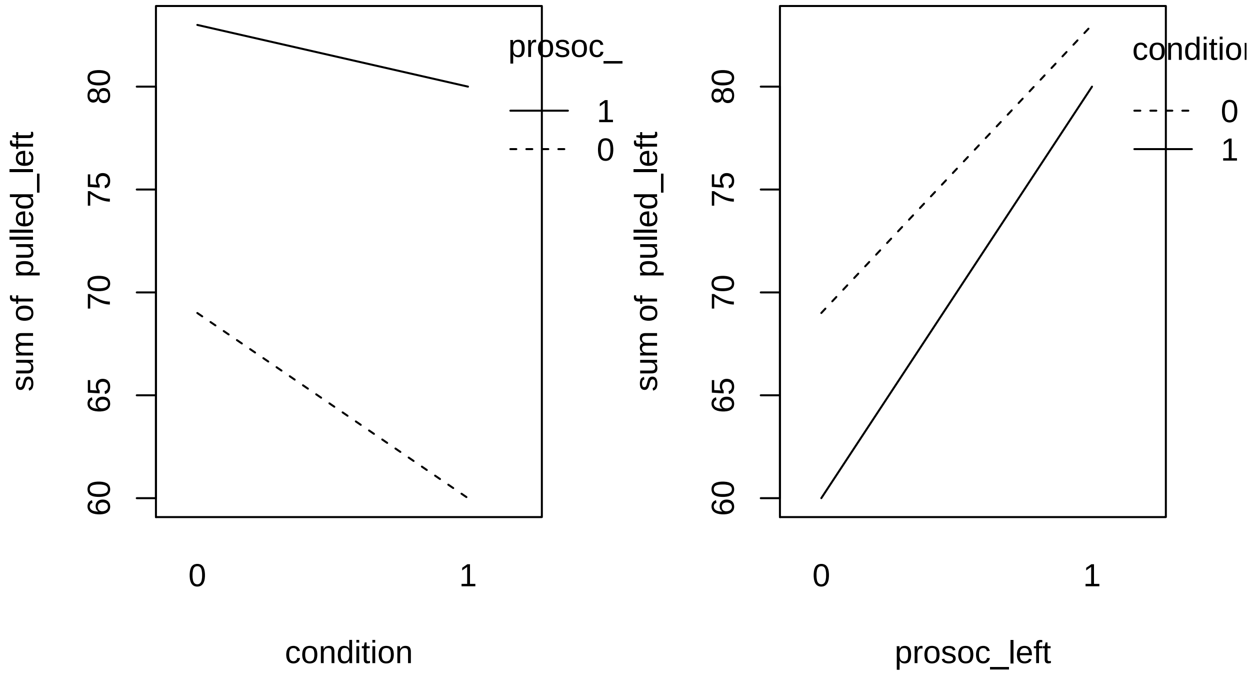
Logistic model - Monkey Experimental Data

```
ilogit <- function(x) exp(x)/(1+exp(x))
```

```
rm(list=ls())

library(faraway)
library(ggplot2)

# Set your working directory
setwd(getwd())
monk_data <- read.delim('monkey_data.txt', header=TRUE, sep = "")
monk_data_df <- read.delim('monkey_data.txt', header=TRUE, sep = "")

monk_data_df$actor <- factor(monk_data_df$actor);
monk_data_df$condition <- factor(monk_data_df$condition);
monk_data_df$prosoc_left <- factor(monk_data_df$prosoc_left);
```

```
old.par <- par(mfrow=c(1, 2))
with(monk_data_df, interaction.plot(condition, prosoc_left, pulled_left, sum))
with(monk_data_df, interaction.plot(prosoc_left, condition, pulled_left, sum))
```



```r
par(old.par)
```

