This is an R package for analyzing the fixation pattern of eye movement data. The corresponding method is called "Spatio-temporal Fixation Pattern Analysis" (FPA; see references for details). This package provides R implementation which greatly simplifies the FPA analysis; it includes several major functions and some sample data (see "*fpa manual.pdf* " for details).

Installation

This package uses some functions from packages of "reshape" and "fields"; so before installing this package, please make sure you have installed the other two. If not, you may start install them and their supporting packages:

```
install.packages("reshape")
install.packages("fields")
install.packages("spam")
install.packages("maps")
```

Then you may download the 'fpa' package on GitHub (https://github.com/LawrenceCao/fpa) to your working directory, and then use the code:

```
install.packages("devtools")
library(devtools)
install("fpa")
```

to install it. Or directly use the code:

```
install_github("LawrenceCao/fpa/fpa", dependencies=TRUE)
```

Preprocessing

The package provides some sample data to work with. Let us take a look at the raw data first:

```
library(fpa)
data(rawdata)
head(rawdata)
```

and R returns:

```
  List Subject Condition Item Region Fix_Start Fix_End
1    1       1         1    3      1      1341    1481
2    1       1         1    3      1      1503    1744
3    1       1         1    3      3      1913    2034
4    1       1         1    3      3      2070    2483
5    1       1         1    3      4      2521    2839
6    1       1         1    3      5      2864    3104
```

The variables List, Subject, Condition and Item are commonly used variables in an experiment. If you do not divide your materials into lists, just use “1” for List in every row. Region is the id of areas defined by researchers (may be a word in a sentence, or an area in a picture). Fix_Start and Fix_End are the start and end time of each fixation, which can be easily obtained from eye-tracker software. Therefore, this is fixation time data.

The ft2fp function is used to convert fixation time data to fixation probabilities, which is the basic index in FPA analysis. Please use exact same number and names of variables (e.g., “List” rather than “list”) when you prepare your own raw data; otherwise, errors may occur for returned data frame.

Here is an example of using ft2fp function:

```
newdata <- ft2fp(rawdata,8,2500,50, Normalize = TRUE, ExcludeNoRegress = FALSE,
ExcludeFirstPass = TRUE)
```

The first argument of the function is rawdata, which is the fixation time data shown above.

The second argument is the critical region defined by the researcher. The function will only retain the data after the subjects passed the critical region for the first time. This is because the fixation pattern will be clearer if we synchronize the starting point for all trials; and it is usually the fixation pattern after passing certain region that interests the researches most. In this example, the critical region is the last one, region 8.

The third and fourth arguments determine the time course to show in obtained fixation pattern. The third one is the time course from the starting point; and the fourth one is the time interval. In this example, the fixation probabilities from starting point (0 ms) to end point (2500ms) will be shown, and each time interval will be 50 ms.

The other three arguments are not necessary to be provided by users. The default value for `Normalize` is `TRUE`, meaning that the duration of each fixation will be adjusted depending on the relative reading speed of subjects, which will make the pattern clearer. The default value for `ExcludeNoRegress` is `FALSE`, which means that trials with no regression after the first-pass of critical region will not be discarded. The default value for `ExcludeFirstPass` is `TRUE`, which means that the first-pass fixations on critical region itself will be discarded, which will eliminate the effect of variation in first-pass time. You can ignore those variables if you do not want make changes on default values.

The function returns a data frame which contains the fixation probability data. Let us take a look at the new data:

```
head(newdata)
```

and R returns:

```
 list subject condition region N Time y fix_prob
1    1       1         1      1 2    0 0      0.0
2    1       1         1      2 2    0 1      0.5
3    1       1         1      3 2    0 0      0.0
4    1       1         1      4 2    0 1      0.5
5    1       1         1      5 2    0 0      0.0
6    1       1         1      6 2    0 0      0.0
```

Apart from basic information like `list`, `subject`, `condition`, and `region`, there several new variables generated. `Time` is the time points generated according to researcher's customized time course. `N` is the total number of valid trials in that condition; `y` is the number of trials with fixation on that region at that time. N and y are basic information needed for further analysis like empirical transformation and multi-level regressions. `fix_prob` is the fixation probability for each spatio-temporal unit, and is calculated by dividing y by N.

Visualization

After preprocessing, the big part of FPA analysis has been done. The next work is to visualize the results and make sense of the fixation pattern.

