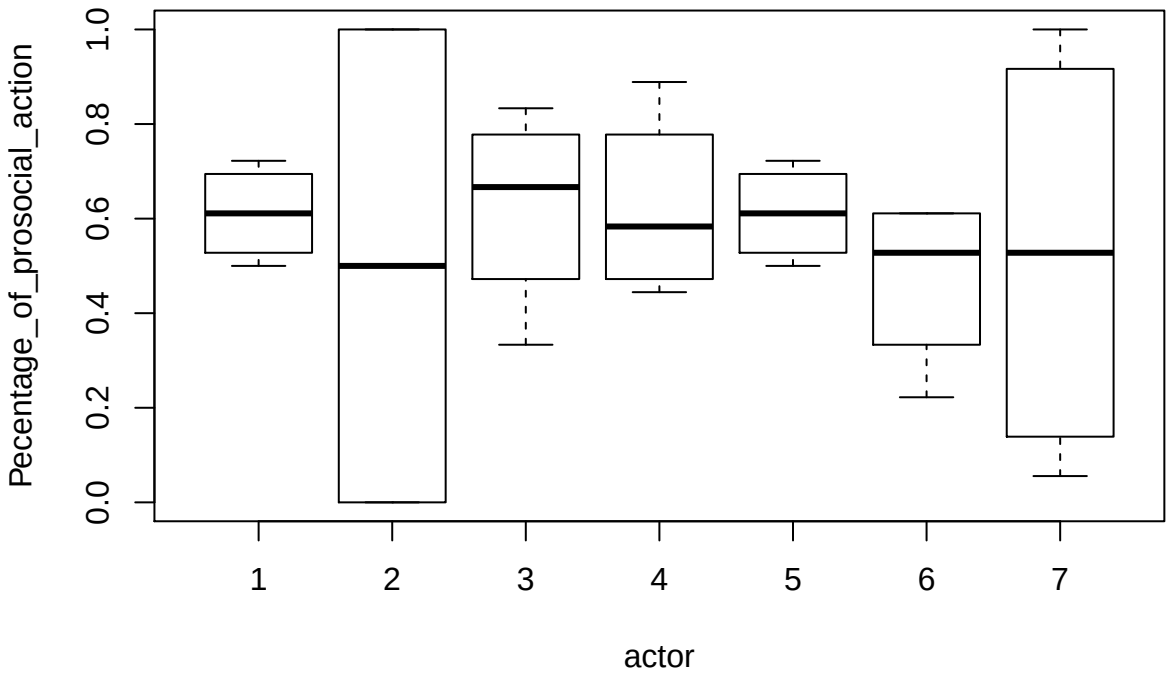
1. On average, chimapnzees choose the prosocial option sightly more frequently in condition 1 (when there is another chimpanzee in opposite). Since the mean incrases and the median decrease, the relationship between them is not clear.

3.1.2 Percentage of prosocial action and Actor

```
# It can be seen form the data that the mean prosocial_action for each actor are diffenent.
prosocial = xtabs(prosocial_action ~ actor, data = dataset)
(prosocial_percentage = transform(prosocial, Freq=Freq/72))
```

```
##   actor      Freq
## 1     1 0.6111111
## 2     2 0.5000000
## 3     3 0.6250000
## 4     4 0.6250000
## 5     5 0.6111111
## 6     6 0.4722222
## 7     7 0.5277778
```

```
#Boxplot of prosocial action and condition
plot(Pecentage_of_prosocial_action~actor,x)
```



