STA 207 Progress Report

Dawei Wang

2023-03-09

Abstract

This study analyzed a subset of data collected by Steinmetz et al. (2019). The goal of this study is to find out how the neural activity in the visual cortex is modulated by certain visual stimuli and how this information can be utilized to predict the outcome of the visual stimuli. Therefore, the spike trains of neurons in the visual cortex, which were recorded as the activity of the neurons in the mice's visual cortex during the trials, were collected to analyze the relationship between visual stimuli and mice neural activity.

Introduction

To understand how the brain processes sensory information under different conditions, scholars study the modulation of visual responses by behavioral state in the mouse visual cortex. Frenkel et al. (2006) finded that mice reared in complete darkness had impaired visual acuity and reduced responsiveness of visual cortex neurons to visual stimuli compared to those reared in normal lighting conditions. Wang et al. (2007) used electrophysiological recordings to measure the responses of individual neurons to visual stimuli and confirm their functional properties. Cristopher et al. (2010) designed an experiment in which they recorded neural activity in the visual cortex of mice while presenting visual stimuli under different behavioral conditions and find that that the visual responses of the mice were modulated by their behavioral state.

Background

Steinmetz et al. used Neuropixels probes to record from approximately 30,000 neurons in 42 brain regions of mice performing a visual discrimination task. They found that neurons encoding visual stimuli and upcoming choices occupied restricted regions in the neocortex, basal ganglia, and midbrain. In this study, we will focused on the spike trains of neurons in the visual cortex, which were recorded as the activities of the neurons. Five sessions (among 39 sessions) from two mice (among 10 mice) were studied.In each session, the mouse was presented with visual stimuli on two screens positioned on both sides of it. Several hundred trials were conducted, and the visual stimuli were randomly varied in contrast levels that ranged from no stimulus to levels of 0.25, 0.5, and 1. The mouse used a wheel controlled by its forepaws to make decisions based on the visual stimuli, and a reward or penalty was given based on the accuracy of the mouse's decisions. The spike trains of neurons in the visual cortex were recorded as the activity of the neurons in the mice's visual cortex. Our primary research questions are: 1. How do visual cortex neurons respond to stimuli presented on the left and right? 2. Can we predict trial outcomes by analyzing neural activities and stimuli? To answer the first question, we built a mixed effect model comprising two fixed-effect factors, left contrast and right contrast, and a random intercept for each session. For the second question, we developed regression models to identify the best-performing model. To evaluate the prediction performance, we used sensitivity and specificity measures based on the first 100 trials in Session 1.

Descriptive analysis

Selection of outcome variables

In each trial of a specific session, the spike trains of certain number of visual neurons were recorded while the mouse was exposed to a combination of visual stimuli. As we can see in the following table

session	number of neurons	number of trials
1	178	214
2	533	251
3	228	228
4	120	249
5	99	254

The number of neurons in different sessions are different.

On the other hand, large part of the neurons didn't fire in the recorded 0.4 second in specific trial, but never fired neurons during each session are rare.

For example, the following table shows the no firing rate in first 5 trial of session1. (number of neurons that didn't fire in certain trial/ total number of neurons in some session).

session 1	
Trial ID	No firing rate
1	44.9%
2	54.5%
3	48.3%
4	57.8%
5	41.6%

And the never fired rate (number of neurons never fired during some session/ number of neurons in that session) in each session is quite low:

Session ID	Never firing rate
1	0.56%
2	0.75%
3	3.95%
4	1.67%
5	0%

```
# for(Session_ID in seq(1,5))
# {
# #number of trials in each session
# n.trials=length(session[[Session_ID]]$spks)
# #number of neurons in each session
# n.neurons=dim(session[[Session_ID]]$spks[[1]])[1]
# # number of time
# number_of_time=dim(session[[Session_ID]]$spks[[1]])[2]
#
#
# # initiate the time reduced matrix---the number of spikes of each neuron in each trial was summed by
```

```
# sum_spks_neuron <- matrix(0,nrow =n.trials,ncol=n.neurons)
#
# for(i in seq(1,n.trials))
#   sum_spks_neuron[i,]=rowSums(session[[Session_ID]]$spks[[i]])
#
# print(table(colSums(sum_spks_neuron)))
# }
```

To analyze the relationship between visual stimuli and mice neural activity, we need to make the outcome variables of different session comparable. Since there is a considerable (about half) no firing rate for each trial and the overall never firing rate is quite low for each session, it is meaningful to compute the mean firing rate of each trial. To be more detailed, the considerable no firing rate means the firing rate is not contribute by small number of abnormal neurons and the low never firing rate ensure that in each trial most of neurons have contributed to the mean firing rate of some trials.

The firing rate is defined as:

$$firing\ rate = \frac{total\ spikes\ of\ all\ neurons\ in\ certain\ trial}{number\ of\ neurons \times number\ of\ trials}$$

