STA 141A: Final Project

Macy Chen - 923176901

Abstract

This project analyze neural activity to predict decision-making outcomes in mice based on visual stimuli and corresponding neural responses. Using spike trains from the visual cortex of four mice across 18 sessions, we aim to build a predictive model that classifies trial outcomes as success or failure. The project is structured into three phases: exploratory data analysis to understand the dataset, data integration to address variability across sessions, and model training and evaluation. By leveraging neural activity and stimulus information, this work provides insights into the relationship between neural activity and decision-making behavior in visual tasks for mice.

Introduction

In this project, we leverage a dataset from Steinmetz et al. (2019) to explore the relationship between neural activity in the visual cortex and decision-making behavior in mice. The dataset comprises spike trains from neurons in the visual cortex of four mice across 18 experimental sessions, where mice were presented with varying contrast levels of visual stimuli and required to make decisions using a wheel controlled by their forepaws. The primary objective is to develop a predictive model that uses neural activity data (spike trains) and stimulus information (left and right contrast levels) to classify trial outcomes as success (1) or failure (-1).

The primary goal of this project is to build a predictive model that uses neural activity data and stimulus information to classify trial outcomes. To achieve this, we adopt a structured approach divided into three parts. First, we conduct exploratory data analysis to characterize the dataset, including neural spike rates, success rates, and variability across sessions and mice. Second, we integrate data across sessions by identifying shared patterns and addressing session-specific differences to enhance predictive performance. Finally, we train and evaluate a predictive model using test sets from two sessions to assess its ability to generalize across different experimental conditions.

By analyzing the neural correlates of decision-making, we aim to contribute to a deeper understanding of how sensory information is processed and translated into behavior.

Exploratory Analysis

Data processing

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v purrr     1.0.2
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
##
```

```
## Attaching package: 'xgboost'
##
##
## The following object is masked from 'package:dplyr':
##
##     slice
##
##
## Loading required package: lattice
##
##
## Attaching package: 'caret'
##
##
## The following object is masked from 'package:purrr':
##
##     lift
##
##
## Type 'citation("pROC")' for a citation.
##
##
## Attaching package: 'pROC'
##
##
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
session = list()
for(i in 1:18){
  session[[i]] = readRDS(paste('./Data/session',i,'.rds',sep=''))
}

summary(session[[1]])
```

```
##                 Length Class  Mode
## contrast_left   114    -none- numeric
## contrast_right  114    -none- numeric
## feedback_type   114    -none- numeric
## mouse_name        1    -none- character
## brain_area      734    -none- character
## date_exp          1    -none- character
## spks            114    -none- list
## time            114    -none- list
```

```
n_session = length(session)
metadata = tibble(
  session_id = rep(0, n_session),
  mouse_name = rep('name', n_session),
  date_exp = rep('dt', n_session),
  n_brain_area = rep(0, n_session),
  n_neurons = rep(0, n_session),
  n_trials = rep(0, n_session),
  success_rate = rep(0, n_session)
)

for(i in 1:n_session){
  tmp = session[[i]];
  metadata[i, 1] = i;
  metadata[i, 2] = tmp$mouse_name;
  metadata[i, 3] = tmp$date_exp;
  metadata[i, 4] = length(unique(tmp$brain_area));
  metadata[i, 5] = dim(tmp$spks[[1]])[1];
  metadata[i, 6] = length(tmp$feedback_type);
  metadata[i, 7] = mean(tmp$feedback_type + 1) / 2
}
metadata
```

```
## # A tibble: 18 x 7
##    session_id mouse_name date_exp   n_brain_area n_neurons n_trials success_rate
##         <dbl> <chr>      <chr>             <dbl>     <dbl>    <dbl>        <dbl>
##  1          1 Cori       2016-12-14            8       734      114        0.605
##  2          2 Cori       2016-12-17            5      1070      251        0.633
##  3          3 Cori       2016-12-18           11       619      228        0.662
##  4          4 Forssmann  2017-11-01           11      1769      249        0.667
##  5          5 Forssmann  2017-11-02           10      1077      254        0.661
##  6          6 Forssmann  2017-11-04            5      1169      290        0.741
##  7          7 Forssmann  2017-11-05            8       584      252        0.671
##  8          8 Hench      2017-06-15           15      1157      250        0.644
##  9          9 Hench      2017-06-16           12       788      372        0.685
## 10         10 Hench      2017-06-17           13      1172      447        0.620
## 11         11 Hench      2017-06-18            6       857      342        0.795
## 12         12 Lederberg  2017-12-05           12       698      340        0.738
## 13         13 Lederberg  2017-12-06           15       983      300        0.797
## 14         14 Lederberg  2017-12-07           10       756      268        0.694
## 15         15 Lederberg  2017-12-08            8       743      404        0.765
## 16         16 Lederberg  2017-12-09            6       474      280        0.718
## 17         17 Lederberg  2017-12-10            6       565      224        0.830
## 18         18 Lederberg  2017-12-11           10      1090      216        0.806
```

```
get_trial_data <- function(session_id, trial_id) {
  # Retrieve spikes for the specified trial
  spikes <- session[[session_id]]$spks[[trial_id]]

  # Check for missing values
  if (any(is.na(spikes))) {
    message("Missing value in session ", session_id, ", trial ", trial_id)
    return(NULL)  # Return NULL if there are missing values
  }

  # Calculate neuron spike sums
  neuron_spike_sum <- rowSums(spikes)

  # Create a tibble with neuron_spike, brain_area, and calculate region_sum_spike, region_count, and re
  trial_tibble <- tibble(
    neuron_spike = neuron_spike_sum,
    brain_area = session[[session_id]]$brain_area
  ) %>%
    group_by(brain_area) %>%
    summarize(
      region_sum_spike = sum(neuron_spike),
      region_count = n(),
      region_mean_spike = mean(neuron_spike)
    )

  # Add columns for additional trial information
  trial_tibble <- trial_tibble %>%
    add_column(
      trial_id = trial_id,
      contrast_left = session[[session_id]]$contrast_left[trial_id],
      contrast_right = session[[session_id]]$contrast_right[trial_id],
      feedback_type = session[[session_id]]$feedback_type[trial_id],
      contrast_diff = abs(session[[session_id]]$contrast_left[trial_id] - session[[session_id]]$contras
      mouse_name = session[[session_id]]$mouse_name,
      session_id = session_id
    )

  # Return the data for the specific trial
  return(trial_tibble)
}

# Example: Get data for trial 1 in session 1
s1t1_data <- get_trial_data(1, 1)
s1t1_data
```

```
## # A tibble: 8 x 11
##   brain_area region_sum_spike region_count region_mean_spike trial_id
##   <chr>                 <dbl>        <int>             <dbl>    <dbl>
## 1 ACA                     138          109             1.27         1
## 2 CA3                     104           68             1.53         1
## 3 DG                       79           34             2.32         1
## 4 LS                      269          139             1.94         1
## 5 MOs                     106          113             0.938        1
## 6 SUB                     156           75             2.08         1
## 7 VISp                    300          178             1.69         1
## 8 root                      9           18             0.5          1
## # i 6 more variables: contrast_left <dbl>, contrast_right <dbl>,
## #   feedback_type <dbl>, contrast_diff <dbl>, mouse_name <chr>,
## #   session_id <dbl>
```

```
get_session_data <- function(session_id) {
  # Initialize an empty list to store data for each trial
  trial_data_list <- list()

  # Get the total number of trials in the session
  total_trials <- length(session[[session_id]]$spks)

  # Iterate over all trials in the session and call get_trial_data for each
  for (trial_id in seq_len(total_trials)) {
    # Call the get_trial_data function for each trial
    trial_data <- get_trial_data(session_id, trial_id)
```

```r
    # If the trial data is NULL (i.e., it had missing values), skip it
    if (!is.null(trial_data)) {
      trial_data_list[[trial_id]] <- trial_data
    }
  }

  # Combine all trial data into a single tibble
  session_data <- bind_rows(trial_data_list)

  return(session_data)
}

# Example: Get all trial data for session 1
session1_data <- get_session_data(3)
session1_data
```

```
## # A tibble: 2,508 x 11
##    brain_area region_sum_spike region_count region_mean_spike trial_id
##    <chr>                 <dbl>        <int>             <dbl>    <int>
##  1 CA1                     115           42              2.74        1
##  2 DG                      138           34              4.06        1
##  3 LP                       18            4              4.5         1
##  4 MG                      373          137              2.72        1
##  5 MRN                     231           41              5.63        1
##  6 NB                       72           43              1.67        1
##  7 POST                     70           63              1.11        1
##  8 SPF                      69           15              4.6         1
##  9 VISam                   235          114              2.06        1
## 10 VISp                    151          114              1.32        1
## # i 2,498 more rows
## # i 6 more variables: contrast_left <dbl>, contrast_right <dbl>,
## #   feedback_type <dbl>, contrast_diff <dbl>, mouse_name <chr>,
## #   session_id <dbl>
```