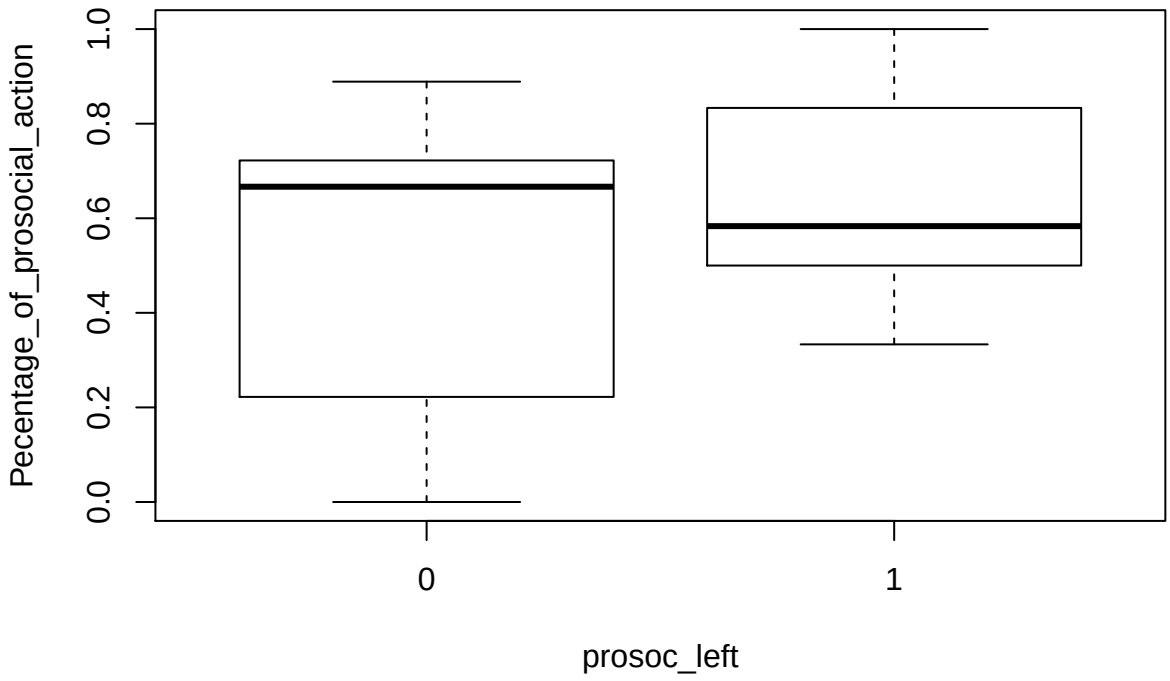
The second plot shows that each actor has different response in the test. The prosocial_action is depend on actor. For example, the second actor has the largest variance and actor 1 and 5 have relatively high median and low variance. The trend is meaningless in this plot because actor is not ordinal class.

3.1.3 Percentage of prosocial action and Prosoc_left

```
prosocial = xtabs(prosocial_action ~ prosoc_left, data = dataset)
(prosocial_percentage = transform(prosocial, Freq=Freq/252))
```

```
##   prosoc_left      Freq
## 1           0 0.4880952
## 2           1 0.6468254
```

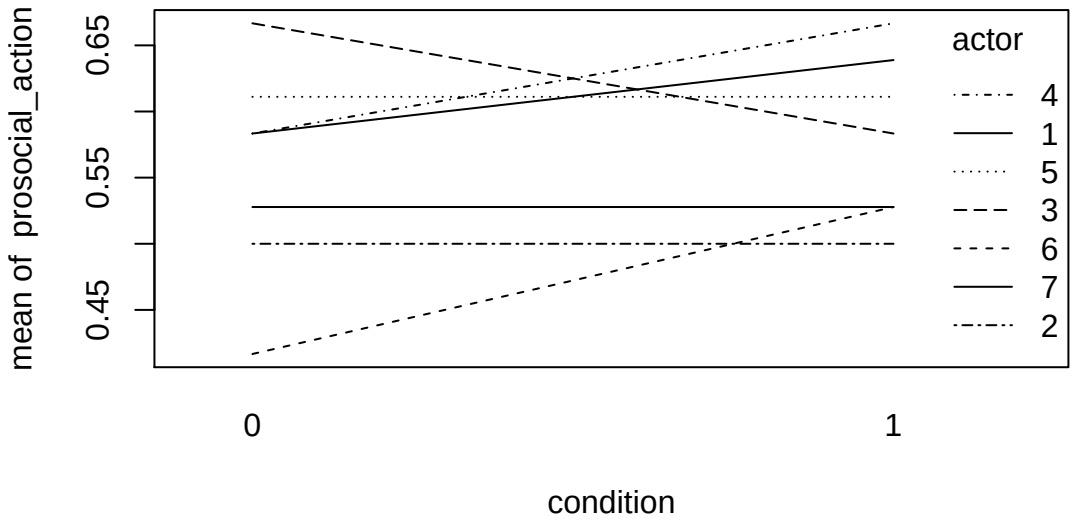
```
#Boxplot of prosocial action and prosoc_left
plot(Pecentage_of_prosocial_action~prosoc_left,x)
```



