

## **Assignment 1: Neuroscience of Decision Making PSY 307 (Monsoon 2024)**

**Name:** Anjaneya Sharma

**Roll Number:** 2021449

---

**Instructions:** Please write your own responses and do not copy or lift text/code from any source (including the paper). If you are referring to credible external sources other than the attached paper for your answers, please cite those sources (within the body of text and provide a reference list at the end) in the APA citation format (<https://www.mendeley.com/guides/apa-citation-guide>). Word limits given are indicative and less than the indicated numbers may also be used.

**Please download this MS word question-cum-response template to TYPE your answers and feel free to add sheets as required. Convert this document to a PDF and rename the file: name\_RollNo. before submitting. Please note that answers in this template only will be evaluated and hand-written or scanned answer sheets will not be evaluated.**

**[Strict deadline for submission: 21 September, Saturday, 11:00 PM]**

---

1. Fill out the google form: <https://forms.gle/a23if8Kbdjs2gRfJ8>
2. An experimenter recorded a SINGLE cortical neuron's activity from two brain regions- H & A- processing visual information as the organism viewed three distinct visual stimuli on the screen. Each stimulus was presented and its neuronal activity was recorded for 30 times. The collected datasets are attached herewith: 'dataset\_H.mat' and 'dataset\_A.mat'.

dataset\_H.mat: Neuronal activity recorded from region H.

dataset\_A.mat: Neuronal activity recorded from region A.

Each dataset consists of 3 columns/cells:

Column 1: Timestamps of action potentials for the face stimulus.

Column 2: Timestamps of action potentials for the text stimulus.

Column 3: Timestamps of action potentials for the speech stimulus.

Each column contains 30 trials and the array size for each trial is different. The timestamps are associated to the occurrence of each action potential, measured in milliseconds from stimulus onset. For example: In the dataset\_A, the first trial has 17 timestamps meaning, action potential/spike was recorded at these 17 timestamps. A negative timestamp (e.g.: -900) means action potential/spike that occurred 900 ms before stimulus onset time (0 ms). Similarly, a positive timestamp (e.g.: 300) indicates spikes that occurred after stimulus onset (0 ms).

**Total neuronal recording duration = 3000 milliseconds; Time = 0 millisecond (stimulus onset); Time = 1000 milliseconds (stimulus offset)**

**Now solve the following. Insert a figure (wherever required) and paste the MATLAB/Python/R code for the same. Any figure must provide all information necessary to interpret it including axes labels, captions/legends (see Fig.3 of the attached paper for a sample; simple figure titles as captions are not enough).**

**[ link for importing MATLAB data arrays into Python and R]**

- <https://www.askpython.com/python/examples/mat-files-in-python>

Note to the reader: I wanted to explore the dataset and understand more about it, hence i used a jupyter notebook, hence kindly ignore the comments in between indicating the cell numbers.

Also, I wrote some code to check out the dataset before starting with the exercises/questions. That code is given below.

```
get_ipython().system('pip install scipy')
```

```
# In[348]:
```

```
import pandas as pd
from scipy.io import loadmat
import numpy as np
import matplotlib.pyplot as plt
```

```
plt.style.use('default')
```

```
# In[349]:
```

```
# info for myself, c1 : face , c2: text , c3: speech
# each contain info about timestamps
stimuli = ['Face', 'Text', 'Speech']
```

```
# In[350]:
```

```
annots_H = loadmat('dataset_H.mat',squeeze_me=True) # used squeeze me to reduce the extra
dimension added by matlab
annots_A = loadmat('dataset_A.mat', squeeze_me=True) # resulting ds is flattened now
```

```
# In[351]:
```

```
type(annots_H)
```

```
# In[352]:
```

```
annots_H.keys()
```

```
# In[353]:
```

```
dfH = annots_H['dataset_H']  
dfA = annots_A['dataset_A']
```

```
# In[354]:
```

```
print(type(dfH))  
print(len(dfH))
```

```
# In[355]:
```

```
dfH
```

```
# In[356]:
```

```
dfH.shape
```

```
# In[357]:
```

```
dfH[2].shape # number of trials
```

```
# In[358]:
```

```
dfH[1][0].shape # number of spikes
```

```
# In[359]:
```

```
#creating a data stimuli dictionary
```

```
data_H_dict = {'Face': dfH[0], 'Text': dfH[1], 'Speech': dfH[2]}
```

