

# A quick eyetrackingR Walkthrough

Analysis Timestamp:

```
date()
```

```
## [1] "Thu Sep 10 23:41:05 2015"
```

## Dependencies

```
library(ggplot2)
library(lme4)
```

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

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
##
## The following objects are masked from 'package:stats':
##
##   filter, lag
##
## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
```

```
library(zoo)
```

```
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##   as.Date, as.Date.numeric
```

## Load dataset

```
load('../data/word-recognition.Rdata')
```

Load and set data options for eyetrackingR library.

```
devtools::load_all("../")
```

```
## Loading eyetrackingR
```

```
# set data options
data_options = set_data_options(
  trial_column = "Trial",
  time_column = "TimeFromTrialOnset",
  trackloss_column = "TrackLoss",
  aoi_columns = c('Animate', 'Inanimate'),
  participant_column = "ParticipantName"
)
```

Verify state of each relevant column.

```
data <- verify_dataset(data, data_options)
```

Deal with trackloss.

We are going to:

- Treat looks outside of our AOI as if they are trackloss
- Calculate the amount of trackloss in each trial
- Remove trials with over 50% trackloss
- Remove all remaining trackloss samples from our dataset (so that, in each sample, the toddler is looking either to one AOI or the other)

```
# convert non-AOI looks to trackloss
data <- convert_non_aoi_to_trackloss(data, data_options)

# analyze amount of trackloss by subjects and trials
trackloss <- trackloss_analysis(data, data_options, window_start = 15500, window_end = 21000)

# show trackloss
trackloss
```

```
## Source: local data frame [155 x 6]
```

```
##
##   ParticipantName      Trial Samples TracklossSamples
##         (fctr)         (fctr)   (dbl)           (dbl)
## 1    ANCAT139 FamiliarBottle    331           162
## 2    ANCAT18  FamiliarBird     331            74
## 3    ANCAT18 FamiliarBottle     331            43
## 4    ANCAT18  FamiliarCow      331           160
## 5    ANCAT18  FamiliarDog      331            96
## 6    ANCAT18 FamiliarHorse     331           165
## 7    ANCAT18 FamiliarSpoon     331            95
## 8    ANCAT22  FamiliarBird     331            14
```

```
## 9          ANCAT22 FamiliarBottle      331          8
## 10         ANCAT22   FamiliarDog      331          55
## ..          ...          ...          ...          ...
## Variables not shown: TracklossForTrial (dbl), TracklossForParticipant
## (dbl)
```

```
data <- clean_by_trackloss(data,
                           data_options,
                           trial_prop_thresh = .25,
                           window_start = 15500,
                           window_end = 21000)
```

```
## Performing Trackloss Analysis...
## Will exclude trials whose trackloss proportion is greater than : 0.25
## ...removed 34 trials.
```

```
# remove all trackloss from remaining trials
data <- remove_trackloss(data, data_options, delete_rows=TRUE)
```

Subset data to examine “response” window, and create “Target” condition based on TrialName

```
# subset to response window
response_window <- subset_by_window(data, data_options, window_start = 15500, window_end = 21000)