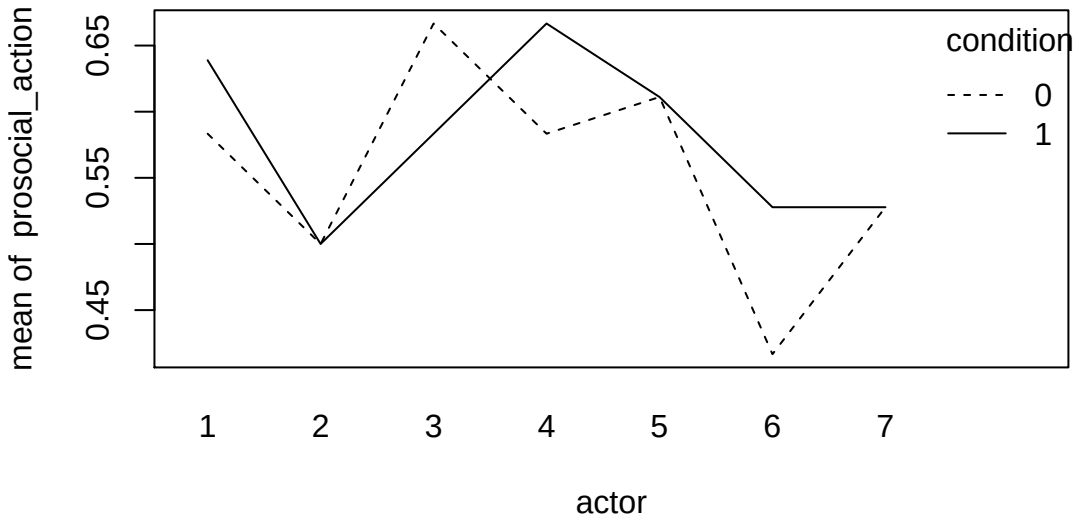
The third plot shows the relationship between prosoc_left and prosocial_action. It can be seen the left and right has different result on prosocial_action. When prosocial option is at right, chimpanzees has 0.4880952 to choose it, but when it is at left, chimpanzees has 0.6468254 probability to choose it.

3.2 Interactions Between Two Factors

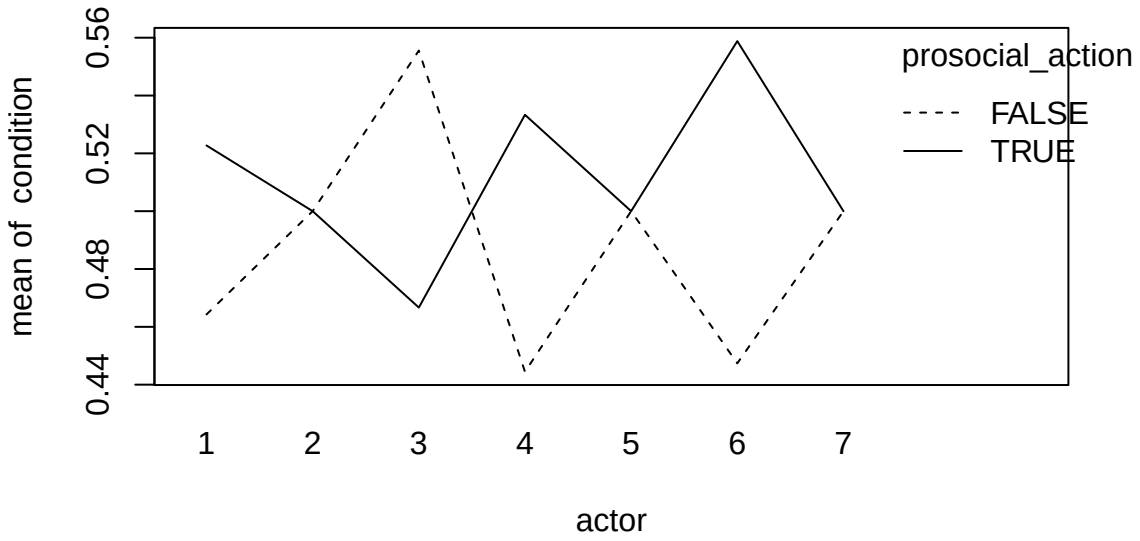
```
with(dataset, interaction.plot(condition,actor,prosocial_action))
```




```
with(dataset, interaction.plot(actor,condition,prosocial_action))
```



```r
with(dataset, interaction.plot(actor,prosocial_action,condition))
```



#It seems that there are some interaction between actor and prosocial_action and all actors have different mean of pro_action.

3.3 Outlier Detection

From the previous plots, the second chimpanzee (actor 2) show strange result. In both conditions, it chooses proaction option at exactly at 0.5. By looking back to the raw dataset. It can be found that the it always choose the left one regardless of any other factors. There is no similar behaviour like this among other chimpanzees and it might be an outlier.

```r
# Second chimpanzee always pulled the left lever
xtabs(~prosoc_left+pulled_left, data =  dataset[dataset$actor == 2,])
```

```
##            pulled_left
## prosoc_left  1
##           0 36
##           1 36
```

```r
# Other chimpanzees pulled both left and right lever
# for example chimpanzees 1
xtabs(~prosoc_left+pulled_left, data =  dataset[dataset$actor == 1,])
```

```
##            pulled_left
## prosoc_left  0  1
##           0 25 11
##           1 17 19
```

This might be an error in data collection process and might be nomal behaviour of chimpanzee species. Based on the data, I cannot decided which one it should be. This problem should be checked with the zoologist and experts with domain knowledges to decided whether to remove data from chimpanzee 2.

