Examining Variation in Neurons' Visual Encoding Mean Firing Rate

Luyang Zhang

3/20/2023

```
knitr::opts_chunk$set(
    echo = FALSE,
    message = FALSE,
    warning = FALSE
)
```

1.Abstract

The neural activity in the visual cortex is analyzed from 1,196 trials from an experiment conducted by Nicholas A. Steinmetz et al. We used three-way mixed effects ANOVA model to test our hypothesis that contrast levels and their interaction bring impact on neurons' mean firing rate. Then our three-way mixed effects model suggested that neurons' mean firing rate after visual stimuli presented is affected by the contrast levels of two visual stimuli and their interaction effect while considering each experiment session is a random factor. Then we conducted model diagnostics and sensitivity analysis including residual plots, normality test, and homoskedasticity test. In the second part of the project, we predicted whether mice receive reward or penalty. Our logistic regression model and linear discriminant analysis gives 77% accuracy with TPR around 96% and FPR around 77%. Lastly, we trained another model based on linear discriminant analysis (LDA) that gives a similar outcome.

2.Introduction

In 2019, Nicholas A. Steinmetz et al performed a neuron encoding research experiment on mice visual cortex and collecting neuron encoding data to investigate neurons' involvement in perceptual decisions. Most previous studies only studies some certain regions in the brain action selection, whereas Steinmetz et al conducted a comprehensive data collection across different regions in mice brains. Since other studies have observed that neurons in many brain regions fire stimuli for movements, rewards, and other tasks, it indicates that neurons across different regions in the brain could have some strong correlation. Thus, their research could uncover some new correlation among neurons. However, their experiment cannot distinguish neuron stimuli for actions and choice-related signals.

3.Data Description

This is a Go-NoGo experiment in which two screens are presented to the subjects. Mice are supposed to react by turning a wheel connected to their front paw when they receive visual stimuli from the screens. The signal of turning and neuron spikes are recorded. Researchers used probes inserted into mice's left hemisphere to receive information from mice's visual cortex and how long subjects fire spikes.

The experiment also excluded the trials in which the movement onset was less than 125 or more than 400 ms post-stimulus onset to avoid the event that wheel were coincidentally turned before mice could react to the visual stimuli. Similarly, trials with 50 to 400 ms post-stimulus onset when no recorded movement were excluded.

Mice receive a water reward if they successfully turn the wheel to indicate the screen showing a higher contrast or they do not turn when no contrasts presented. If the screens give equal contrast, mice earn a reward by arbitrarily turning the wheel to either the left or right direction.

A few sessions were experimented with. Each session has hundreds of trials. Each trial represents one observation. It has two dimensions: the column dimension means neurons, and the row dimension represents time. For example, trial number 1 in session 1 has 178 neurons across 39 timestamps. As a remark, sessions 1 through 5 did not happen in a time order, but trials within each session were in a timely order.

3.1 Descriptive Analysis

We created a new variable called `firing_rate` which stores the mean firing rate for each trial aggregated by the number of neurons and time. This variable is calculated by collapsing the `spks` variable in the original dataset. `spks` is a matrix in each trial. The row dimension records the number of spikes in the visual cortex and the column dimension represents the corresponding time bins defined in `time` variable in the original dataset. The mean firing rate for the i^{th} session and j^{th} trial is calculated as follow:

$$\text{Firing Rate}_{ij} = \frac{\sum_k \sum_l (\text{spikes})_{ijkl}}{\text{number of neuron}_{ij} \times \text{time}_{ij}}$$

where k is the index for neuron and l is the index for time. $k = 1, 2, \ldots, n_k$, $l = 1, 2, \ldots, n_l$.

We prepared the data frame by binding all five sessions' data together using `rbind()` function. Note that in the original dataset, many neuron never reacted to the visual stimuli, but we still included these neurons for the analysis since the study is for correlation among neurons not only for visual encoding.

Table 1: Head of dataset after preprocessing

contrast_left	contrast_right	firing_rate	feedback_type	session
1	0	6.194	1	1
0	0.5	4.087	1	1
1	0.5	5.871	1	1
0	0	4.087	1	1
0.5	1	5.534	-1	1
0	0	3.553	-1	1

Table 1 gives a snapshot of the head portion of the data frame after data manipulation. The variable `firing_rate` is a numeric variable, and the other four variables `contrast_left`, `contrast_right`, `feedback_type`, and `session` are factor variables. This is the data frame that we later used for ANOVA model, where firing rate is the response variable. There are two fixed effects factors: `contrast_left` and `contrast_right`, and one random effect `session`.

session	Number of Trials
1	214
2	251
3	228
4	249
5	254

Table 2 summarized the the number of trials for each session. Then our total number of trials is the five sessions' combined.

The firing rate distribution by different session is shown in the left density curves in **Figure 1**. The shape of distribution and mode for each session is different, so this suggests that `session` could bring a significant effect for influencing the mean firing rate.

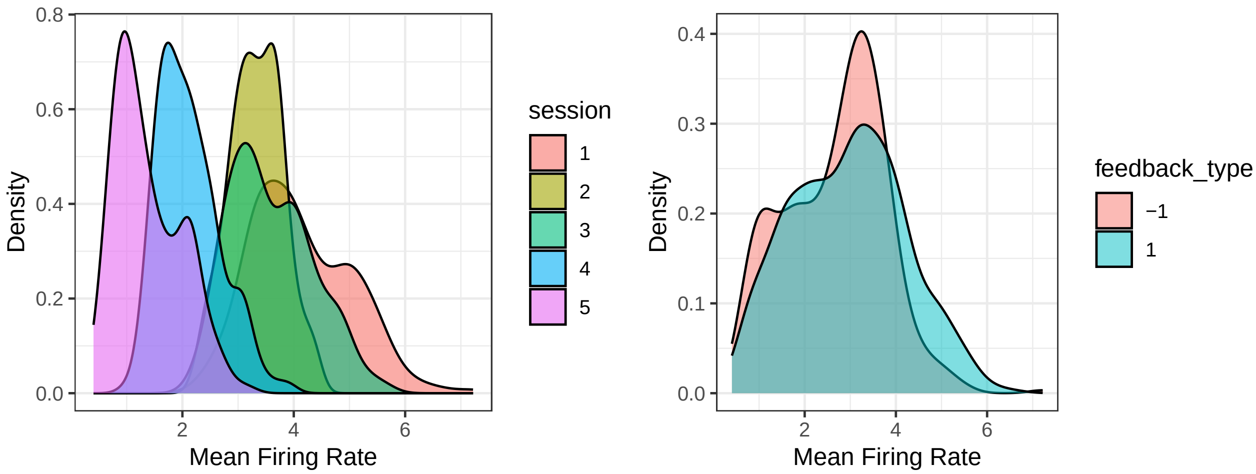

Figure 1: (left)Histogram of Mean Firing Rate by Session; (right)Histogram of Mean Firing Rate by feedback types

The right density curve in **Figure 1** shows the distribution of mean firing rate by feedback type. This density curves suggest that sometimes the stimuli took longer to fire when the feedback type is reward. In the figure, it is noted that the curve for reward tails off slower than type penalty.