# create "Target" condition column based on trial names
response_window$Target <- ifelse(grepl('(Spoon|Bottle)', response_window$Trial), 'Inanimate', 'Animate')
response_window$Target <- factor(response_window$Target)
```

Describe and visualize over results

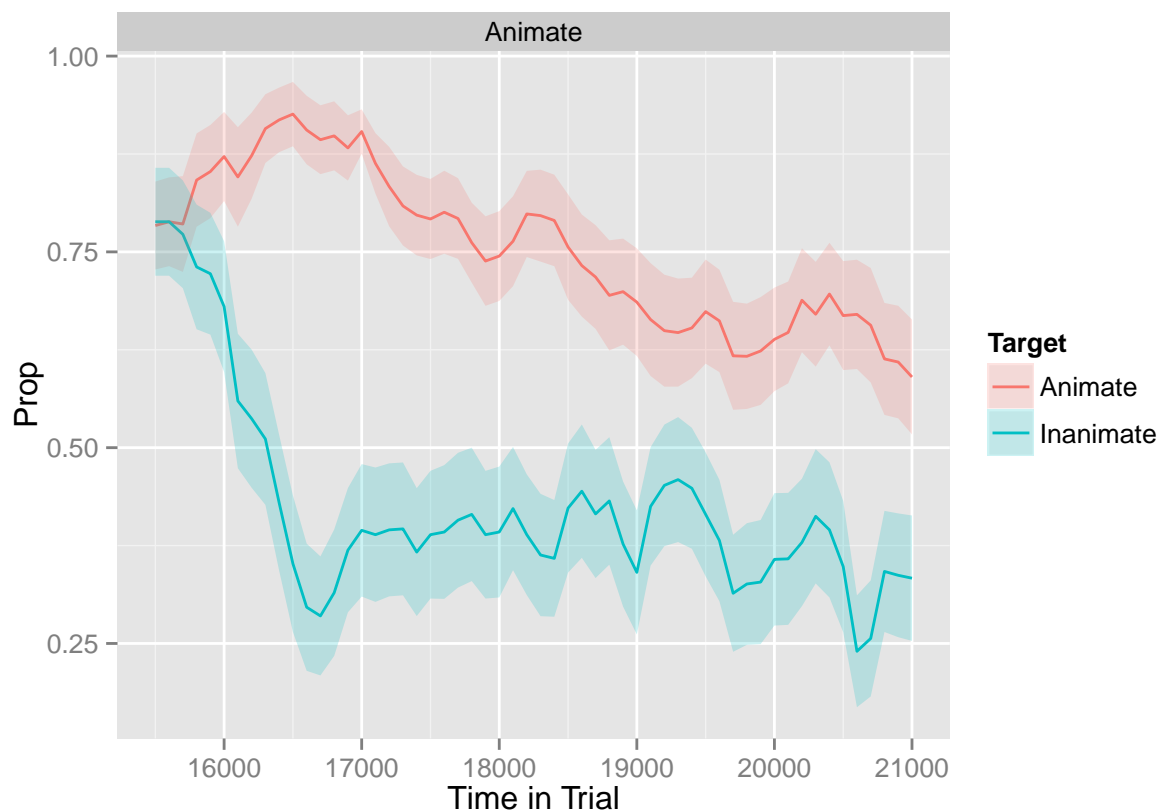
```
(data_summary <- describe_data(response_window, data_options, describe_column='Animate', group_columns=ParticipantName))

## Source: local data frame [54 x 8]
## Groups: ParticipantName [?]
##
##   ParticipantName Target      Mean      SD      Var      Min
##   (fctr)      (fctr)      (dbl)      (dbl)      (dbl)      (dbl)
## 1     ANCAT18  Animate  0.3579767  0.4803407  0.2307271  0.3579767
## 2     ANCAT18  Inanimate 0.0000000  0.0000000  0.0000000  0.0000000
## 3     ANCAT22  Animate  0.7903403  0.4072893  0.1658846  0.7903403
## 4     ANCAT22  Inanimate 0.2843895  0.4514722  0.2038271  0.2843895
## 5     ANCAT23  Animate  0.8156646  0.3879110  0.1504749  0.8156646
## 6     ANCAT23  Inanimate 0.7063758  0.4558043  0.2077576  0.7063758
## 7     ANCAT26  Animate  0.7001675  0.4585687  0.2102852  0.7001675
## 8     ANCAT26  Inanimate 0.4958264  0.5004005  0.2504006  0.4958264
## 9     ANCAT39  Animate  0.8779343  0.3274899  0.1072496  0.8779343
## 10    ANCAT39  Inanimate 0.1848341  0.3884699  0.1509089  0.1848341
## ..          ...          ...          ...          ...          ...
## Variables not shown: Max (dbl), NumTrials (int)
```