```r
session_data_list <- list()

for (i in 1:18) {
  x <- get_session_data(i)
  session_data_list[[i]] <- x
}

# Combine all session data
all_session_data <- bind_rows(session_data_list)
all_session_data$session_id <- factor(all_session_data$session_id)
all_session_data
```

```
## # A tibble: 49,173 x 11
##    brain_area region_sum_spike region_count region_mean_spike trial_id
##    <chr>                 <dbl>        <int>             <dbl>    <int>
##  1 ACA                     138          109             1.27         1
##  2 CA3                     104           68             1.53         1
##  3 DG                       79           34             2.32         1
##  4 LS                      269          139             1.94         1
##  5 MOs                     106          113             0.938        1
##  6 SUB                    156         75              2.08         1
##  7 VISp                   300        178              1.69         1
##  8 root                     9         18              0.5          1
##  9 ACA                     64        109              0.587        2
## 10 CA3                    116         68              1.71         2
## # i 49,163 more rows
## # i 6 more variables: contrast_left <dbl>, contrast_right <dbl>,
## #   feedback_type <dbl>, contrast_diff <dbl>, mouse_name <chr>,
## #   session_id <fct>
```

```r
ggplot(all_session_data, aes(x = session_id, y = brain_area)) +
  geom_point() +
  labs(title = "Brain Areas with Neurons Recorded in Each Session",
       x = "Session",
       y = "Brain Area") +
  theme_minimal()
```

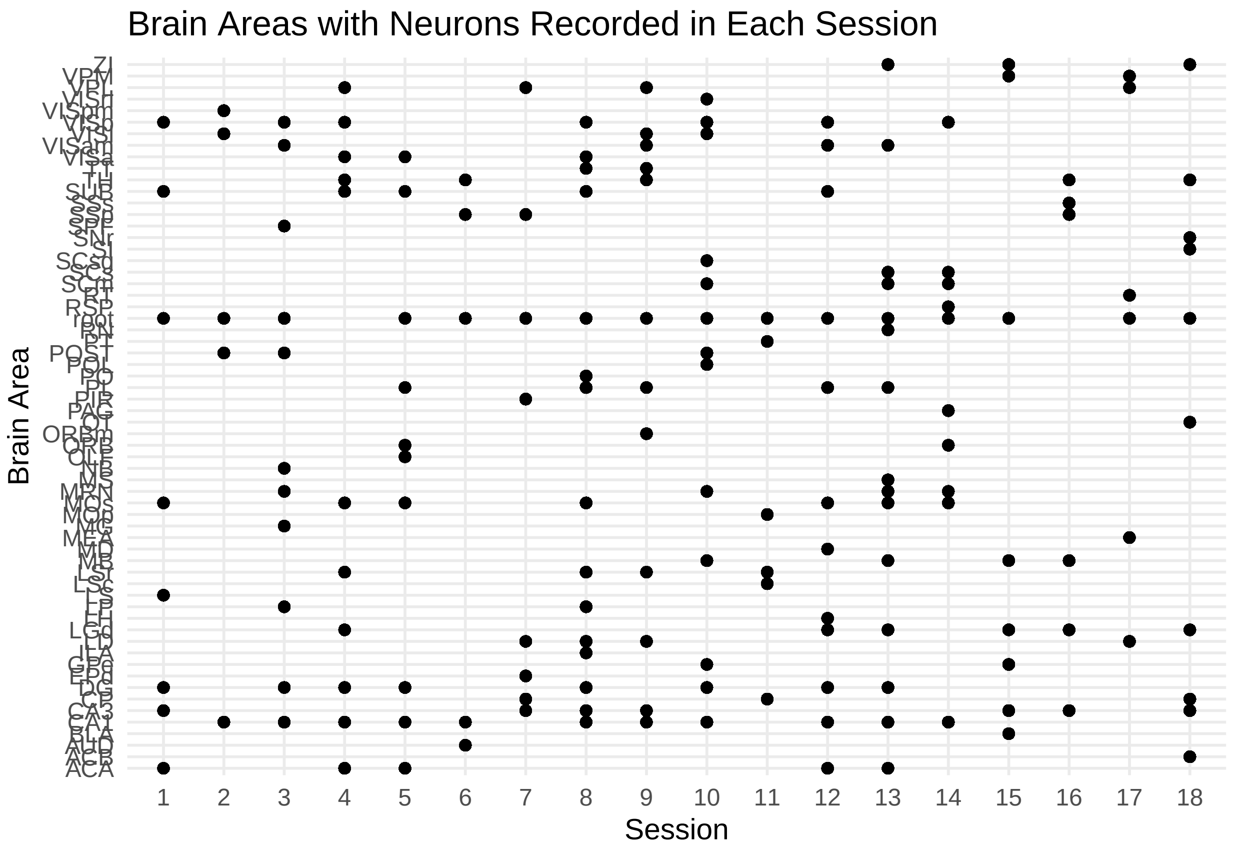


Figure 1.0 - Brain Areas with Neurons Recorded in Each Session

This dot plot visualizes the brain areas where neurons were recorded across 18 experimental sessions. Each black dot represents a recorded neuron in a specific brain area during a given session. Neurons were recorded across a wide range of brain areas, indicating a broad sampling of neural activity. However, not all areas were recorded in every session. The uneven distribution of recorded neurons across sessions might suggest experimental variability or selective recording of specific regions at different times. Some brain areas, such as root and CA1, show more consistent recordings across multiple sessions, while others appear sporadically. This could impact model performance if certain areas contribute more to trial outcomes.

```r
# Function to calculate contract difference distribution
get_contrast_difference <- function(session_data, session_id) {
  session_data %>%
    count(contrast_diff) %>%
    mutate(session_id = session_id,
           percentage = n / sum(n) * 100,
           labels = paste0(round(percentage, 2), "%"))
}

# Compute contract difference distribution for all 18 sessions
contrast_difference_data <- bind_rows(lapply(1:18, function(i) {
  session_data <- get_session_data(i)
  get_contrast_difference(session_data, i)
}))

contrast_difference_data
```

```
## # A tibble: 90 x 5
##    contrast_diff     n session_id percentage labels
##            <dbl> <int>      <int>      <dbl> <chr>
##  1          0      240          1      26.3  26.32%
##  2          0.25   112          1      12.3  12.28%
##  3          0.5    216          1      23.7  23.68%
##  4          0.75   200          1      21.9  21.93%
##  5          1      144          1      15.8  15.79%
##  6          0      435          2      34.7  34.66%
##  7          0.25   120          2       9.56 9.56%
##  8          0.5    240          2      19.1  19.12%
##  9          0.75   140          2      11.2  11.16%
## 10          1      320          2      25.5  25.5%
## # i 80 more rows
```