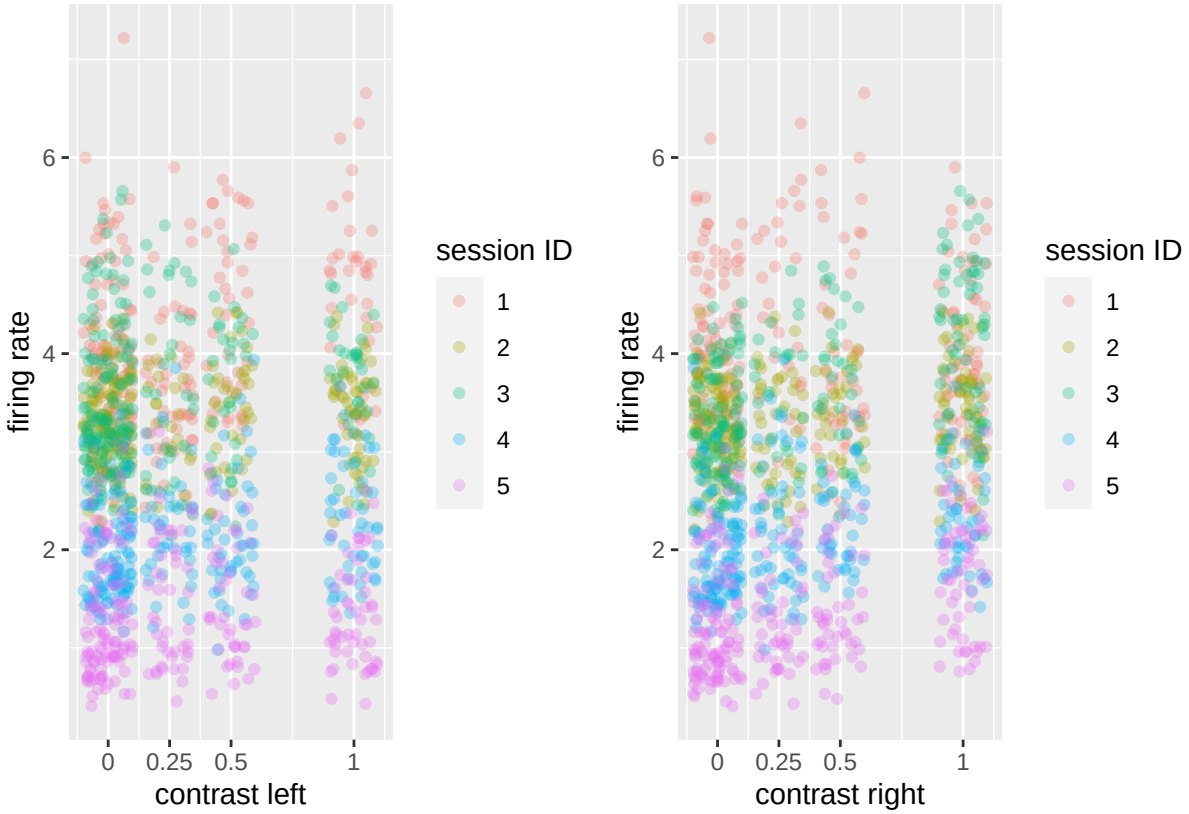
Here are some reasons why mean firing rate is a useful response variable:

- Reflects the average level of neural activity: Mean firing rate provides a measure of the average number of spikes (or action potentials) fired by a neuron over a given time period. This can be useful for characterizing the overall level of activity of a neuron, and how it changes in response to different stimuli or conditions.
- Can be used to compare across different neurons: By computing the mean firing rate across multiple trials or conditions, researchers can compare the activity of different neurons to each other. This can be useful for identifying neurons that respond preferentially to specific types of stimuli or for characterizing how different regions of the brain process information.
- Simple to compute and interpret: Mean firing rate is a relatively simple measure to compute, requiring only the counting of spikes over a given time period. It is also straightforward to interpret, providing a clear measure of the level of activity of a neuron in terms of spikes per second.

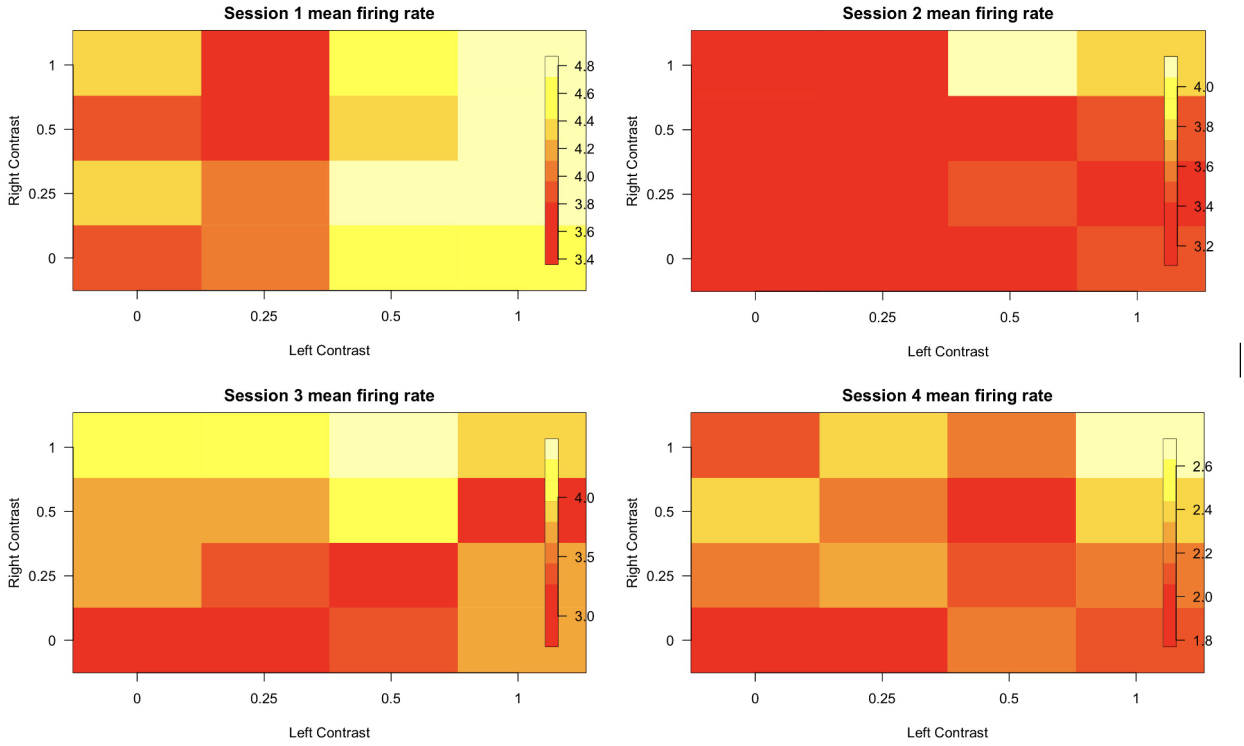
Therefore, we construct five dataframes which contain the session ID, contrast of the left stimulus, contrast of the right stimulus,firing rate of all five sessions then aggregate them into a new dataframe named “Aggregate”.