First, the returned data frame (newdata) contains the fixation probabilities for every subjects. So we need to average them across subjects to obtain the general pattern. The function get_pattern is built for this purpose:

```
pattern <- get_pattern(newdata)
```

Take a look at the structure of returned data:

```
head(pattern)

condition region          0         50        100        150        200
1         1      1 0.14083333 0.14083333 0.14083333 0.15083333 0.16333333
2         1      2 0.13250000 0.10750000 0.10750000 0.09750000 0.13916667
3         1      3 0.10750000 0.10750000 0.10750000 0.11166667 0.08500000
4         1      4 0.08083333 0.08083333 0.07083333 0.03333333 0.02500000
5         1      5 0.02500000 0.02500000 0.02500000 0.02500000 0.04166667
6         1      6 0.17416667 0.16416667 0.16416667 0.20416667 0.19166667
```

The above is part of the head of returned data `pattern`. It retains the condition and region information. And besides, each column contains the averaged fixation probabilities for each region at that time point. There are totally 51 time points (from 0 to 2500ms, 50ms as an interval; only part of them shown above), as customized by the researcher.

Next step is to visualize the pattern. Users can choose to make the plots by themselves based on the data returned by get_pattern, using tools like excel, R, MATLAB and others. If users want to use excel for visualization, they may want to export the data first:

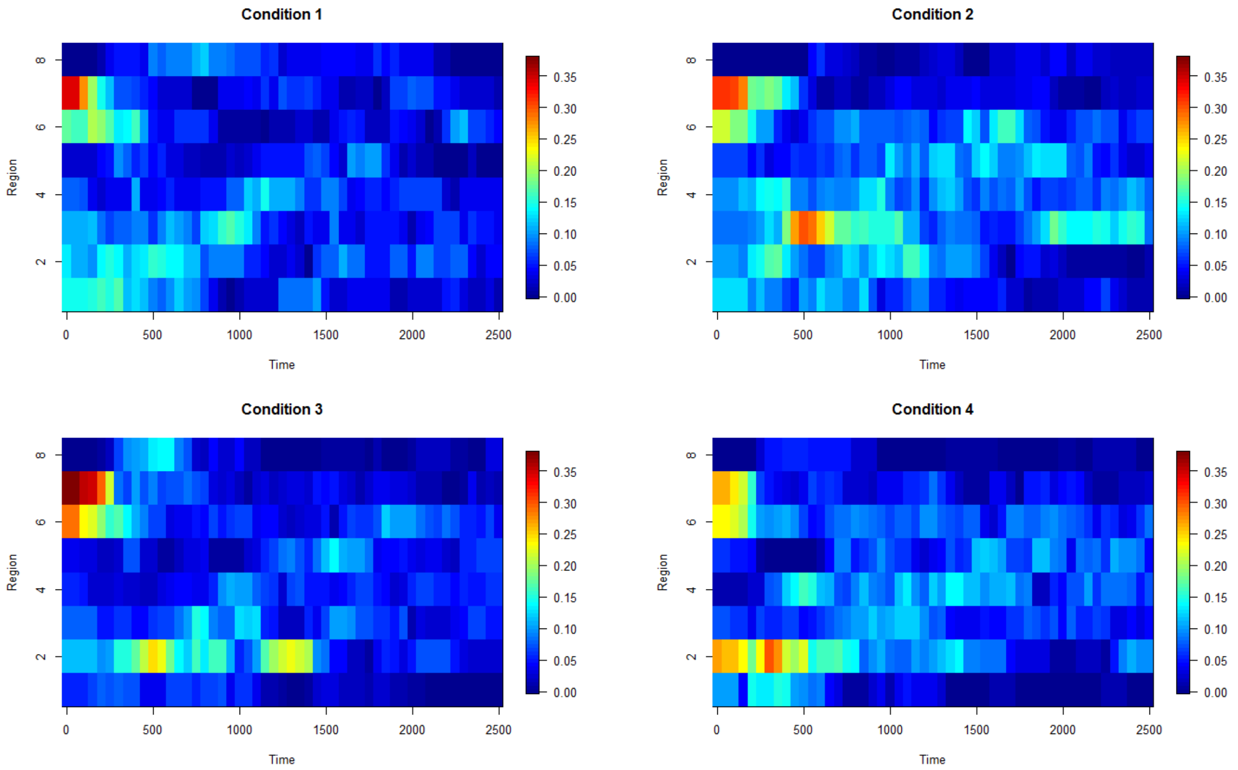
```
write.csv(pattern,"pattern.csv")
```

However, 'fpa' package provides quick tools to visualize the results. For instance:

```
plot_pattern(pattern, Condition="All")
```

The first argument of the function `plot_pattern` is the general pattern data. The second specifies the condition(s) the user want to plot. `Condition` can be a string ("All", indicting all consitions), a number (e.g., 1, meaning only condition 1), or a vector (e.g., c(1,2), meaning that condition 1 and 2 will be plotted).