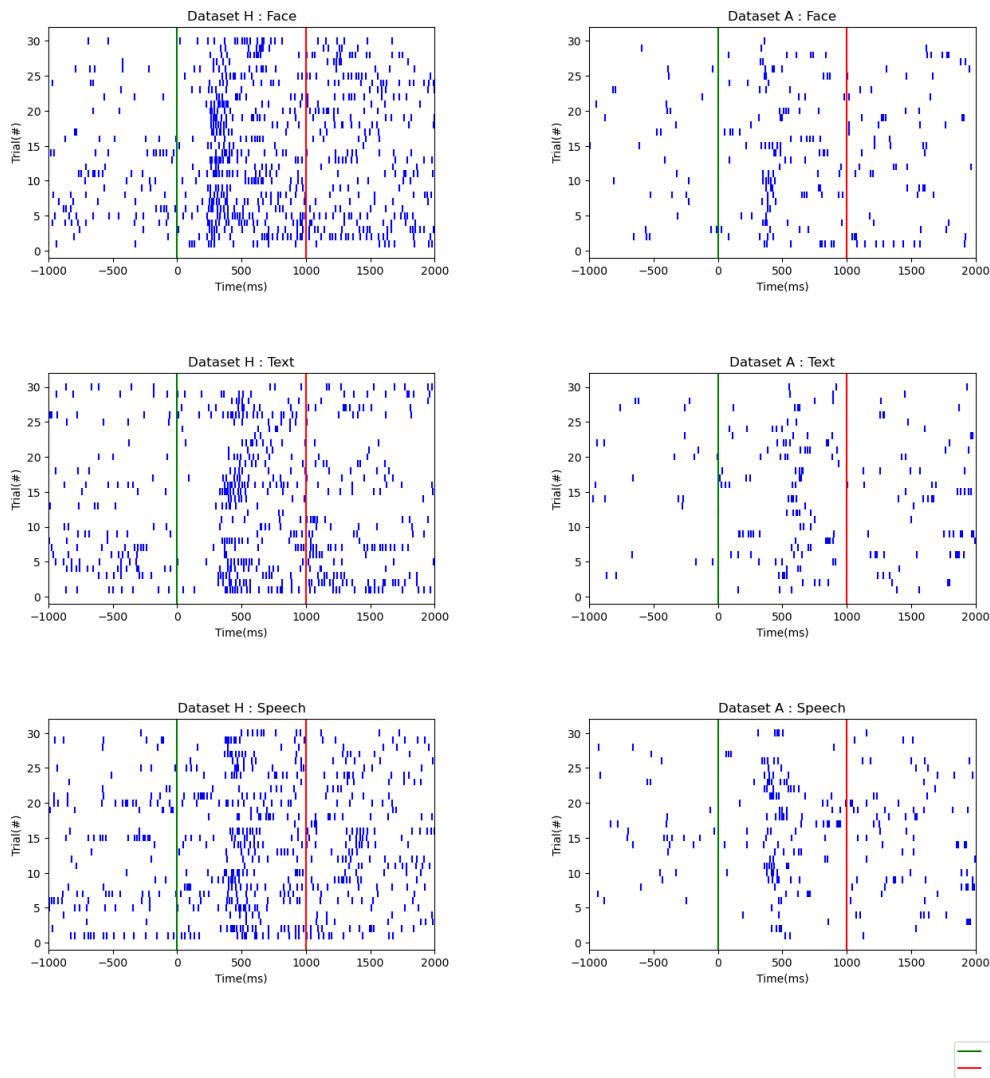
```
data_A_dict = {'Face': dfA[0], 'Text': dfA[1], 'Speech': dfA[2]}
```

```
# In[ ]:
```

- A) Create a Raster plot of the two neurons recorded from two distinct brain regions for ALL SIX of the stimuli (2 X face; 2 X text; 2 X speech) and mark the onset of stimulus onset and offset of the stimulus by a vertical green and red line respectively on the same individual subplots. Mark the spikes (action potentials) with blue colour. [5 marks]**

**Hint: Create a larger figure with six subplots (positioned as 3 rows x 2 columns). Plot dataset\_H in the first column and dataset\_A in the second column of the larger figure. Indicate the stimulus type - face, text, speech on top of each subplot as subplot title**

