# example eyetrackingR

## R Markdown

eyetrackingR is an R package designed to make dealing with eye-tracking data easier. It handles tasks along the pipeline from raw data to analysis and visualization – as illustrated in the eyetrackingR workflow. Check out the vignettes to the left for some gentle introductions to using eyetrackingR for several popular types of analyses, including growth-curve analysis, onset-contingent reaction time analyses, as well as several non-parametric bootstrapping approaches.

```
# load *eyetrackingR* and set data options
library(eyetrackingR)
data("word_recognition")
dataset <- make_eyetrackingr_data(word_recognition,
                                  participant_column = "ParticipantName",
                                  trial_column = "Trial",
                                  time_column = "TimeFromTrialOnset",
                                  trackloss_column = "TrackLoss",
                                  aoi_columns = c('Animate','Inanimate'),
                                  treat_non_aoi_looks_as_missing = TRUE
)
# remove trackloss-ridden trials
dataset_clean <- clean_by_trackloss(dataset,
                                    participant_prop_thresh = 1, trial_prop_thresh = .25,
                                    window_start_time = 15500, window_end_time = 21000)
```

```
## Performing Trackloss Analysis...
```

```
## Will exclude trials whose trackloss proportion is greater than : 0.25
```

```
##  ...removed  33  trials.
```

```
# zoom in on response window
word_window <- subset_by_window(dataset_clean, rezero = FALSE,
                                window_start_time = 15500, window_end_time = 21000)
```