From this interaction plot when compared to condition and prosoc_left - it suggests that there is a linear relationship with both of these variables. The plot suggests that there is no interaction effect across these variables, which would be seen if there was a crossing of the lines, or opposite gradients. The probability of the actors choosing the left lever goes down across both prosocial options (whether its on the left or right) when the condition changes from 0 to 1. This would be consistent with the idea that the monkeys do not favour the prosocial option. In the case where they do, we would expect for the prosocial option of 1, and the condition = 1 would actually show a positive gradient - that is they are more likely to select the left lever given a monkey is on the other side of the table; however, this is not observed in the data. Also, noteworthy is the significant gap between the y values across both conditions - this indicates that which side the prosocial option is on will be important to the model. It seems that experimental condition has only a slight effect, and may not be significant to the model.

Let's build a model to investigate condition and prosocial option, and to double check there is no interaction effects.

```r
binom1 <- glm(cbind(pulled_left,1-pulled_left) ~ (condition+prosoc_left)^2, family=binomial, monk_data_
summary(binom1)
```

```
##
## Call:
## glm(formula = cbind(pulled_left, 1 - pulled_left) ~ (condition +
##     prosoc_left)^2, family = binomial, data = monk_data_df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4663  -1.2595   0.9137   1.0974   1.2181
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)              0.1911     0.1790   1.067   0.2858
## condition1              -0.2864     0.2527  -1.133   0.2571
## prosoc_left1             0.4666     0.2595   1.798   0.0722 .
## condition1:prosoc_left1  0.1821     0.3652   0.499   0.6181
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 685.94  on 503  degrees of freedom
## Residual deviance: 675.05  on 500  degrees of freedom
## AIC: 683.05
##
## Number of Fisher Scoring iterations: 4
```

```r
pchisq(675.05-0, 500, lower.tail = FALSE)
```

```
## [1] 2.6298e-07
```

```
(phihat_binom1 <- sum(residuals(binom1,type="pearson")^2)/(df.residual(binom1)))
```

```
## [1] 1.008
```

The first attempt at modelling considers condition, prosocial, and an interaction term between the two - as predictors for whether the monkey pulls the left lever. Objectively, this is a poor model. None of the predictive factors are significant - and the residual deviance is extremely high, almost the same as the null model deviance. The model is significantly different compared to the saturated model. We need to find a better model. Let's consider using actor as a predictor variable; as when looking at the raw data it is clear that monkeys behave in different ways (particularly actor 2,6,7).

```
binom10 <- glm(cbind(pulled_left, 1-pulled_left) ~ actor + condition + prosoc_left, family=binomial, mon
summary(binom10)
```

```
##
## Call:
## glm(formula = cbind(pulled_left, 1 - pulled_left) ~ actor + condition +
##     prosoc_left, family = binomial, data = monk_data_df)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.33453  -0.93778   0.00015   1.03306   1.69789
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -5.937e-01  2.878e-01  -2.063 0.039086 *
## actor2        1.896e+01  7.535e+02   0.025 0.979929
## actor3       -3.061e-01  3.507e-01  -0.873 0.382756
## actor4       -3.061e-01  3.507e-01  -0.873 0.382756
## actor5        2.840e-15  3.447e-01   0.000 1.000000
## actor6        9.432e-01  3.496e-01   2.698 0.006981 **
## actor7        2.492e+00  4.515e-01   5.520  3.4e-08 ***
## condition1   -2.717e-01  2.132e-01  -1.274 0.202642
## prosoc_left1  7.590e-01  2.147e-01   3.536 0.000407 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 685.94  on 503  degrees of freedom
## Residual deviance: 511.61  on 495  degrees of freedom
## AIC: 529.61
##
## Number of Fisher Scoring iterations: 17
```

```
(phihat_binom10 <- sum(residuals(binom10,type="pearson")^2)/(df.residual(binom10)))
```

```
## [1] 0.8884116
```

```
pchisq(deviance(binom10), df.residual(binom10), lower.tail = FALSE)
```

```
## [1] 0.2934989
```

The model is a significant improvement to the previous model; however, still is quite poor when considering the high residual deviance of 511.61. There is a lot of unexplained variance that the model does not capture. The model is deemed adequate from the pchisq test - it is not significantly different when compared to the saturated model. The estimate of phi indicates there is no overdispersion.