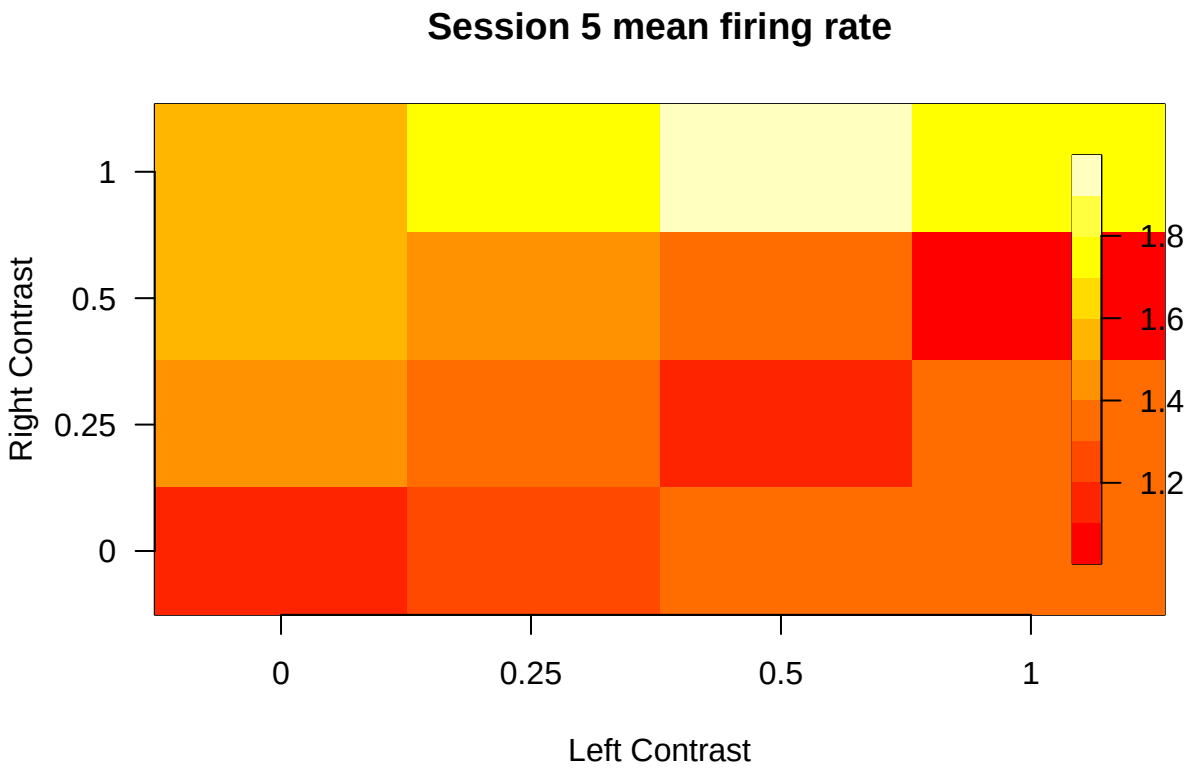
Selection of factors

The following figure summarizes firing rate by contrast of the left stimulus, contrast of the right stimulus.



The above scatter plots show that the mean firing rate distributions in different sessions differ from each other, but in each session the distribution of neurons' mean firing rate is relatively concentrated at certain contrast left/right level. So we decide to consider the session ID as a random effect.

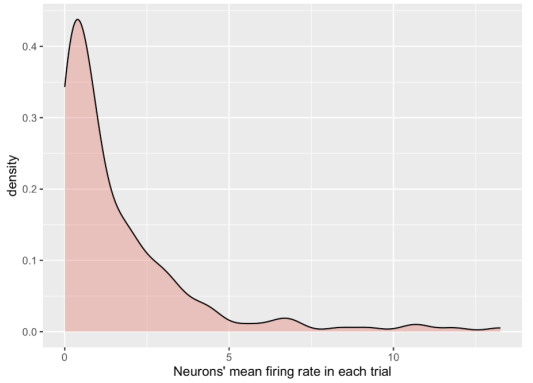


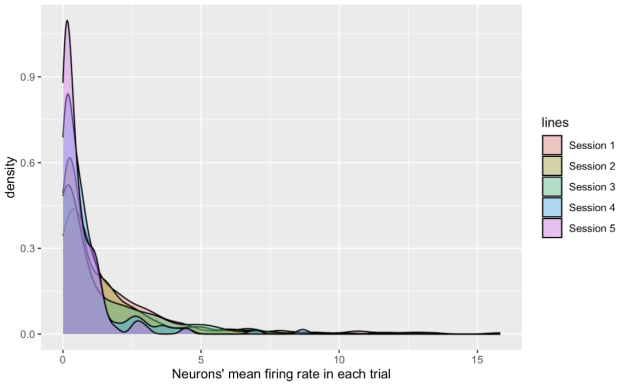


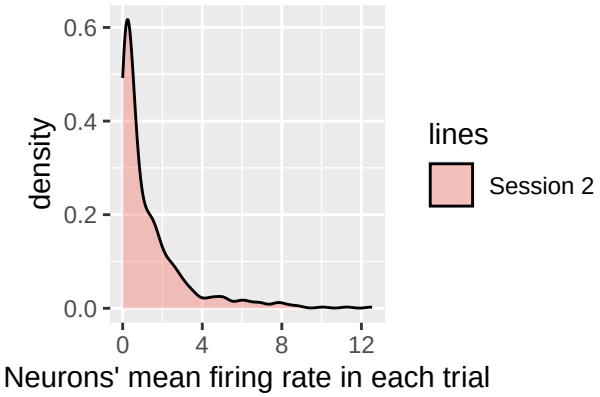
The above color maps depict the mean firing rate in each sessions given the combination of contrasts presented. They show that the firing rate tend to be higher as the left/ritght contrast level ascending. However, they are not symmetry about the $Left\ Contrast = Right\ Contrast$ line, which means that the same level of left and right contrast tend to have different impact on the mean firing rate to mice.