```
# Plot the distribution
ggplot(contrast_difference_data, aes(x = contrast_diff, y = n)) +
  geom_bar(stat = "identity") +
  facet_wrap(~ session_id, nrow = 6, ncol = 3) +  # Arrange in 3x6 grid
  geom_text(aes(label = labels, y = n + 50), size = 3) + # Adjust y position for label visibility
  labs(title = "Contrast Difference Distribution",
       x = "Contrast Difference",
       y = "Count") +
  scale_x_continuous(breaks = seq(0, 1, by = 0.25)) +  # Set x-axis breaks to go by 0.25
  theme_minimal()
```

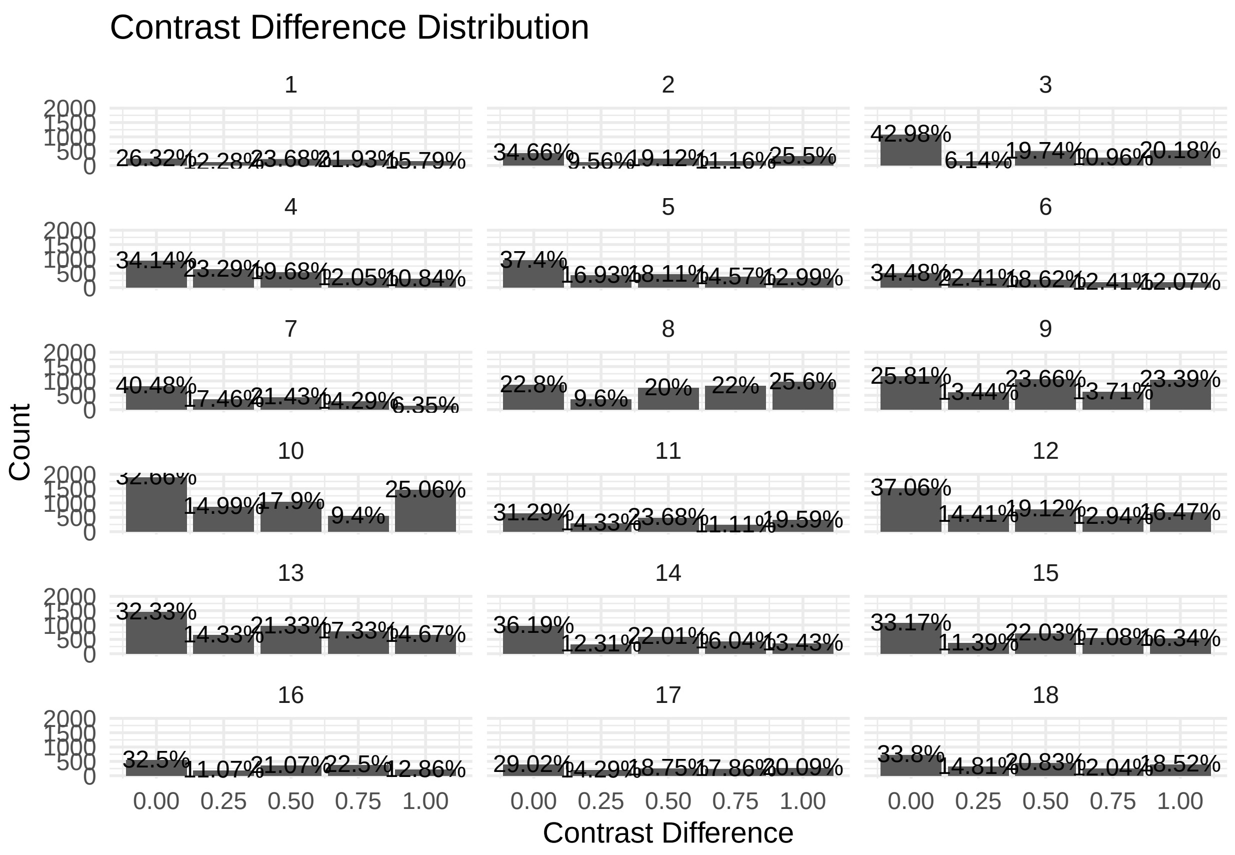


Figure 2.0 - Distribution of contrast difference (relative difference in contrast between the left and right stimuli) across sessions

This histogram displays the distribution of contrast differences across 18 experimental sessions. The contrast difference represents the relative difference in contrast between the left and right stimuli, influencing the mouse's decision-making process.

There is variability in contrast difference distributions across sessions, indicating that stimulus presentation was not uniform over time. Some sessions, such as session 3, 8, and 10, show a strong bias toward specific contrast differences, with a higher proportion of trials at a contrast difference of 1.0. Other sessions, such as session 1, 12, 17, have a more even distribution of contrast differences, which may indicate a more balanced experimental design in those cases. In many sessions, the highest contrast difference (1.00) appears most frequently, meaning that trials with strong contrast imbalances were more prevalent. This could make classification easier since high contrast differences likely lead to clearer decision-making by the mice. Howver, sessions with an overrepresentation of specific contrast differences might lead to biased predictive models. If the model sees fewer trials with intermediate contrast differences (e.g., 0.25, 0.5), it may struggle to generalize well in those cases.

```r
# Function to compute success rate per contrast_diff for a session
get_success_rate_per_contrast_diff <- function(session_data) {
  session_data %>%
    group_by(session_id, contrast_diff) %>%
    summarize(success_rate = mean(feedback_type == 1),
              .groups = "drop") %>% # Compute proportion of successful trials
    arrange(session_id, contrast_diff)
}

# Compute success rate per contrast diff for all 18 sessions
success_rate_per_contrast_diff_data <- bind_rows(lapply(1:18, function(i) {
  session_data <- get_session_data(i)
  get_success_rate_per_contrast_diff(session_data)
}))

success_rate_per_contrast_diff_data
```

```
## # A tibble: 90 x 3
##    session_id contrast_diff success_rate
##         <int>         <dbl>        <dbl>
##  1          1          0           0.6
##  2          1          0.25        0.571
##  3          1          0.5         0.630
##  4          1          0.75        0.36
##  5          1          1           0.944
##  6          2          0           0.644
##  7          2          0.25        0.625
##  8          2          0.5         0.792
##  9          2          0.75        0.536
## 10          2          1           0.547
## # i 80 more rows
```

```r
# Plot success rate per contrast diff for each session
ggplot(success_rate_per_contrast_diff_data, aes(x = contrast_diff, y = success_rate)) +
  geom_bar(stat = "identity", show.legend = FALSE) +
  facet_wrap(~ session_id, nrow = 6, ncol = 3) +  # Arrange in 3x6 grid
  labs(
    title = "Success Rate Per Contrast Difference for Each Session",
    x = "Contrast Difference",
    y = "Success Rate"
  ) +
  scale_x_continuous(breaks = seq(0, 1, by = 0.25)) +
  theme_minimal()
```

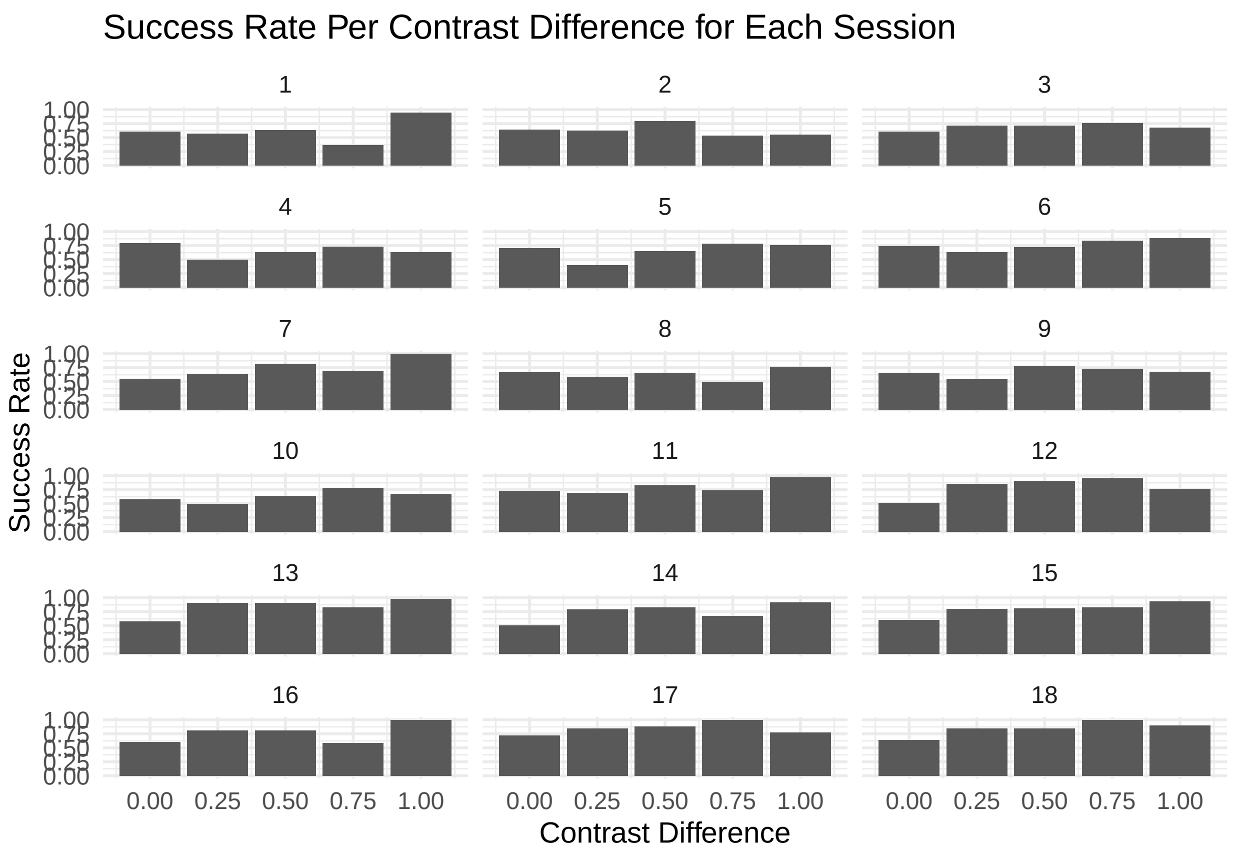


Figure 3.0 - Success Rate Per Contrast Difference Across Sessions

This bar graph illustrates the relationship between contrast difference and trial success rate across 18 experimental sessions. The success rate represents the proportion of successful trials for each contrast difference level.

For most of the sessions, it can be seen that higher contrast differences ten to lead to higher success rates. This trend suggests that when the difference between left and right stimuli is more pronounced, the mice make more accurate decisions. While the general pattern holds across most sessions, some sessions display higher success rates for lack of contrast. This may indicate that other factors, such as time and brain area, play a role in performance beyond just the contrast levels