Penalty has more mean firing rate concentrated together as it has a notable higher peak value, though the mode for both types are roughly the same.

Table 3: Top five

contrast_left	contrast_right	firing_rate
1	0	6.194
1	0.5	5.871
0.5	0	5.562
0.5	0.5	5.534
0.5	1	5.534

Table 4: Last five

contrast_left	contrast_right	firing_rate
0	0	4.087
0	0.5	4.087
0	0.25	4.368
0.25	0.5	4.396
0.25	1	4.41

Table 3 and **Table 4** summarized the top five and least five mean firing rate by different levels of contrasts. It is not likely that only the difference between the two levels has some decisive effects on the mean firing rate. In the table, we noticed the difference between two contrasts can be 0, 0.25, 0.5 and 1. Thus, we conducted more rigorous study later to investigate the factors affecting mean firing rate.

Table 5: Number of observation of the right contrast

contrast_right	n
0	522
0.25	195
0.5	192
1	287

Table 6: Number of observation of the left contrast

contrast_left	n
0	591
0.25	189
0.5	192
1	224

Table 5 and **Table 6** shows number of observations for each levels of contrast right and left respectively. Both screens presented more zero contrast level than other three levels.

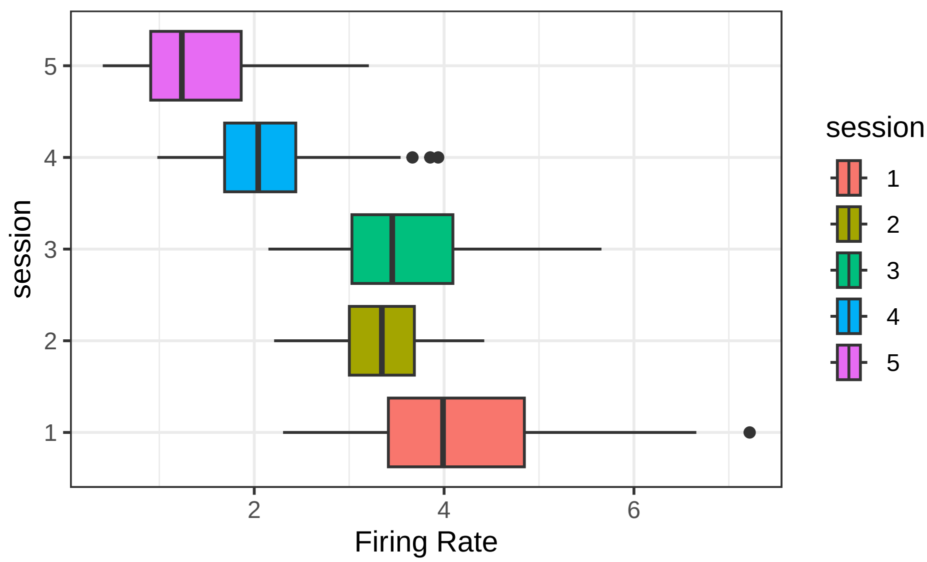

Figure 2: Boxplot of firing rate by session

Figure 2 is the boxplots of firing rate by each session. There are a three outliers in session 4 and one outlier in session 1. These outliers are all have higher than usual firing rates. And we can see that the mean and IQR for each session is quite different.

Figure 3 shows the mean plots and interaction plots for the variables. The mean plots for left and right contrast shows when increasing contrast levels, the mean firing rate tend to increase. Both mean plots for contrasts show when contrast level is 0.5 or 1, the mean firing rate are higher than when level is 0 or 0.25.

The mean plots for feedback type shows a reward feedback has a longer mean firing rate. This aligns with our previous density curves for firing rate and feedback type.

The interaction plot shows the mean of firing rate when both contrast levels vary. Though it is hard to see a pattern, we conducted a more rigorous study in the next section of inferential analysis.