3.Contengency Table

Use contengency table to test the dependency of prosocial_action, condition and and actor. Using three way contengency table to test the mutually independency of the 3 factors. Poisson regression is used because poisson regression with log link will gives chi-sqare distribution.

H0 = prosocial_action & condition & actor are independent

H1 = prosocial_action & condition & actor are not independent

```
x = xtabs( ~ condition + prosocial_action, data = dataset)
(x = data.frame(x))
```

```
##   condition prosocial_action Freq
## 1         0            FALSE  112
## 2         1            FALSE  106
## 3         0             TRUE  140
## 4         1             TRUE  146
```

The cell counts need to be at least 5 for the deviance to has chi-square distribution as describe in the lectue. From the data count above, it can be seen that the minimum count is 13.

```
#fit poisson model
model0 = glm(Freq ~ condition + prosocial_action, family = poisson, data = x)
#deviance of the poission regression model
deviance(model0)
```

```
## [1] 0.2910418
```

```
#degrees of freedom
df.residual(model0)
```

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

```
# calculate the p-value
pchisq(deviance(model0), df = df.residual(model0), lower.tail = FALSE)
```

```
## [1] 0.5895537
```

The p-value is 0.9598556, it is much larger than 0.05, thus we cannot reject H0 and prosocial_action & condition & action are independent. Since we are interested in whether prosocial_action is related the condition (whether there is another chimpanzee in another side), from the result we can say that prosocial_action is independent of condition. And chimpanzee don't behave similarly as human.

4. Binomial Regression

4.1 Fitting data with binomial regression

If we let the count be response variable, and all four factors (condition, prosoc_left, pulled_left, actor) as explanation variable. The count has binomial distribution with n equals to 18 and p is the probability of choosing the prosocial_action. Since the n is only 18 and p is large, we cannot use poisson distribution to approximate the binomial model.

```
x = xtabs( ~ condition + prosoc_left + pulled_left+ actor, data = dataset)
x = data.frame(x)
#set last column to n = 18
x["total"] = 18
(x = x[x$prosoc_left == x$pulled_left,])
```

```
##    condition prosoc_left pulled_left actor Freq total
## 1          0           0           0     1   12    18
## 2          1           0           0     1   13    18
## 7          0           1           1     1    9    18
## 8          1           1           1     1   10    18
## 9          0           0           0     2    0    18
## 10         1           0           0     2    0    18
## 15         0           1           1     2   18    18
## 16         1           1           1     2   18    18
## 17         0           0           0     3   13    18
## 18         1           0           0     3   15    18
## 23         0           1           1     3   11    18
## 24         1           1           1     3    6    18
## 25         0           0           0     4   12    18
## 26         1           0           0     4   16    18
## 31         0           1           1     4    9    18
## 32         1           1           1     4    8    18
## 33         0           0           0     5   12    18
## 34         1           0           0     5   13    18
## 39         0           1           1     5   10    18
## 40         1           1           1     5    9    18
## 41         0           0           0     6    4    18
## 42         1           0           0     6    8    18
## 47         0           1           1     6   11    18
## 48         1           1           1     6   11    18
## 49         0           0           0     7    4    18
## 50         1           0           0     7    1    18
## 55         0           1           1     7   15    18
## 56         1           1           1     7   18    18
```

```
#the data have 28 rows
dim(x)
```

```
## [1] 28  6
```

```
#fit binomial regression model
fullmodel = glm(cbind(total-Freq,Freq) ~ (condition + prosoc_left +actor)^2, family = binomial, data = x

#use AIC to select model
finalmodel = step(fullmodel)
```

```
## Start:  AIC=127.33
## cbind(total - Freq, Freq) ~ (condition + prosoc_left + actor)^2
##
##                         Df Deviance    AIC
## - condition:actor        6   13.193 117.30
## - condition:prosoc_left  1   13.017 127.12
## <none>                       11.231 127.33
## - prosoc_left:actor      6  173.039 277.14
##
## Step:  AIC=117.3
## cbind(total - Freq, Freq) ~ condition + prosoc_left + actor +
##     condition:prosoc_left + prosoc_left:actor
##
##                         Df Deviance    AIC
## - condition:prosoc_left  1   15.002 117.11
## <none>                       13.193 117.30
## - prosoc_left:actor      6  175.035 267.14
##
## Step:  AIC=117.1
## cbind(total - Freq, Freq) ~ condition + prosoc_left + actor +
##     prosoc_left:actor
##
##                     Df Deviance    AIC
## - condition          1   15.414 115.52
## <none>                   15.002 117.11
## - prosoc_left:actor  6  176.199 266.30
##
## Step:  AIC=115.52
## cbind(total - Freq, Freq) ~ prosoc_left + actor + prosoc_left:actor
##
##                     Df Deviance    AIC
## <none>                   15.414 115.52
## - prosoc_left:actor  6  176.502 264.61
```