```r
# Function to compute success rate over time for a session
get_success_rate_over_time <- function(session_data, bin_size) {
  session_data %>%
    arrange(trial_id) %>%
    mutate(
      bin = (trial_id - 1) %/% bin_size + 1  # Assign each trial to a bin
    ) %>%
    group_by(session_id, bin) %>%
    summarize(
      success_rate = mean(feedback_type == 1),  # Compute success rate
      .groups = "drop"
    )
}

# Compute success rate for all 18 sessions
success_rate_over_time_data <- bind_rows(lapply(1:18, function(i) {
  session_data <- get_session_data(i)
  get_success_rate_over_time(session_data, 20)
}))

success_rate_over_time_data
```

```
## # A tibble: 262 x 3
##    session_id   bin success_rate
##         <int> <dbl>        <dbl>
##  1          1     1        0.65
##  2          1     2        0.8
##  3          1     3        0.6
##  4          1     4        0.7
##  5          1     5        0.55
##  6          1     6        0.214
##  7          2     1        0.6
##  8          2     2        0.8
##  9          2     3        0.75
## 10          2     4        0.8
## # i 252 more rows
```

```r
# Plot success rate over time for each session
ggplot(success_rate_over_time_data, aes(x = factor(bin), y = success_rate)) +
  geom_bar(stat = "identity", show.legend = FALSE) +
  facet_wrap(~ session_id, nrow = 6, ncol = 3) +  # Arrange in 3x6 grid
  labs(
    title = "Success Rate Over Time for Each Session",
    x = "Time Bin",
    y = "Success Rate"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))  # Rotate x-axis labels
```

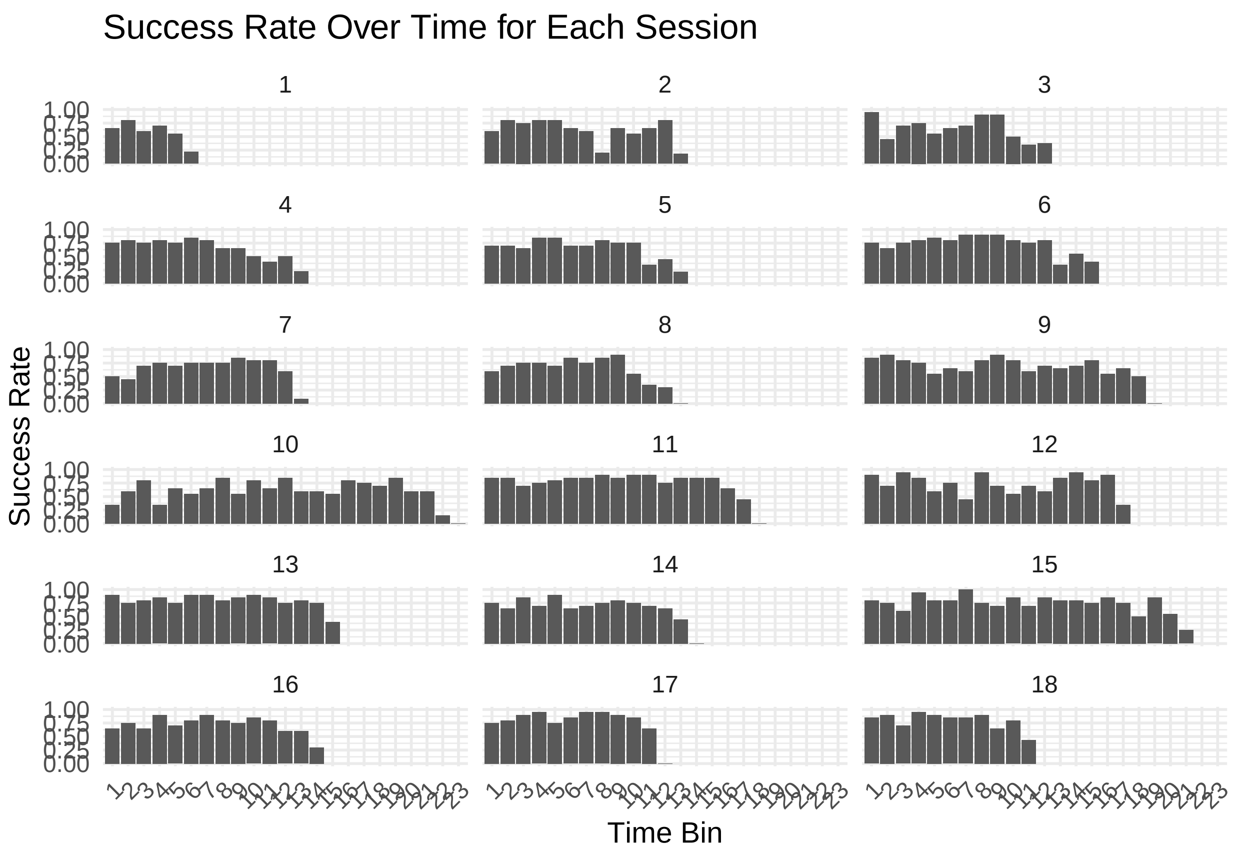


Figure 4.0 - Success rate over time for each Session

This bar chart demonstrates the relationship between time and trial success rate across 18 experimental sessions. Many sessions show a downward trend in success rate as time progresses, suggesting that performance declines within a session. This pattern may indicate fatigue, reduced attention, or motivation loss over time. Furthermore, earlier sessions generally show a sharper decline, with later times having lower success rates compared to later sessions. This pattern could suggest learning or adaptation over multiple days.

```r
# Function to compute success rate per mouse over time
get_mouse_success_rate <- function(all_session_data, bin_size = 20) {
  all_session_data %>%
    arrange(trial_id) %>%
    mutate(
      bin = (trial_id - 1) %/% bin_size + 1  # Assign trials to bins
    ) %>%
    group_by(mouse_name, bin) %>%
    summarize(
      success_rate = mean(feedback_type == 1),  # Compute success rate
      .groups = "drop"
    )
}

# Compute success rate per mouse
mouse_success_rate_data <- get_mouse_success_rate(all_session_data)
head(mouse_success_rate_data)
```

```
## # A tibble: 6 x 3
##   mouse_name   bin success_rate
##   <chr>      <dbl>        <dbl>
## 1 Cori           1        0.777
## 2 Cori           2        0.640
## 3 Cori           3        0.677
## 4 Cori           4        0.744
## 5 Cori           5        0.602
## 6 Cori           6        0.537
```