Resulting Plots are shown below:-



Code for the following is given below:-

##(in continuation with the note before) moreover, since this was the first part, i also created plots for each of the datasets (hence the extra code)

### Part A (Raster Plots)      # represents Bold text in jupyter notebook

#I've made a generic function made to avoid using the same set of commands or running loops

def raster(spike\_times, ax, title):

    for trial\_idx, trial\_data in enumerate(spike\_times):

        ax.vlines(trial\_data, trial\_idx + 0.5, trial\_idx + 1.5, color='blue')

    ax.axvline(0, color='green', label='Onset of stimulus')

    ax.axvline(1000, color='red', label='Offset of stimulus')

#given information : stimulus onset - 0 ms ; stimulus offset - 1000ms

```
ax.set_xlim([-1000, 2000])
ax.set_title(title)
ax.set_xlabel('Time(ms)')
ax.set_ylabel('Trial(#)')
```

# In[ ]:

# In[ ]:

# In[361]:

#individual plots for dataset\_H

```
fig_H, axs_H = plt.subplots(3, 1, figsize=(8, 10))
raster(data_H[0], axs_H[0], 'Dataset H: Face')
raster(data_H[1], axs_H[1], 'Dataset H: Text')
raster(data_H[2], axs_H[2], 'Dataset H: Speech')
plt.subplots_adjust(wspace=0.4, hspace=0.5) #just increased spacing between the graphs
```

# In[362]:

# Create individual plots for dataset\_A

```
fig_A, axs_A = plt.subplots(3, 1, figsize=(10, 10))
raster(data_A[0], axs_A[0], 'Dataset A: Face')
raster(data_A[1], axs_A[1], 'Dataset A: Text')
raster(data_A[2], axs_A[2], 'Dataset A: Speech')
plt.subplots_adjust(wspace=0.4, hspace=0.5) #just increased spacing between the graphs
```

```
# In[ ]:
```

```
# In[ ]:
```

```
# In[363]:
```

```
#making the combined plot consisting of 6 plots
```

```
fig, axs = plt.subplots(3, 2, figsize=(15,15))
```

```
# axs[0, 0].get_shared_y_axes().join(axs[0, 0], axs[1, 0], axs[2, 0]) # Align y-axis for region H plots
```

```
raster(data_H[0], axs[0, 0], 'Dataset H : Face')
```

```
raster(data_H[1], axs[1, 0], 'Dataset H : Text')
```

```
raster(data_H[2], axs[2, 0], 'Dataset H : Speech')
```

```
# axs[0, 1].get_shared_y_axes().join(axs[0, 1], axs[1, 1], axs[2, 1])
```

```
raster(data_A[0], axs[0, 1], 'Dataset A : Face')
```

```
raster(data_A[1], axs[1, 1], 'Dataset A : Text')
```

```
raster(data_A[2], axs[2, 1], 'Dataset A : Speech')
```

```
handles, labels = axs[0, 0].get_legend_handles_labels()
```

```
fig.legend(handles, labels, loc='lower right')
```

```
plt.subplots_adjust(wspace=0.4, hspace=0.5) #just increased spacing between the combined graphs
```

```
plt.show()
```