As the model infers that there is significant improvement to be had, we must consider whether a binomial model on each trial is the correct model to use. Upon reflection of the dataset and how it is structured, the model being fitted is firstly not going to be a robust model for a technical reason, but also does not address the research question of whether monkeys choose prosocial options or not. The model being built is predicting each bernoulli trial, that is a prediction is being made for each 504 trials - and most likely due to limitations on what we have available as predictor variables will never fit a precise model. Furthermore, for each of the 18 trials for each condition, prosocial option, and monkey, the same prediction is being made for each of these trials. There is a lot of error in the model from this, you have to sum the deviances, and also there is a lot of redundancy!

If we assume each trial as an independant and identically distributed bernoulli trial, then the sum of these bernoulli's will give rise to a binomial distribution with n = 18 for each condition, prosocial option, and actor. This assumption will need to be investigated using diagnostic plots after the model is fitted, as some sort of learning effect could be at play in between the trials - but this seems unlikely as they are not rewarded or punished for choosing one option over the other, so there is most likely nothing to learn. However, one must be open minded to other potential issues that diagnostics might show.

```
binomial_monk_data <- (counts_3_factor <- ftable(pulled_left ~ condition + prosoc_left + actor, monk_da
binomial_monk_data <- as.data.frame(binomial_monk_data)
binomial_monk_data <- binomial_monk_data[29:56, ]

binomial_monk_data$actor <- factor(binomial_monk_data$actor);
binomial_monk_data$condition <- factor(binomial_monk_data$condition);
binomial_monk_data$prosoc_left <- factor(binomial_monk_data$prosoc_left);

binomial_model1 <- glm(cbind(Freq, 18-Freq) ~ actor + condition + prosoc_left, family=binomial, binomia
summary(binomial_model1)
```

```
##
## Call:
## glm(formula = cbind(Freq, 18 - Freq) ~ actor + condition + prosoc_left,
##     family = binomial, data = binomial_monk_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4709  -0.4052   0.0001   0.3297   1.7804
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -5.937e-01  2.878e-01  -2.063 0.039086 *
## actor2        2.202e+01  3.481e+03   0.006 0.994954
## actor3       -3.061e-01  3.507e-01  -0.873 0.382756
## actor4       -3.061e-01  3.507e-01  -0.873 0.382756
## actor5       -6.454e-17  3.447e-01   0.000 1.000000
## actor6        9.432e-01  3.496e-01   2.698 0.006981 **
## actor7        2.492e+00  4.515e-01   5.520  3.4e-08 ***
## condition1   -2.717e-01  2.132e-01  -1.274 0.202642
## prosoc_left1  7.590e-01  2.147e-01   3.536 0.000407 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 193.657  on 27  degrees of freedom
## Residual deviance:  19.325  on 19  degrees of freedom
## AIC: 109.43
##
## Number of Fisher Scoring iterations: 18
```

```
pchisq(deviance(binomial_model1), df.residual(binomial_model1), lower.tail = FALSE)
```

```
## [1] 0.436179
```

```
phihat_binomial_model1 <- sum(residuals(binomial_model1, type="pearson")^2)/(df.residual(binomial_model

binomial_model2 <- glm(cbind(Freq, 18-Freq) ~ actor + prosoc_left, family=binomial, binomial_monk_data)
summary(binomial_model2)
```

```
##
## Call:
## glm(formula = cbind(Freq, 18 - Freq) ~ actor + prosoc_left, family = binomial,
##     data = binomial_monk_data)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.59540  -0.54330   0.00014   0.40946   1.99021
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -7.265e-01  2.686e-01  -2.705 0.006834 **
## actor2        2.201e+01  3.496e+03   0.006 0.994975
## actor3       -3.048e-01  3.499e-01  -0.871 0.383755
## actor4       -3.048e-01  3.499e-01  -0.871 0.383755
## actor5       -7.729e-16  3.439e-01   0.000 1.000000
## actor6        9.391e-01  3.488e-01   2.692 0.007099 **
## actor7        2.483e+00  4.506e-01   5.510 3.59e-08 ***
## prosoc_left1  7.559e-01  2.142e-01   3.529 0.000417 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 193.657  on 27  degrees of freedom
## Residual deviance:  20.953  on 20  degrees of freedom
## AIC: 109.06
##
## Number of Fisher Scoring iterations: 18
```

```r
### final model - adequacy test
pchisq(deviance(binomial_model2), df.residual(binomial_model2), lower.tail = FALSE)
```

```
## [1] 0.3998885
```

```r
### not significantly different
pchisq(deviance(binomial_model2)-deviance(binomial_model1), 1, lower.tail = FALSE)
```

```
## [1] 0.2019302
```

```r
binomial_model3 <- glm(cbind(Freq, 18-Freq) ~ prosoc_left, family=binomial, binomial_monk_data)
summary(binomial_model3)
```

```
##
## Call:
## glm(formula = cbind(Freq, 18 - Freq) ~ prosoc_left, family = binomial,
##     data = binomial_monk_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.6156  -1.5830  -0.7973   2.3276   4.9098
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.04763    0.12602   0.378  0.70548
## prosoc_left1  0.55749    0.18235   3.057  0.00223 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 193.66  on 27  degrees of freedom
## Residual deviance: 184.21  on 26  degrees of freedom
## AIC: 260.32
##
## Number of Fisher Scoring iterations: 4
```

```r
### highly significantly different ###
pchisq(deviance(binomial_model3)-deviance(binomial_model1), df.residual(binomial_model3) - df.residual(
```