The fianl model after model selection is y ~ prosoc_left + actor + prosoc_left:actor

```
summary(finalmodel)
```

```
##
## Call:
## glm(formula = cbind(total - Freq, Freq) ~ prosoc_left + actor +
##     prosoc_left:actor, family = binomial, data = x)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7699  -0.2581   0.0000   0.2538   1.1946
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -8.210e-01  3.618e-01  -2.269  0.02326 *
## prosoc_left1         7.098e-01  4.923e-01   1.442  0.14939
## actor2               2.247e+01  5.089e+03   0.004  0.99648
## actor3              -4.318e-01  5.400e-01  -0.800  0.42396
## actor4              -4.318e-01  5.400e-01  -0.800  0.42396
## actor5              -1.555e-15  5.117e-01   0.000  1.00000
```

```
## actor6               1.514e+00  5.059e-01   2.993  0.00276 **
## actor7               2.646e+00  6.026e-01   4.390 1.13e-05 ***
## prosoc_left1:actor2 -4.402e+01  7.197e+03  -0.006  0.99512
## prosoc_left1:actor3  6.542e-01  7.173e-01   0.912  0.36173
## prosoc_left1:actor4  6.542e-01  7.173e-01   0.912  0.36173
## prosoc_left1:actor5  9.103e-16  6.962e-01   0.000  1.00000
## prosoc_left1:actor6 -1.855e+00  6.959e-01  -2.666  0.00769 **
## prosoc_left1:actor7 -4.932e+00  9.156e-01  -5.387 7.16e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 197.206  on 27  degrees of freedom
## Residual deviance:  15.414  on 14  degrees of freedom
## AIC: 115.52
##
## Number of Fisher Scoring iterations: 18
```

```
deviance(finalmodel)
```

```
## [1] 15.4145
```

```
df.residual(finalmodel)
```

```
## [1] 14
```

```
pchisq(deviance(finalmodel), df = df.residual(finalmodel), lower.tail = FALSE)
```

```
## [1] 0.3504197
```

```
anova(finalmodel,fullmodel,test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: cbind(total - Freq, Freq) ~ prosoc_left + actor + prosoc_left:actor
## Model 2: cbind(total - Freq, Freq) ~ (condition + prosoc_left + actor)^2
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        14     15.415
## 2         6     11.231  8   4.1838   0.8402
```

The model comparision test between fullmodel and final model gives p-value = 0.9968, thus the final model is adequate. The fianl model after model selection is y ~ prosoc_left + actor + prosoc_left:actor. The number of times choosing prosocial action does dependent on condition because it is not in the final model.

Check overdispersion

```
phihat = sum(residuals(finalmodel,type = "pearson")^2)/finalmodel$df.residual
```

The phihat is 0.3075466 < 1, there is some underdispersion and there is smaller variability than expected. In this case, the test has decreased power and the p-value tend to be less significant. However, underdispersion will not cause problem like overdispersion and we don't need to use the quansibinomial to redo the model fitting.