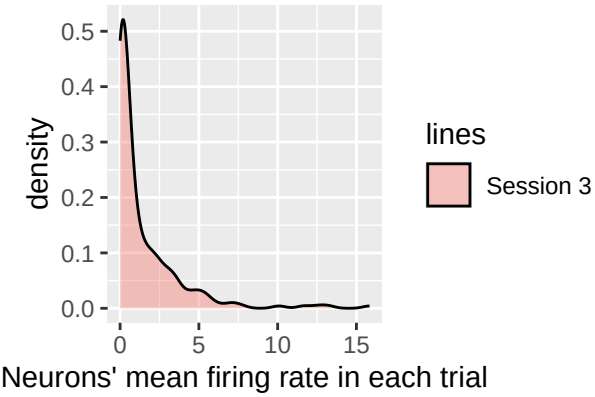
mean firing rate reduced by time

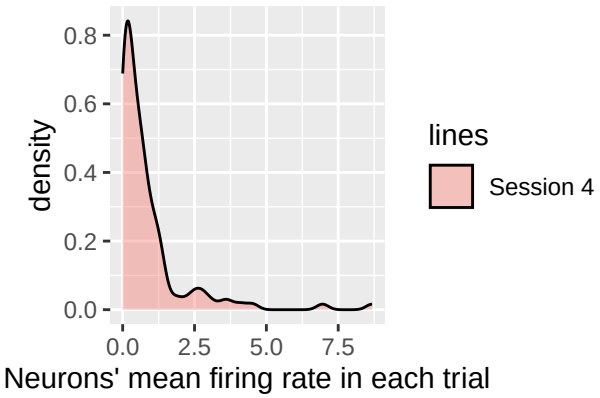
Furthermore, we are interested in the distribution of neurons mean firing rate over 0.4 second, so we calculated the mean firing rate of each neuron in each trials of all sessions. Then we draw the density plot for each session to see the distribution of mean firing rate of each neuron.