```
# aggregate across trials within subjects in time analysis
response_time <- make_time_sequence_data(response_window, data_options, time_bin_size = 100,
                                         predictor_columns = c("Target"),
                                         aoi = c("Animate"))
)
```

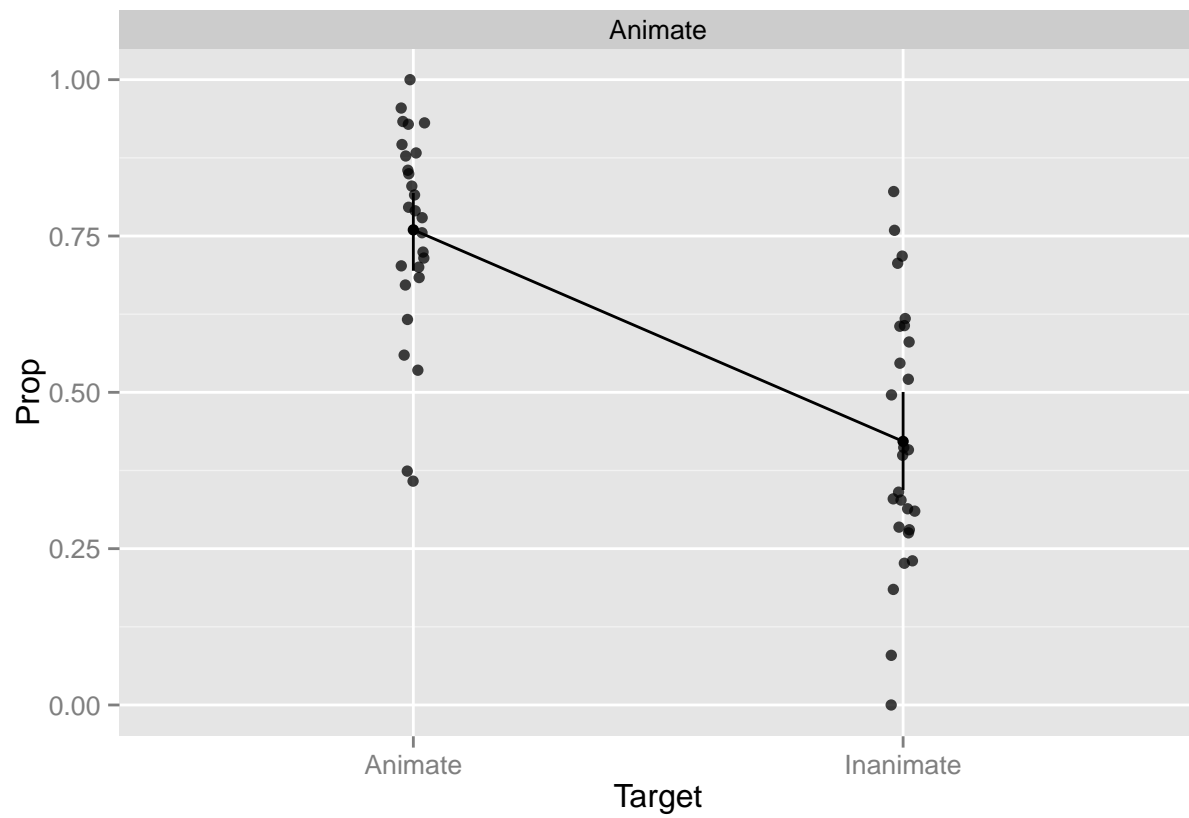
```
## Warning in make_time_sequence_data(response_window, data_options,
## time_bin_size = 100, : With the current time-bin size, the final time-
## bin is very small. Consider choosing a different time-bin size or using
## subset_by_window to remove this portion of the trial.
```

```
# visualize time results
plot(response_time, predictor_column = "Target")
```

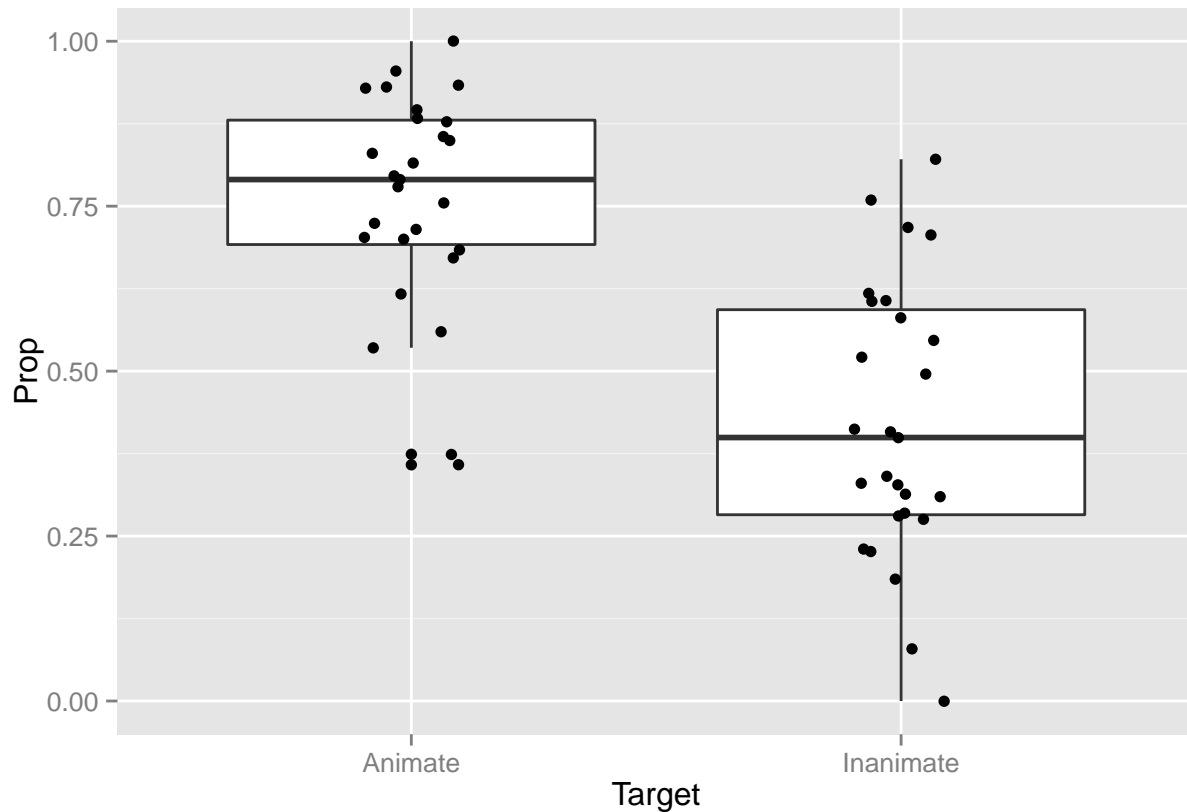