Diagonistic plot

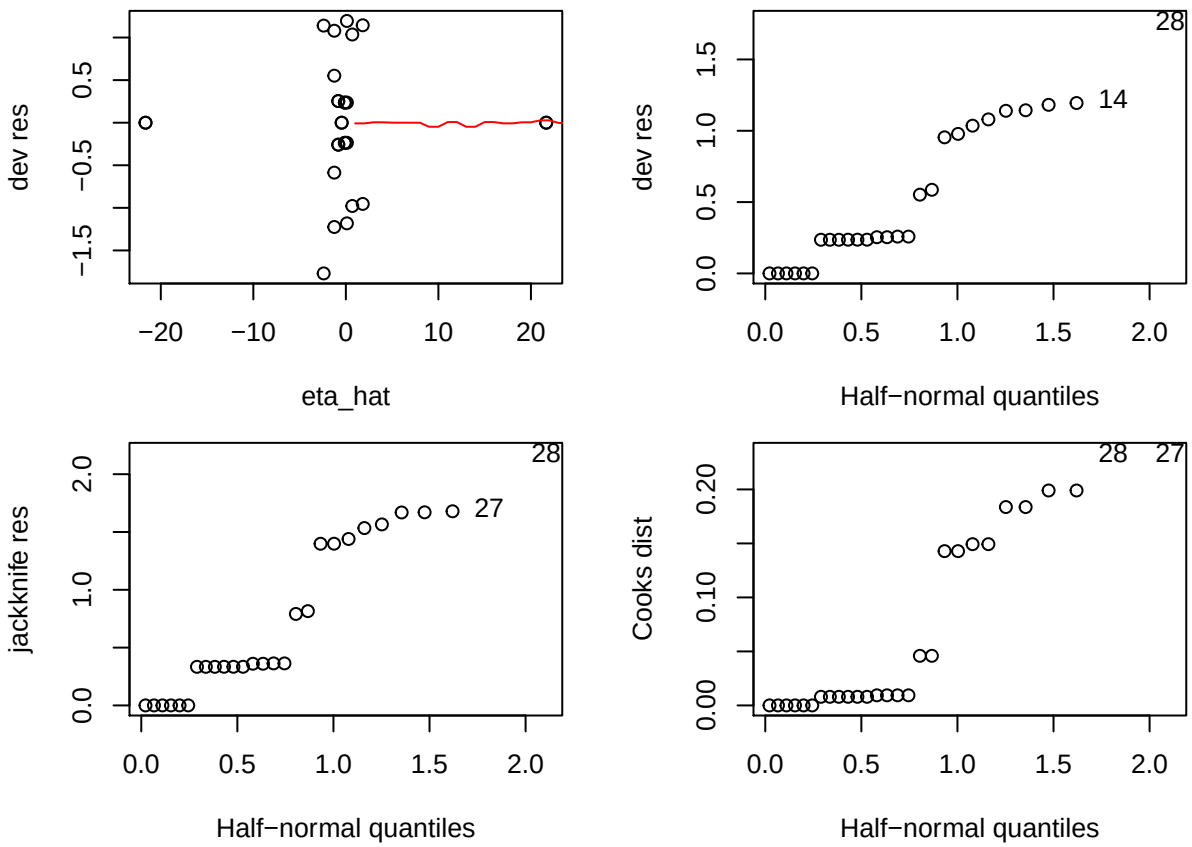
```
library(faraway) # for halfnorm function
(D_res <- residuals(finalmodel))
```

```
##             1             2             7             8             9
##  2.537886e-01 -2.581409e-01  2.358844e-01 -2.363725e-01  1.191847e-04
##            10            15            16            17            18
##  1.191847e-04 -1.191847e-04 -1.191847e-04  5.519314e-01 -5.862540e-01
##            23            24            25            26            31
## -1.181904e+00  1.194603e+00  1.079799e+00 -1.224915e+00 -2.358844e-01
##            32            33            34            39            40
##  2.363725e-01  2.537886e-01 -2.581409e-01 -2.363725e-01  2.358844e-01
##            41            42            47            48            49
##  1.035615e+00 -9.779990e-01  5.575504e-08  5.575504e-08 -9.539987e-01
##            50            55            56
##  1.143727e+00  1.139990e+00 -1.769861e+00
```

```
P_res <- residuals(finalmodel, type="pearson")
lever <- influence(finalmodel)$hat
J_res <- rstudent(finalmodel)
Cooks <- cooks.distance(finalmodel)
(eta_hat <- predict(finalmodel, type="link"))
```

```
##           1           2           7           8           9          10
##  -0.8209806  -0.8209806  -0.1112256  -0.1112256  21.6531921  21.6531921
##          15          16          17          18          23          24
## -21.6531920 -21.6531920  -1.2527630  -1.2527630   0.1112256   0.1112256
##          25          26          31          32          33          34
##  -1.2527630  -1.2527630   0.1112256   0.1112256  -0.8209806  -0.8209806
##          39          40          41          42          47          48
##  -0.1112256  -0.1112256   0.6931472   0.6931472  -0.4519851  -0.4519851
##          49          50          55          56
##   1.8245493   1.8245493  -2.3978953  -2.3978953
```

```
par(mfrow=c(2,2))
par(mar=c(4,4,1,2))
plot(eta_hat, D_res, ylab="dev res")
lines(predict(loess(D_res ~ eta_hat)), col="red")
halfnorm(D_res, ylab="dev res")
halfnorm(J_res, ylab="jackknife res")
halfnorm(Cooks, ylab="Cooks dist")
```



The first plot has some data points have extremely small value on the left. From the eta_hat data, it can be found that it is due to the second actor always choose the left lever.