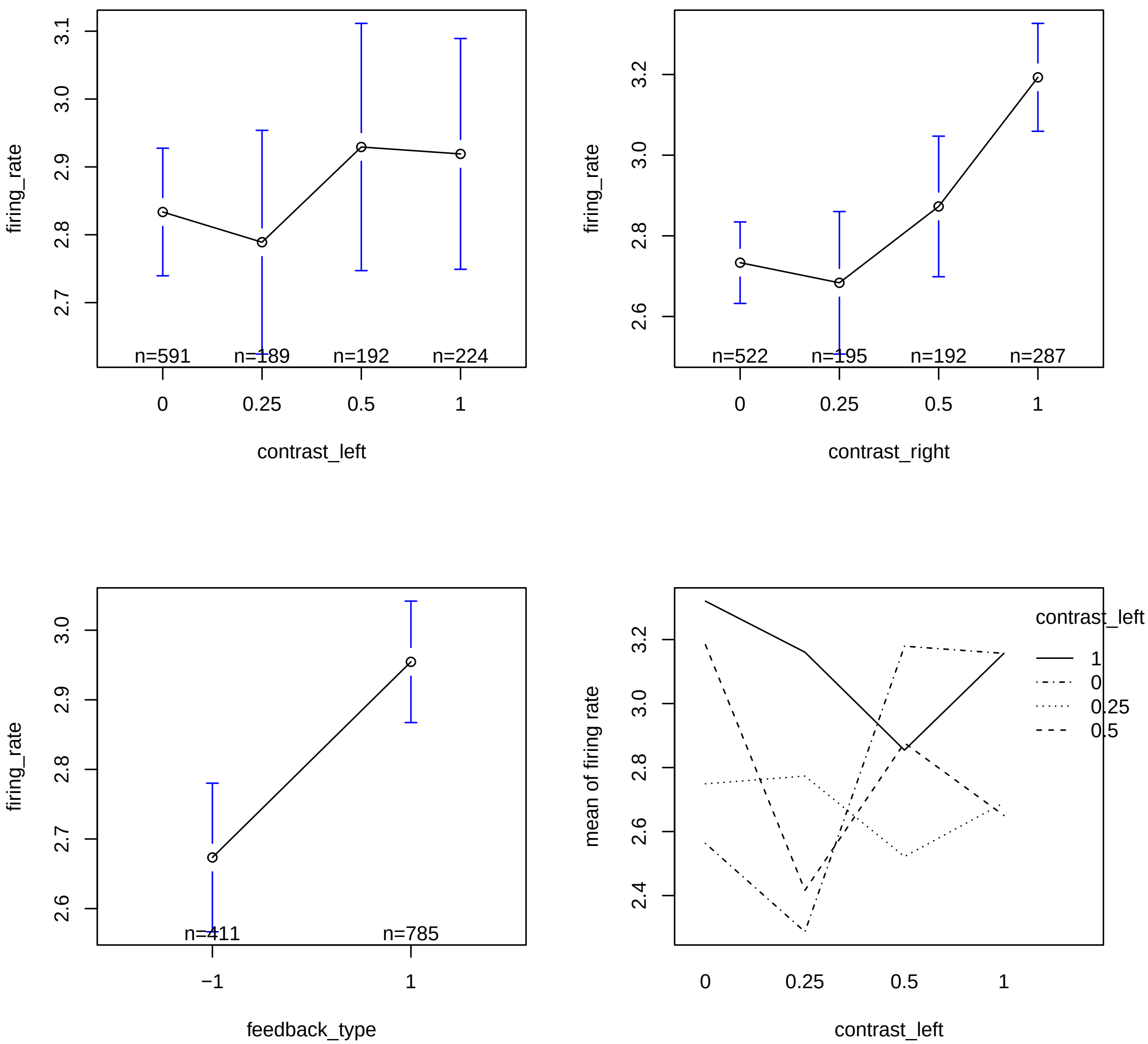

Figure 3: mean plots and interaction plot

4. Inferential Analysis

4.1 Mixed Effect Model

To explain the variation in the mean firing rate, we used ANOVA models. We fit the mean firing rate to left contrast, right contrast and session. The full model is:

$$Y_{ijkl} = \mu + \alpha_i + \beta_j + \gamma_k + (\alpha\beta)_{ij} + \epsilon_{ijkl}$$

where α_i and β_j are the fixed-effect factors of i^{th} left contrast and j^{th} right contrast, respectively. γ_k is the random effect of sessions, $(\alpha\beta)_{ij}$ is an interaction term of contrast left and contrast right. ϵ_{ijk} is the error terms. Specifically, we set up the a random intercept as session.

This model satisfies the following assumptions: (1) $\sum \alpha_i = 0$, (2)$\sum \beta_j = 0$, (3) γ_k are i.i.d $N(0, \sigma_\gamma^2)$, (4) $\sum(\alpha\beta)_{ij} = 0$, (5) $\epsilon_{ijkl} \overset{i.i.d}{\sim} N(0,\ \sigma^2)$

Table 7(a)

contrast levels	right 0	right 0.25	right 0.5	right 0.5
left 0	2.644	0.3295	0.3103	0.4754
left 0.25	0.1232	0.3296	0.4181	-0.2869
left 0.5	-0.2242	-0.3086	0.3269	-0.2799
left 1	-0.1156	4207	-0.3441	-0.1902

Table 7(b)

Groups	Name	Variance	Standard Deviation
Session	(Intercept)	1.2667	1.1255
Residual		0.3995	0.6321
Number of obs: 1196	Groups: session , 5		

Table 7(a) shows the fitted coefficients estimate by each pair of contrast levels, and **Table 7(b)** shows the variance and standard deviation for the random effect `session`. Then we are interested in testing if interaction term is significant in the model.

4.2 Model Selection

4.2.1 Testing interaction term

$$H_0 : (\alpha\beta)_{ij} = 0 \ vs.\ H_1 :\ (\alpha\beta)_{ij} \neq 0 \text{ for at least one pair of i, j}$$

Namely, we compared these two models:

full model:

$$Y_{ijk} = \mu + \alpha_i + \beta_j + \gamma_k + (\alpha\beta)_{ij} + \epsilon_{ijk}$$

additive-only model:

$$Y_{ijk} = \mu + \alpha_i + \beta_j + \eta_{k(ij)} + \epsilon_{ijk}$$

Table 9: ANOVA table for the additive model and full model

	npar	AIC	BIC	logLik	deviance	Chisq	Df	Pr(>Chisq)
model_add	9	2349	2395	-1166	2331	NA	NA	NA
model_full	18	2350	2441	-1157	2314	17.52	9	0.04112

Noted the reduced model is a nested model to the full model, we use `anova()` function to perform a likelihood ratio test. **Table 9** gives the summary output. The test statistic is the ratio of the maximum likelihood of the parameter of interest under null hypothesis and under the entire parameter space. It follows a Chi-square distribution which degrees of freedom is the number of difference in parameter between the two models.

In our case, since the p-value for our likelihood ratio test is $0.041 < 0.05$, so we reject the null hypothesis that interaction does not exist. Thus, we prefer accepting the full model.

4.2.2 Testing random effect Next, we tested whether the random effect is significant.

$$H_0 : \sigma^2 = 0 \ vs. \ H_1 : \ \sigma^2 \neq 0$$

We now compare these two models:

full model:

$$Y_{ijk} = \mu + \alpha_i + \beta_j + \eta_{k(ij)} + (\alpha\beta)_{ij} + \epsilon_{ijk}$$

fixed effects with interaction term model:

$$Y_{ijk} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \epsilon_{ijk}$$

Table 10: ANOVA Table for fixed-effect-only model and full model (continued below)

	npar	AIC	BIC	logLik	deviance	Chisq	Df
model_fixed_full	17	3788	3874	-1877	3754	NA	NA
model_full	18	2350	2441	-1157	2314	1440	1

	Pr(>Chisq)
model_fixed_full	NA
model_full	4.619e-315

Table 10 shows the output of the summary for the model comparison. Here, we use the same likelihood ratio test for testing random effect. The p-value shows significant, so we reject the null hypothesis. Then we believe the random effect of `session` exists in the model. Therefore, our final model is still the full model, which states:

$$Y_{ijk} = \mu + \alpha_i + \beta_j + \eta_{k(ij)} + (\alpha\beta)_{ij} + \epsilon_{ijk}$$

