
A COMPREHENSIVE ANALYSIS ON THE PROSOCIAL BEHAVIOR OF CHIMPANZEES

Calvin Zhuoqun Huang

908525 | Tutorial information: Anubhav Kaphle, Thur 9-10

The University of Melbourne

Parkville, VIC 3010

calvin.huang@unimelb.edu.au

zhuoqunh@student.unimelb.edu.au

calvineye0627@gmail.com

September 14, 2019

ABSTRACT

Prosocial behaviour as a means of benefiting others is a social behaviour that is widely demonstrated in Homo sapiens. However, whether it is observed in the population of the chimpanzee is an unknown but interesting question. This totally (not) novel paper performs comprehensive analysis on a chimpanzee dataset to evaluate the prosocial preference of 7 individual entities. By demonstrating the effect of having a controlled experiment, with the assist of modern applied statistical tools, such as GLM and Interaction Test, we conclude that chimpanzee typically will not behave prosocially.

Keywords Chimpanzee · joke · analysis

1 Introduction

I do not bother writing a whole chunk of background information, just going to copy-paste assignment spec and some memes :)

> When human students participate in an experiment like this, they nearly always choose the lever linked to two pieces of food, the prosocial option, but only 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.



2 Study Goal/Procedure

> The question is whether a focal chimpanzee behaves similarly, choosing the prosocial option more often when another animal is present. If you conclude that chimpanzees behave differently, describe their behaviour (i.e., interpret your final model).

The significance of each attribute are summarised below:

Attribute	Significance
pulled_prosocial	(prosoc_left == pulled_left) response of interest
actor	high change there exists difference between each individual
condition	the primary attribute of interest, does this have influence of prosocial choice?
pulled_left	also used to decide handedness (same side/same condition, what choice is taken)
prosoc_left	used to study handedness

The analysis of this dataset is divided into the following stages

1. Preprocess the dataset.
2. Does each chimpanzee has handedness? Alternatively, in other words, is there a difference between the portion of left/right choices between each chimpanzee?
3. Does condition affect the prosocial choice (like it does for human)?
4. Does chimpanzee have tendency of choosing the prosocial option regardless of the other individual? That is, how does their choice compare to completely random choice ($p=0.5$)?

3 Preprocessing and Summary statistics

3.1 Preprocessing

The preprocessing carried out is very basic as shown below. Check inline comments for explanation.

```
## Read data
data = read.csv("assign2.txt", sep=" ")
## Process data
# Create new column on if prosocial
data$pulled_prosocial = as.numeric(data$prosoc_left == data$pulled_left)
# Create new column of 1s for faster aggregation
data$const = 1
# Aggregate data based on each individual
data.aggreated = aggregate(~actor+condition+pulled_prosocial, sum, d=data)
```

3.2 Summary Statistics

```
aggregate(~actor+condition+prosoc_left, sum, d=data)

##      actor condition prosoc_left pulled_left pulled_prosocial const
## 1         1         0           0           6           12      18
## 2         2         0           0          18           0      18
## 3         3         0           0           5          13      18
## 4         4         0           0           6          12      18
## 5         5         0           0           6          12      18
## 6         6         0           0          14           4      18
## 7         7         0           0          14           4      18
## 8         1         1           0           5          13      18
## 9         2         1           0          18           0      18
## 10        3         1           0           3          15      18
## 11        4         1           0           2          16      18
## 12        5         1           0           5          13      18
## 13        6         1           0          10           8      18
## 14        7         1           0          17           1      18
## 15        1         0           1           9           9      18
## 16        2         0           1          18          18      18
## 17        3         0           1          11          11      18
## 18        4         0           1           9           9      18
## 19        5         0           1          10          10      18
## 20        6         0           1          11          11      18
## 21        7         0           1          15          15      18
## 22        1         1           1          10          10      18
## 23        2         1           1          18          18      18
## 24        3         1           1           6           6      18
```

##	25	4	1	1	8	8	18
##	26	5	1	1	9	9	18
##	27	6	1	1	11	11	18
##	28	7	1	1	18	18	18

The above summary shows that the experiment design is balanced; all types of condition has the same count of the data point.

4 Handedness

4.1 Plot

Begin investigation by plotting the factor level interaction plot of this dataset.

```
# Investigate if the side of lever the chimpanzee pulled depends on the condition*prosoc_left
## Plot 1: individual/condition*prosoc_left
ggplot(data = data,
  aes(x = interaction(factor(condition), factor(prosoc_left)),
    y = pulled_left, colour = factor(actor),
    group=factor(actor))) +
  stat_summary(fun.y=mean, geom="point") +
  stat_summary(fun.y=mean, geom="line") + xlab("Condition, Prosoc_left") +
  ylab("p(pulled_left)") + ylim(0, 1)
```

From this plot we see there are certainly tendency for each chimpanzee to favour lever on different sides (or in other words, having handedness). Especially individual 2 and 7, who almost always pulls left lever. And having the prosocial option on the left hand side seems to significantly impact the decision making of some of the chimpanzees.