4.2 After remove the outlier, fitting data with binomial regression

```
#remove actor
x = xtabs( ~ condition + prosoc_left + pulled_left+ actor, data = dataset[dataset$actor !=2,])
x = data.frame(x)
#set last column to n = 18
x["total"] = 18
(x = x[x$prosoc_left == x$pulled_left,])
```

```
##    condition prosoc_left pulled_left actor Freq total
## 1          0           0           0     1   12    18
## 2          1           0           0     1   13    18
## 7          0           1           1     1    9    18
## 8          1           1           1     1   10    18
## 9          0           0           0     3   13    18
## 10         1           0           0     3   15    18
## 15         0           1           1     3   11    18
## 16         1           1           1     3    6    18
## 17         0           0           0     4   12    18
## 18         1           0           0     4   16    18
## 23         0           1           1     4    9    18
## 24         1           1           1     4    8    18
## 25         0           0           0     5   12    18
## 26         1           0           0     5   13    18
```

```
## 31         0           1           1     5   10    18
## 32         1           1           1     5    9    18
## 33         0           0           0     6    4    18
## 34         1           0           0     6    8    18
## 39         0           1           1     6   11    18
## 40         1           1           1     6   11    18
## 41         0           0           0     7    4    18
## 42         1           0           0     7    1    18
## 47         0           1           1     7   15    18
## 48         1           1           1     7   18    18
```

```
#the data have 28 rows
dim(x)
```

```
## [1] 24  6
```

```
#fit binomial regression model
fullmodel = glm(cbind(total-Freq,Freq) ~ (condition + prosoc_left +actor)^2, family = binomial, data = x

#use AIC to select model
finalmodel = step(fullmodel)
```

```
## Start:  AIC=121.33
## cbind(total - Freq, Freq) ~ (condition + prosoc_left + actor)^2
##
##                         Df Deviance    AIC
## - condition:actor        5   13.193 113.30
## - condition:prosoc_left  1   13.017 121.12
## <none>                       11.231 121.33
## - prosoc_left:actor      5   86.140 186.24
##
## Step:  AIC=113.3
## cbind(total - Freq, Freq) ~ condition + prosoc_left + actor +
##     condition:prosoc_left + prosoc_left:actor
##
##                         Df Deviance    AIC
## - condition:prosoc_left  1   15.002 113.11
## <none>                       13.193 113.30
## - prosoc_left:actor      5   88.002 178.10
##
## Step:  AIC=113.1
## cbind(total - Freq, Freq) ~ condition + prosoc_left + actor +
##     prosoc_left:actor
##
##                     Df Deviance    AIC
## - condition          1   15.414 111.52
## <none>                   15.002 113.11
## - prosoc_left:actor  5   89.383 177.49
##
## Step:  AIC=111.52
## cbind(total - Freq, Freq) ~ prosoc_left + actor + prosoc_left:actor
##
##                     Df Deviance    AIC
## <none>                   15.414 111.52
## - prosoc_left:actor  5   89.730 175.83
```