```
# aggregate by subject across the response window
response_window_agg_by_sub <- make_time_window_data(response_window,
                                                    data_options,
                                                    aois='Animate',
                                                    predictor_columns=c('Target','Age','MCDI_Total'),
                                                    summarize_by = "ParticipantName")

# take a quick peek at mean proportion by Target condition
plot(response_window_agg_by_sub, predictor_columns="Target")
```



```
# visualize our aggregated data with a boxplot (data should match previous plot)  
ggplot(response_window_agg_by_sub, aes(x=Target, y=Prop)) +  
  geom_boxplot() +  
  geom_point(position=position_jitter(.1))
```



```
# show condition means
describe_data(response_window_agg_by_sub, describe_column = "Prop", group_columns = "Target")
```

```
## Source: local data frame [2 x 6]
##
##      Target      Mean      SD      Var      Min      Max
##      (fctr)    (dbl)    (dbl)    (dbl)    (dbl)    (dbl)
## 1  Animate  0.7598246  0.1645562  0.02707875  0.7598246  0.7598246
## 2 Inanimate 0.4215059  0.2093654  0.04383387  0.4215059  0.4215059
```

## Simple t-test

```
# simple t-test between conditions
t.test(ArcSin ~ Target, data=response_window_agg_by_sub, paired=TRUE)
```

```
##
## Paired t-test
##
## data: ArcSin by Target
## t = 7.2616, df = 26, p-value = 1.034e-07
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.2817599 0.5042545
## sample estimates:
## mean of the differences
##          0.3930072
```

## Mixed-effects models on windowed data

```
response_window_agg <- make_time_window_data(response_window,
                                             data_options,
                                             aois='Animate',
                                             predictor_columns=c('Target','Age','MCDI_Total'))

# mixed-effects linear model on subject*trial data
response_window_agg$TargetC <- ifelse(response_window_agg$Target == 'Animate', .5, -.5)
response_window_agg$TargetC <- scale(response_window_agg$TargetC, center=T, scale=F)

model <- lmer(Elog ~ TargetC + (1 + TargetC | Trial) + (1 | ParticipantName), data = response_window_agg,
             summary(model))

## Linear mixed model fit by maximum likelihood ['lmerMod']
## Formula: Elog ~ TargetC + (1 + TargetC | Trial) + (1 | ParticipantName)
## Data: response_window_agg
##
##      AIC      BIC   logLik deviance df.resid
##    502.1    521.6   -244.0    488.1     114
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.8298 -0.6062 -0.1254  0.4959  2.6248
##
## Random effects:
## Groups           Name      Variance Std.Dev.  Corr
## ParticipantName (Intercept) 4.953e-01 7.038e-01
## Trial            (Intercept) 0.000e+00 0.000e+00
##                TargetC      6.307e-15 7.941e-08  NaN
## Residual                2.920e+00 1.709e+00
## Number of obs: 121, groups: ParticipantName, 27; Trial, 6
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)   1.1413    0.2086    5.470
## TargetC       2.4855    0.3245    7.661
##
## Correlation of Fixed Effects:
##      (Intr)
## TargetC 0.005

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

## Single term deletions
##
## Model:
## Elog ~ TargetC + (1 + TargetC | Trial) + (1 | ParticipantName)
##      Df      AIC      LRT Pr(Chi)
## <none>    502.05
## TargetC  1 509.02 8.9602 0.002759 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