4.2 Model and verification

```
# Full model
data.handedness.bmodel = glm(pulled_left ~ factor(actor)+factor(condition)
  +factor(prosoc_left), data = data, family=poisson)
summary(data.handedness.bmodel)

##
```

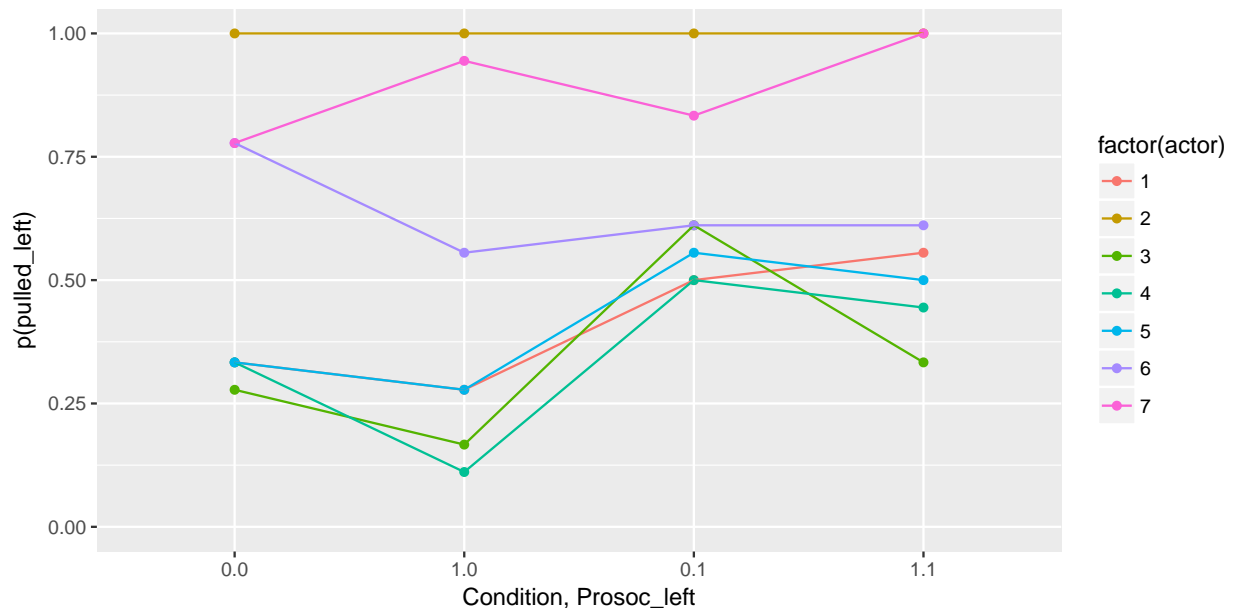


Figure 1: Factor Plot for Handedness

```
## Call:
## glm(formula = pulled_left ~ factor(actor) + factor(condition) +
##      factor(prosoc_left), family = poisson, data = data)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.43748  -0.86223  -0.03281   0.48835   1.01758
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -9.590e-01  2.020e-01  -4.747 2.07e-06 ***
## factor(actor)2     8.755e-01  2.173e-01   4.029 5.61e-05 ***
## factor(actor)3    -1.823e-01  2.708e-01  -0.673 0.500772
## factor(actor)4    -1.823e-01  2.708e-01  -0.673 0.500772
## factor(actor)5    -5.229e-12  2.582e-01   0.000 1.000000
## factor(actor)6     4.274e-01  2.347e-01   1.821 0.068541 .
## factor(actor)7     7.577e-01  2.213e-01   3.424 0.000616 ***
## factor(condition)1 -8.224e-02  1.171e-01  -0.702 0.482647
## factor(prosoc_left)1 2.339e-01  1.178e-01   1.985 0.047125 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##      Null deviance: 318.76  on 503  degrees of freedom
## Residual deviance: 262.67  on 495  degrees of freedom
## AIC: 864.67
##
## Number of Fisher Scoring iterations: 5
```

Our result from visualisation is confirmed from this fitted model. The impact of the condition is shallow, while some individuals have some preference compared with others. The strange observation is that the prosocial side also have a significant impact on the choice but not the condition.

We progress our investigation by refining the model with step function

```
# Step model
data.handedness.bmodel.step = step(data.handedness.bmodel)

## Start:  AIC=864.67
## pulled_left ~ factor(actor) + factor(condition) + factor(prosoc_left)
##
##              Df Deviance    AIC
## - factor(condition)    1   263.16 863.16
## <none>                   262.67 864.67
## - factor(prosoc_left)    1   266.64 866.64
## - factor(actor)          6   314.30 904.30
##
## Step:  AIC=863.16
## pulled_left ~ factor(actor) + factor(prosoc_left)
##
##              Df Deviance    AIC
## <none>                   263.16 863.16
## - factor(prosoc_left)    1   267.13 865.13
## - factor(actor)          6   314.79 902.79

summary(data.handedness.bmodel.step)

##
## Call:
## glm(formula = pulled_left ~ factor(actor) + factor(prosoc_left),
##      family = poisson, data = data)
##
## Deviance Residuals:
```

```
##      Min      1Q      Median      3Q      Max
## -1.40882 -0.85808  0.00763   0.52214  0.98831
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -9.993e-01  1.941e-01  -5.149 2.62e-07 ***
## factor(actor)2     8.755e-01  2.173e-01   4.029 5.61e-05 ***
## factor(actor)3    -1.823e-01  2.708e-01  -0.673 0.500772
## factor(actor)4    -1.823e-01  2.708e-01  -0.673 0.500772
## factor(actor)5    -4.199e-15  2.582e-01   0.000 1.000000
## factor(actor)6     4.274e-01  2.347e-01   1.821 0.068541 .
## factor(actor)7     7.577e-01  2.213e-01   3.424 0.000616 ***
## factor(prosoc_left)1 2.339e-01  1.178e-01   1.985 0.047125 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##      Null deviance: 318.76  on 503  degrees of freedom
## Residual deviance: 263.16  on 496  degrees of freedom
## AIC: 863.16
##
## Number of Fisher Scoring iterations: 5
```

