Course Project: Consulting for Milestone I

```r
suppressWarnings(library(tidyverse))
suppressWarnings(library(knitr))
suppressWarnings(library(dplyr))
```

Common quesions

1. **Are there any specific plots you want us to include in the reports?**

No. Your report, your story! The project description offers some thoughts on what kinds of information to include, (i) describe the data structures across sessions (e.g., number of neurons, number of trials, stimuli conditions, feedback types), (ii) explore the neural activities during each trial, (iii) explore the changes across trials, and (iv) explore homogeneity and heterogeneity across sessions and mice. But you have the freedom to choose what plots to draw.

2. **Do we need to study neuroscience to work on this project?**

No. The course project is playground for you to experiment with big and complex data set. We do not expect you to know neuroscience at all. However, you may find that picking up some backgrounds of the data might help, as is the case for real-world statistics/data science problem.

3. **The data set is too complicated that I don't know where to start to turn it into a data frame.**

For almost every real data set, we first need to understand the data set (its structure, questions of interest, key features, etc.) before formatting them into a standard format (e.g., a data frame in R). It might be misleading since, in many classrooms, you will be given a clean, toy data set. As you will see below, the "standard format" might vary depending on what you want to achieve.

Data structure

We will investigate the structure of the data at three levels: the full dataset, one session, and one trial. When exploring the data set on your own, you may want to take the bottom-up approach, which means to examine and understand the data from one trial before scaling it up to sessions.

Dataset

A total of 18 RDS files are provided that contain the experiment records from 18 sessions on 5 mice. We can begin our analysis by summarizing the information from the sessions.

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

From the project description or using `names()`, we can find that there are 8 variables in each session. We can report mouse_name, date_exp, number of brain_area, number of neurons, number of trials, success rate for each session (6 in total).

You may (correctly) wonder how to know what features to summarize and report. Keep in mind that this is a data analysis project but not an exam. It means that you need to iterate between exploration and analysis.

```
# Summarize the information across sessions:


# Knowing what summary we want to report, we can create a tibble:
# All values in this function serve only as place holders

n.session=length(session)

# in library tidyverse
meta <- tibble(
  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]];
  meta[i,1]=tmp$mouse_name;
  meta[i,2]=tmp$date_exp;
  meta[i,3]=length(unique(tmp$brain_area));
  meta[i,4]=dim(tmp$spks[[1]])[1];
  meta[i,5]=length(tmp$feedback_type);
  meta[i,6]=mean(tmp$feedback_type+1)/2;
  }
```

We can print the table in R markdown using `kable()` in the `knitr` package.

```
# In package knitr
kable(meta, format = "html", table.attr = "class='table table-striped'",digits=2)
```

mouse_name	date_exp	n_brain_area	n_neurons	n_trials	success_rate
Cori	2016-12-14	8	734	114	0.61
Cori	2016-12-17	5	1070	251	0.63
Cori	2016-12-18	11	619	228	0.66
Forssmann	2017-11-01	11	1769	249	0.67

mouse_name	date_exp	n_brain_area	n_neurons	n_trials	success_rate
Forssmann	2017-11-02	10	1077	254	0.66
Forssmann	2017-11-04	5	1169	290	0.74
Forssmann	2017-11-05	8	584	252	0.67
Hench	2017-06-15	15	1157	250	0.64
Hench	2017-06-16	12	788	372	0.69
Hench	2017-06-17	13	1172	447	0.62
Hench	2017-06-18	6	857	342	0.80
Lederberg	2017-12-05	12	698	340	0.74
Lederberg	2017-12-06	15	983	300	0.80
Lederberg	2017-12-07	10	756	268	0.69
Lederberg	2017-12-08	8	743	404	0.76
Lederberg	2017-12-09	6	474	280	0.72
Lederberg	2017-12-10	6	565	224	0.83
Lederberg	2017-12-11	10	1090	216	0.81

```
# 1. You may want to change the column names of the table to avoid words like n_brain_area
# 2. You may want to add a footnote and title of any tables in your report
```

You can add and remove columns in your report. What we do not want to see is that you copy the above code, or any code, but have no clues why to use them in your report. To be clear: using code from course notes, discussion, or consulting session will not be counted as plagiarism. However, including results you cannot explain in your report will result in low grades.

Session

We can pick one of the sessions to understand its structure. You can pick any sessions with the exceptions of Sessions 1 and 18, where 100 trials were randomly removed (see project description). Let’s take a look at Session 2.