## Growth curve analysis

```
# growth curve analysis on time series
response_time$TargetC <- ifelse(response_time$Target == 'Animate', .5, -.5)
response_time$TargetC <- scale(response_time$TargetC, center=T, scale=F)

model <- lmer(Elog ~ TargetC*(ot1 + ot2 + ot3 + ot4 + ot5) + (1 | Trial) + (1 | ParticipantName), data = response_time,
summary(model)
```

```
## Linear mixed model fit by maximum likelihood ['lmerMod']
## Formula: Elog ~ TargetC * (ot1 + ot2 + ot3 + ot4 + ot5) + (1 | Trial) +
##      (1 | ParticipantName)
##      Data: response_time
##
##      AIC      BIC    logLik deviance df.resid
## 28269.0 28370.8 -14119.5 28239.0      6545
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.6080 -0.8771  0.2872  0.6795  2.0702
##
## Random effects:
##  Groups             Name             Variance Std.Dev.
## ParticipantName (Intercept) 0.42628  0.6529
## Trial              (Intercept) 0.02103  0.1450
## Residual                                4.27329  2.0672
## Number of obs: 6560, groups: ParticipantName, 27; Trial, 6
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)   0.7148    0.1416   5.047
## TargetC       1.7340    0.1368  12.675
## ot1          -2.9040    0.1934 -15.019
## ot2           0.2677    0.1937   1.382
## ot3          -0.6158    0.1933  -3.186
## ot4          -0.2628    0.1927  -1.364
## ot5           0.5999    0.1931   3.106
## TargetC:ot1  -1.0124    0.4011  -2.524
## TargetC:ot2  -1.9643    0.4019  -4.887
## TargetC:ot3   3.8307    0.4010   9.553
## TargetC:ot4  -2.3421    0.4001  -5.854
## TargetC:ot5  -0.4444    0.4010  -1.108
##
## Correlation of Fixed Effects:
##              (Intr) TargetC ot1    ot2    ot3    ot4    ot5    TrgC:1 TrgC:2
## TargetC      -0.023
## ot1           0.003  0.002
## ot2           0.006  0.003  0.024
## ot3           0.001  0.002  0.032  0.021
## ot4           0.003  0.001  0.010  0.028  0.022
## ot5           0.000  0.000  0.012  0.011  0.024  0.020
## TargetC:ot1   0.001  0.006  0.005  0.009  0.005  0.004  0.003
## TargetC:ot2   0.001  0.008  0.009  0.005  0.008  0.007  0.007  0.019
```



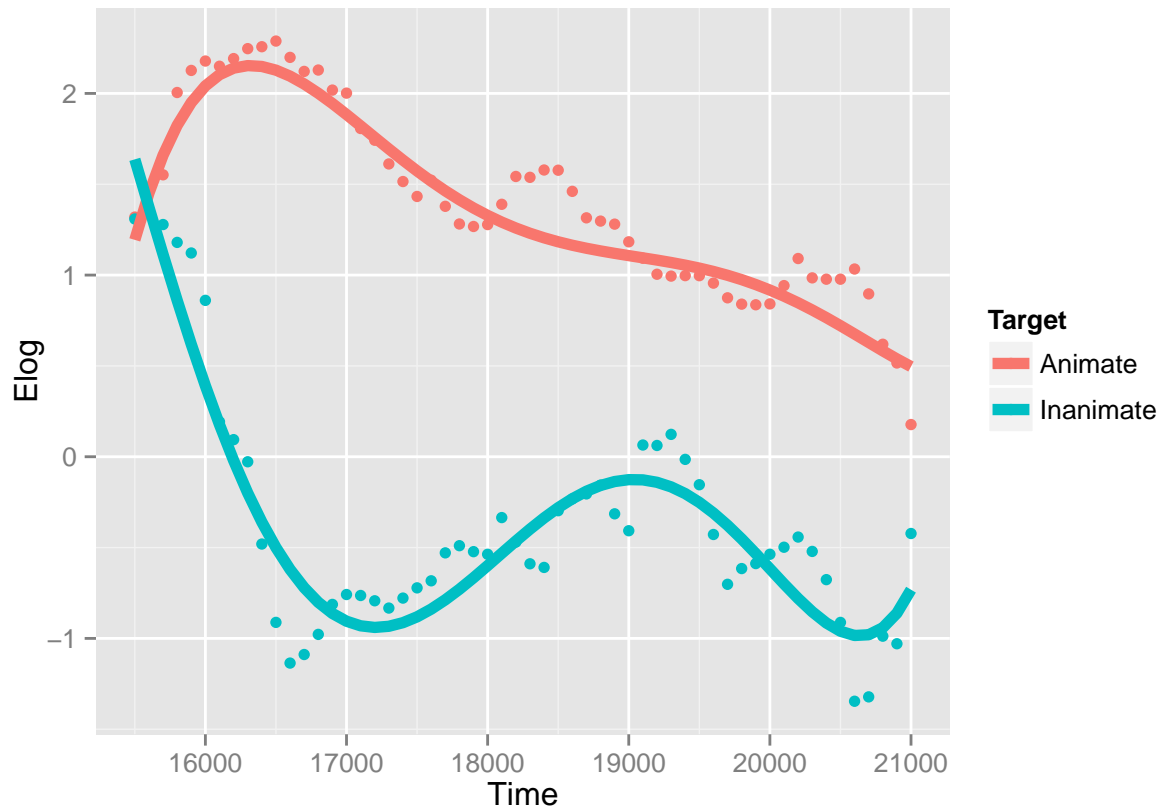
```
## TargetC:ot3 0.001 0.003 0.005 0.008 0.006 0.010 0.005 0.028 0.017
## TargetC:ot4 0.000 0.004 0.004 0.007 0.010 0.004 0.007 0.007 0.024
## TargetC:ot5 0.000 0.000 0.003 0.007 0.005 0.007 0.005 0.010 0.007
##           TrgC:3 TrgC:4
## TargetC
## ot1
## ot2
## ot3
## ot4
## ot5
## TargetC:ot1
## TargetC:ot2
## TargetC:ot3
## TargetC:ot4 0.017
## TargetC:ot5 0.021 0.016
```

```
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>          28269
## TargetC         1 28286  19.298 1.118e-05 ***
## ot1             1 28489 221.772 < 2.2e-16 ***
## ot2             1 28269   1.910 0.167002
## ot3             1 28277  10.140 0.001451 **
## ot4             1 28269   1.860 0.172635
## ot5             1 28277   9.643 0.001901 **
## TargetC:ot1     1 28273   6.368 0.011622 *
## TargetC:ot2     1 28291  23.841 1.047e-06 ***
## TargetC:ot3     1 28358  90.636 < 2.2e-16 ***
## TargetC:ot4     1 28301  34.175 5.038e-09 ***
## TargetC:ot5     1 28268   1.228 0.267717
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# visualize GCM model
```

```
ggplot(response_time, aes(x=Time, y=Elog, color=Target)) +
  stat_summary(fun.y=mean, geom="point") +
  stat_summary(aes(y=predict(model,response_time,re.form=NA)), fun.y=mean, geom="line", linetype='solid')
```