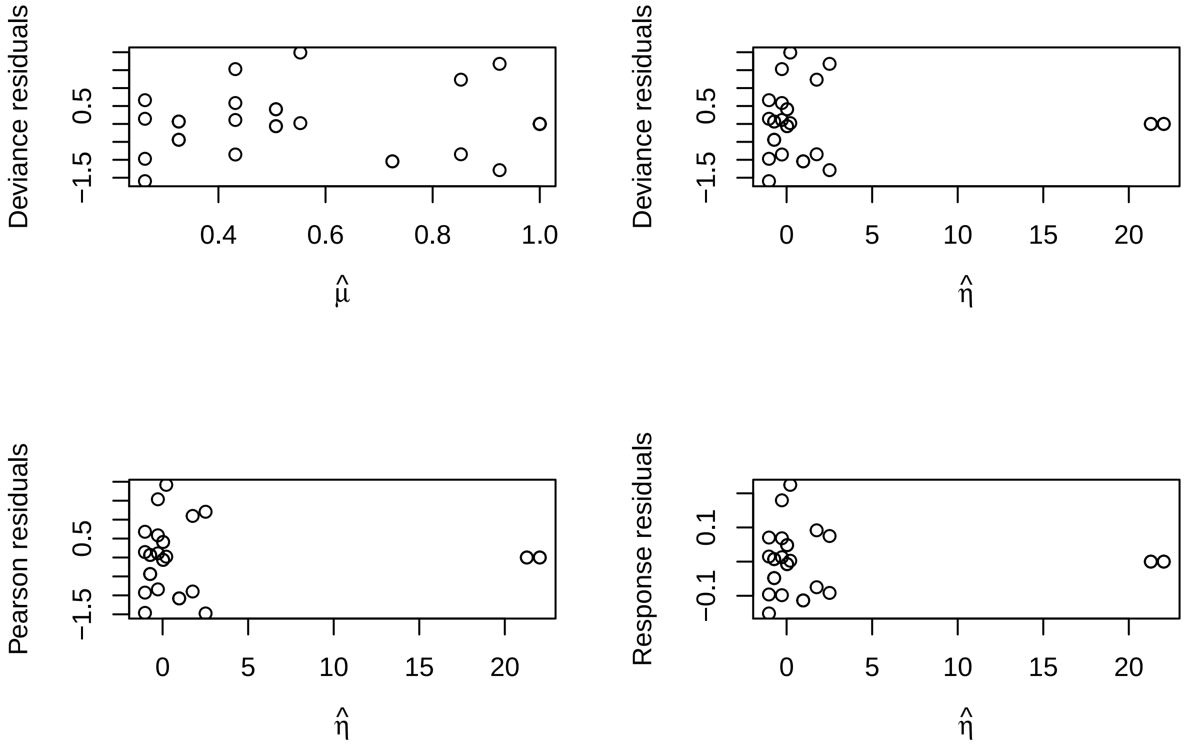
```
## [1] 3.00057e-32
```

After summing the bernoulli trials of pulled_left, making a new data_frame, and fitting a new model it can be seen that is a much more powerful model to describe the data. Essentially, we now have 18 trials for each condition, and the model is determining the probability of this binomial random variable equalling the total amount of times the monkey pulled left for a condition and prosocial option (e.g. out of 18 trials they pulled left 5 times, so 18 choose 5). With a residual deviance of 19.325 on 19 degrees of freedom, an AIC of 109.43, and a chi-squared model adequacy p-value of 0.436179, and no overdispersion - this is an adequate or very good model.

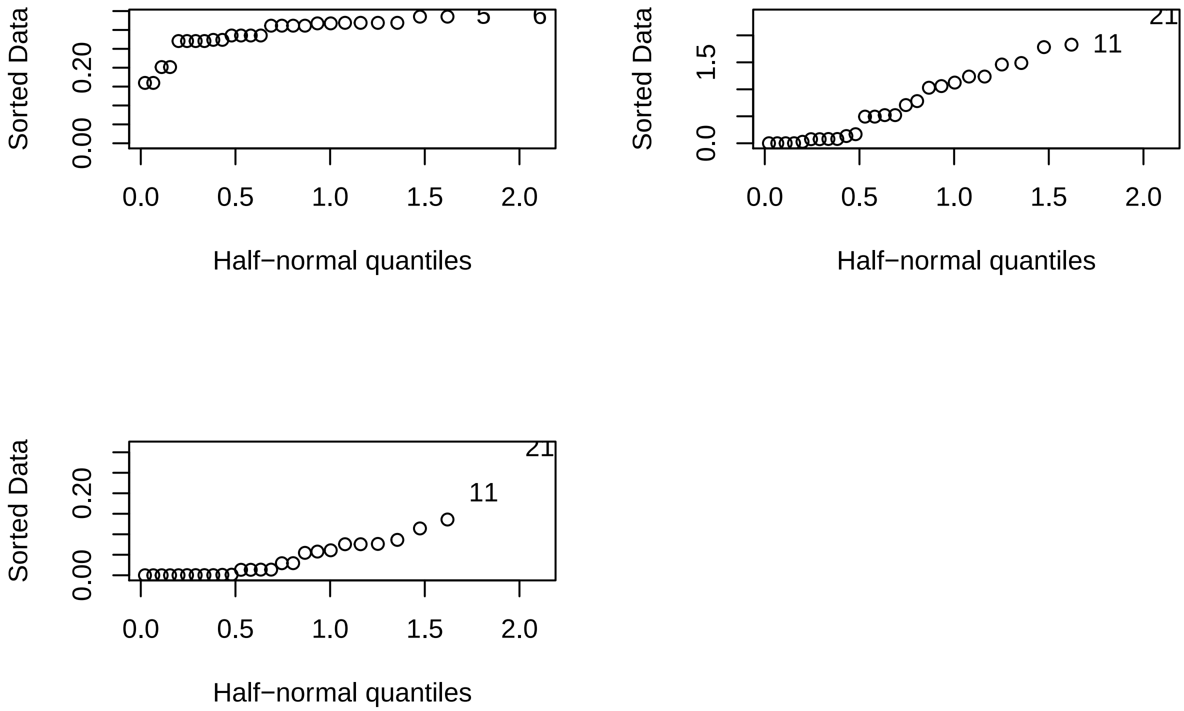
The co-efficient provided for the condition has a p-value of of 0.2026, indicating that it is not significant to the model. Let's fit a few model on just actor and prosocial variables and use a chi-square test for model determination. The statistic is calculated as 0.20193 indicating that the bigger model (with condition) is not significantly different to the model without it. We conclude that condition is irrelevant to the model, and proceed with the model including only actor + prosocial as covariates (binomial_model2). Lastly, as actor has only significant co-efficients for 3/7 of the terms, let us determine whether actor is significant to the model. The p-value for the chi-squared test indicates that actor is highly significant to the model.