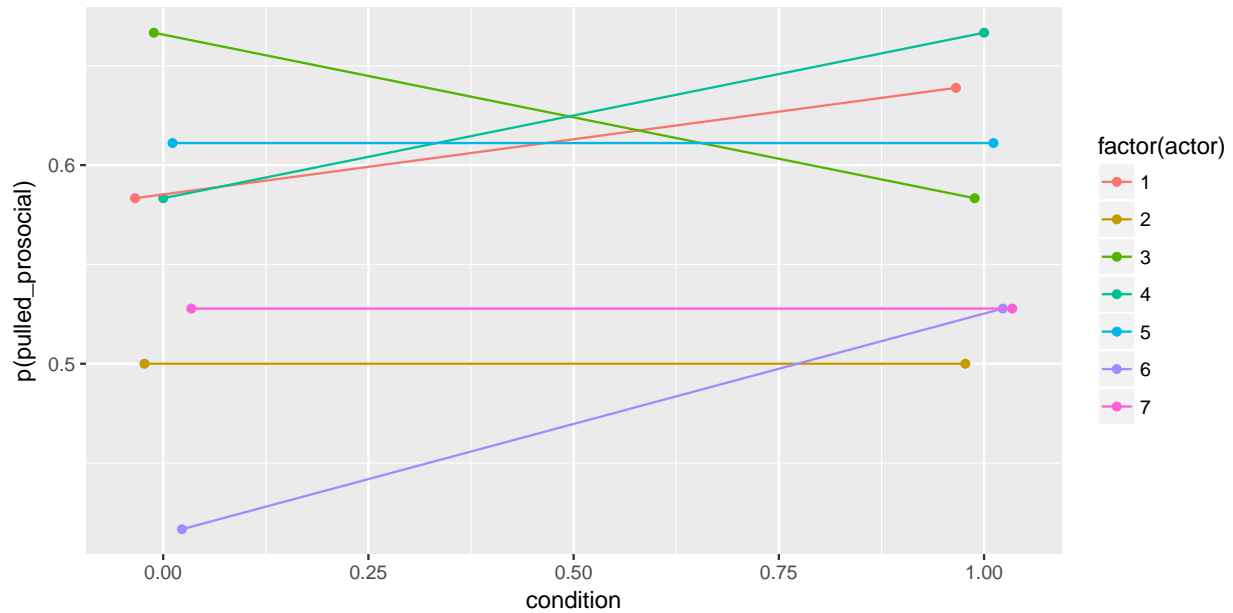
Again, after the selection using AIC with step function, the result is as expected, the condition got dropped, leaving only the other two factors being significant. Based on this model and the above plot, we will divide the chimpanzee into two groups: The neutral group (6), the left handedness group (2, 7) and the right handedness group (1, 3,4,5)

5 Inference on Prosocial Choice

5.1 Plot

Plot the relation between condition and mean value of $p(\text{pulled_prosocial})$ grouped by each actor

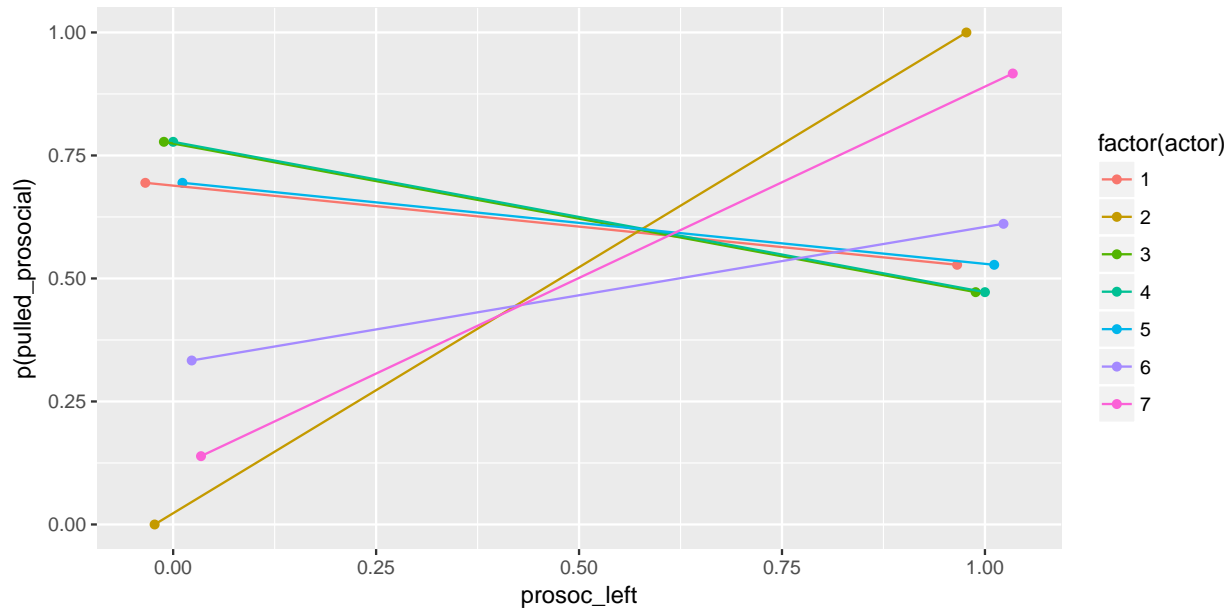
```
# Plot the individual plot
ggplot(data = data,
       aes(x = condition,
           y = pulled_prosocial, colour = factor(actor),
           group=factor(actor))) +
  stat_summary(fun.y=mean, geom="point",
              position=position_dodge(width=0.08)) +
  stat_summary(fun.y=mean, geom="line",
              position=position_dodge(width=0.08)) +
  xlab("condition") +
  ylab("p(pulled_prosocial) ")
```



There are three individuals having increased prosocial choice when there is another individual on the other side, while one individual have decreased rate. And the rest having non changing rate. It suggests there aren't lots of interaction between those two attributes, suggesting initially chimpanzee is not prosocial.