The 1070 neurons in Session 2 are located in CA1, VISl, root, VISpm, POST of the mouse brain. We can visualize the activities of these areas across the 251 trials.

One question that you will almost certain run into is how to define “activities”. This is an open-ended question for you to craft your own answers — you can define multiple summary statistics as different activities of neurons!

Here we take the average number of spikes across neurons in each area as the activities. (Again, we do not want to see this summary in your report if you do not have reasons for why to use it.)

We can start our analysis on one trial.

```r
i.s=2 # indicator for this session

i.t=1 # indicator for this trial

spk.trial = session[[i.s]]$spks[[i.t]]
area=session[[i.s]]$brain_area

# We need to first calculate the number of spikes for each neuron during this trial
spk.count=apply(spk.trial,1,sum)

# for(i in 1:dim(spk.trial)[1]){
#  spk.count[i]=sum(spk.trial[i,])
# }

# Next we take the average of spikes across neurons that live in the same area

# You can use tapply() or group_by() in dplyr

# tapply():
spk.average.tapply=tapply(spk.count, area, mean)


# dplyr:
# To use dplyr you need to create a data frame
tmp <- data.frame(
  area = area,
  spikes = spk.count
)
# Calculate the average by group using dplyr
spk.average.dplyr =tmp %>%
  group_by(area) %>%
  summarize(mean= mean(spikes))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

You may want to turn the code above into a function in order to apply it across trials

```
# Wrapping up the function:

average_spike_area<-function(i.t,this_session){
  spk.trial = this_session$spks[[i.t]]
  area= this_session$brain_area
  spk.count=apply(spk.trial,1,sum)
  spk.average.tapply=tapply(spk.count, area, mean)
  return(spk.average.tapply)
  }

# Test the function
average_spike_area(1,this_session = session[[i.s]])
```

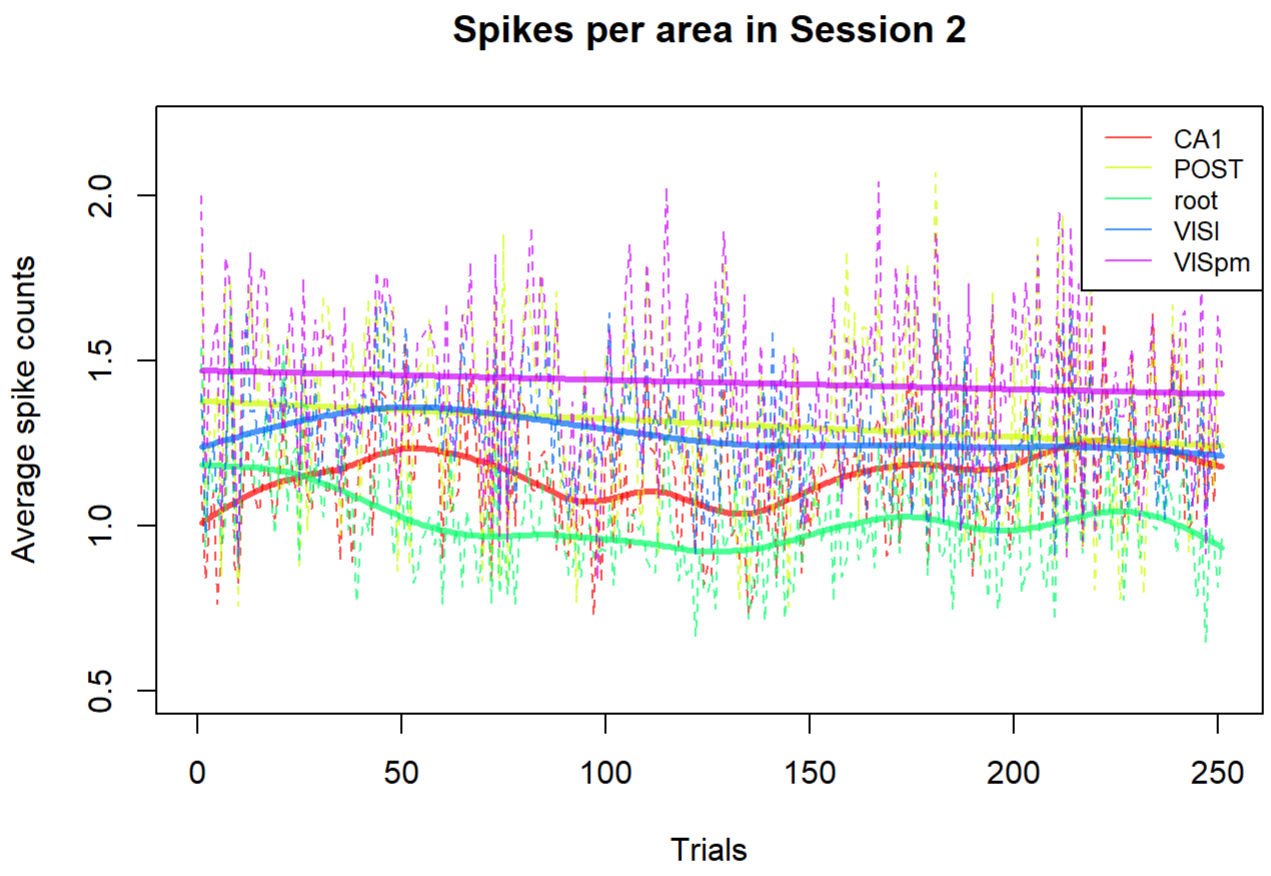
```
##      CA1     POST     root     VISl    VISpm
## 1.121053 1.816754 1.538462 1.398268 2.000000
```

Now we are ready to apply the function across all trials and create a data frame for downstream analysis, including visualization.