The model of pulled_left (frequency) ~ actor + prosoc_left - has an AIC of 109.06, and a residual deviance of 20.953 on 20 degrees of freedom. A model adequacy test shows a p-value of 0.399 - demonstrating that the model explains an adequate amount of the deviance as compared to the saturated ('perfect') model.

```r
par(mfrow=c(2,2))

plot(residuals(binomial_model2) ~ predict(binomial_model2,type="response"),
xlab=expression(hat(mu)), ylab="Deviance residuals")


plot(residuals(binomial_model2) ~ predict(binomial_model2,type="link"),
xlab=expression(hat(eta)), ylab="Deviance residuals")


plot(residuals(binomial_model2,type="pearson") ~ predict(binomial_model2, type="link"),
xlab=expression(hat(eta)), ylab="Pearson residuals")


plot(residuals(binomial_model2,type="response") ~ predict(binomial_model2, type="link"),
xlab=expression(hat(eta)), ylab="Response residuals")
```



```
par(mfrow=c(2,2))
halfnorm(influence(binomial_model2)$hat)
halfnorm(rstudent(binomial_model2))
halfnorm(cooks.distance(binomial_model2))
```

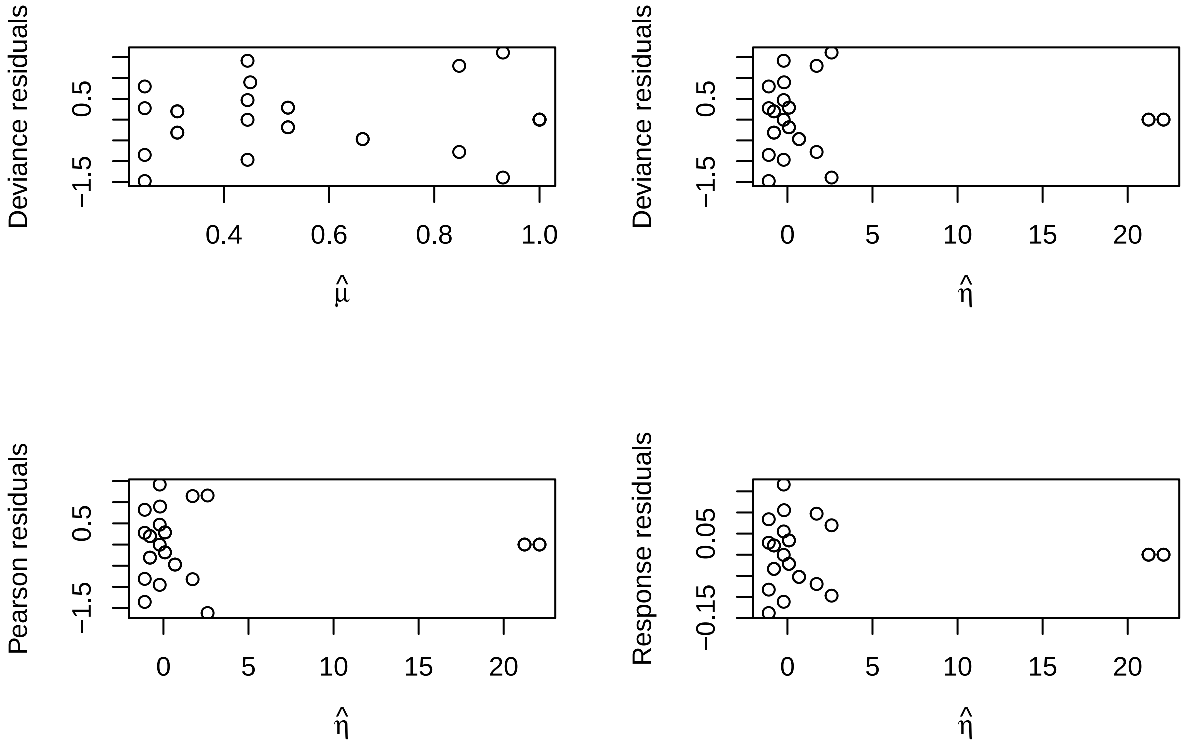


From the diagnostic plots, we see that there are extremely high eta's for a particular monkey. From the cook's distance plot, point 21 seems to be having a high effect on the model - let's remove it and build a new model and see if this is significant. Note, we can't perform a nested model adequacy test as they are now two different datasets (the null models have changed their df).