```
# Plot success rate over time for each mouse
ggplot(mouse_success_rate_data, aes(x = factor(bin), y = success_rate)) +
  geom_bar(stat = "identity", show.legend = FALSE) +
  facet_wrap(~ mouse_name, nrow = 2, ncol = 2) +  # Arrange in 2x2 grid
  labs(
    title = "Success Rate Over Time for Each Mouse",
    x = "Time Bin",
    y = "Success Rate"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))  # Rotate x-axis labels
```

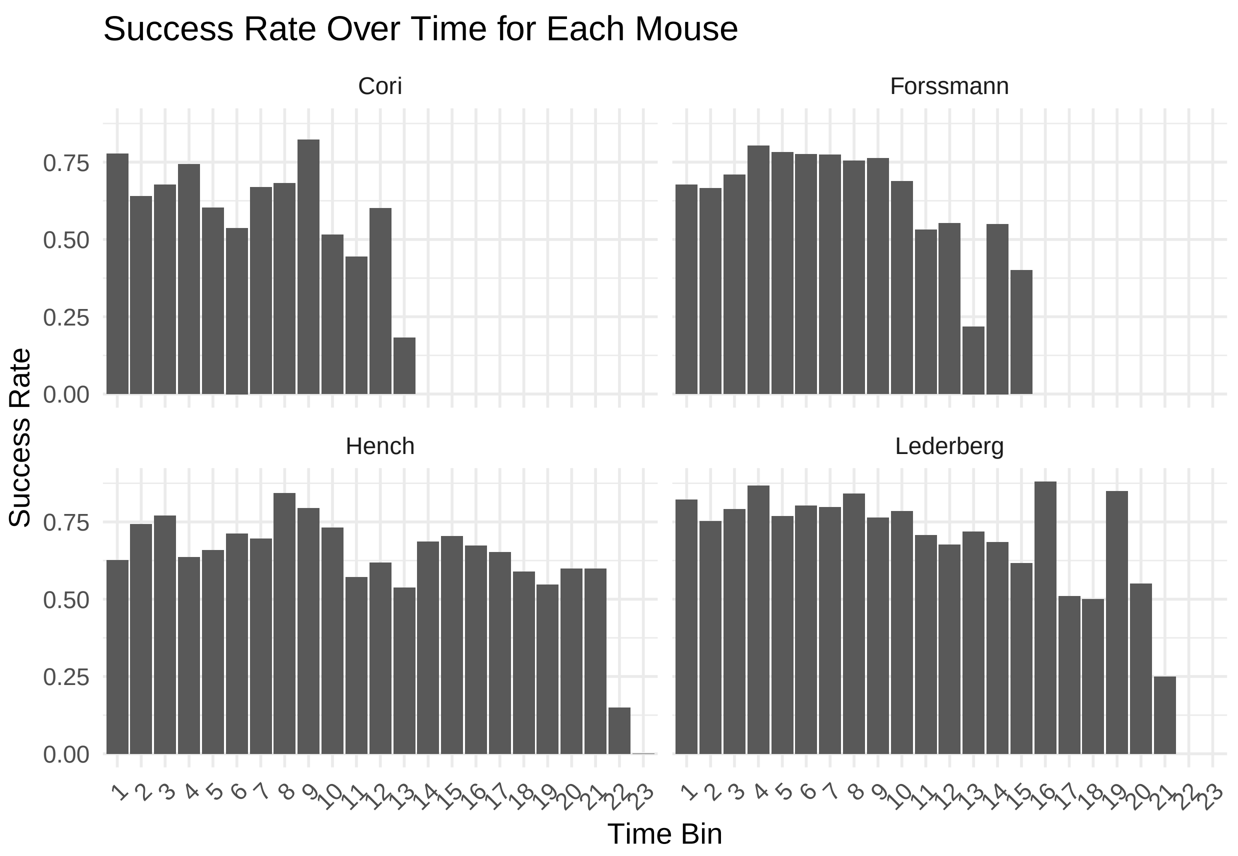


Figure 4.0 - Success rate over time for each mouse

This bar chart demonstrates the relationship between time and trial success rate for the four different mouse. All sessions show a general decrease in success rate as time progresses, suggesting that performance declines over time for all mice. This pattern with the findings from figure 3.0, suggesting that suggesting that the decline in success rate over time is a consistent trend across both individual mice and overall sessions. This reinforces the idea that factors such as fatigue, motivation or attention loss may play a role in diminishing performance.

```r
# Function to compute average spike rate per session over time
get_session_spike_rate <- function(session_data, bin_size = 1) {
  session_data %>%
    arrange(trial_id) %>%
    mutate(
      bin = (trial_id - 1) %/% bin_size + 1,  # Assign trials to bins
      spike_rate = region_sum_spike / region_count  # Compute avg spike rate per trial
    ) %>%
    group_by(session_id, bin) %>%
    summarize(
      avg_spike_rate = mean(spike_rate),  # Compute mean spike rate per bin
      .groups = "drop"
    )
}

# Compute success rate for all 18 sessions
session_spike_rate_data <- bind_rows(lapply(1:18, function(i) {
  session_data <- get_session_data(i)
  get_session_spike_rate(session_data)
}))

session_spike_rate_data
```

```
## # A tibble: 5,081 x 3
##    session_id   bin avg_spike_rate
##         <int> <dbl>          <dbl>
##  1          1     1           1.53
##  2          1     2           1.28
##  3          1     3           1.95
##  4          1     4           1.51
##  5          1     5           1.47
##  6          1     6           1.10
##  7          1     7           2.21
##  8          1     8           1.70
##  9          1     9           1.57
## 10          1    10           1.77
## # i 5,071 more rows
```

```r
# Plot spike rate over time
ggplot(session_spike_rate_data, aes(x = bin, y = avg_spike_rate)) +
  geom_line() +
  geom_smooth(method = "loess") +  # Trend line
  facet_wrap(~ session_id, nrow = 3, ncol = 6) +  # Arrange in 3x6 grid
  labs(
    title = "Neuron Spike Rate Over Time for Each Session",
    x = "Time Bin",
    y = "Average Neuron Spike Rate",
  ) +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

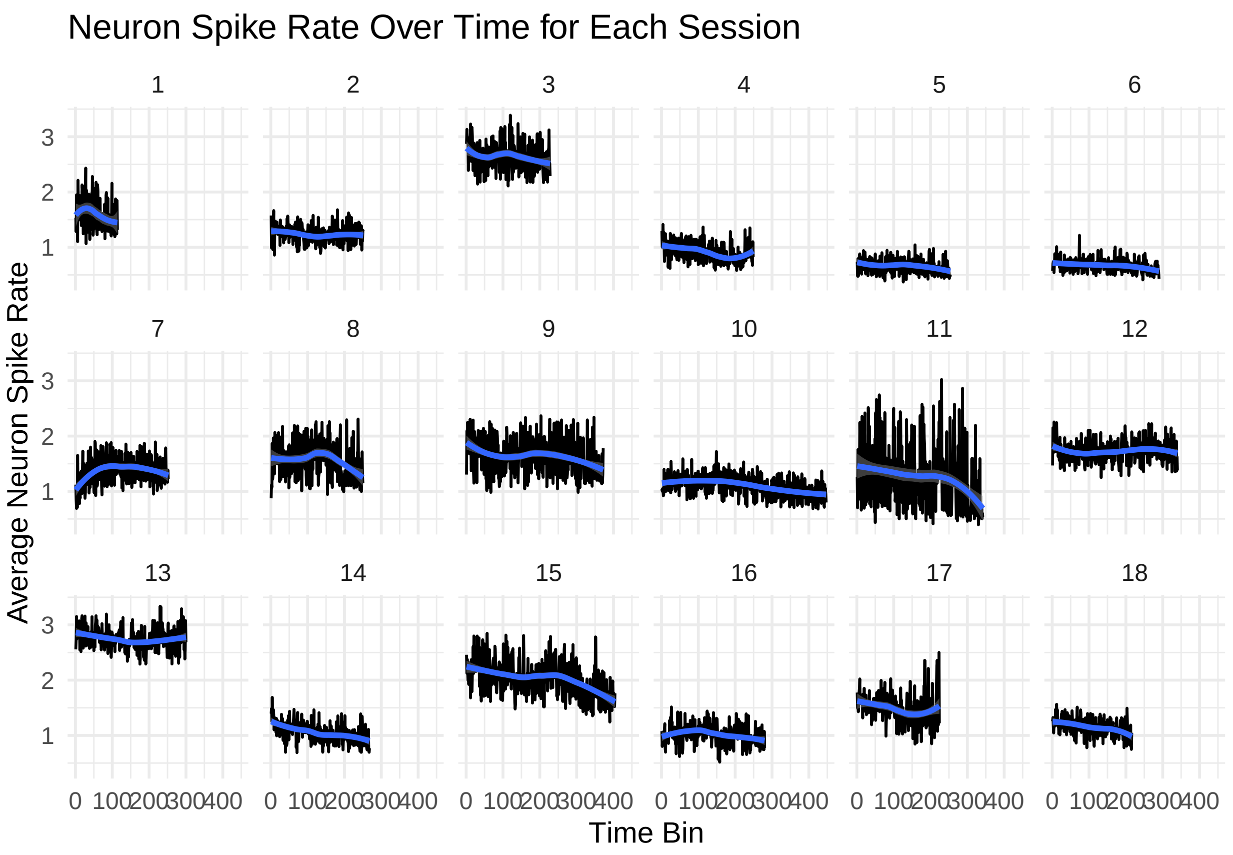


Figure 5.0 - Neuron spike rate over time for each session

This line graph illustrates how neuron spike rates change over time across different experimental sessions. A general decrease in neuron spike rate is observed in most sessions, indicating a possible reduction in neural activity as the session progresses. The decline could be associated with factors such as fatigue, adaptation to the task, or reduced engagement over time. This trend aligns with the decline in trial success rate over time observed in previous figures, implying that reduced neural activity might contribute to poorer task performance.