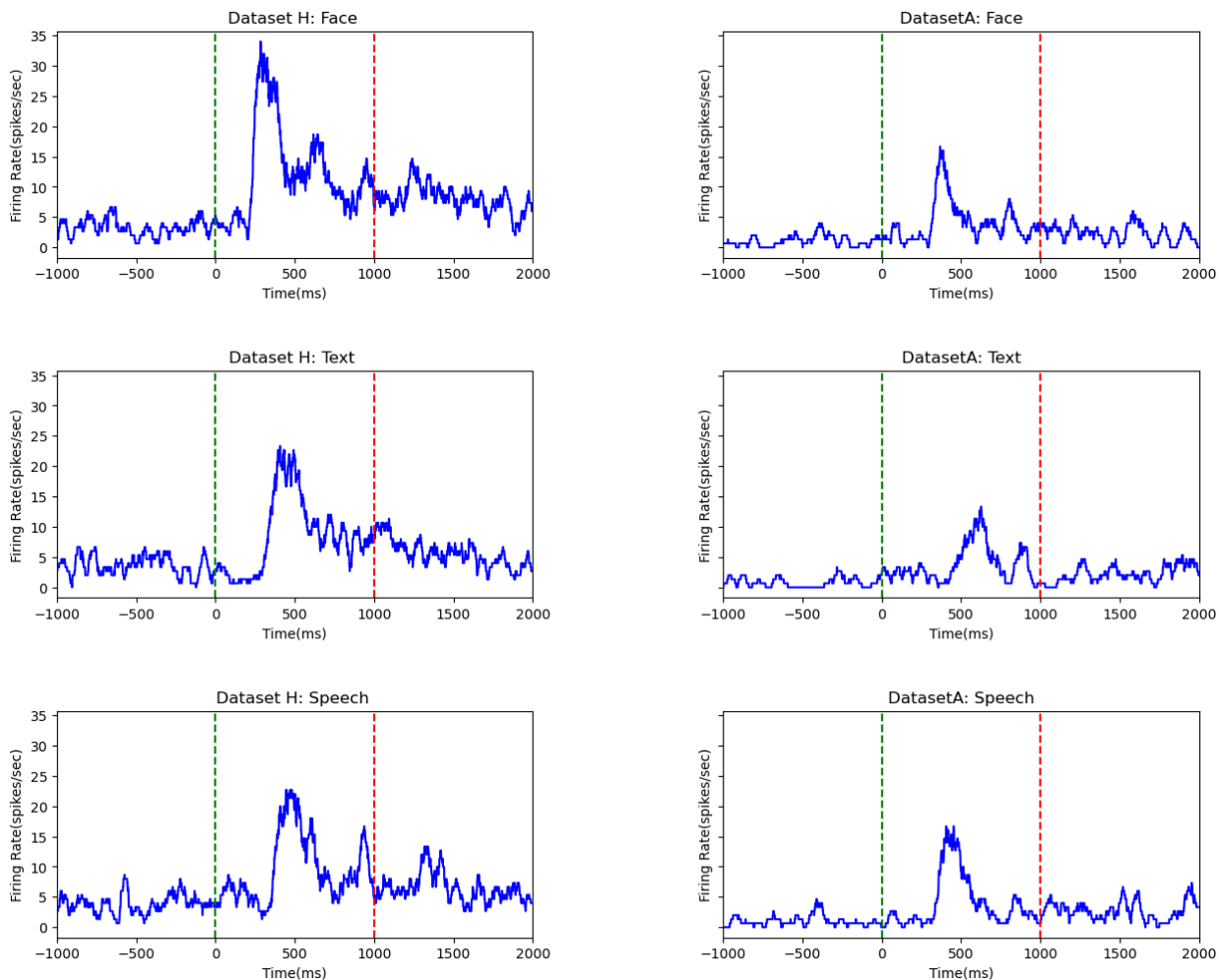
```
# In[ ]:
```

```
# In[ ]:
```

**B) Create a Peri Stimulus Time Plot of two neurons recorded from two distinct brain regions for ALL SIX of the stimuli and mark the onset of stimulus and offset of the stimulus by a vertical green and red line respectively on the same individual subplots. Before computing the histogram, smooth the data for each subplot over a time window of 50 ms. [5 marks]**

**Hint: Create a larger figure with six subplots (positioned as 3 rows x 2 columns); Smooth the data by moving average method; a line plot would suffice and depict the trend. Plot dataset\_H in the first column and dataset\_A in the second column of the larger figure. Indicate the stimulus type - person, text, speech on top of each subplot as subplot title**

Plots for the following part are given below





Code for the plots above is given below:-

```
# ## Part B (Peri stimulus Time Plots)
```

```
# In[ ]:
```

```
# In[ ]:
```

```
# In[364]:
```

```
#same approach followed as done in part A,
```

```
#i.e created a generic function for plotting the peristimulus plots, and
```

```
#later on plotting individual plots and then the combined plots
```

```
def peristimulus(spike_times_list, ax, title, bin_width=1, smoothing_window=50, add_legend=False):
```

```
    all_spikes = []
```

```
    for trial in spike_times_list:
```

```
        spike_times_trial = np.squeeze(trial)
```

```
        spike_times_trial = np.atleast_1d(spike_times_trial)
```

```
        if spike_times_trial.size == 0: #logic followed here is that i've skipped trials with no spikes
```

```

continue
all_spikes.extend(spike_times_trial)

time_range = (-1000, 2000)
bins = np.arange(time_range[0], time_range[1] + bin_width, bin_width)

counts, bin_edges = np.histogram(all_spikes, bins=bins)

firing_rate = counts / len(spike_times_list) * (1000 / bin_width)

#as mentioned in teh assignment, using moving
#average method to smoothen

window_size = int(smoothing_window / bin_width)
window = np.ones(window_size) / window_size
smoothed_rate = np.convolve(firing_rate, window, mode='same')

ax.plot(bins[:-1], smoothed_rate, color='blue')

ax.axvline(x=0, color='green', linestyle='--', label='Onset of stimulus')
ax.axvline(x=1000, color='red', linestyle='--', label='Offset of stimulus')

ax.set_xlim([-1000, 2000])
ax.set_xlabel('Time(ms)')
ax.set_ylabel('Firing Rate(spikes/sec)')

ax.set_title(title)