## Onset-contingent analysis

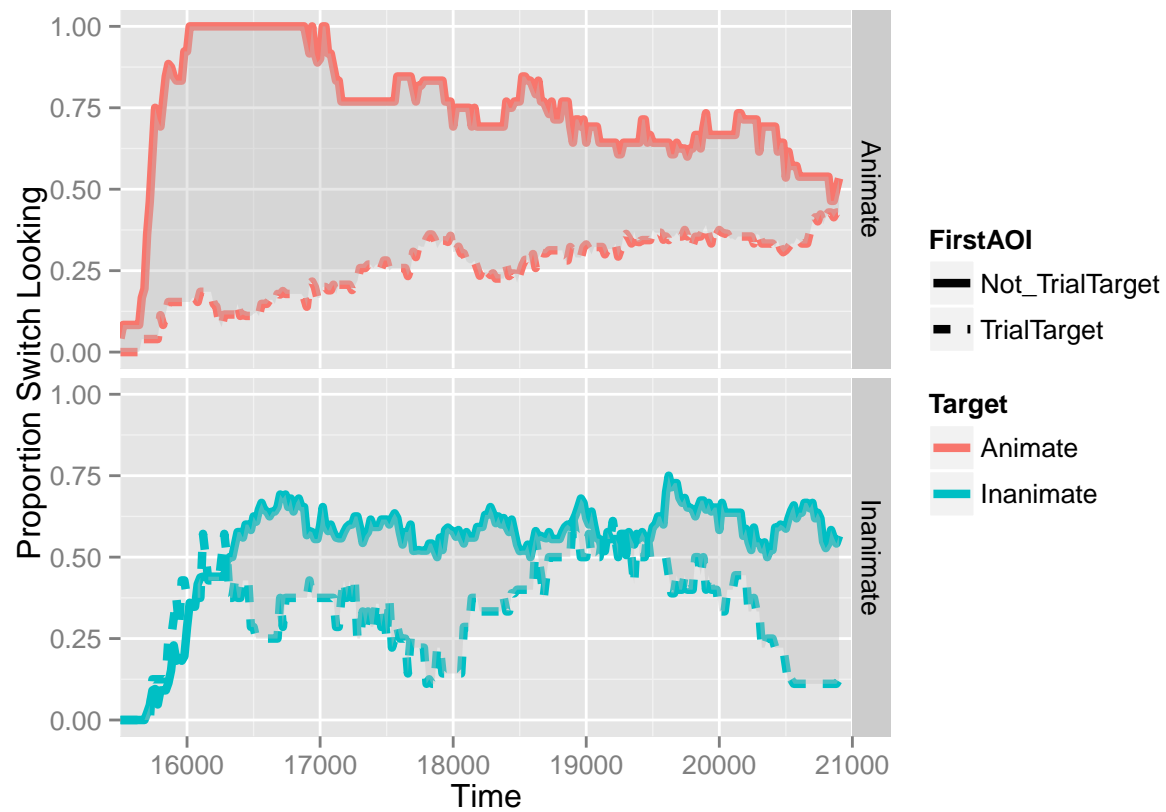
```
# recode AOIs to target & distractor
response_window$TrialTarget <- ifelse(response_window$Target == 'Animate', response_window$Animate, response_window$Inanimate)
response_window$TrialDistractor <- ifelse(response_window$Target == 'Animate', response_window$Inanimate, response_window$Animate)

onsets <- make_onset_data(response_window, data_options, onset_time = 15500, fixation_window_length = 1000)

plot(onsets, predictor_columns = "Target")
```

```
## Warning: Removed 10 rows containing missing values (geom_path).
```

```
## Warning: Removed 10 rows containing missing values (geom_path).
```