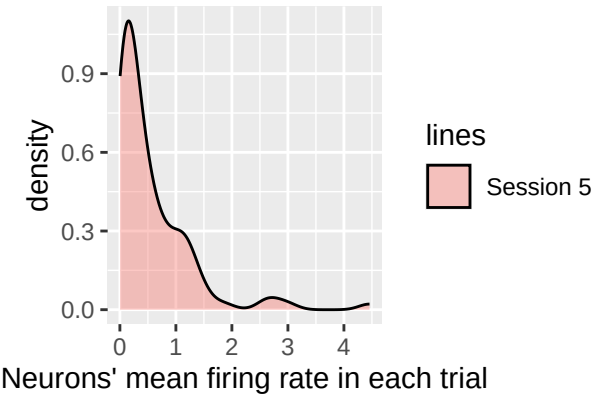






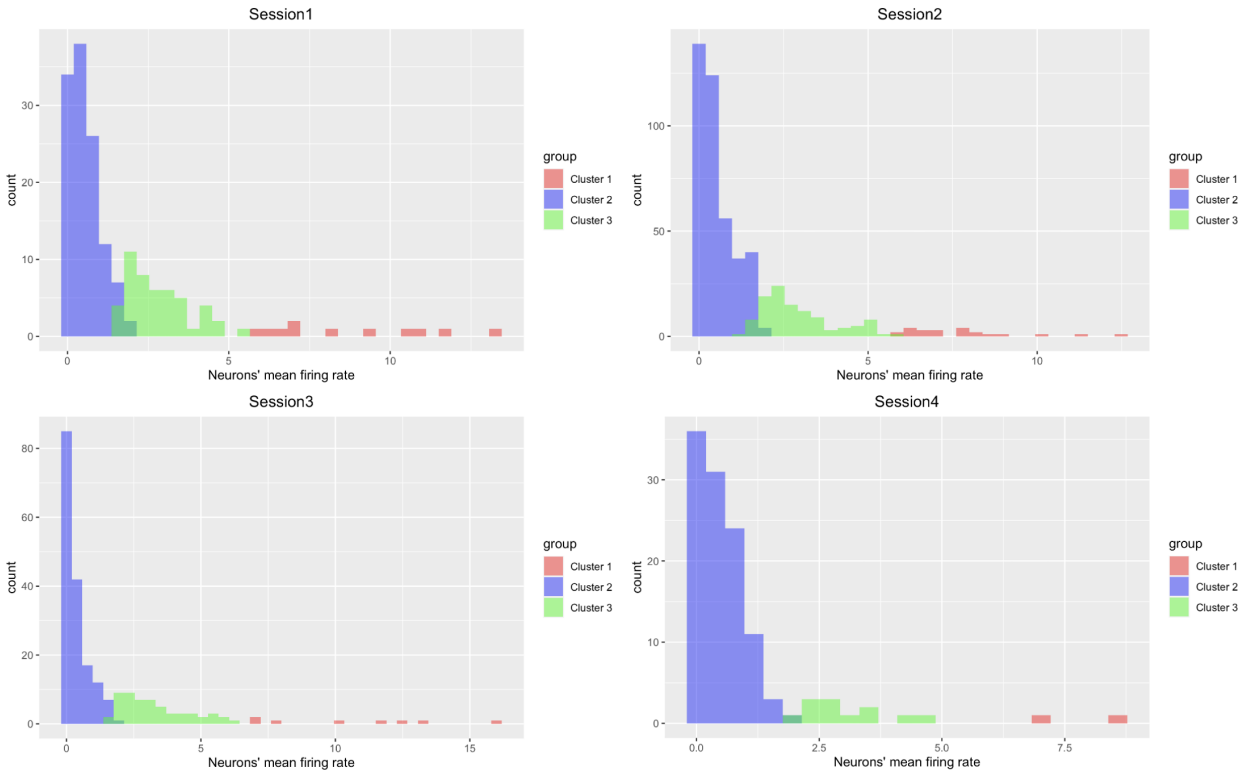






From the above density plots, we guess that there are three types of neurons in rats visual cortex that have different response level ranges when rats are exposed to visual stimulus. So we decide to divide all the neurons into 3 clusters. Since we guess three types of neurons have different mean firing rate, the mean firing rate under $4 \times 4 = 16$ kinds of visual stimulus combination were calculated for each neuron as there coordinate — each neuron has a 16 dimensional vector whose entries are the mean firing rate under specific visual stimulus.

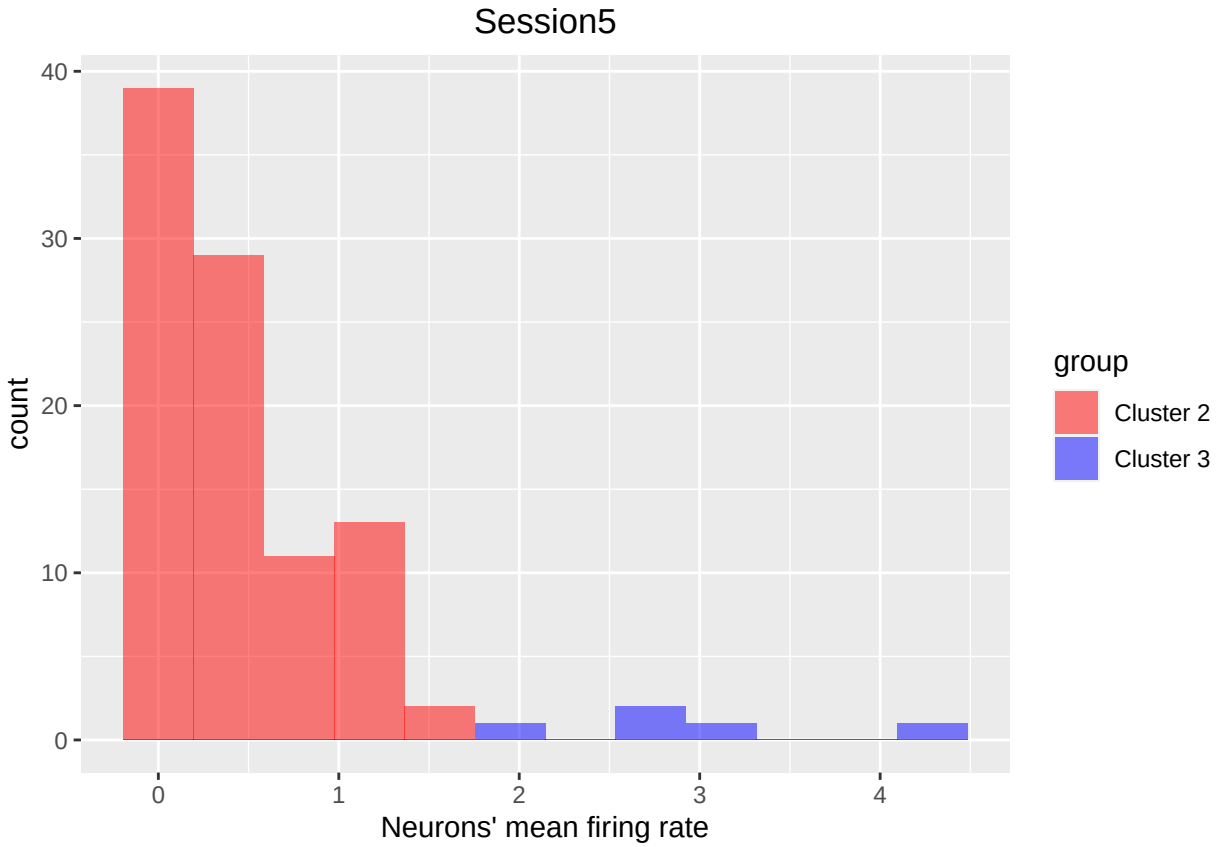
To divide them into 3 clusters, we choose K-means cluster algorithm and derive the labels vector for them. According to the cluster labels, we draw the histograms of neurons in 5 sessions.




```
## Warning in cbind(c(NA, session[[Session_ID]]$mouse_name), c(NA,
## session[[Session_ID]]$feedback_type), : number of rows of result is not a
## multiple of vector length (arg 1)

## Warning in cbind(c(NA, session[[Session_ID]]$mouse_name), c(NA,
## session[[Session_ID]]$feedback_type), : number of rows of result is not a
## multiple of vector length (arg 1)

## Warning in cbind(c(NA, session[[Session_ID]]$mouse_name), c(NA,
## session[[Session_ID]]$feedback_type), : number of rows of result is not a
## multiple of vector length (arg 1)
```



As we can see from the above histograms, in all the session the range of mean firing rate of three clusters have tiny overlap, which means that the cluster labels we get can adequately divide the neurons in all the sessions into three clusters with different mean firing rate range.