However, some sessions exhibit higher overall spike rates, suggesting variability in neural engagement across different experimental runs. Additionally, certain sessions display large fluctuations (spikes) in neuron activity, which may correspond to periods of heightened engagement, changes in stimulus processing, or bursts of decision-making activity. When considering Figure 1.0 (Brain Areas with Neurons Recorded in Each Session), the observed decline in spike rate over time could indicate that certain brain regions show diminishing activity more rapidly than others, potentially affecting task performance. Moreover, the presence of fluctuations in neuron activity in some sessions might be linked to recordings from specific highly active brain areas.

```r
# Function to compute success rate per mouse over time
get_mouse_spike_rate <- function(all_session_data) {
  all_session_data %>%
    arrange(trial_id) %>%
    mutate(
      spike_rate = region_sum_spike / region_count  # Compute avg spike rate per trial
    ) %>%
    group_by(mouse_name, trial_id) %>%
    summarize(
      avg_spike_rate = mean(spike_rate),  # Compute mean spike rate per bin
      .groups = "drop"
    )
}

# Compute success rate per mouse
mouse_spike_rate_data <- get_mouse_spike_rate(all_session_data)
mouse_spike_rate_data
```

```
## # A tibble: 1,392 x 3
##    mouse_name trial_id avg_spike_rate
##    <chr>         <int>          <dbl>
##  1 Cori              1           2.16
##  2 Cori              2           2.07
##  3 Cori              3           2.29
##  4 Cori              4           2.14
##  5 Cori              5           2.03
##  6 Cori              6           1.66
##  7 Cori              7           2.33
##  8 Cori              8           2.26
##  9 Cori              9           2.04
## 10 Cori             10           1.99
## # i 1,382 more rows
```

```r
# Plot spike rate over time for each mouse
ggplot(mouse_spike_rate_data, aes(x = trial_id, y = avg_spike_rate)) +
  geom_line() +
  geom_smooth(method = "loess") +  # Trend line
  facet_wrap(~ mouse_name, nrow = 2, ncol = 2) +  # Arrange in 2x2 grid
  labs(
    title = "Average Neuron Spike Rate Over Time for Each Mouse",
    x = "Time Bin",
    y = "Average Neuron Spike Rate"
  ) +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

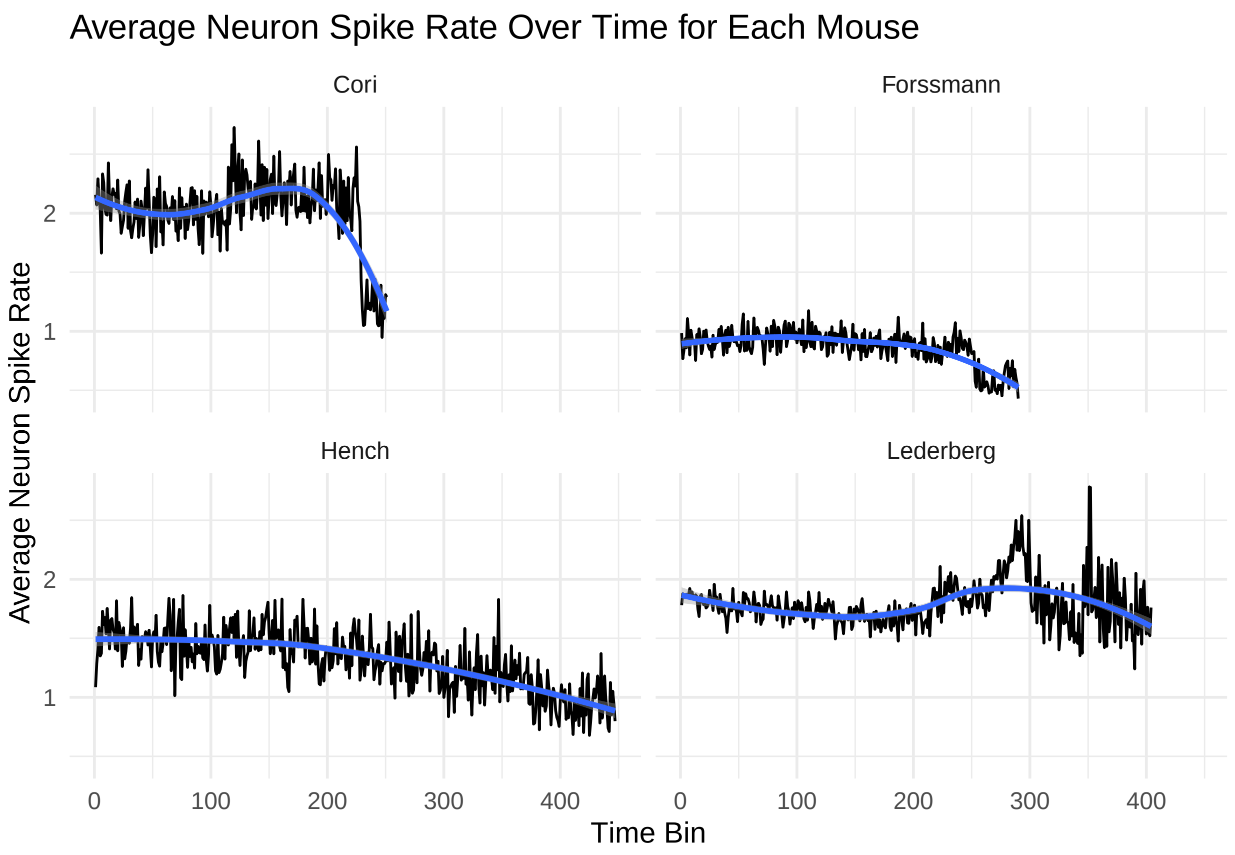


Figure 6.0 - Neuron spike rate over time for each mouse

This line graph illustrates the neuron spike rate trends over time for each individual mouse. Across all mice, a decline in neuron spike rate over time is observed, suggesting that neural activity diminishes as the session progresses. This could be due to factors such as fatigue, adaptation, or decreased engagement with the task over time.

However, some mice, such as Cori and Lederberg exhibit consistently higher spike rates compared to others, indicating potential differences in neural excitability or brain region involvement. This variation may be linked to differences in the brain areas recorded (Figure 1.0) or individual differences in cognitive or motor processing.

When comparing this to Figure 5.0 (Neuron Spike Rate Over Time for Each Session), the overall trend remains consistent, reinforcing the idea that neuron spike rates decline with time. This aligns with behavioral performance declines (Figure 4.0), suggesting that reduced neural activity could contribute to decreasing trial success rates over time.

Data Integration

```
for(i in 1:18){
  n_trials = length(session[[i]]$feedback_type)
  avg_spikes_all = numeric(n_trials)

  for(j in 1:n_trials){
    spk_trial = session[[i]]$spks[[j]]
    total_spikes = apply(spk_trial, 1, sum)
    avg_spikes_all[j] = mean(total_spikes)
  }
  session[[i]]$avg_spks = avg_spikes_all
}
```

```
model_data <- tibble()

for (session_id in 1:18) {

  n_trials = length(session[[session_id]]$feedback_type)
  tmp = session[[session_id]]

  # Initialize trials tibble for the current session
  trials <- tibble(
    mouse_name = rep('mouse_name', n_trials),
    avg_spks = rep(0, n_trials),
    contrast_left = rep(0, n_trials),
    contrast_right = rep(0, n_trials),
    feedback_type = rep(0, n_trials),
    session_ID = rep(0, n_trials)
  )

  # Populate trials tibble with session data
  for (j in 1:n_trials) {
    trials[j, 1] = tmp$mouse_name
    trials[j, 2] = tmp$avg_spks[j]
    trials[j, 3] = tmp$contrast_left[j]
    trials[j, 4] = tmp$contrast_right[j]
    trials[j, 5] = tmp$feedback_type[j]
    trials[j, 6] = session_id
  }

  model_data = rbind(model_data, trials)
}

model_data
```

```
## # A tibble: 5,081 x 6
##    mouse_name avg_spks contrast_left contrast_right feedback_type session_ID
##    <chr>         <dbl>         <dbl>          <dbl>         <dbl>      <dbl>
##  1 Cori           1.58           0             0.5              1          1
##  2 Cori           1.31           0             0                1          1
##  3 Cori           1.84           0.5           1               -1          1
##  4 Cori           1.38           0             0               -1          1
##  5 Cori           1.43           0             0               -1          1
##  6 Cori           1.09           0             0                1          1
##  7 Cori           2.10           1             0.5              1          1
##  8 Cori           1.70           0.5           0                1          1
##  9 Cori           1.51           0             0                1          1
## 10 Cori           1.73           0.5           0.25             1          1
## # i 5,071 more rows
```

```
pca_data <- model_data %>%
  select(avg_spks, contrast_left, contrast_right, feedback_type, session_ID)
pca_data <- scale(pca_data)
```