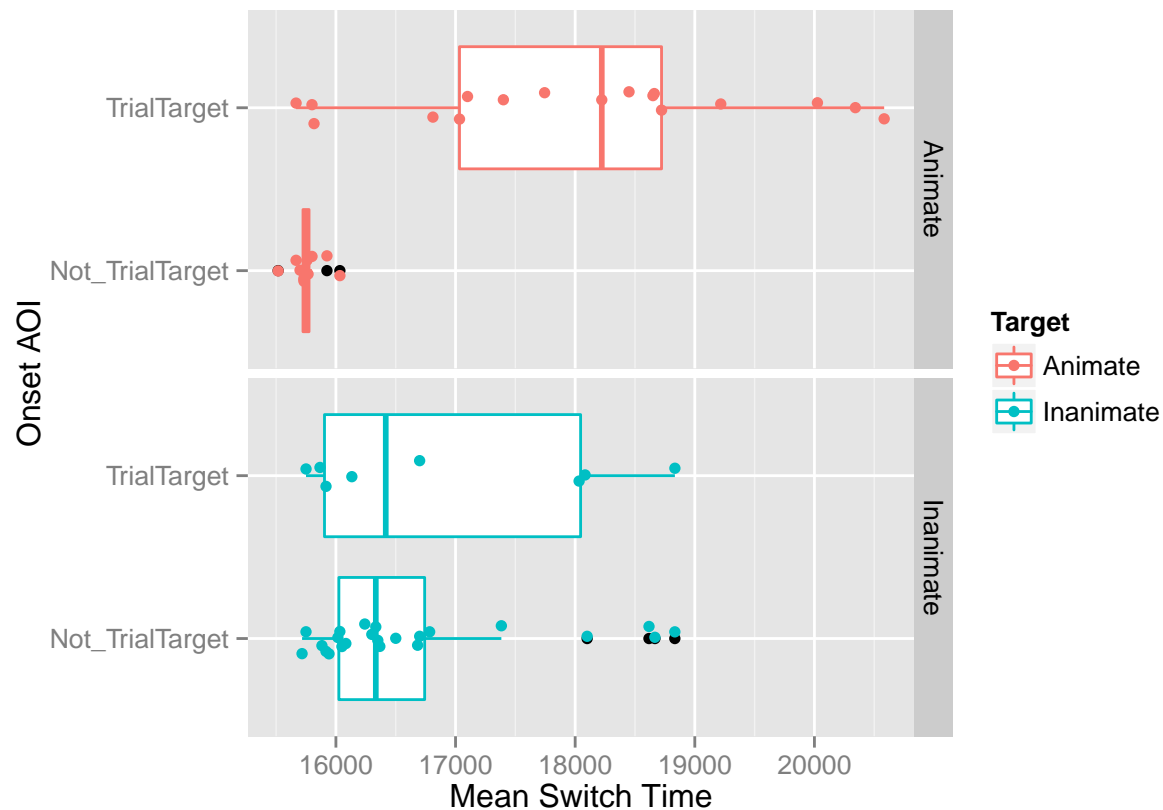


```
# compare switch times
onset_switches <- make_switch_data(onsets, predictor_columns = "Target")

plot(onset_switches, predictor_columns = "Target")
```

```
## Warning: Removed 8 rows containing non-finite values (stat_boxplot).
```

```
## Warning: Removed 8 rows containing missing values (geom_point).
```



```
# center predictor:
onset_switches$FirstAOIC <- ifelse(onset_switches$FirstAOI == 'TrialTarget', .5, -.5)
onset_switches$FirstAOIC <- scale(onset_switches$FirstAOIC, center=TRUE, scale=FALSE)

model <- lmer(FirstSwitch ~ FirstAOIC + (1 + FirstAOIC | Trial) + (1 + FirstAOIC | ParticipantName), data=onset_switches)
summary(model)
```

```
## Linear mixed model fit by maximum likelihood ['lmerMod']
## Formula:
## FirstSwitch ~ FirstAOIC + (1 + FirstAOIC | Trial) + (1 + FirstAOIC | ParticipantName)
## Data: onset_switches
##
##      AIC      BIC    logLik deviance df.resid
## 1742.0   1765.7   -862.0   1724.0      94
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.3951 -0.5914 -0.1854  0.2959  2.3788
##
## Random effects:
##      Groups             Name           Variance Std.Dev. Corr
## ParticipantName (Intercept)  320262     565.9
##                  FirstAOIC   1566606   1251.6   0.90
## Trial              (Intercept)   29484     171.7
##                  FirstAOIC     832048    912.2   1.00
## Residual                        657783    811.0
```

```
## Number of obs: 103, groups: ParticipantName, 26; Trial, 6
##
## Fixed effects:
##           Estimate Std. Error t value
## (Intercept) 16980.9      167.3   101.5
## FirstAOIC    1494.9      482.0     3.1
##
## Correlation of Fixed Effects:
##           (Intr)
## FirstAOIC 0.685
```