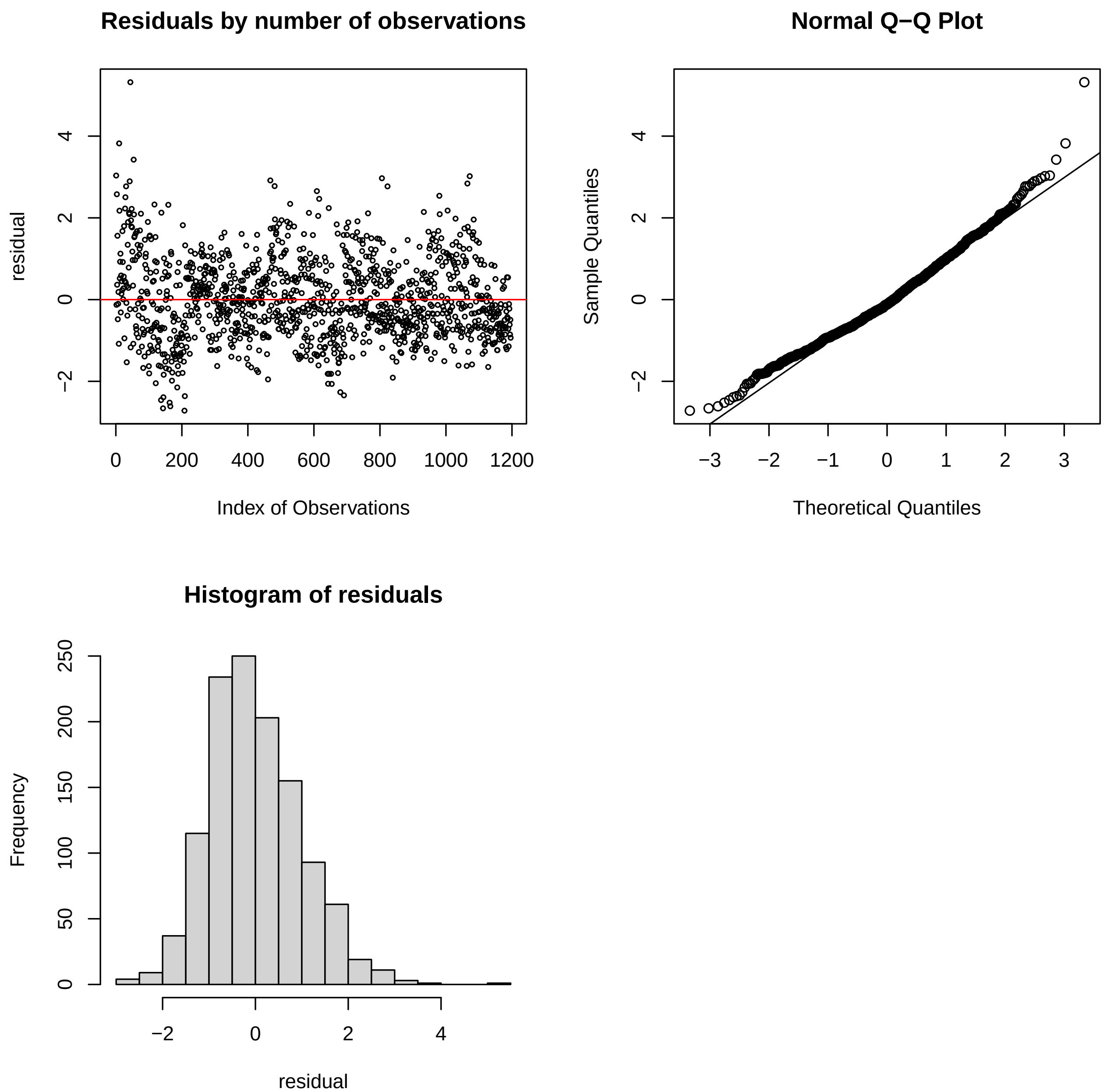

Figure 4: Summary plots for residuals

4.3 Sensitivity Analysis and Model Diagnostics

4.3.1 Residual Studies **Figure 4** shows the distribution of residual versus the number of trials. Noted that trials within each session are conducted in a sequence, it implies that as the experiment went for each trial, the mean firing rate tend to decrease. In other words, mean firing rate has a downward trend. The Normal Q-Q plot seems many residuals departed from normality's reference line especially those around lower quartile, and we did another rigorous test in the next step. And the histogram of residuals shows only one peak but the shape is slightly right-skewed.

4.3.2 Normality Test

Table 12: Shapiro-Wilk Test Result

Test statistic	P value
0.9852	1.105e-09 * * *

Table 12 gives the Shapiro-Wilk normality test for residual. It also shows significant p-value, so the residuals are not normally distributed and then violate the model assumption.

There could exist many other factors that the experiment was not able to record or are neglected. One possible explanation for a shortened mean firing rate could be that mice have learned the experiment mechanism, so it took fewer time for them to react so the stimuli.

4.3.3 Homoskedasticity Test

Table 13: Levene's Test for response

	Df	F value	Pr(>F)
group	15	1.117	0.3356
	1180	NA	NA

Table 14: Levene's Test for the residuals of the full model

	Df	F value	Pr(>F)
group	15	3.696	2.192e-06
	1180	NA	NA

We used Levene's Test by `leveneTest()` function in `car` package. We first test the response variable `firing_rate` with respect to the contrast levels with interactions. **Table 13** shows the test result, and p-value are not significant. So the data distribution itself satisfy the model assumption of homogeneity. Then we test the residual from the ANOVA model against contrast levels with interaction, as if the residuals are the response variable. **Table 14** shows that the p-value became significant. So the residuals are not homogeneity.

4.3.4 Outliers **Figure 5** is the box plot of residuals of the full model. There are several outliers that are unusually high, and there is a extremely small residual.

In conclusion, the ANOVA model gives some insight about how mean firing rate respond when we have different levels of contrast levels. It confirms that session should be considered as a random effect, so there is some variation due to sampling at each session of experiment.

In future studies, to reconcile the non-stationary issue with the data within each session, time series analysis could be done to incorporate autocorrelation between trials into the model.

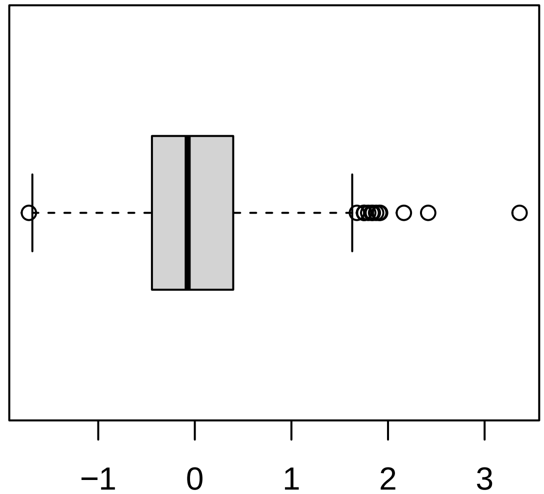

Figure 5: Boxplot for the residuals of the full model

5. Logistic Regression

In this section, we predicted the outcome of the feedback by training a classifier. Firstly, we split the data where the first 100 trials are in the test set and the rest of them are in the training set. We have 1096 observations combined from five sessions for training a classifier.

We choose the response variable as `feedback_type` and `contrast_right`, `contrast_left`, `firing_rate` and `session` as predictors.

The model is as follows:

$$\text{logit}(\pi_i) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4$$

where x_1 is `contrast_left`, x_2 is `contrast_right`, x_3 is `firing_rate`, and x_4 is `session`. π_i is the odds of mice getting a reward.