Inferential analysis (Q1)

In the inferential analysis, we consider a mixed effect model where the two fixed-effect factors are left contrast and right contrast, and a random intercept is included for each session. As a result, Q1 reduces to test the null hypothesis that the two factors have no interaction effect.

$$f_{ijk} = \mu_{...} + \alpha_i + \beta_j + \gamma_k + (\beta\gamma)_{jk}$$

where $\alpha_i, i = 1, \cdots, 5$ is the factor effect of different sessions, $\beta_j, j = 1, 2, 3, 4$ is the factor effect of contrast of the right stimulus, $\gamma_k, k = 1, 2, 3, 4$ is the factor effect of contrast of the right stimulus, $(\beta\gamma)_{jk}$ is the interaction effect.

```
cluster_index_1 <- list()
cluster_index_2 <- list()
cluster_index_3 <- list()
for(Session_ID in seq(1,5))
{
cluster_1 <- labeled_neuron_list[[Session_ID]][2:dim(labeled_neuron_list[[Session_ID]])[1],which(labele
class(cluster_1) <- "numeric"
cluster_1 <- rowSums(cluster_1)/dim(labeled_neuron_list[[Session_ID]])[1]
cluster_index_1[[Session_ID]] <- cluster_1
```

```
cluster_2 <- labeled_neuron_list[[Session_ID]][2:dim(labeled_neuron_list[[Session_ID]])[1],which(labele
class(cluster_2) <- "numeric"
cluster_2 <- rowSums(cluster_2)/dim(labeled_neuron_list[[Session_ID]])[1]
cluster_index_2[[Session_ID]] <- cluster_2

cluster_3 <- labeled_neuron_list[[Session_ID]][2:dim(labeled_neuron_list[[Session_ID]])[1],which(labele
class(cluster_3) <- "numeric"
cluster_3 <- rowSums(cluster_3)/dim(labeled_neuron_list[[Session_ID]])[1]
cluster_index_3[[Session_ID]] <- cluster_3
}
```

```
Session_series <- rep(1:5, c(214, 251, 228,249,254))
left_series <- c(session[[1]]$contrast_left,session[[2]]$contrast_left,session[[3]]$contrast_left,sessi
right_series <- c(session[[1]]$contrast_right,session[[2]]$contrast_right,session[[3]]$contrast_right,s
feed_back_type_series <- c(session[[1]]$feedback_type,session[[2]]$feedback_type,session[[3]]$feedback_
```

session	number of neurons	number of trials
1	178	214
2	533	251
3	228	228
4	120	249
5	99	254

```
cluster_1_mfr <- c(cluster_index_1[[1]],cluster_index_1[[2]],cluster_index_1[[3]],cluster_index_1[[4]],
additive_1 <- lmer(cluster_1_mfr~as.factor(right_series)+as.factor(left_series)+(1|Session_series))
no_additive_1 <- lmer(cluster_1_mfr~as.factor(right_series)*as.factor(left_series)+(1|Session_series))
anova(additive_1,no_additive_1)
```

```
## refitting model(s) with ML (instead of REML)
```

```
## Data: NULL
## Models:
## additive_1: cluster_1_mfr ~ as.factor(right_series) + as.factor(left_series) + (1 | Session_series)
## no_additive_1: cluster_1_mfr ~ as.factor(right_series) * as.factor(left_series) + (1 | Session_series
##               npar     AIC     BIC logLik deviance  Chisq Df Pr(>Chisq)
## additive_1       9 -2369.5 -2323.7 1193.7  -2387.5
## no_additive_1   18 -2365.6 -2274.1 1200.8  -2401.6 14.156  9     0.1169
```

```
cluster_2_mfr <- c(cluster_index_2[[1]],cluster_index_2[[2]],cluster_index_2[[3]],cluster_index_2[[4]],
additive_2 <- lmer(cluster_2_mfr~as.factor(right_series)+as.factor(left_series)+(1|Session_series))
no_additive_2 <- lmer(cluster_2_mfr~as.factor(right_series)*as.factor(left_series)+(1|Session_series))
anova(additive_2,no_additive_2)
```

```
## refitting model(s) with ML (instead of REML)
```

```
## Data: NULL
## Models:
## additive_2: cluster_2_mfr ~ as.factor(right_series) + as.factor(left_series) + (1 | Session_series)
## no_additive_2: cluster_2_mfr ~ as.factor(right_series) * as.factor(left_series) + (1 | Session_series
##               npar     AIC     BIC logLik deviance  Chisq Df Pr(>Chisq)
## additive_2       9 -2138.5 -2092.7 1078.2  -2156.5
## no_additive_2   18 -2128.6 -2037.1 1082.3  -2164.6 8.1649  9     0.5176
```

```
cluster_3_mfr <- c(cluster_index_3[[1]],cluster_index_3[[2]],cluster_index_3[[3]],cluster_index_3[[4]],c
additive_3 <- lmer(cluster_3_mfr~as.factor(right_series)+as.factor(left_series)+(1|Session_series))
no_additive_3 <- lmer(cluster_3_mfr~as.factor(right_series)*as.factor(left_series)+(1|Session_series))
anova(additive_3,no_additive_3)
```

```
## refitting model(s) with ML (instead of REML)
```

```
## Data: NULL
## Models:
## additive_3: cluster_3_mfr ~ as.factor(right_series) + as.factor(left_series) + (1 | Session_series)
## no_additive_3: cluster_3_mfr ~ as.factor(right_series) * as.factor(left_series) + (1 | Session_series
##               npar     AIC     BIC logLik deviance  Chisq Df Pr(>Chisq)
## additive_3       9 -1319.2 -1273.5 668.61  -1337.2
## no_additive_3   18 -1315.7 -1224.1 675.84  -1351.7 14.458  9      0.107
```

Sensitivity analysis (Q1)