Additionally, the relation between `prosoc_left` and `pulled_prosocial` is also plotted. The reason is `prosoc_left` has demonstrated great effect on `pulled_left` in the handedness test. It might also be significant for this relation.

```
ggplot(data = data,
  aes(x = prosoc_left,
    y = pulled_prosocial, colour = factor(actor),
    group=factor(actor))) +
  stat_summary(fun.y=mean, geom="point",
    position=position_dodge(width=0.08)) +
  stat_summary(fun.y=mean, geom="line",
    position=position_dodge(width=0.08)) +
  xlab("prosoc_left") +
  ylab("p(pulled_prosocial) ")
```



There is a much stronger relation between those two than between the condition and pulled_prosocial

5.2 Contingency Test

Start by testing independence using contingency table and we chose a three way test.

```
# Phrase test like this:
#   Factor1: isProsocial
#   Factor2: Conditioned
#   Factor3: The chimp ID
data.aggregated.pmodel1 = glm(const ~ actor + condition + pulled_prosocial,
                              data = data.aggregated, family=poisson)
summary(data.aggregated.pmodel1)

##
## Call:
## glm(formula = const ~ actor + condition + pulled_prosocial, family = poisson,
##      data = data.aggregated)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.26120  -0.40522  -0.00993   0.35669   1.30553
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.745e+00  1.204e-01  22.79  < 2e-16 ***
## actor        1.150e-16  2.227e-02   0.00  1.00000
## condition    3.922e-16  8.909e-02   0.00  1.00000
## pulled_prosocial 2.715e-01  8.991e-02   3.02  0.00253 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##      Null deviance: 18.9075  on 27  degrees of freedom
## Residual deviance:  9.7048  on 24  degrees of freedom
## AIC: 149.83
##
## Number of Fisher Scoring iterations: 4
# Test independence
```

```

deviance(data.agggregated.pmodel1)

## [1] 9.704807

df.residual(data.agggregated.pmodel1)

## [1] 24

pchisq(deviance(data.agggregated.pmodel1),
        df=df.residual(data.agggregated.pmodel1), lower.tail = FALSE)

## [1] 0.9956601

# Close to 1, so we conclude independent

# Use table
data.agggregated.xtabs1 <- xtabs(const ~ actor + pulled_prosocial + condition,
                                data = data.agggregated)

summary(data.agggregated.xtabs1)

## Call: xtabs(formula = const ~ actor + pulled_prosocial + condition,
##             data = data.agggregated)
## Number of cases in table: 504
## Number of factors: 3
## Test for independence of all factors:
##   Chisq = 9.668, df = 19, p-value = 0.9607

```

5.3 Reduce from Full Interaction Model (with prosocial side)

We start the model fitting with all the interactions and gradually step it down. Binomial model is chosen as it evaluating the chance of picking prosocial options. Or in other words, it's a Bernoulli Random variable $Ber(y_i|\theta_{x_i})$, so binomial model makes sense.

```

# The binomial model (full interaction)
# Try with all factors, (not including pulled_left as it's not something we want to study)
data.bmodel1 = glm(cbind(pulled_prosocial, 1-pulled_prosocial)~factor(actor)*
                   factor(condition)*factor(prosoc_left), data=data,
                   family = "binomial")

summary(data.bmodel1)

##
## Call:
## glm(formula = cbind(pulled_prosocial, 1 - pulled_prosocial) ~
##      factor(actor) * factor(condition) * factor(prosoc_left),
##      family = "binomial", data = data)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.09629  -0.90052   0.00013   0.90052   2.40432
##
## Coefficients:
##
##              Estimate
## (Intercept)      6.931e-01
## factor(actor)2    -1.926e+01
## factor(actor)3      2.624e-01
## factor(actor)4      1.163e-14
## factor(actor)5      1.223e-14
## factor(actor)6    -1.946e+00
## factor(actor)7    -1.946e+00
## factor(condition)1  2.624e-01
## factor(prosoc_left)1 -6.931e-01
## factor(actor)2:factor(condition)1 -2.624e-01
## factor(actor)3:factor(condition)1  3.916e-01
## factor(actor)4:factor(condition)1  1.124e+00