Table 15: Summary of the coefficients in the logistic regression model

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-2.455	0.4835	-5.078	3.817e-07
contrast_left0.25	-0.2042	0.198	-1.031	0.3023
contrast_left0.5	-0.4582	0.1944	-2.357	0.0184
contrast_left1	-0.2716	0.1896	-1.433	0.152
contrast_right0.25	-1.006	0.1982	-5.077	3.843e-07
contrast_right0.5	-0.9256	0.1974	-4.689	2.743e-06
contrast_right1	-0.9277	0.184	-5.043	4.593e-07
firing_rate	0.9695	0.1245	7.784	7.009e-15
session2	0.4861	0.2538	1.915	0.05545
session3	0.3306	0.2562	1.291	0.1968
session4	1.88	0.3194	5.885	3.984e-09
session5	2.592	0.3851	6.731	1.68e-11

Using `summary()` function, we can access the estimations for each coefficients in the logistic regression model. **Table 15** provides the summary output. Noted that all estimators for contrast levels are negative, so the baseline which both contrasts equal to zero gives the highest logit function. Thus the mice is easier to receive rewards compared to any other combination of contrast levels. Also, We found p-values for session 3 is not significant, so we fit another logistic regression model which `seesion` is excluded. We then conducted another likelihood ratio test using `anova()` function.

5.1 Model Selection

$$H_0 : \beta_4 = 0 \; vs. \; H_1 : \; \beta_4 \neq 0$$

Table 16: ANOVA Table for the Likelihood Ratio Test for the full logistic model and the model without a factor variable session

Resid. Df	Resid. Dev	Df	Deviance	Pr(>Chi)
1088	1374	NA	NA	NA
1084	1313	4	60.92	1.858e-12

Table 16 give the likelihood ratio test summary output. We tested if the coefficient for `session` is zero, so we set β_4 to 0 in the null hypothesis. The p-value shows significant, so we reject our null hypothesis and conclude that we accept the full model.

5.2 Model Testing

Then we test the model using the 100 observation of session 1 as the test dataset. `predict()` gave the predicted probability of receiving reward for each observation. Then we set our threshold as 0.5 and tabulated the confusion matrix (**Table 17**. The logistic regression model predict most reward type correct–only two observations are classified as penalty. However, more observation recorded as penalty are classified as reward. Thus, it has a True Positive Rate (TPR) of 97.3% and a False Positive Rate (FPR) of 76.92%. The overall accuracy is 78%.

Table 17: confusion matrix for logistic regression

	Prediction	
	-1	1
Observation		
-1	6	20
1	2	72

5.3 Logistic Model Diagnostics

5.3.1 Pearson residuals and deviance residuals These residuals are used for testing if the If the two kinds of residuals are not quite similar to each other, the model may suffer from potential lack-of-fit. In figure 6 two kinds of residuals are similar to each other, so the model does not show strong evidence of lack-of-fit.

5.3.2 Residual plots Secondly, we check the residual vs. fitted values plots, at **Figure 7**. The key here is to check patterns for residuals. The red curves are smoother for the fitted values. They are roughly around zero except at extreme values.

5.3.3 Leverage Points and Cook's Distance The fitted values of a linear model can be defined as $\hat{Y} = HY$. The diagonal of the hat matrix H are called leverages. These leverage points give some implication on model's goodness-of-fit. We looked for high leverage points which indicates the prediction can be unusual.

We used `hatvalues()` function in `stats` package to extract the diagonals of the hat matrix in the logistic regression model. In figure, there are two points that shows high leverage.