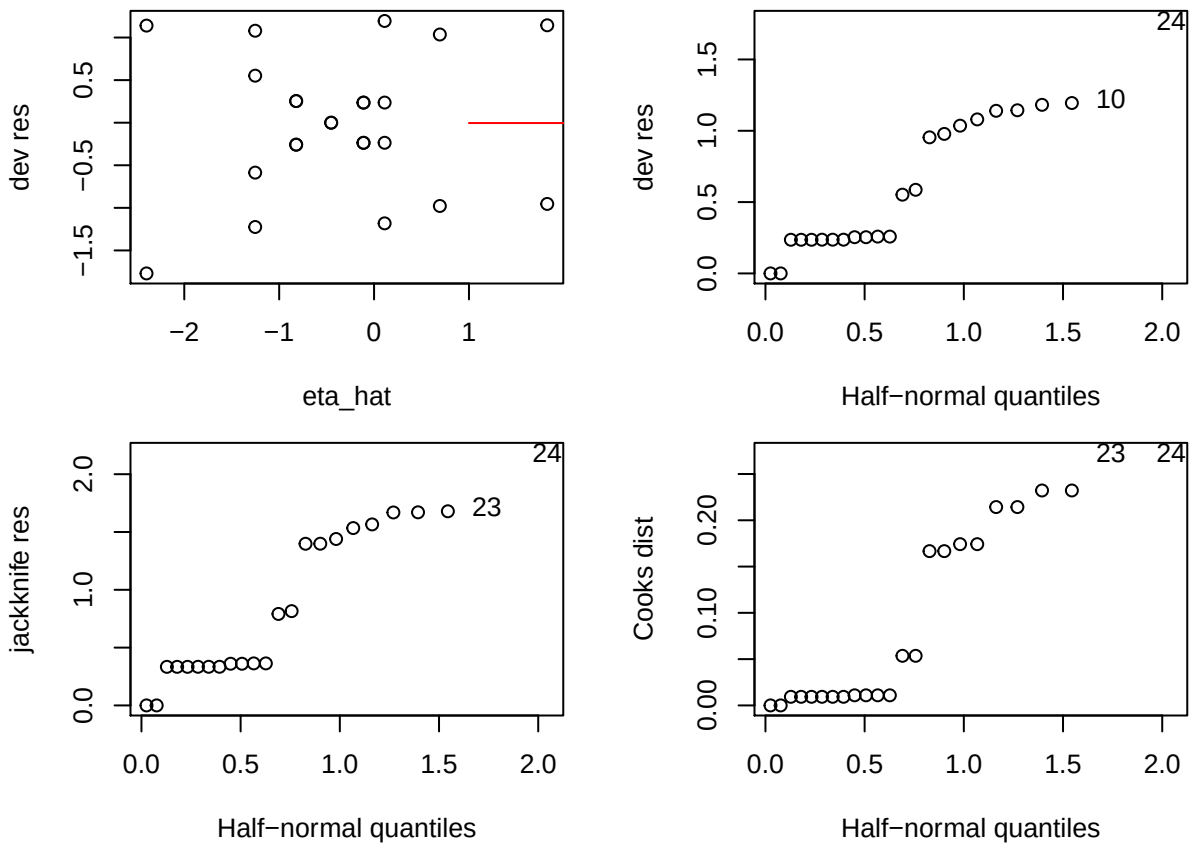
```r
library(faraway) # for halfnorm function
(D_res <- residuals(finalmodel))
```

```
##             1             2             7             8             9
##  2.537886e-01 -2.581409e-01  2.358844e-01 -2.363725e-01  5.519314e-01
##            10            15            16            17            18
## -5.862540e-01 -1.181904e+00  1.194603e+00  1.079799e+00 -1.224915e+00
##            23            24            25            26            31
## -2.358844e-01  2.363725e-01  2.537886e-01 -2.581409e-01 -2.363725e-01
##            32            33            34            39            40
##  2.358844e-01  1.035615e+00 -9.779990e-01 -5.771195e-08 -5.771195e-08
##            41            42            47            48
## -9.539987e-01  1.143727e+00  1.139990e+00 -1.769861e+00
```

```r
P_res <- residuals(finalmodel, type="pearson")
lever <- influence(finalmodel)$hat
J_res <- rstudent(finalmodel)
Cooks <- cooks.distance(finalmodel)
(eta_hat <- predict(finalmodel, type="link"))
```

```
##          1          2          7          8          9         10
## -0.8209806 -0.8209806 -0.1112256 -0.1112256 -1.2527630 -1.2527630
##         15         16         17         18         23         24
##  0.1112256  0.1112256 -1.2527630 -1.2527630  0.1112256  0.1112256
##         25         26         31         32         33         34
## -0.8209806 -0.8209806 -0.1112256 -0.1112256  0.6931472  0.6931472
##         39         40         41         42         47         48
## -0.4519851 -0.4519851  1.8245493  1.8245493 -2.3978953 -2.3978953
```

```r
par(mfrow=c(2,2))
par(mar=c(4,4,1,2))
plot(eta_hat, D_res, ylab="dev res")
lines(predict(loess(D_res ~ eta_hat)), col="red")
halfnorm(D_res, ylab="dev res")
halfnorm(J_res, ylab="jackknife res")
halfnorm(Cooks, ylab="Cooks dist")
```




```
summary(finalmodel)
```

```
##
## Call:
## glm(formula = cbind(total - Freq, Freq) ~ prosoc_left + actor +
##     prosoc_left:actor, family = binomial, data = x)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7699  -0.3402   0.0000   0.3283   1.1946
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)            -8.210e-01  3.618e-01  -2.269  0.02326 *
## prosoc_left1            7.098e-01  4.923e-01   1.442  0.14939
## actor3                 -4.318e-01  5.400e-01  -0.800  0.42396
## actor4                 -4.318e-01  5.400e-01  -0.800  0.42396
## actor5                 -6.478e-16  5.117e-01   0.000  1.00000
## actor6                  1.514e+00  5.059e-01   2.993  0.00276 **
## actor7                  2.646e+00  6.026e-01   4.390 1.13e-05 ***
## prosoc_left1:actor3     6.542e-01  7.173e-01   0.912  0.36173
## prosoc_left1:actor4     6.542e-01  7.173e-01   0.912  0.36173
## prosoc_left1:actor5    -2.601e-16  6.962e-01   0.000  1.00000
## prosoc_left1:actor6    -1.855e+00  6.959e-01  -2.666  0.00769 **
## prosoc_left1:actor7    -4.932e+00  9.156e-01  -5.387 7.16e-08 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 95.847  on 23  degrees of freedom
## Residual deviance: 15.414  on 12  degrees of freedom
## AIC: 111.52
##
## Number of Fisher Scoring iterations: 5
```

```
deviance(finalmodel)
```

```
## [1] 15.4145
```

```
df.residual(finalmodel)
```

```
## [1] 12
```

```
pchisq(deviance(finalmodel), df = df.residual(finalmodel), lower.tail = FALSE)
```

```
## [1] 0.2195474
```

```
anova(finalmodel,fullmodel,test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: cbind(total - Freq, Freq) ~ prosoc_left + actor + prosoc_left:actor
## Model 2: cbind(total - Freq, Freq) ~ (condition + prosoc_left + actor)^2
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        12     15.415
## 2         5     11.231  7   4.1838   0.7584
```

5.Conclusion