```

```
if add_legend:
    ax.legend()
```

```
fig, axs = plt.subplots(3, 2, figsize=(15, 12), sharex=False, sharey=True)
```

```
for row, stimulus in enumerate(stimuli):
```

```
    ax_H = axs[row, 0]
```

```
    peristimulus(data_H_dict[stimulus], ax_H, f'Dataset H: {stimulus} ', add_legend=add_legend)
```

```
    ax_A = axs[row, 1]
```

```
    peristimulus(data_A_dict[stimulus], ax_A, f'DatasetA: {stimulus}')
```

```
plt.subplots_adjust(wspace=0.4, hspace=0.5) #just increased spacing between the combined graphs
plt.show()
```

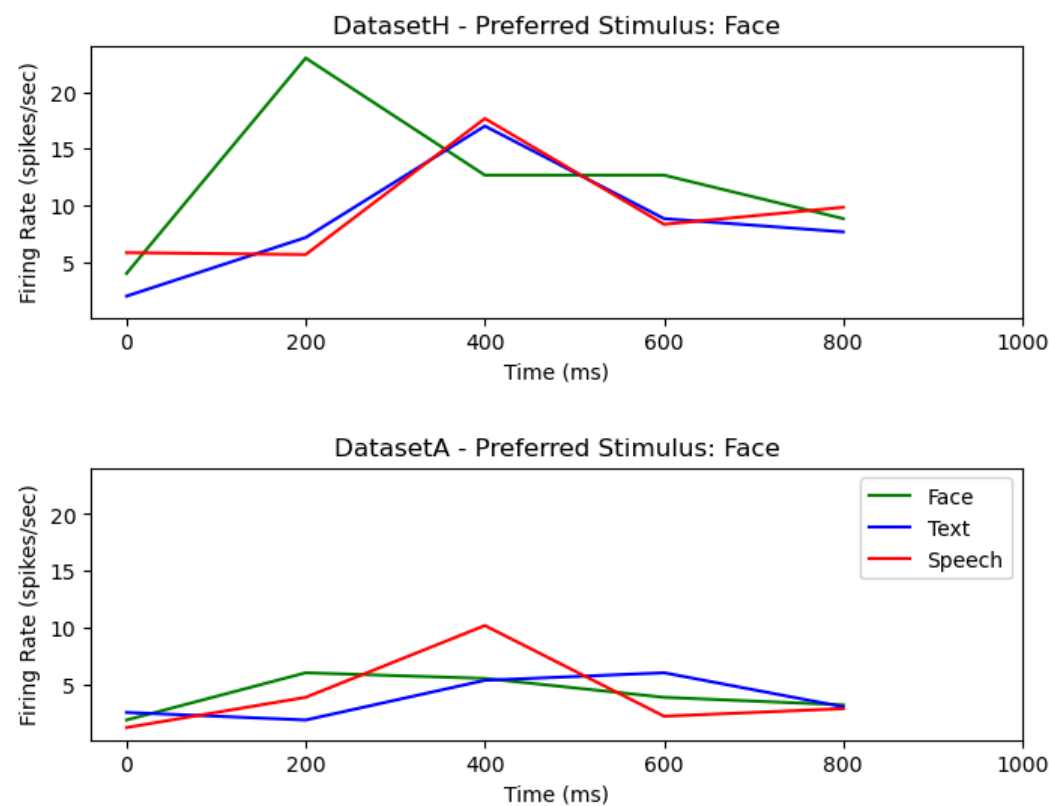
**C) Create figures representing the preferred stimulus type of two neurons (dataset\_H & dataset\_A) recorded from two distinct brain regions from the average firing rate of the neuron (between 0 – 1000 ms and all trials) for each stimulus type using a binwidth of 200 ms. Plot the curves of each stimulus type recorded from the same neuron in one graph. There will be two separate graphs for the two neurons/datasets. Mark the curves for stimulus type -person in red colour, text in green colour, and speech in blue colour. Computationally calculate the 'most preferred stimulus type' for each neuron and report it on the title of the plot. Also, report the order of stimulus preference for each neuron. [5+2+1 marks]**