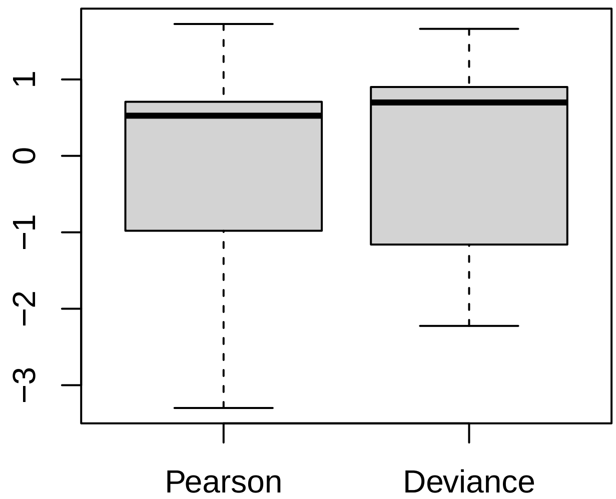

Figure 6: Boxplots of the two kinds of residuals for the logistic model

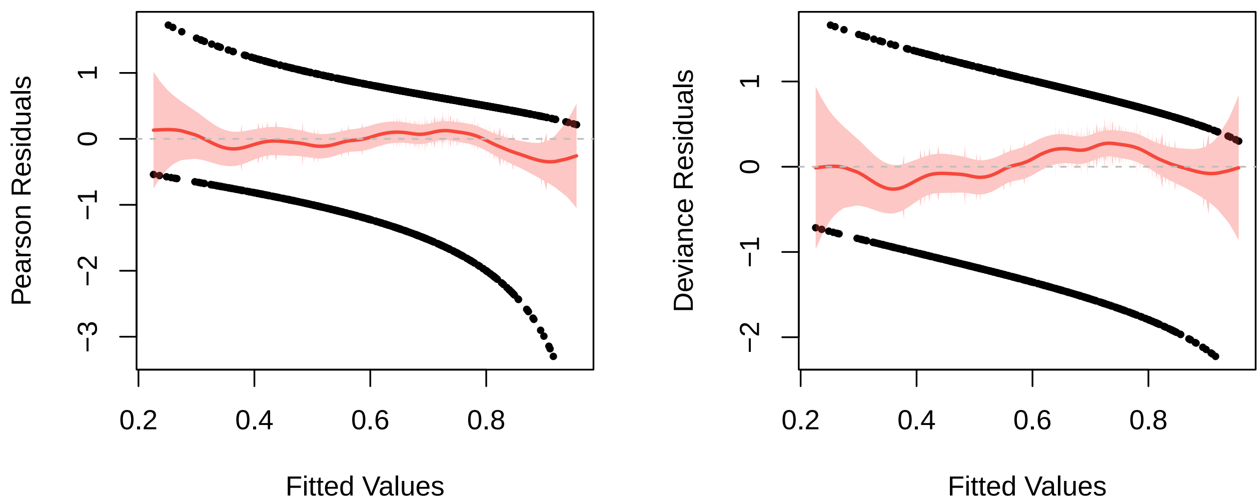

Figure 7: Residual vs. Fitted values plots

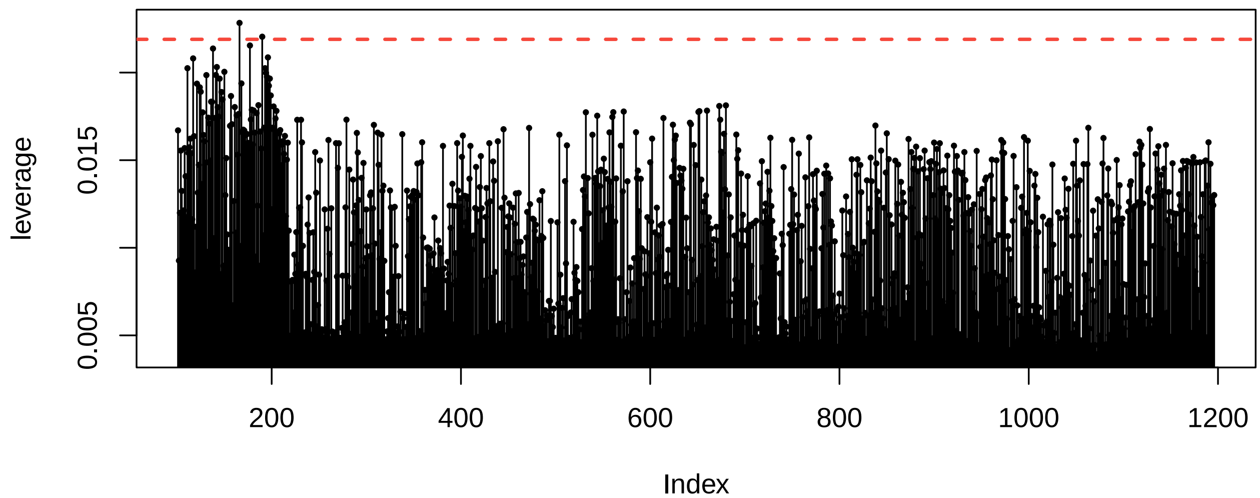

Figure 8: Cook's Distance of residuals

Finally, we plotted influential points based on Cook's distance. The red triangles represent high leverage points, and the blue index number indicates the top three among them. Based on **Figure 8**, and the previous leverage points study, only very few observations have high leverage and high Cook's distance.

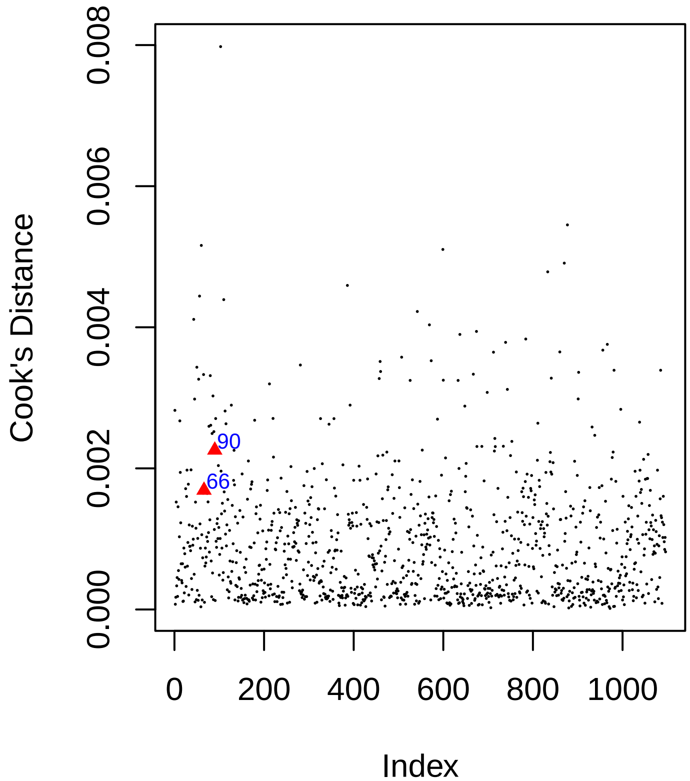

Figure 9: **Figure 10**

6. Linear Discriminant Analysis

We used Linear Discriminant Analysis (LDA) as an alternative classifier for predicting feedback type. In this approach, we estimate the distribution of each predictors given the response classes, and we used Bayes' Theorem to calculated the probability of either classes given the values in predictors.

$$P(Y = 1|X = \vec{x}) = \frac{\pi_1 f_1(\vec{x})}{\pi_1 f_1(\vec{x}) + \pi_2 f_2(\vec{x})}$$

where π_1 is the prior distribution for randomly choose a observation from feedback class 1 (reward), π_2 is the prior distribution for randomly choose a observation from feedback class 2 (penalty), $f_1(\vec{x})$ is the conditional distribution of the predictor matrix $\mathbf{X}$ given the response is class 1, $f_2(\vec{x})$ is the conditional distribution of the predictor matrix $\mathbf{X}$ given the response is class 2.

LDA assumed each predictor follows a normal distribution and two classes in the response have identical covariance matrices.

We use `lda()` function in `MASS` package to fit LDA models. This time we fit `feedback_type` to all four predictors, the same predictors as the previous logistic regression model.

Table 13: Confusion matrix of the LDA classifier

	Prediction	
	-1	1
Observation		
-1	6	20
1	2	72

Then we predicted the outcome using the same test set. It gives the same confusion matrix (see **Table 13**).