```r
pca_result <- prcomp(pca_data, center = TRUE, scale. = TRUE)

pca_scores <- as.data.frame(pca_result$x)
pcamodel_data <- cbind(model_data, pca_scores[, 1:2])  # Keep first two PCs

ggplot(pcamodel_data, aes(x = PC1, y = PC2, color = mouse_name)) +
  geom_point() +
  labs(title = "PCA: PC1 vs. PC2", x = "PC1", y = "PC2") +
  theme_minimal()
```

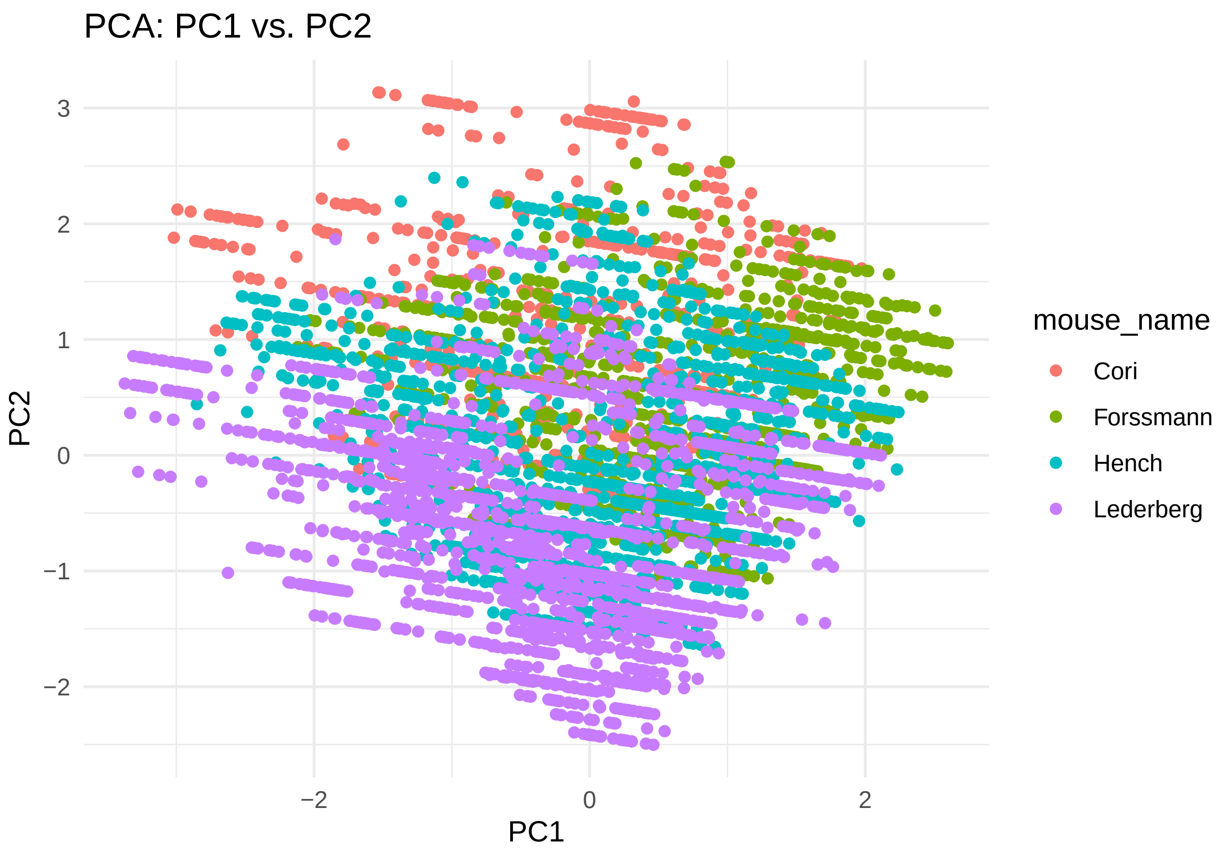
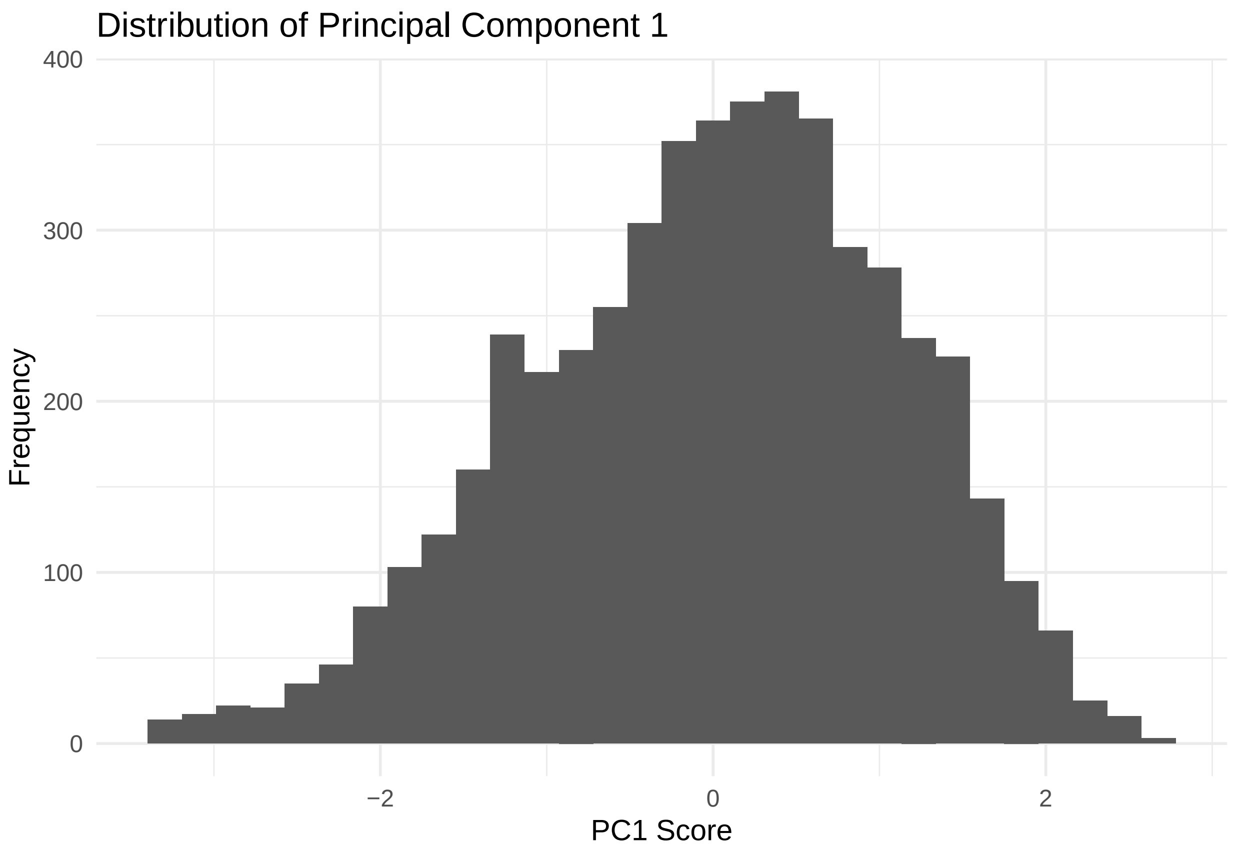


Figure 7.0 - Principal Component Analysis (PCA) plot along the first two principal components

The PCA plot reveals group-specific trends among the four mouse groups—Lederberg, Hench, Forssmann, and Cori—each showing unique distributions along the principal components. Lederberg (purple) exhibits the widest spread along PC1 and extends into the lower PC2 range. This suggests variation within this group and characteristics captured by PC2. Hench (blue) and Forssmann (green) is spread across both principal components, overlapping with multiple groups, showing a less distinct clustering pattern. Meanwhile, Cori (red) is primarily positioned in the upper section, suggesting that this group possesses more distinct features that contribute positively to PC2. The significant overlap among the four mouse groups in the PCA plot suggests that the features used for dimension reduction share common variance across groups. This implies that while some group-specific patterns exist, there is no strict separation between the groups, meaning they may have similar underlying characteristics and the measured variables do not fully distinguish the groups.

```r
PC1 <- pca_result$x[, 1]

ggplot(model_data, aes(x = PC1)) +
  geom_histogram(bins = 30) +
  labs(title = "Distribution of Principal Component 1",
       x = "PC1 Score",
       y = "Frequency") +
  theme_minimal()
```



Predictive Modeling

```
set.seed(123) # for reproducibility

train_index <- createDataPartition(model_data$feedback_type, p = .8,
                                   list = FALSE,
                                   times = 1)
train_data <- model_data[train_index, ]
test_data <- model_data[-train_index, ]

train_matrix <- as.matrix(train_data %>% select(contrast_left, contrast_right, avg_spks, session_ID))
test_matrix <- as.matrix(test_data %>% select(contrast_left, contrast_right, avg_spks, session_ID))

train_labels <- ifelse(train_data$feedback_type == -1, 0, 1)
test_labels <- ifelse(test_data$feedback_type == -1, 0, 1)

xgb_model <- xgboost(data = train_matrix, label = train_labels,
                     objective = "binary:logistic", nrounds = 10)
```