**What kind of neuronal information coding is represented by these neurons and why? [2 marks]**

**Hint: Create a larger figure with two subplots (positioned as 1 row x 2 columns).**

Plots for the two neurons/datasets are given below:-

Face (12.233 (spikes/second)), Speech (9.467 (spikes/second)), Text (8.533 (spikes/second))  
Face (4.067 (spikes/second)), Speech (4.033 (spikes/second)), Text (3.733 (spikes/second))



hence we can determine from the statistics that dataset H prefers Face and dataset A prefers Face stimulus

Code for the plots and calculation above is given below:-

### Part C (Preferred stimulus type of 2 neurons)

In[407]:

```
def compute_average_firing_rate(spike_times_list, bins):  
    trial_firing_rates = []
```

```

for trial in spike_times_list:
    spike_times_trial = np.squeeze(trial)
    spike_times_trial = np.atleast_1d(spike_times_trial)

    spikes_in_window = spike_times_trial[(spike_times_trial >= 0) & (spike_times_trial <= 1000)]
    counts, _ = np.histogram(spikes_in_window, bins=bins)

    firing_rate = counts * (1000 / (bins[1] - bins[0]))
    trial_firing_rates.append(firing_rate)

    mean_firing_rate = np.mean(trial_firing_rates, axis=0)
    return mean_firing_rate

```

```

bin_width = 200#(ms)
bins = np.arange(0, 1000 + bin_width, bin_width)

fig, axs = plt.subplots(2, 1, figsize=(8, 6), sharey=True)

```

```

preferred_stimuli = {}
op = [] # for printing the final answers below
printdictH = {}
printdictA = {}
for idx, (data_dict, region) in enumerate(zip([data_H_dict, data_A_dict], ['H', 'A'])):
    ax = axs[idx]
    mean_firing_rates = {}

    for stimulus in stimuli:
        #taking the average over all bins(in the code beloww)
        spike_times_list = data_dict[stimulus]

```

```
mean_firing_rate = compute_average_firing_rate(spike_times_list, bins)
mean_firing_rates[stimulus] = np.mean(mean_firing_rate)
```

```
ax.plot(bins[:-1], mean_firing_rate, color=colors[stimulus], label=stimulus)
```

```
sorted_stimuli = sorted(mean_firing_rates.items(), key=lambda x: x[1], reverse=True) #finding
the stimulus preferred in each case
```

```
preferred_stimulus = sorted_stimuli[0][0]
```

```
preferred_stimuli[region] = (preferred_stimulus, sorted_stimuli)
```

```
ax.set_title(f'Dataset{region} - Preferred Stimulus: {preferred_stimulus}')
```

```
ax.set_xlabel('Time (ms)')
```

```
ax.set_ylabel('Firing Rate (spikes/sec)')
```

```
ax.set_xticks(bins)
```

```
order = ', '.join([f'{stimulus} ({round(rate,3)} (spikes/second))' for stimulus, rate in
sorted_stimuli])
```

```
# print(f'{stimulus}: {round(rate,3)}')
```

```
first_word = order.split()[0]
```

```
op.append(first_word)
```

```
print(order)
```

```
plt.subplots_adjust(wspace=0.4, hspace=.55) #just increased spacing between the combined graphs
```

```
ax.legend()
```

```
plt.show()
```

```
print(f'Hence we can determine from the statistics that dataset H prefers {op[0]} and dataset A prefers {op[1]} stimulus')
```

```
# In[ ]:
```

```
# In[ ]:
```

```
# In[ ]:
```

```
# In[ ]:
```

```
# In[ ]:
```

```
# In[ ]:
```

The type of coding represented by these neurons is rate coding.

We know that the preferred stimulus type is not determined by firing rate, but by the average firing rate, and higher it is, relatively more strong is the response of the neuron to the stimuli, thereby showing the response through the rate of firing of neurons.