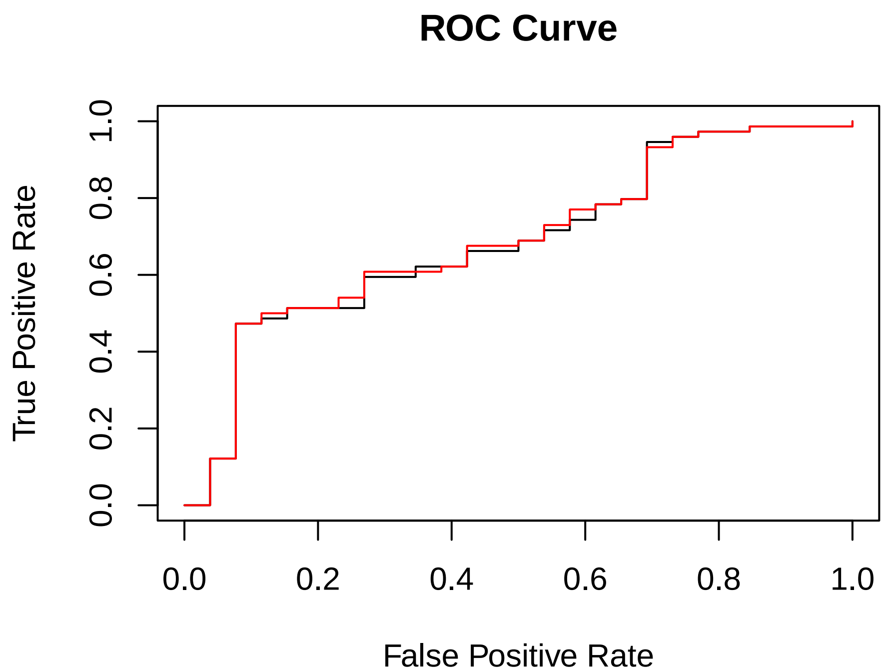

Figure 10: ROC Curves for logistic regression and LDA classifier

6.1 ROC curves We then compared the two classifier using ROC curve from `ROCR` package, which the red curve indicates LDA model and black curve indicates logistic regression. ROC Curve plots the true positive rate against the false positive rate for a classier. It gives a graphic view of how it performs under all threshold. The two curves are very close to each for almost all threshold, so they are not much different for predicting feedback type for the first 100 observations in session 1's data. This conclusion aligned with our findings that the two confusion matrix are also the same.

7. Conclusion

This study explained how two visual stimuli influence the mean firing rate of mice neuron responds . The experiment contains three variables: contrast levels of left and right screen and session. We treated the session as a random factor as the experiment is supposedly a sampling process. Using the ANOVA model, we can now tell that contrast levels and their interaction are all statistically significant in influencing the mean firing rate. Moreover, the likelihood ratio test for model comparison shows that the interaction term and random effect are statistically significant. The mean firing rate also suffers from the random effect of the session. In the experiment, researchers collected data from a population of neurons. And there exists some variation due to sampling by session to session.

In addition, we fit a logistic regression model to predict the feedback type of whether a mouse receives a reward or penalty based on the reaction to the contrasts and mean firing rate. This model has an accuracy of 78% with a True Positive Rate of 97.3% and a False Positive Rate of 76.92%.

Using Linear Discriminant Analysis, we trained another classifier for feedback type. This LDA model performs very similarly to the logistic regression model. It has an accuracy of 77% with a True Positive Rate of 95.95% and a False Positive Rate of 76.92%. Furthermore, in the future, more statistical learning tools can be used in predicting feedback types, such as quadratic discriminant analysis or support vector machine, and use cross-validation or other criteria to test and validate more models.

Reference

Acknowledgement: Christopher Li, Lulu Xue, STA 207 Lecture notes and discussion notes.

Faraway, J. (2006). *Extending the Linear Model with R--Generalized Linear, Mixed Effects and Nonparametric Regression Models*. Chapman & Hall/CRC.

Meier, L. *ANOVA and Mixed Models–A Short Introduction Using R*. CRC Press.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An Introduction to Statistical Learning with Applications in R*. Springer.

Steinmetz, N.A., Zatka-Haas, P., Carandini, M. et al. Distributed coding of choice, action and engagement across the mouse brain. Nature 576, 266–273 (2019). https://doi.org/10.1038/s41586-019-1787-x