R plots:

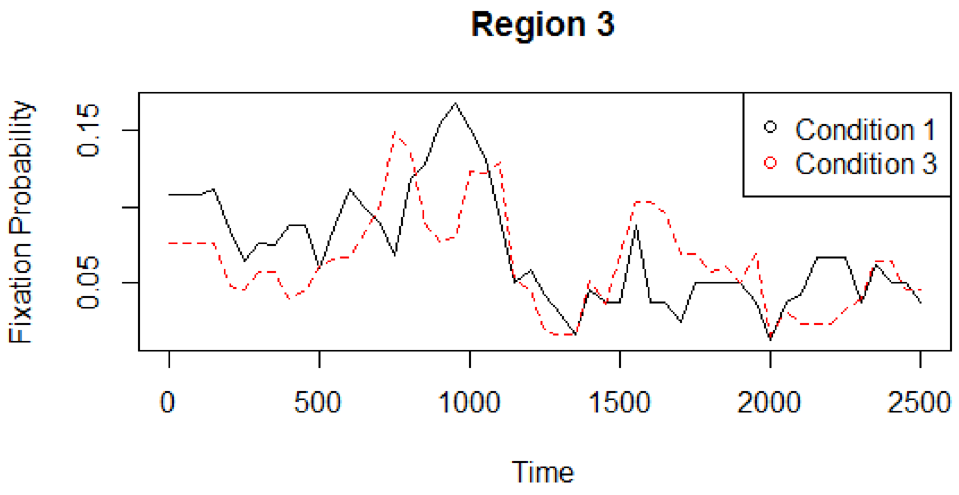


These are 3-d plots, with x for time, y for region, and the color representing the average fixation probabilities. Four subplots correspond to four conditions in this example. As we can observe, most subjects regress to region 7 after they fixated on critical region (region 8), and then they return to the beginning of sentence (region 1 and 2), and then go through other regions. There are differences in the general patterns between conditions.

The above plots give the general pattern, but more detailed information may not be clear enough. For instance, I may want to know if there is a difference in fixation probability at region 3 between condition 1 and condition 3. The above picture may not be clear enough. Then we may use the function lineplot:

```
lineplot(pattern, Region=3, Condition=c(1,3))
```

and R shows:



This plot gives more detailed information on region 3 for condition 1 and 3. We can observe that the fixation probability on region 3 differs depending on time: at earlier stage (before 1000ms), condition 1 has generally higher fixation probability; at later stage (1500ms-2000ms), condition 3 is a little bit higher.

Higher-level Analyses

Because of the binary nature of dependent variable, and other complex dependencies in the experiment, I suggest use the multi-level logistic regression (MLR) model developed by Barr (see reference) for drawing conclusions in FPA. In short words, the fixation probability need to be empirically transformed, and then submit to multi-level linear regression models.

Users can compare the fixation probabilities across conditions at certain spatio-temporal units (e.g., between all conditions at region 3 for the time between 1500 and 2000ms). First subset the data obtained in ft2fp function:

```
newdata$Time <- as.integer(newdata$Time)

subdata <- subset(newdata, c(region ==3, Time <= 2000 & Time >= 1500))
```

After that you can conduct MLR on the new dataset, by referring to the MLR guide (http://talklab.psy.gla.ac.uk/tvw/elogit-wt.html). The current version of ‘fpa’ does not provide batch analyses tools of MRL, but it may be created in later versions.

References

Barr, D. J. (2008). Analyzing ‘visual world’eyetracking data using multilevel logistic regression. *Journal of memory and language*, *59*(4), 457-474.

Cao, J., Wang, J., Wang, S., & Chen, H.-C. (2016). Spatio-temporal fixation pattern analysis: A new method to analyze eye movement reading data. The 16th International Conference of Eastern Asian Languages Processing.