```



```

## factor(actor)5:factor(condition)1 -2.041e-14
## factor(actor)6:factor(condition)1 7.673e-01
## factor(actor)7:factor(condition)1 -1.843e+00
## factor(actor)2:factor(prosoc_left)1 3.783e+01
## factor(actor)3:factor(prosoc_left)1 1.896e-01
## factor(actor)4:factor(prosoc_left)1 -2.697e-14
## factor(actor)5:factor(prosoc_left)1 2.231e-01
## factor(actor)6:factor(prosoc_left)1 2.398e+00
## factor(actor)7:factor(prosoc_left)1 3.555e+00
## factor(condition)1:factor(prosoc_left)1 -3.922e-02
## factor(actor)2:factor(condition)1:factor(prosoc_left)1 3.922e-02
## factor(actor)3:factor(condition)1:factor(prosoc_left)1 -1.760e+00
## factor(actor)4:factor(condition)1:factor(prosoc_left)1 -1.570e+00
## factor(actor)5:factor(condition)1:factor(prosoc_left)1 -4.463e-01
## factor(actor)6:factor(condition)1:factor(prosoc_left)1 -9.904e-01
## factor(actor)7:factor(condition)1:factor(prosoc_left)1 1.858e+01
## Std. Error z value
## (Intercept) 5.000e-01 1.386
## factor(actor)2 1.537e+03 -0.013
## factor(actor)3 7.259e-01 0.361
## factor(actor)4 7.071e-01 0.000
## factor(actor)5 7.071e-01 0.000
## factor(actor)6 7.559e-01 -2.574
## factor(actor)7 7.559e-01 -2.574
## factor(condition)1 7.259e-01 0.361
## factor(prosoc_left)1 6.872e-01 -1.009
## factor(actor)2:factor(condition)1 2.174e+03 0.000
## factor(actor)3:factor(condition)1 1.097e+00 0.357
## factor(actor)4:factor(condition)1 1.157e+00 0.971
## factor(actor)5:factor(condition)1 1.027e+00 0.000
## factor(actor)6:factor(condition)1 1.036e+00 0.741
## factor(actor)7:factor(condition)1 1.381e+00 -1.334
## factor(actor)2:factor(prosoc_left)1 2.174e+03 0.017
## factor(actor)3:factor(prosoc_left)1 9.914e-01 0.191
## factor(actor)4:factor(prosoc_left)1 9.718e-01 0.000
## factor(actor)5:factor(prosoc_left)1 9.733e-01 0.229
## factor(actor)6:factor(prosoc_left)1 1.014e+00 2.366
## factor(actor)7:factor(prosoc_left)1 1.093e+00 3.254
## factor(condition)1:factor(prosoc_left)1 9.870e-01 -0.040
## factor(actor)2:factor(condition)1:factor(prosoc_left)1 3.075e+03 0.000
## factor(actor)3:factor(condition)1:factor(prosoc_left)1 1.461e+00 -1.204
## factor(actor)4:factor(condition)1:factor(prosoc_left)1 1.495e+00 -1.051
## factor(actor)5:factor(condition)1:factor(prosoc_left)1 1.396e+00 -0.320
## factor(actor)6:factor(condition)1:factor(prosoc_left)1 1.410e+00 -0.702
## factor(actor)7:factor(condition)1:factor(prosoc_left)1 1.537e+03 0.012
## Pr(>|z|)
## (Intercept) 0.16566
## factor(actor)2 0.99001
## factor(actor)3 0.71777
## factor(actor)4 1.00000
## factor(actor)5 1.00000
## factor(actor)6 0.01005 *
## factor(actor)7 0.01005 *
## factor(condition)1 0.71777
## factor(prosoc_left)1 0.31313
## factor(actor)2:factor(condition)1 0.99990
## factor(actor)3:factor(condition)1 0.72119
## factor(actor)4:factor(condition)1 0.33148
## factor(actor)5:factor(condition)1 1.00000
## factor(actor)6:factor(condition)1 0.45895
## factor(actor)7:factor(condition)1 0.18207
## factor(actor)2:factor(prosoc_left)1 0.98612
## factor(actor)3:factor(prosoc_left)1 0.84832
## factor(actor)4:factor(prosoc_left)1 1.00000

```

```
## factor(actor)5:factor(prosoc_left)1 0.81865
## factor(actor)6:factor(prosoc_left)1 0.01800 *
## factor(actor)7:factor(prosoc_left)1 0.00114 **
## factor(condition)1:factor(prosoc_left)1 0.96830
## factor(actor)2:factor(condition)1:factor(prosoc_left)1 0.99999
## factor(actor)3:factor(condition)1:factor(prosoc_left)1 0.22841
## factor(actor)4:factor(condition)1:factor(prosoc_left)1 0.29345
## factor(actor)5:factor(condition)1:factor(prosoc_left)1 0.74917
## factor(actor)6:factor(condition)1:factor(prosoc_left)1 0.48242
## factor(actor)7:factor(condition)1:factor(prosoc_left)1 0.99036
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 689.49 on 503 degrees of freedom
## Residual deviance: 492.28 on 476 degrees of freedom
## AIC: 548.28
##
## Number of Fisher Scoring iterations: 17
```

Checking the model parameter and p-value carefully, one can easily observe that lots of interaction terms are not significant. Thus a step function will certainly be ideal for reducing the number of noise parameters in the model. Additionally, from the significance of interaction between actor and condition, we can see that except for actor 7, other actors all seems indifferent to if anyone is on the other side.

5.3.1 Reduced model

The variable selection is carried out with AIC, we choose AIC as it has scales well with smaller datasets and can either add or remove parameters and genuinely does not stuck in a BAD local optimum.

```
data.bmodel1.step = step(data.bmodel1)

## Start:  AIC=548.28
## cbind(pulled_prosocial, 1 - pulled_prosocial) ~ factor(actor) *
##   factor(condition) * factor(prosoc_left)
##
##                                     Df Deviance   AIC
## - factor(actor):factor(condition):factor(prosoc_left)  6   503.51 547.51
## <none>                                                  492.28 548.28
##
## Step:  AIC=547.51
## cbind(pulled_prosocial, 1 - pulled_prosocial) ~ factor(actor) +
##   factor(condition) + factor(prosoc_left) + factor(actor):factor(condition) +
##   factor(actor):factor(prosoc_left) + factor(condition):factor(prosoc_left)
##
##                                     Df Deviance   AIC
## - factor(actor):factor(condition)          6   505.48 537.48
## - factor(condition):factor(prosoc_left)     1   505.30 547.30
## <none>                                     503.51 547.51
## - factor(actor):factor(prosoc_left)         6   665.32 697.32
##
## Step:  AIC=537.48
## cbind(pulled_prosocial, 1 - pulled_prosocial) ~ factor(actor) +
##   factor(condition) + factor(prosoc_left) + factor(actor):factor(prosoc_left) +
##   factor(condition):factor(prosoc_left)
##
##                                     Df Deviance   AIC
## - factor(condition):factor(prosoc_left)     1   507.29 537.29
## <none>                                     505.48 537.48
## - factor(actor):factor(prosoc_left)         6   667.32 687.32
##
## Step:  AIC=537.29
```