Appendix

```
# Data Import
pytsession=list()
session = list()
for(i in 1:5){
  session[[i]]=readRDS(paste('/Users/luyang/rstudio/STA 207/project/session',i,'.rds',sep=''))

}
library(dplyr)
library(ggplot2)
library(ROCR)
library(pander)
library(tidyverse)
library(gridExtra)
library(gplots)
library(DescTools)
library(car)
library(astsa)
library(MASS)


ID=1
t=0.4 # from Background

n.trials=length(session[[ID]]$spks)
n.neurons=dim(session[[ID]]$spks[[1]])[1]

firingrate1=numeric(n.trials)

for(i in 1:n.trials){
  firingrate1[i]=sum(session[[ID]]$spks[[i]])/n.neurons/t
}
```

```r
ID=2
t=0.4 # from Background

n.trials=length(session[[ID]]$spks)
n.neurons=dim(session[[ID]]$spks[[1]])[1]

firingrate2=numeric(n.trials)

for(i in 1:n.trials){
  firingrate2[i]=sum(session[[ID]]$spks[[i]])/n.neurons/t
}

ID=3
t=0.4 # from Background

n.trials=length(session[[ID]]$spks)
n.neurons=dim(session[[ID]]$spks[[1]])[1]

firingrate3=numeric(n.trials)

for(i in 1:n.trials){
  firingrate3[i]=sum(session[[ID]]$spks[[i]])/n.neurons/t
}

ID=4
t=0.4 # from Background

n.trials=length(session[[ID]]$spks)
n.neurons=dim(session[[ID]]$spks[[1]])[1]

firingrate4=numeric(n.trials)

for(i in 1:n.trials){
  firingrate4[i]=sum(session[[ID]]$spks[[i]])/n.neurons/t
}

ID=5
t=0.4 # from Background

n.trials=length(session[[ID]]$spks)
n.neurons=dim(session[[ID]]$spks[[1]])[1]

firingrate5=numeric(n.trials)

for(i in 1:n.trials){
  firingrate5[i]=sum(session[[ID]]$spks[[i]])/n.neurons/t
}

ssn1 <- data.frame(contrast_left = session[[1]]$contrast_left,
                   contrast_right = session[[1]]$contrast_right,
                   firing_rate = firingrate1,
                   feedback_type = session[[1]]$feedback_type,
                   session = 1)
```

```r
ssn2 <- data.frame(contrast_left = session[[2]]$contrast_left,
                   contrast_right = session[[2]]$contrast_right,
                   firing_rate = firingrate2,
                   feedback_type = session[[2]]$feedback_type,
                   session = 2)

ssn3 <- data.frame(contrast_left = session[[3]]$contrast_left,
                   contrast_right = session[[3]]$contrast_right,
                   firing_rate = firingrate3,
                   feedback_type = session[[3]]$feedback_type,
                   session = 3)

ssn4 <- data.frame(contrast_left = session[[4]]$contrast_left,
                   contrast_right = session[[4]]$contrast_right,
                   firing_rate = firingrate4,
                   feedback_type = session[[4]]$feedback_type,
                   session = 4)

ssn5 <- data.frame(contrast_left = session[[5]]$contrast_left,
                   contrast_right = session[[5]]$contrast_right,
                   firing_rate = firingrate5,
                   feedback_type = session[[5]]$feedback_type,
                   session = 5)

df <- rbind(ssn1, ssn2, ssn3, ssn4, ssn5)
df$session <- as.factor(df$session)
df$feedback_type <- as.factor(df$feedback_type)
df$contrast_left <- as.factor(df$contrast_left)
df$contrast_right <- as.factor(df$contrast_right)
```

```r
head(df) %>% pander(caption = '**Table 1**')

library(sqldf)
library(RSQLite)
sqldf('SELECT SESSION, COUNT(SESSION) AS "Number of Trials" FROM df GROUP BY SESSION') %>% pander(title

library(ggplot2)
grid.arrange(ggplot(df) + geom_density(aes(x = firing_rate, fill = session), alpha = 0.6) + xlab('firin
             ggplot(df) + geom_density(aes(x = firing_rate, fill = feedback_type), alpha = 0.5)+ xlab('
             ncol = 2)

query1 <- sqldf('SELECT contrast_left, contrast_right, firing_rate
      FROM df GROUP BY contrast_left, contrast_right
      ORDER BY firing_rate DESC
      LIMIT 5')

query2 <- sqldf('SELECT contrast_left, contrast_right, firing_rate
      FROM df GROUP BY contrast_left, contrast_right
      ORDER BY firing_rate ASC
      LIMIT 5')

#pander(multicolumn = 2, multirow = 2, list(query1, query2), width = 'fit')
```

```r
pander(query1, caption = "**Table 3(a)**")
pander(query2, caption = "**Table 3(b)**")


ggplot(df) + geom_boxplot(aes(x = firing_rate, y = session, fill= session)) + theme_bw() + xlab('Firing

df %>% group_by(contrast_right) %>% count() %>% pander(caption = "**Table 4(a)**")
df %>% group_by(contrast_left) %>% count() %>% pander(caption = "**Table 4(b)**")


library(lme4)
library(lmerTest)

model_full <- lmerTest::lmer(firing_rate ~ contrast_right*contrast_left + (1|session), data = df)

model_fixed_full <- lm(firing_rate ~ contrast_right*contrast_left, data = df)

model_add <- lmerTest::lmer(firing_rate ~ contrast_right + contrast_left + (1|session), data = df)

model_fixed <- lm(firing_rate ~ contrast_right+contrast_left, data = df)

coefs <- summary(model_full)$coef[,1]
coef_mat <- matrix(coefs, ncol = 4, byrow = T)
rownames(coef_mat) <- c('left 0', 'left 0.25', 'left 0.5', 'left 1')
colnames(coef_mat) <- c('right 0', 'right 0.25', 'right 0.5', 'right 1')


anova(model_full, model_add) %>% pander(caption = "**Table 6**")
anova(model_full, model_fixed_full) %>% pander(caption = "**Table 7**")

par(mfrow = c(2,2))
#plot(model_add, which = 1)
plot(summary(model_full)$residual, ylab = 'residual', type = 'p', cex = 0.5);abline(h=0, col = 'red')
qqnorm(summary(model_full)$residual);qqline(summary(model_add)$residual)
hist(summary(model_full)$residual, xlab = 'residual', main = 'Histogram of residuals')

shapiro.test(resid(model_full)) %>% pander(caption = "**Table 8**")
leveneTest(firing_rate~contrast_right*contrast_left, data = df) %>% pander(caption = "**Table 9(a)**")
leveneTest(resid(model_full)~contrast_right*contrast_left, data = df)%>% pander(caption = "**Table 9(b)
boxplot(resid(model_full), horizontal = T)

df_train <- df[101:nrow(df), ]
df_test <- df[1:100, ]

library(MASS)
logit_model <- glm(feedback_type ~ ., family = binomial(), data=df_train)

summary(logit_model)$coef %>% pander(digits = 4, caption = "**Table 10**")

h0_logit_model <- glm(feedback_type ~ .-session, family = binomial(), data = df_train)
anova(h0_logit_model, logit_model, test = 'Chi') %>%pander(caption = "**Table 11**")

logit_pred <- ifelse(predict(logit_model, df_test, type = "response") <.5, "-1", "1")
```

```r
logit_cm=table(Feedback=df_test$feedback_type,Prediction=logit_pred)
logit_cm
sum(diag(logit_cm))/sum(logit_cm)

logit_cm[2,2]/(logit_cm[2,2]+logit_cm[2,1]) # TPR
logit_cm[1,2]/(logit_cm[1,2]+logit_cm[1,1]) # FPR

res.P = residuals(logit_model, type = "pearson")
res.D = residuals(logit_model, type = "deviance")
boxplot(cbind(res.P, res.D), names = c("Pearson", "Deviance"))

par(mfrow=c(1,2))
plot(logit_model$fitted.values, res.P, pch=16, cex=0.6, ylab='Pearson Residuals', xlab='Fitted Values')
lines(smooth.spline(logit_model$fitted.values, res.P, spar=0.9), col=2)
abline(h=0, lty=2, col='grey')
plot(logit_model$fitted.values, res.D, pch=16, cex=0.6, ylab='Deviance Residuals', xlab='Fitted Values')
lines(smooth.spline(logit_model$fitted.values, res.D, spar=0.9), col=2)
abline(h=0, lty=2, col='grey')

leverage = hatvalues(logit_model)
plot(names(leverage), leverage, xlab="Index", type="h")
points(names(leverage), leverage, pch=16, cex=0.5)
p = length(coef(logit_model))
n = nrow(df_train)
infPts = which(leverage>2*p/n)
abline(h=2*p/n,col=2,lwd=2,lty=2)

cooks = cooks.distance(logit_model)
plot(cooks, ylab="Cook's Distance", pch=16, cex=0.2)
points(infPts, cooks[infPts], pch=17, cex=0.8, col='red') # influential points
susPts = as.numeric(names(sort(cooks[infPts], decreasing=TRUE))[1:3]) - 100
text(susPts, cooks[susPts], susPts, adj=c(-0.1,-0.1), cex=0.7, col='blue')

lda_fit <- lda(feedback_type~contrast_left + contrast_right + session + firing_rate, data = df_train)
lda_pred <- predict(lda_fit, dplyr::select(df_test, -feedback_type))


lda_cm=table(Feedback=df_test$feedback_type, Prediction=lda_pred$class)

sum(diag(lda_cm))/sum(lda_cm)
lda_cm

lda_cm[2,2]/(lda_cm[2,2]+lda_cm[2,1]) # TPR
lda_cm[1,2]/(lda_cm[1,2]+lda_cm[1,1]) # FPR

library(ROCR)

# Logistic
logit_probs <- predict(logit_model, df_test, type = "response")
prediction_logit <- prediction(predictions = logit_probs, labels = df_test$feedback_type)

perf_logit <- performance(prediction.obj = prediction_logit, 'tpr', 'fpr')
```

```r
# LDA
prediction_lda <- prediction(predictions = lda_pred$posterior[, 2],
labels = df_test$feedback_type)

perf_lda <- performance(prediction.obj = prediction_lda, 'tpr', 'fpr')
perf_lda

plot(perf_lda, main = "ROC Curve", xlab = "False Positive Rate", ylab = "True Positive Rate")
plot(perf_logit, col = 'red', add = T)
```