Predictive modeling (Q2)

logistic regression

```
n <- length(left_series)

df <- data.frame(
  feed_back_type_series = (feed_back_type_series[101:n] + 1) / 2,
  left_series = left_series[101:n],
  right_series = right_series[101:n],
  Session_series=Session_series[101:n],
  cluster_1_mfr=cluster_1_mfr[101:n],
  cluster_2_mfr=cluster_2_mfr[101:n],
  cluster_3_mfr=cluster_3_mfr[101:n]
)

# Fit a logistic regression model
model <- glm(feed_back_type_series ~ left_series + right_series+Session_series+cluster_1_mfr+cluster_2_m

new_df <- data.frame(
  feed_back_type_series = (feed_back_type_series[1:100] + 1) / 2,
  left_series = left_series[1:100],
  right_series = right_series[1:100],
  Session_series=Session_series[1:100],
  cluster_1_mfr=cluster_1_mfr[1:100],
  cluster_2_mfr=cluster_2_mfr[1:100],
  cluster_3_mfr=cluster_3_mfr[1:100]
)

# Create a vector of actual binary outcomes for the testing set
actual <- new_df$feed_back_type_series

# Create a vector of predicted probabilities for the testing set
predicted_probs <- predict(model, newdata = new_df, type = "response")

# Convert the predicted probabilities into binary predictions using a threshold value of 0.5
```

```r
predicted <- ifelse(predicted_probs >= 0.5, 1, 0)

# sum(abs(predicted-actual))

# Calculate the ROC curve and AUC
roc_data <- roc(actual, predicted_probs)
```

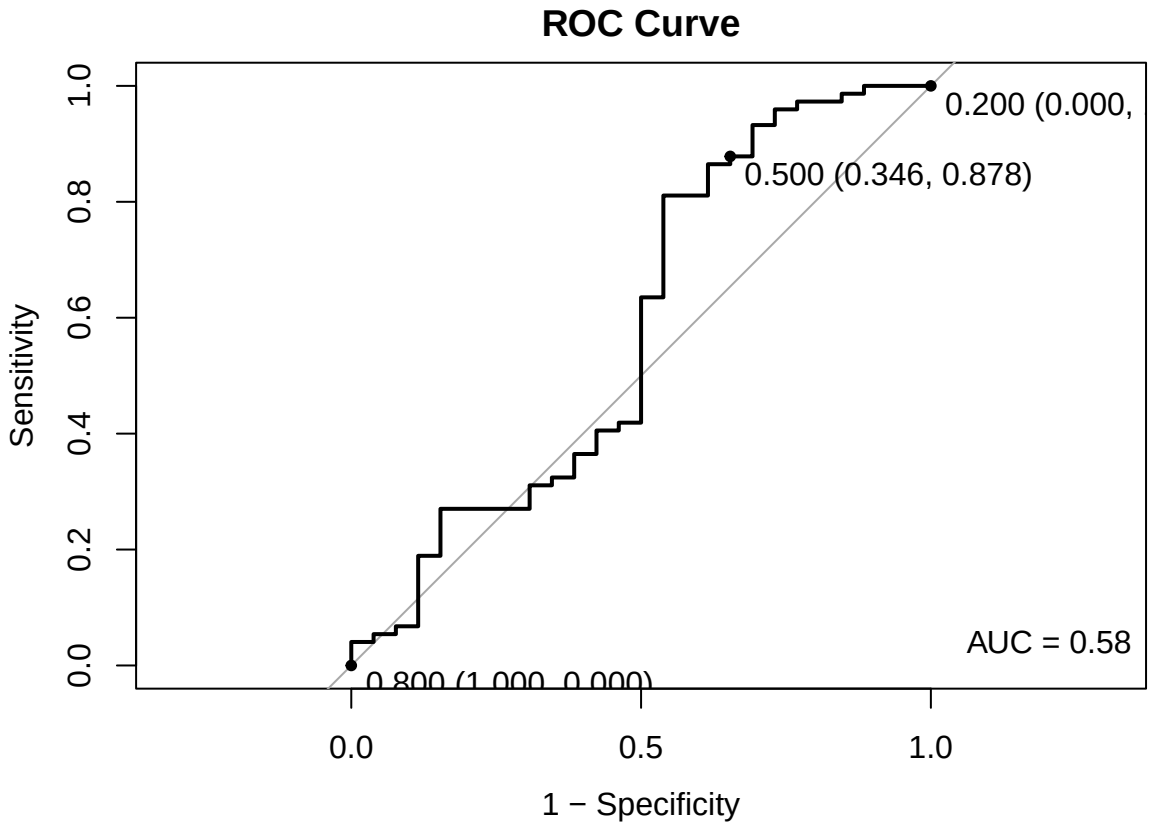
```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```r
auc <- auc(roc_data)

# Plot the ROC curve
plot(roc_data, main = "ROC Curve", print.thres = c(0.2, 0.5, 0.8), legacy.axes = TRUE)

# Add the AUC value to the plot
legend("bottomright", legend = paste("AUC =", round(auc, 2)), bty = "n")
```




```r
df_all_cluster <- data.frame(
  feed_back_type_series = (feed_back_type_series[101:n] + 1) / 2,
  left_series = Aggregate$`contrast left`[101:n],
  right_series = Aggregate$`contrast right`[101:n],
  Session_series=Aggregate$`session ID`[101:n],
  mfr=Aggregate$`firing rate`[101:n]
)
```

```r
model_all_cluster <- glm(feed_back_type_series ~ left_series + right_series+Session_series+mfr, data=df

new_df_all_cluster <- data.frame(
  feed_back_type_series = (feed_back_type_series[1:100] + 1) / 2,
  left_series = Aggregate$`contrast left`[1:100],
  right_series = Aggregate$`contrast right`[1:100],
  Session_series=Aggregate$`session ID`[1:100],
  mfr=Aggregate$`firing rate`[1:100]
)



# Create a vector of actual binary outcomes for the testing set
actual <- new_df_all_cluster$feed_back_type_series

# Create a vector of predicted probabilities for the testing set
predicted_probs <- predict(model_all_cluster, newdata = new_df_all_cluster, type = "response")

# Convert the predicted probabilities into binary predictions using a threshold value of 0.5
predicted <- ifelse(predicted_probs >= 0.5, 1, 0)

sum(abs(predicted-actual))
```

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

```r
# Calculate the ROC curve and AUC
roc_data <- roc(actual, predicted_probs)
```