```
n.trial=length(session[[i.s]]$feedback_type)
n.area=length(unique(session[[i.s]]$brain_area ))
# Alternatively, you can extract these information in the meta that we created before.

# We will create a data frame that contain the average spike counts for each area, feedback typ
e,  the two contrasts, and the trial id

trial.summary =matrix(nrow=n.trial,ncol= n.area+1+2+1)
for(i.t in 1:n.trial){
  trial.summary[i.t,]=c(average_spike_area(i.t,this_session = session[[i.s]]),
                          session[[i.s]]$feedback_type[i.t],
                        session[[i.s]]$contrast_left[i.t],
                        session[[i.s]]$contrast_right[i.s],
                        i.t)
}

colnames(trial.summary)=c(names(average_spike_area(i.t,this_session = session[[i.s]])), 'feedbac
k', 'left contr.','right contr.','id' )

# Turning it into a data frame
trial.summary <- as_tibble(trial.summary)
```

Once a standard data frame is created, we can use `ggplot2` to create the visualization. If data is not well-formatted, you may want to use `plot` in base R for quick exploration. We will use base R here.

```r
area.col=rainbow(n=n.area,alpha=0.7)
# In base R, I usually initiate a blank plot before drawing anything on it
plot(x=1,y=0, col='white',xlim=c(0,n.trial),ylim=c(0.5,2.2), xlab="Trials",ylab="Average spike c
ounts", main=paste("Spikes per area in Session", i.s))