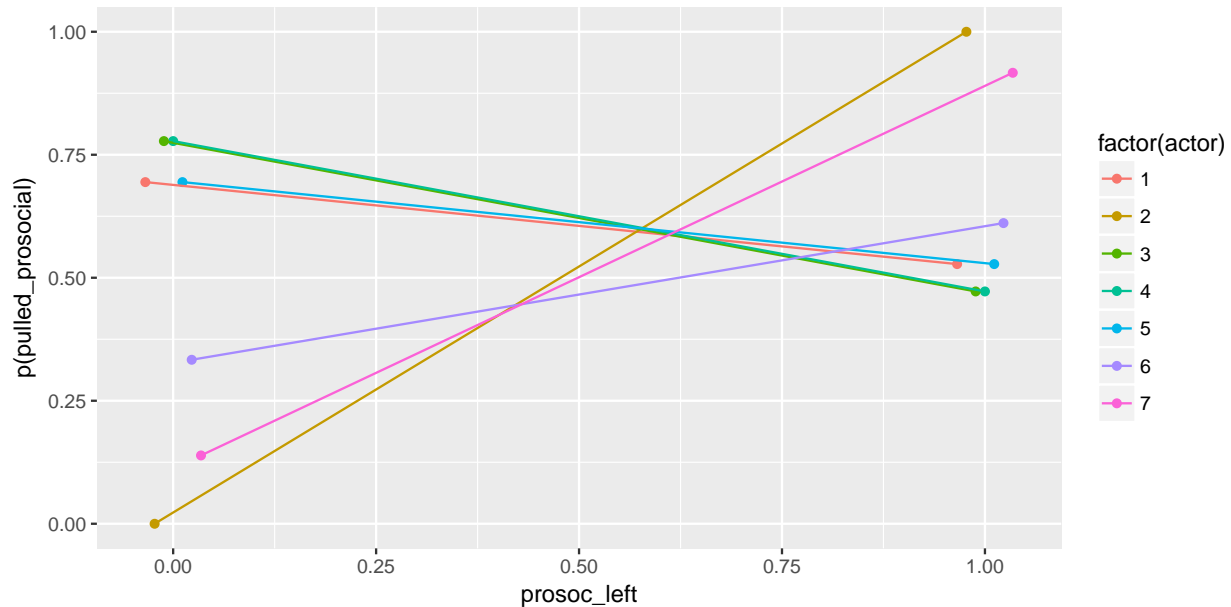
```
## cbind(pulled_prosocial, 1 - pulled_prosocial) ~ factor(actor) +
##     factor(condition) + factor(prosoc_left) + factor(actor):factor(prosoc_left)
##
##               Df Deviance    AIC
## - factor(condition)           1   507.70 535.70
## <none>                        507.29 537.29
## - factor(actor):factor(prosoc_left) 6   668.48 686.48
##
## Step: AIC=535.7
## cbind(pulled_prosocial, 1 - pulled_prosocial) ~ factor(actor) +
##     factor(prosoc_left) + factor(actor):factor(prosoc_left)
##
##               Df Deviance    AIC
## <none>                        507.70 535.70
## - factor(actor):factor(prosoc_left) 6   668.79 684.79

summary(data.bmodel1.step)

##
## Call:
## glm(formula = cbind(pulled_prosocial, 1 - pulled_prosocial) ~
##     factor(actor) + factor(prosoc_left) + factor(actor):factor(prosoc_left),
##     family = "binomial", data = data)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.22931  -1.13056   0.00013   0.85398   1.98700
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)      8.210e-01  3.618e-01   2.269  0.02326
## factor(actor)2    -1.939e+01  1.087e+03  -0.018  0.98577
## factor(actor)3     4.318e-01  5.400e-01   0.800  0.42396
## factor(actor)4     4.318e-01  5.400e-01   0.800  0.42396
## factor(actor)5    -6.606e-15  5.117e-01   0.000  1.00000
## factor(actor)6    -1.514e+00  5.059e-01  -2.993  0.00276
## factor(actor)7    -2.646e+00  6.026e-01  -4.390 1.13e-05
## factor(prosoc_left)1 -7.098e-01  4.923e-01  -1.442  0.14939
## factor(actor)2:factor(prosoc_left)1  3.784e+01  1.537e+03   0.025  0.98036
## factor(actor)3:factor(prosoc_left)1 -6.542e-01  7.173e-01  -0.912  0.36173
## factor(actor)4:factor(prosoc_left)1 -6.542e-01  7.173e-01  -0.912  0.36173
## factor(actor)5:factor(prosoc_left)1 -3.771e-15  6.962e-01   0.000  1.00000
## factor(actor)6:factor(prosoc_left)1  1.855e+00  6.959e-01   2.666  0.00769
## factor(actor)7:factor(prosoc_left)1  4.932e+00  9.156e-01   5.387 7.16e-08
##
## (Intercept) *
## factor(actor)2
## factor(actor)3
## factor(actor)4
## factor(actor)5
## factor(actor)6 **
## factor(actor)7 ***
## factor(prosoc_left)1
## factor(actor)2:factor(prosoc_left)1
## factor(actor)3:factor(prosoc_left)1
## factor(actor)4:factor(prosoc_left)1
## factor(actor)5:factor(prosoc_left)1
## factor(actor)6:factor(prosoc_left)1 **
## factor(actor)7:factor(prosoc_left)1 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
## Null deviance: 689.49 on 503 degrees of freedom
## Residual deviance: 507.70 on 490 degrees of freedom
## AIC: 535.7
##
## Number of Fisher Scoring iterations: 17
```

Only attribute actor, prosoc_left and the interaction between those two attributes are left. If we compare the result with previous figure (Because Knitr with autoref doesn't seem to work for me, so instead of a hyperlink I will put the figure here for easier comparison)



Because here, we are using actor 1 as our `contr.treatment`, so all measurement are relative to actor 1. From the p-values, it's not hard to observe only actor 1, 6, 7 has some significance. We will analyse the reason below.