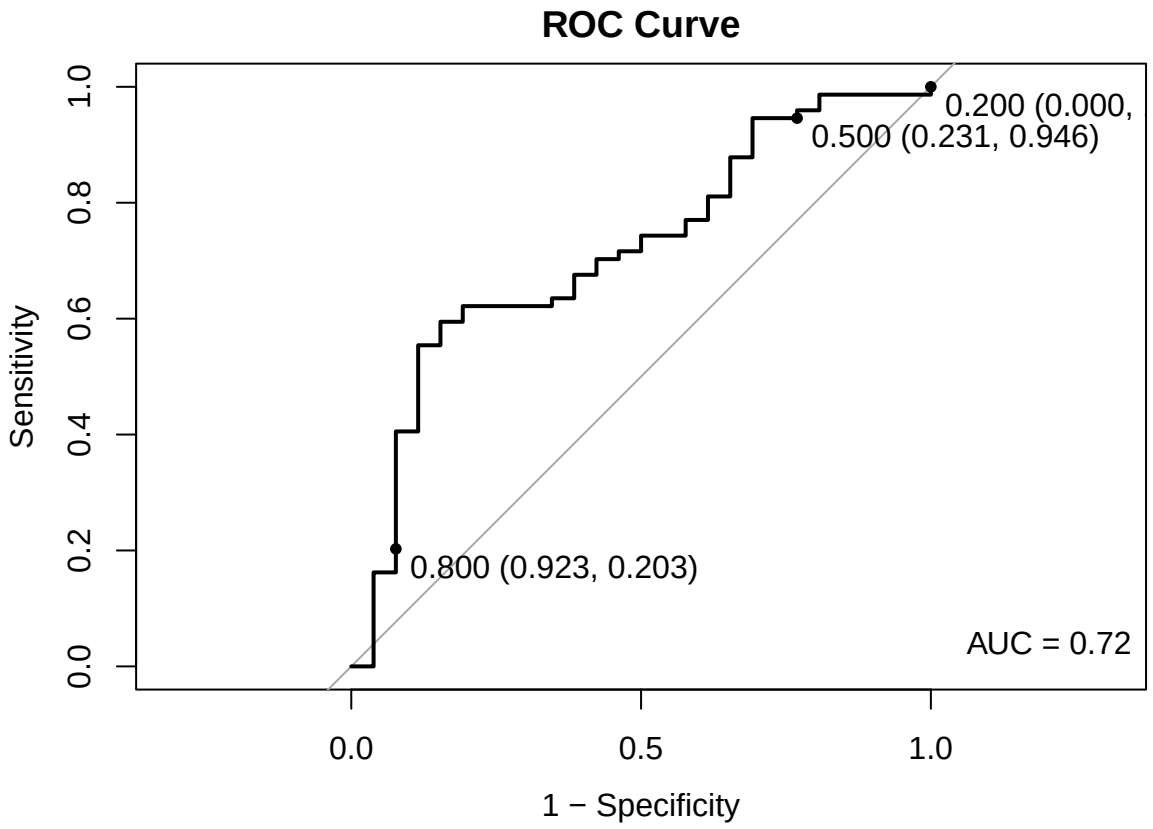
```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```r
auc <- auc(roc_data)

# Plot the ROC curve
plot(roc_data, main = "ROC Curve", print.thres = c(0.2, 0.5, 0.8), legacy.axes = TRUE)

# Add the AUC value to the plot
legend("bottomright", legend = paste("AUC =", round(auc, 2)), bty = "n")
```



LDA

Reference

```
# Load the required packages
# library(ggplot2)
#
# for(i in seq(1,number_of_trial))
#   sum_spks_1=sum_spks_1+session[[1]]$spks[[i]]
#
# sum_spks_1=sum_spks_1/number_of_trial
#
# sum_spks_1=data.frame(time = rep(1:39, dim(session[[1]]$spks[[1]])[1]),
#                   neuron_id = rep(1:dim(session[[1]]$spks[[1]])[1], each = 39),
#                   spike_value = c(sum_spks_1))
#
# # Create a sample dataset
# n <- length(session[[1]]$spks) # Number of spikes
# spikes <- sum_spks_1
# # Define a color palette
# my_colors <- c("#E69F00", "#56B4E9", "#009E73", "#F0E442", "#0072B2", "#D55E00", "#CC79A7")
#
# # Plot spike raster
# ggplot(spikes, aes(x = time, y = neuron_id)) +
#   geom_point(size = 1, alpha = 0.5, shape = '.') +
#   scale_y_continuous(expand = c(0, 0.5)) + # Expand y-axis limits slightly
#   scale_color_manual(values = my_colors) + # Use custom color palette
```

```
#   theme_classic() + # Apply classic theme
#   labs(x = "Time (ms)", y = "Neuron ID", title = "Spike raster plot") + # Add axis labels and title
#   theme(plot.title = element_text(hjust = 0.5, size = 16), # Center plot title and increase font size
#         axis.text = element_text(size = 12), # Increase font size of axis labels
#         axis.title = element_text(size = 14)) # Increase font size of axis titles
```

Steinmetz NA, Zatka-Haas P, Carandini M, Harris KD. Distributed coding of choice, action and engagement across the mouse brain. Nature. 2019 Dec;576(7786):266-273. doi: 10.1038/s41586-019-1787-x. Epub 2019 Nov 27. PMID: 31776518; PMCID: PMC6913580.

Niell, C. M., & Stryker, M. P. (2010). Modulation of visual responses by behavioral state in mouse visual cortex. Neuron, 65(4), 472-479.

Wang, Q., & Burkhalter, A. (2007). Area map of mouse visual cortex. Journal of Comparative Neurology, 502(3), 339-357.

Frenkel, M. Y., Sawtell, N. B., Diogo, A. C. M., Yoon, B., Neve, R. L., & Bear, M. F. (2006). Instructive effect of visual experience in mouse visual cortex. Neuron, 51(3), 339-349.