for(i in 1:n.area){
  lines(y=trial.summary[[i]],x=trial.summary$id,col=area.col[i],lty=2,lwd=1)
  lines(smooth.spline(trial.summary$id, trial.summary[[i]]),col=area.col[i],lwd=3)
  }
legend("topright",
  legend = colnames(trial.summary)[1:n.area],
  col = area.col,
  lty = 1,
  cex = 0.8
)
```



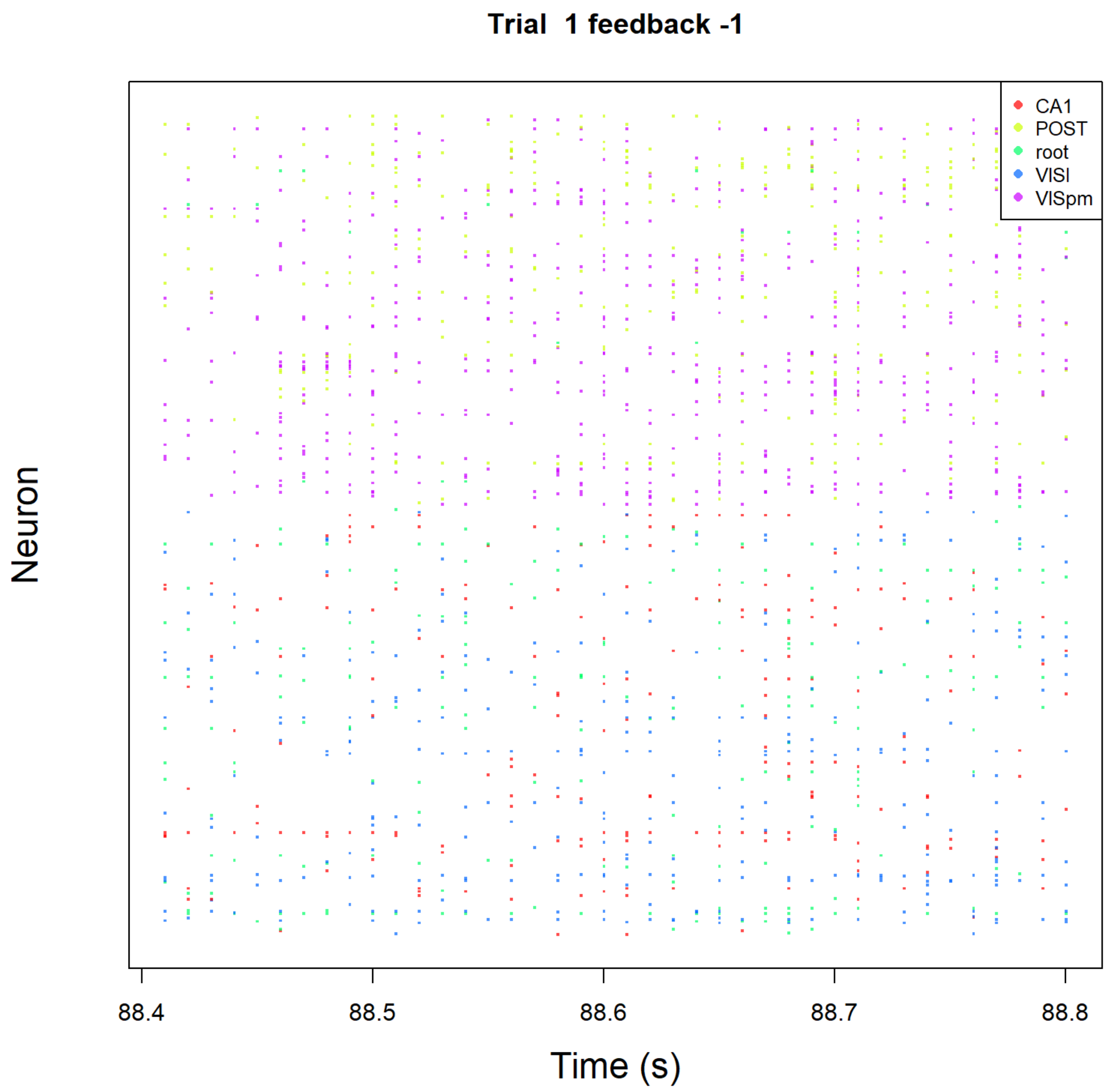
What do you learn from this plot? How do you want to improve this plot (e.g, including the feedback information)? In your report, you need to add a caption for each plot along with some explanation. Again, do not include any results that you do not plan to explain or you do not understand.

Trial

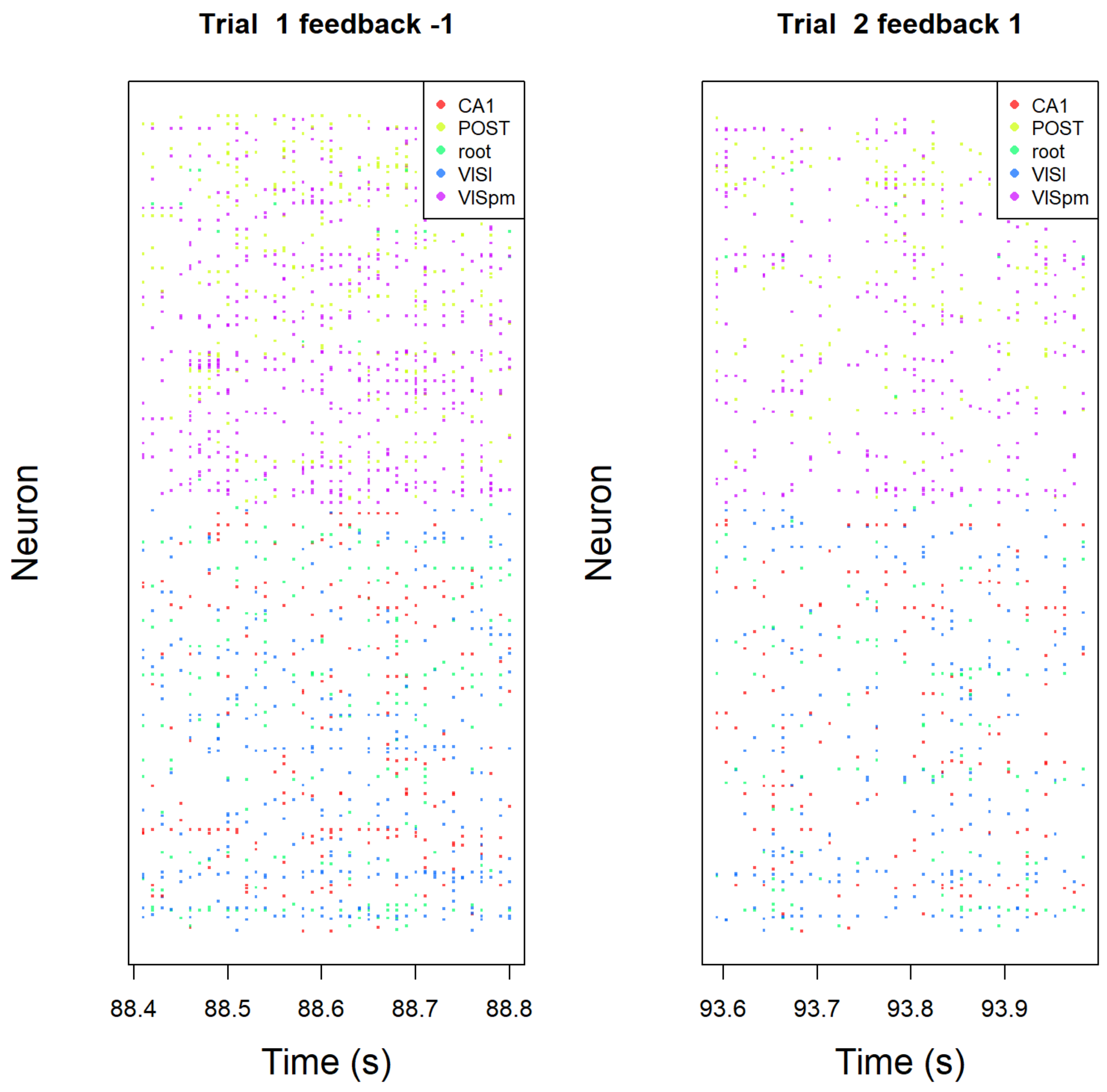
At the trial level, we can visualize the activities of all neurons in this session.