```
## [1]  train-logloss:0.621927
## [2]  train-logloss:0.578853
## [3]  train-logloss:0.550712
## [4]  train-logloss:0.535591
## [5]  train-logloss:0.514585
## [6]  train-logloss:0.502697
## [7]  train-logloss:0.495982
## [8]  train-logloss:0.491304
## [9]  train-logloss:0.484158
## [10] train-logloss:0.474202
```

Prediction Performance

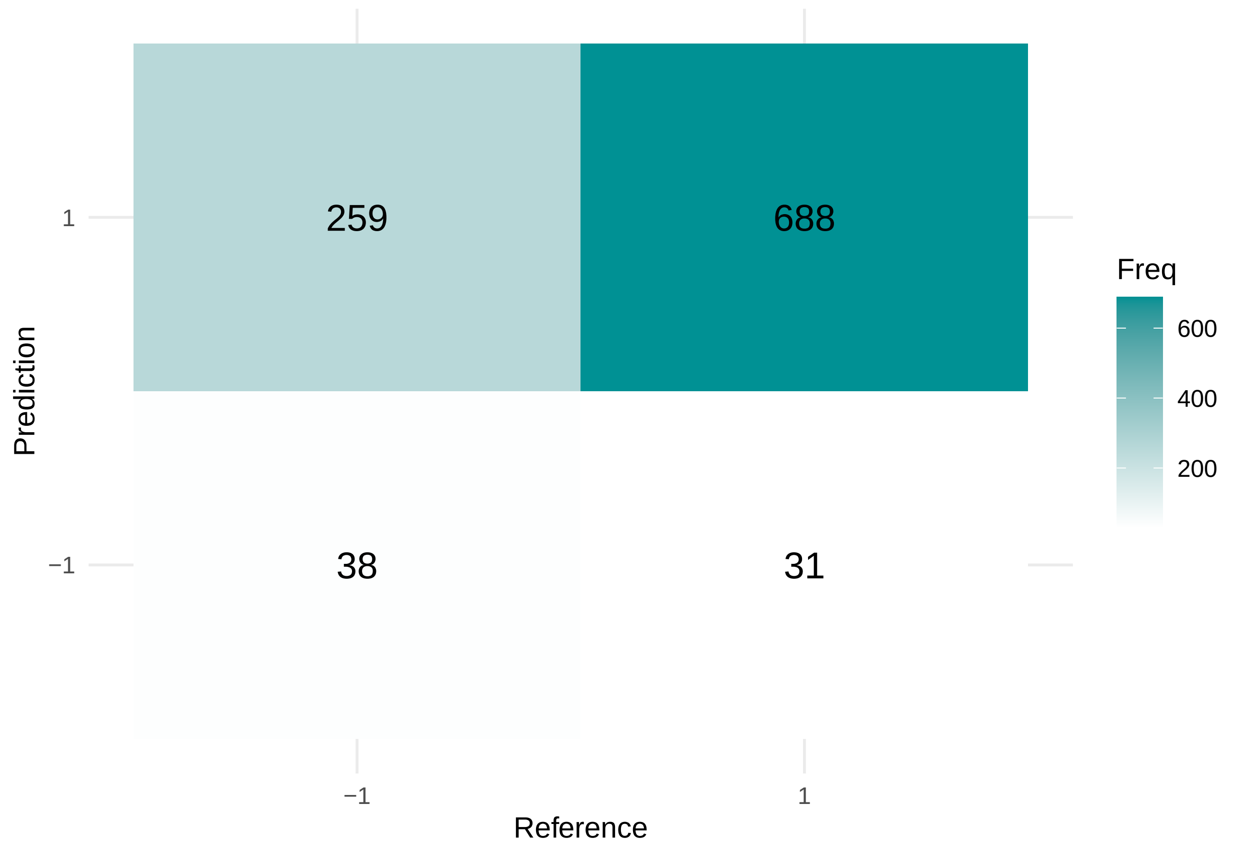
```r
test_pred <- predict(xgb_model, test_matrix)
predicted_labels <- as.numeric(ifelse(test_pred > 0.5, 1, 0))

accuracy <- mean(predicted_labels == test_labels)
accuracy
```

```
## [1] 0.7145669
```

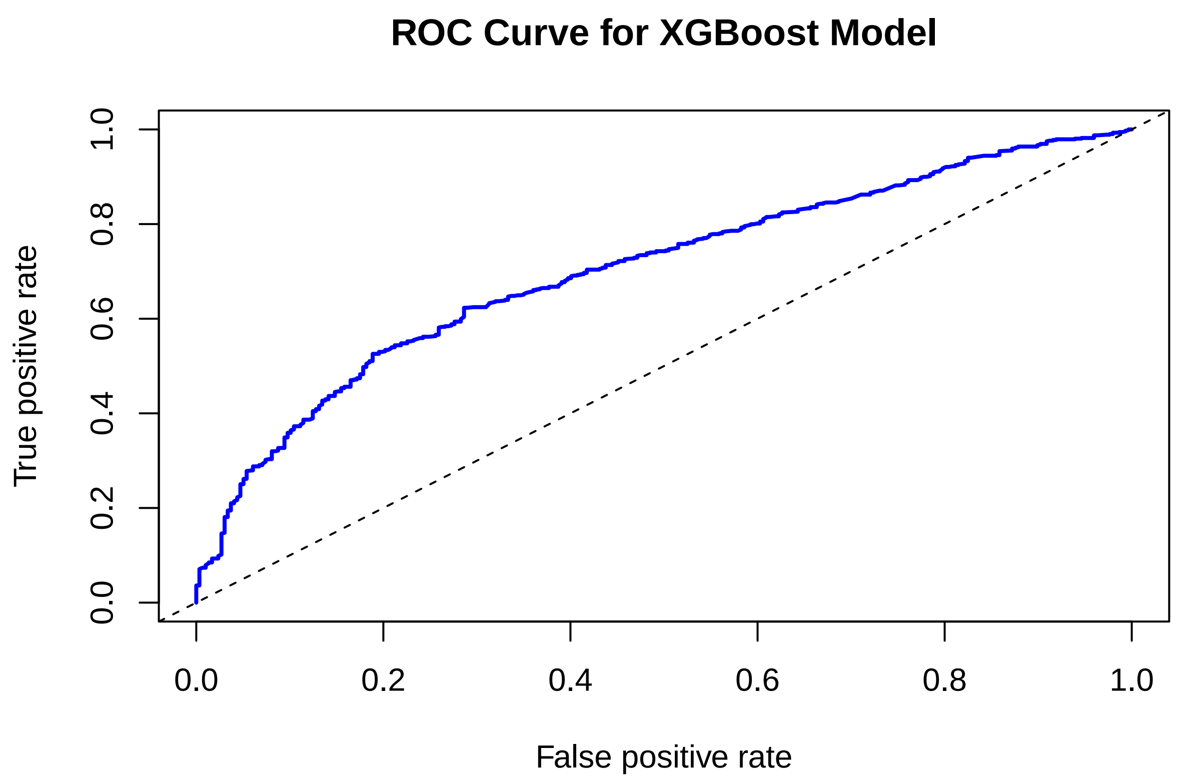
```r
conf_matrix <- confusionMatrix(as.factor(predicted_labels), as.factor(test_labels))
cm_table <- as.data.frame(conf_matrix$table)

# Plot the confusion matrix
ggplot(cm_table, aes(Reference, Prediction, fill = Freq)) +
  geom_tile() +
  geom_text(aes(label = Freq), color = "black", size = 5) +
  scale_fill_gradient(low = "white", high = "#009194") +
  labs(x = "Reference", y = "Prediction") +
  scale_x_discrete(labels = c("-1", "1")) +
  scale_y_discrete(labels = c("-1", "1")) +
  theme_minimal()
```



```r
test_pred <- predict(xgb_model, newdata = test_matrix)

pred <- prediction(test_pred, test_labels)
perf <- performance(pred, "tpr", "fpr")
plot(perf, col = "blue", lwd = 2, main = "ROC Curve for XGBoost Model")
abline(a = 0, b = 1, lty = 2)
```



```r
auroc <- roc(test_labels, test_pred)
```

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

## Setting direction: controls < cases
```

```r
auroc
```

```
##
## Call:
## roc.default(response = test_labels, predictor = test_pred)
##
## Data: test_pred in 297 controls (test_labels 0) < 719 cases (test_labels 1).
## Area under the curve: 0.7024
```

Discussion

The predictive model built using the XGBoost algorithm demonstrated a moderate level of performance in predicting trial outcomes based on neural activity and visual stimuli. The model achieved an accuracy of approximately 71.46%, indicating that it was able to correctly classify the feedback type (success or failure) in about 71% of the cases. Additionally, the area under the receiver operating characteristic curve (AUROC) was 0.7024, suggesting that the model's ability to distinguish between the two classes (success and failure) is reasonably good, though there is still room for improvement.

These results highlight the potential of using neural activity data, specifically spike trains, in conjunction with stimuli information to predict behavioral outcomes. The model's performance suggests that there are patterns in the neural responses that correlate with the success or failure of decisions in visual tasks. However, further refinement of the model could lead to improved accuracy and better predictive power.