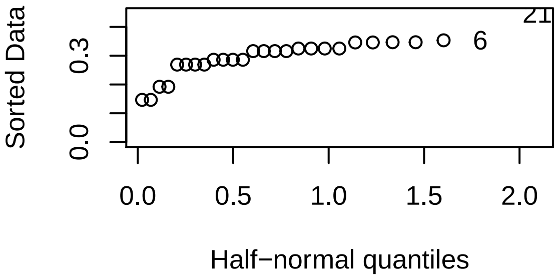
```
binomial_model4 <- glm(cbind(Freq, 18-Freq) ~ actor + prosoc_left, family=binomial, binomial_monk_data,
summary(binomial_model4)
```

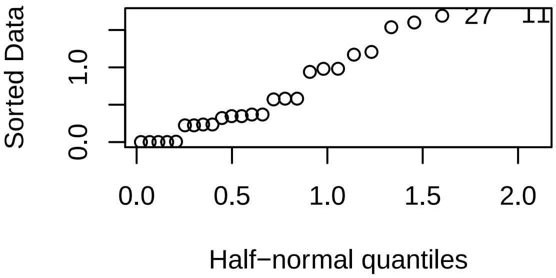
```
##
## Call:
## glm(formula = cbind(Freq, 18 - Freq) ~ actor + prosoc_left, family = binomial,
##     data = binomial_monk_data, subset = c(-21))
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.47632  -0.39012   0.00009   0.28707   1.61073
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -7.930e-01  2.721e-01  -2.914  0.00357 **
## actor2        2.202e+01  3.462e+03   0.006  0.99492
## actor3       -3.084e-01  3.520e-01  -0.876  0.38099
## actor4       -3.084e-01  3.520e-01  -0.876  0.38099
## actor5        3.613e-16  3.460e-01   0.000  1.00000
## actor6        5.930e-01  3.748e-01   1.582  0.11361
## actor7        2.507e+00  4.528e-01   5.536 3.10e-08 ***
## prosoc_left1  8.802e-01  2.215e-01   3.975 7.04e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 190.407  on 26  degrees of freedom
## Residual deviance:  15.312  on 19  degrees of freedom
## AIC: 100.4
##
## Number of Fisher Scoring iterations: 18
```