```
plot.trial<-function(i.t,area, area.col,this_session){

    spks=this_session$spks[[i.t]];
    n.neuron=dim(spks)[1]
    time.points=this_session$time[[i.t]]

    plot(0,0,xlim=c(min(time.points),max(time.points)),ylim=c(0,n.neuron+1),col='white', xlab='T
ime (s)',yaxt='n', ylab='Neuron', main=paste('Trial ',i.t, 'feedback', this_session$feedback_typ
e[i.t] ),cex.lab=1.5)
    for(i in 1:n.neuron){
        i.a=which(area== this_session$brain_area[i]);
        col.this=area.col[i.a]

        ids.spike=which(spks[i,]>0) # find out when there are spikes
        if( length(ids.spike)>0 ){
            points(x=time.points[ids.spike],y=rep(i, length(ids.spike) ),pch='.',cex=2, col=col.
this)
        }


    }

legend("topright",
  legend = area,
  col = area.col,
  pch = 16,
  cex = 0.8
  )
  }
```

```
varname=names(trial.summary);
area=varname[1:(length(varname)-4)]
plot.trial(1,area, area.col,session[[i.s]])
```



```
varname=names(trial.summary);
area=varname[1:(length(varname)-4)]
par(mfrow=c(1,2))
plot.trial(1,area, area.col,session[[i.s]])
plot.trial(2,area, area.col,session[[i.s]])
```



```
par(mfrow=c(1,1))
```

What do you learn from this plot? How do you want to improve this plot (e.g, including the feedback information)?

Next steps

We have gone over the data structure once. What do you think we should do next?

Here are some thoughts:

- Summary of insights: what do you find interesting/important/intriguing from your exploration?

- Visualization: Can you summarize your findings at the trial-level into a few numbers in order to visualize their patterns across trials? Likewise, is it possible to visualize the patterns across sessions?
- How to explore the difference across sessions?
- Challenges to overcome: What are the main challenges for Milestone II and the final predictive model? How do you plan to solve them and what methods do you want to use?
- Report writing: how to put down all your thoughts and discovery in a report?