Observe `prosoc_left1` has p value of 0.15, this indicates there are switching side of `prosoc_option` has some significance to actor 1. From the plot, the baseline and trend of individual 3,4,5's behavior closely resembles actor 1, this is the reason why they don't get significance for either actor or the interaction of `prosoc_side`. The same analysis can be applied to individual 7 and 2. And individual 6 is different from the rest of the groups. And these corresponds with the handedness analysis we have above.

The model also suggests condition have no importance on the model.

5.3.2 Diagnostics

There are some diagnostics produced in my code, but they are not important for the final conclusion, so I will not put it here. Only dispersion is listed below:

```
# Check overdispersion
# Estimate Phi
n = dim(data)[1]
p = 2
(data.bmodel1.step.phihat = sum(residuals(data.bmodel1.step, type =
  "pearson")^2) / (n-p))

## [1] 0.8605578
```

The value is less than 1, so no overdispersion

5.4 Reduce from Full Interaction Model (without prosocial side)

We now remove the `prosoc_left` as it's not important to our analysis (only significant because we want to see how much the handedness affects the prosocial choice).

```

# The binomial model (full interaction)
# Try with all factors, (not including pulled_left as it's not something we want to study)
data.bmodel2 = glm(cbind(pulled_prosocial, 1-pulled_prosocial) ~
                    factor(actor)*factor(condition), data=data,
                    family = "binomial")
summary(data.bmodel2)

##
## Call:
## glm(formula = cbind(pulled_prosocial, 1 - pulled_prosocial) ~
##      factor(actor) * factor(condition), family = "binomial", data = data)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.4823  -1.2250   0.9005   1.0383   1.3232
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)      3.365e-01  3.381e-01   0.995   0.320
## factor(actor)2    -3.365e-01  4.748e-01  -0.709   0.478
## factor(actor)3     3.567e-01  4.892e-01   0.729   0.466
## factor(actor)4    -1.888e-15  4.781e-01   0.000   1.000
## factor(actor)5     1.155e-01  4.808e-01   0.240   0.810
## factor(actor)6    -6.729e-01  4.781e-01  -1.408   0.159
## factor(actor)7    -2.252e-01  4.751e-01  -0.474   0.635
## factor(condition)1  2.341e-01  4.844e-01   0.483   0.629
## factor(actor)2:factor(condition)1 -2.341e-01  6.760e-01  -0.346   0.729
## factor(actor)3:factor(condition)1 -5.907e-01  6.885e-01  -0.858   0.391
## factor(actor)4:factor(condition)1  1.226e-01  6.885e-01   0.178   0.859
## factor(actor)5:factor(condition)1 -2.341e-01  6.844e-01  -0.342   0.732
## factor(actor)6:factor(condition)1  2.136e-01  6.785e-01   0.315   0.753
## factor(actor)7:factor(condition)1 -2.341e-01  6.765e-01  -0.346   0.729
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 689.49  on 503  degrees of freedom
## Residual deviance: 679.78  on 490  degrees of freedom
## AIC: 707.78
##
## Number of Fisher Scoring iterations: 4

```

Non of the attributes are significant.

Step the model:

```

data.bmodel2.step = step(data.bmodel2)

## Start:  AIC=707.78
## cbind(pulled_prosocial, 1 - pulled_prosocial) ~ factor(actor) *
##      factor(condition)
##
##              Df Deviance    AIC
## - factor(actor):factor(condition)  6   681.69 697.69
## <none>                                679.78 707.78
##
## Step:  AIC=697.69
## cbind(pulled_prosocial, 1 - pulled_prosocial) ~ factor(actor) +
##      factor(condition)
##
##              Df Deviance    AIC
## - factor(actor)      6   689.20 693.20
## - factor(condition)  1   681.98 695.98
## <none>                681.69 697.69
##