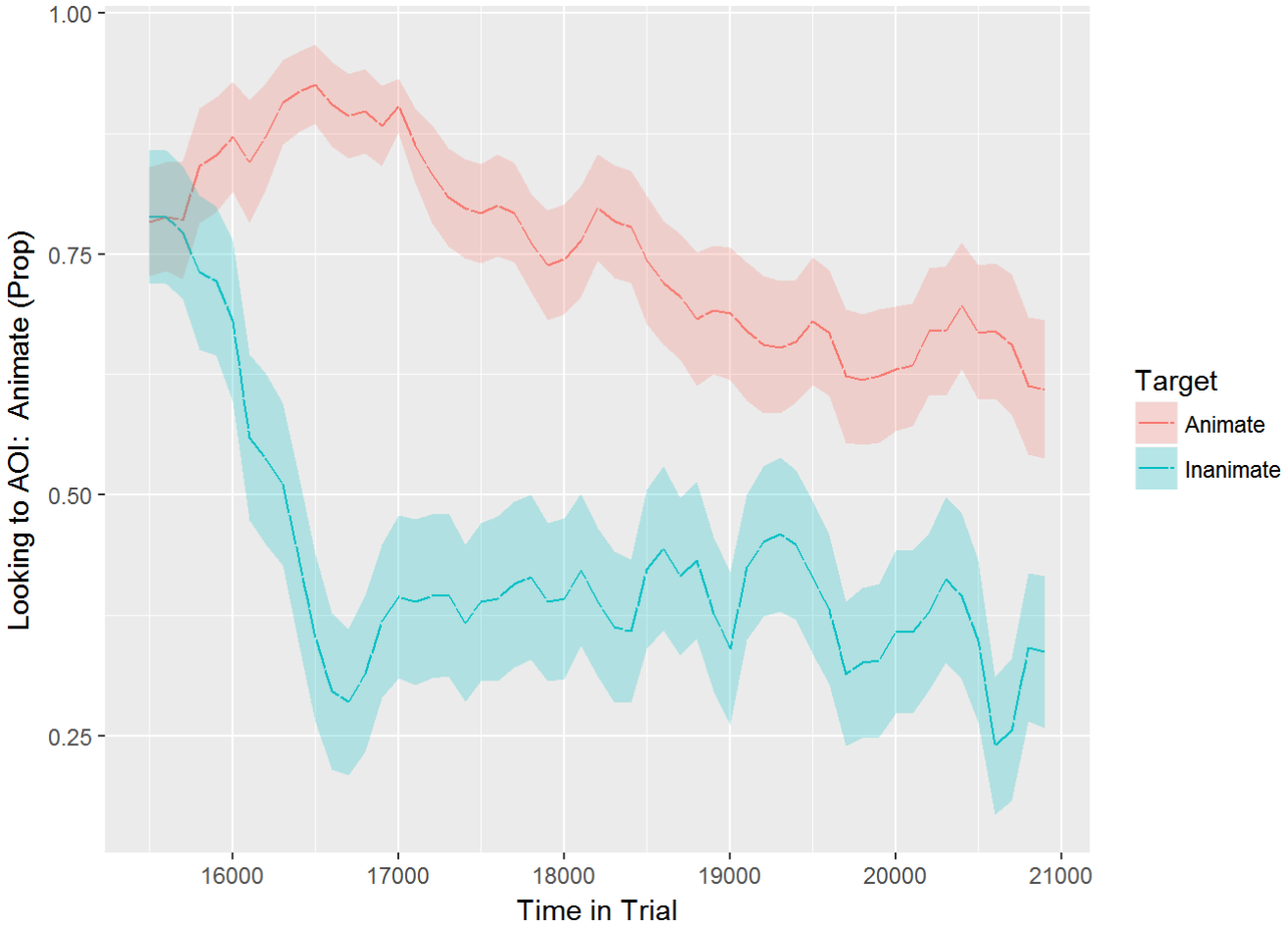
```
## Avg. window length in new data will be 5500
```

```
# create a column indicating what type of trial:
word_window$Target <- as.factor( ifelse(test = grepl('(Spoon|Bottle)', word_window$Trial),
                                        yes = 'Inanimate',
                                        no  = 'Animate') )

# convert data into a series of time-bins:
word_time <- make_time_sequence_data(word_window, time_bin_size = 100,
                                     predictor_columns = "Target", aois = c("Animate"))
plot(word_time, predictor_column = "Target")
```

```
## Warning: Removed 37 rows containing non-finite values (stat_summary).

## Warning: Removed 37 rows containing non-finite values (stat_summary).
```




```
# An important step in performing regression analysis is to center predictors (in order to make
 parameter estimates more interpretable)
word_time$TargetC <- ifelse(word_time$Target == 'Animate', .5, -.5)
word_time$TargetC <- word_time$TargetC - mean(word_time$TargetC)

# perform a growth-curve analysis
library(lme4)
```

```
## Loading required package: Matrix
```

```
model <- lmer(Elog ~ TargetC*(ot1 + ot2 + ot3 + ot4 + ot5) + (1 | Trial) + (1 |
ParticipantName), data = word_time, REML = FALSE)
broom::tidy(model, effects="fixed")
```

```
##           term   estimate std.error   statistic
## 1  (Intercept)  0.7203296 0.1354306   5.3188097
## 2      TargetC  1.7087978 0.1285352  13.2943981
## 3          ot1 -2.8035128 0.1860393 -15.0694667
## 4          ot2  0.2320570 0.1860393   1.2473547
## 5          ot3 -0.5809525 0.1860393  -3.1227409
## 6          ot4 -0.3634757 0.1860393  -1.9537579
## 7          ot5  0.5969192 0.1860393   3.2085652
## 8  TargetC:ot1 -0.9024639 0.3874279  -2.3293726
## 9  TargetC:ot2 -1.8994471 0.3874279  -4.9027118
## 10 TargetC:ot3  4.0342392 0.3874279  10.4128784
## 11 TargetC:ot4 -1.8634407 0.3874279  -4.8097748
## 12 TargetC:ot5 -0.0775141 0.3874279  -0.2000736
```

```
drop1(model,~.,test="Chi")
```

```
## Single term deletions
##
## Model:
## Elog ~ TargetC * (ot1 + ot2 + ot3 + ot4 + ot5) + (1 | Trial) +
##     (1 | ParticipantName)
##             Df   AIC     LRT   Pr(Chi)
## <none>         28831
## TargetC      1 28849  19.838 8.428e-06 ***
## ot1          1 29052 223.313 < 2.2e-16 ***
## ot2          1 28830   1.556  0.212294
## ot3          1 28839   9.744  0.001799 **
## ot4          1 28833   3.816  0.050763 .
## ot5          1 28839  10.287  0.001340 **
## TargetC:ot1  1 28834   5.424  0.019864 *
## TargetC:ot2  1 28853  23.993 9.667e-07 ***
## TargetC:ot3  1 28936 107.557 < 2.2e-16 ***
## TargetC:ot4  1 28852  23.094 1.543e-06 ***
## TargetC:ot5  1 28829   0.040  0.841423
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```