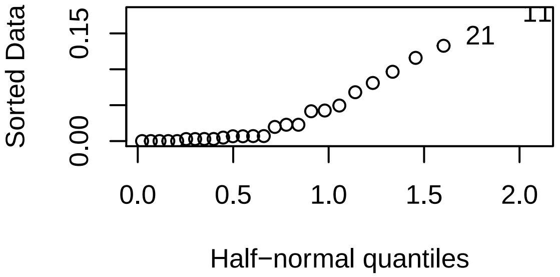
```r
### test remove actor 2 data as well ###
binomial_model5 <- glm(cbind(Freq, 18-Freq) ~ actor + prosoc_left, family=binomial, binomial_monk_data,

par(mfrow=c(2,2))
plot(residuals(binomial_model4) ~ predict(binomial_model4,type="response"),
xlab=expression(hat(mu)), ylab="Deviance residuals")

plot(residuals(binomial_model4) ~ predict(binomial_model4,type="link"),
xlab=expression(hat(eta)), ylab="Deviance residuals")

plot(residuals(binomial_model4,type="pearson") ~ predict(binomial_model4, type="link"),
xlab=expression(hat(eta)), ylab="Pearson residuals")

plot(residuals(binomial_model4,type="response") ~ predict(binomial_model4, type="link"),
xlab=expression(hat(eta)), ylab="Response residuals")
```



```r
par(mfrow=c(2,2))
halfnorm(influence(binomial_model4)$hat)
halfnorm(rstudent(binomial_model4))
halfnorm(cooks.distance(binomial_model4))
```







The last model has a new AIC of 100.4 compared to 109.06 for the previous model which included point 21. The residual deviance is 15.312 on 19 degrees of freedom. Actor7 has a highly significant coefficient, also with prosoc_left being highly significant. The diagnostic plots still show that actor2 gives high eta calculations. If we recreate the model without points 21,5,6,7,8 - then the model becomes 'worse' with an AIC of 109 as compared to 100 with actor 2 included. We conclude that the model with an AIC of 100 is the last and best model.

Summary:

The final model is a binomial model with logistic link function, which predicts the total amount of times a monkey pulls the left lever with covariates of actor and prosocial side as predictors. Upon diagnostic analysis it was observed that point 21 had a high cook's distance and effect on the model, and once removed provided the best overall model. It can be concluded that the individual monkey has an effect on whether they pull the left lever or not - particularly with actor7 having significantly different behaviour to the others. Actor7 shows increases across condition, when the prosocial option is on the left or right side. This shows that condition is important for actor7 and is thus being factored into that individuals co-efficient, but it can't necessarily be linked to the behaviour of monkeys choosing the prosocial option overall.

It was found that monkeys do not factor in whether a monkey is on the other side of the table or not as to whether they choose the prosocial option or not. A highly significant co-efficient was determined for the prosocial factor. If the prosocial option is on the left side, the monkeys are more likely to pick that side. If the prosocial option is on the right side, they are more likely to pull the right lever. This is independant of whether or not there is a monkey on the other side. No interaction effects between actor and prosoc_left option was observed.

It is possibly to theorize a reason for this behaviour; however, we cannot test it without further experimentation. A main theory to consider is that the monkeys are simply gluttons, and are picking the side which has two plates of food on it as they wish to eat both plates of food;regardless of whether a monkey is over the other end or not. This is the simplest, and most likely explanation. As this was a robust finding over all the conditions, it also suggests that the monkeys haven't learnt that they are not going to get two plates of food. You would first have to design an experiment to determine that they understand that they will not be getting the second plate of food. Our model is assuming that each trial is independant, but an experiment which somehow determines if a learning effect takes place would help determine whether the monkeys are prosocial or not. A way to test this is to get the monkeys before each trial to somehow indicate which side the plates of food will end up at after the pull the lever. The current data and analysis indicates that monkeys do not consider the prosocial option like how humans do.