```

```
## Step: AIC=693.2
## cbind(pulled_prosocial, 1 - pulled_prosocial) ~ factor(condition)
##
##           Df Deviance   AIC
## - factor(condition) 1   689.49 691.49
## <none>                689.20 693.20
##
## Step: AIC=691.49
## cbind(pulled_prosocial, 1 - pulled_prosocial) ~ 1

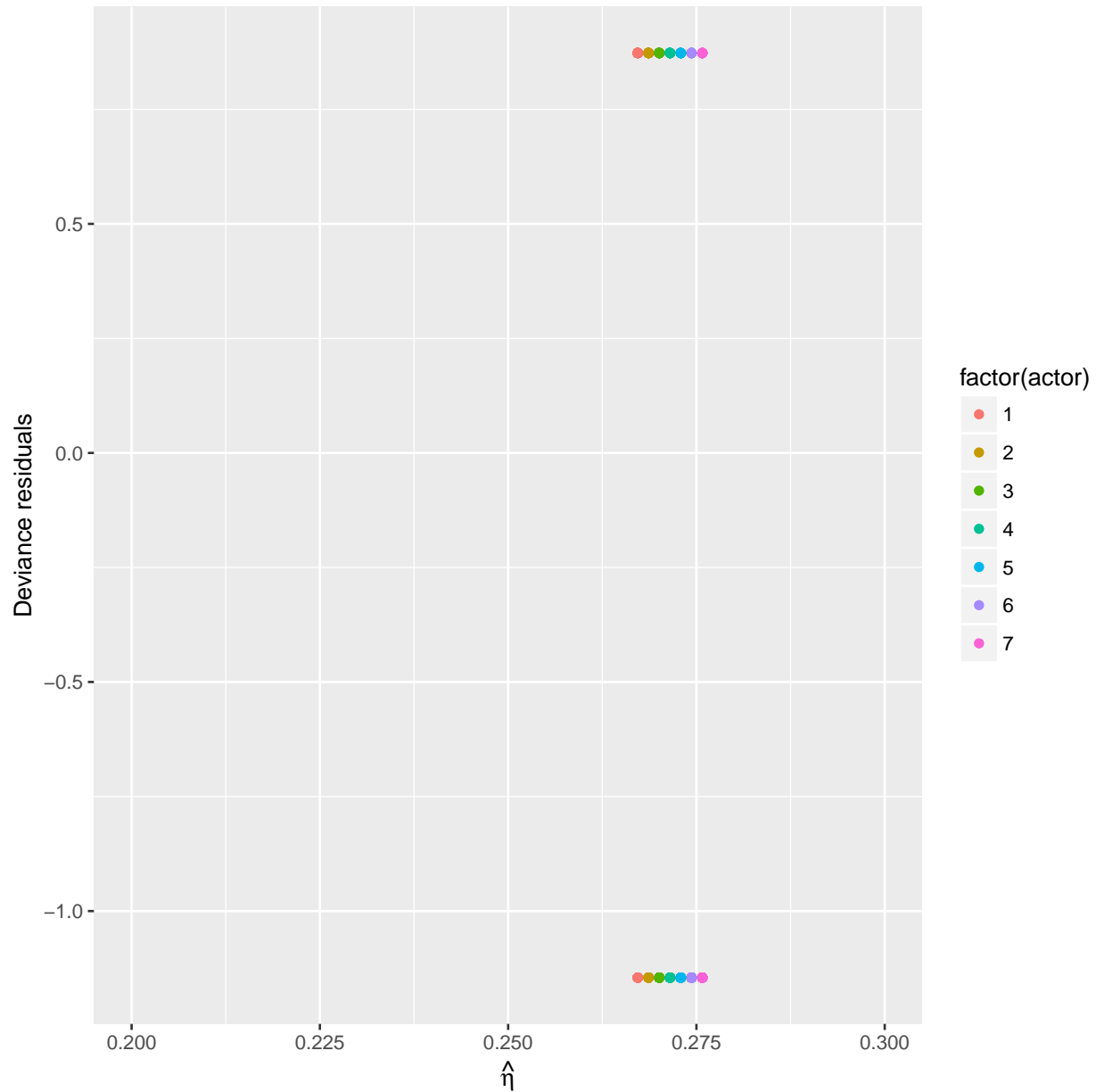
summary(data.bmodel2.step)

##
## Call:
## glm(formula = cbind(pulled_prosocial, 1 - pulled_prosocial) ~
##      1, family = "binomial", data = data)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.295  -1.295   1.065   1.065   1.065
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.27150    0.08991    3.02  0.00253 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 689.49  on 503  degrees of freedom
## Residual deviance: 689.49  on 503  degrees of freedom
## AIC: 691.49
##
## Number of Fisher Scoring iterations: 4
```

There is only intercept term left. And this suggests neither actor nor condition have an effect on the prosocial choice of chimpanzee.

5.4.1 Diagnostics

```
# Check residuals
ggplot(data = data,
       aes(x = predict(data.bmodel2.step, type="link", data=data),
           y = residuals(data.bmodel2.step, data=data, type="pearson"),
           colour=factor(actor),
           group=factor(actor))) +
  geom_point(position=position_dodge(width=0.01)) +
  xlab(expression(hat(eta))) +
  ylab("Deviance residuals") +
  xlim(0.2, 0.3)
```



Because the residual plots will only have two points, it's boring to look at, so I will just put it here and give no additional comment ;)

```
# Check overdispersion
# Estimate Phi
n = dim(data)[1]
p = 2
(data.bmodel2.step.phihat = sum(residuals(data.bmodel2.step, type =
  "pearson")^2) / (n-p))
## [1] 1.003984
```

The value is exactly 1, so no overdispersion.

In addition we also check confidence interval:

```
# Check mean
data.bmodel2.step.predict = predict(data.bmodel2.step, type="link", se.fit=TRUE)
```

```

# Check conf interval (link)
critval <- 1.96 ## approx 95% CI
upr <- data.bmodel2.step.predict$fit + (critval * data.bmodel2.step.predict$se.fit)
lwr <- data.bmodel2.step.predict$fit - (critval * data.bmodel2.step.predict$se.fit)
fit <- data.bmodel2.step.predict$fit
distinct(data.frame(upr, fit, lwr))

##           upr           fit           lwr
## 1 0.4477187 0.2714967 0.09527477

# Check conf interval (response)
f = data.bmodel2$family$linkinv
distinct(data.frame(f(upr), f(fit), f(lwr)))

##      f.upr.      f.fit.      f.lwr.
## 1 0.6100967 0.5674603 0.5238007

```

The predicted value and confidence interval is around 0.55 and doesn't include 0.5. This indicates chimpanzee population have the more chance of picking the prosocial option compared to the non-prosocial option regardless if there is an individual on the other side. As the CI not including 0.5 is equivalent to significance test, We won't conduct a separate test.

6 Summary

In essence, from the fitted binomial models/independence tests and all diagnostics. We here answer the questions above:

- Chimpanzee individuals does have strong handedness.
- Chimpanzee individuals' choice of prosocial option is affected by the position of the prosocial option. (left:2,7; right:1,3,4,5; no obvious preference:6)
- Chimpanzee individuals' choice of prosocial option is independent of existence of individual on the other side (or the condition predictor).
- Chimpanzee have the tendency of choosing more prosocial option (0.55) than the non-prosocial option in general. And is statistically significant compared to random choice (0.5)

Ideally more indicative bar plot with CI should be made.