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

```
## Single term deletions
##
## Model:
## FirstSwitch ~ FirstAOIC + (1 + FirstAOIC | Trial) + (1 + FirstAOIC |
##   ParticipantName)
##           Df      AIC      LRT Pr(Chi)
## <none>         1742.0
## FirstAOIC  1 1746.2 6.2459 0.01245 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

## Bootstrapped splines analysis.

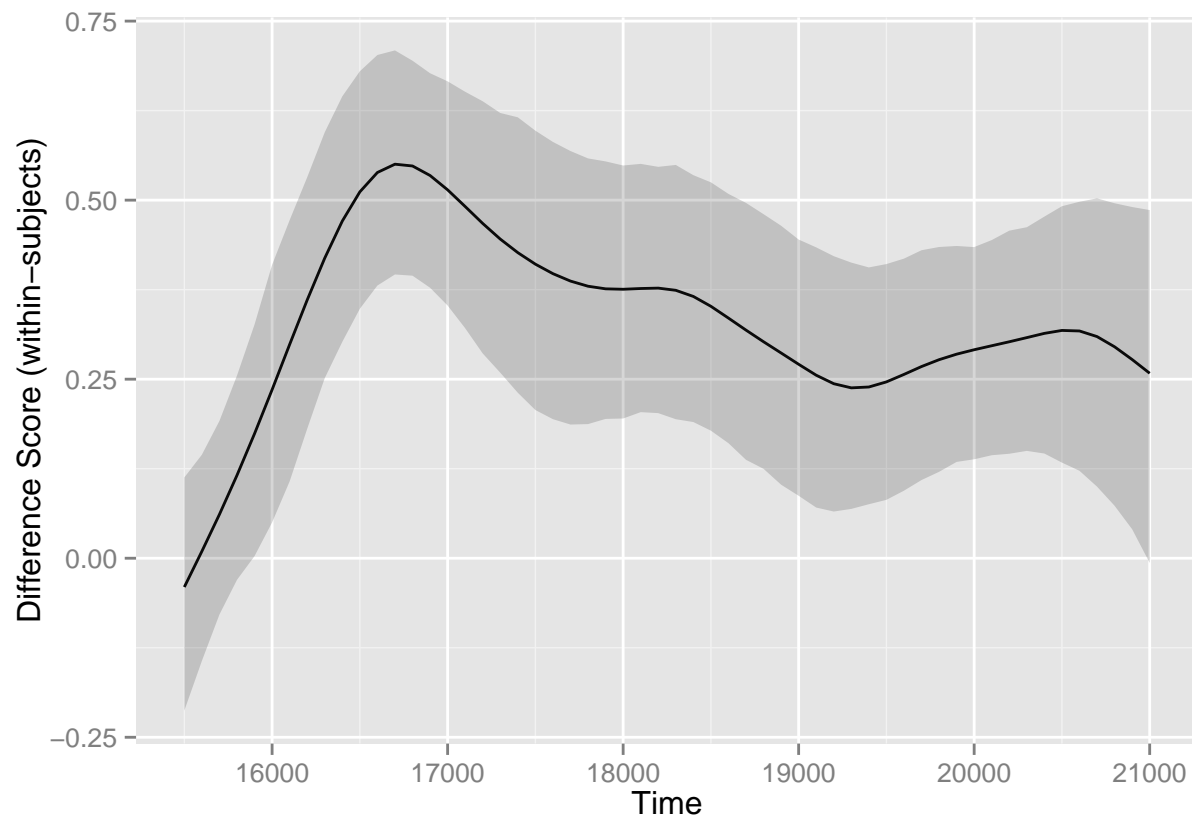
Do a proper within-subjects analysis for the familiar trials.

```
bootstrapped_familiar <- make_boot_splines_data(response_time,
                                                predictor_column = 'Target',
                                                within_subj = TRUE,
                                                samples = 1000,
                                                resolution = 100,
                                                alpha = .05,
                                                smoother = 'smooth.spline')
```

```
## Loading required package: pbapply
## Preparing dataframe...
## Resampling ...
```

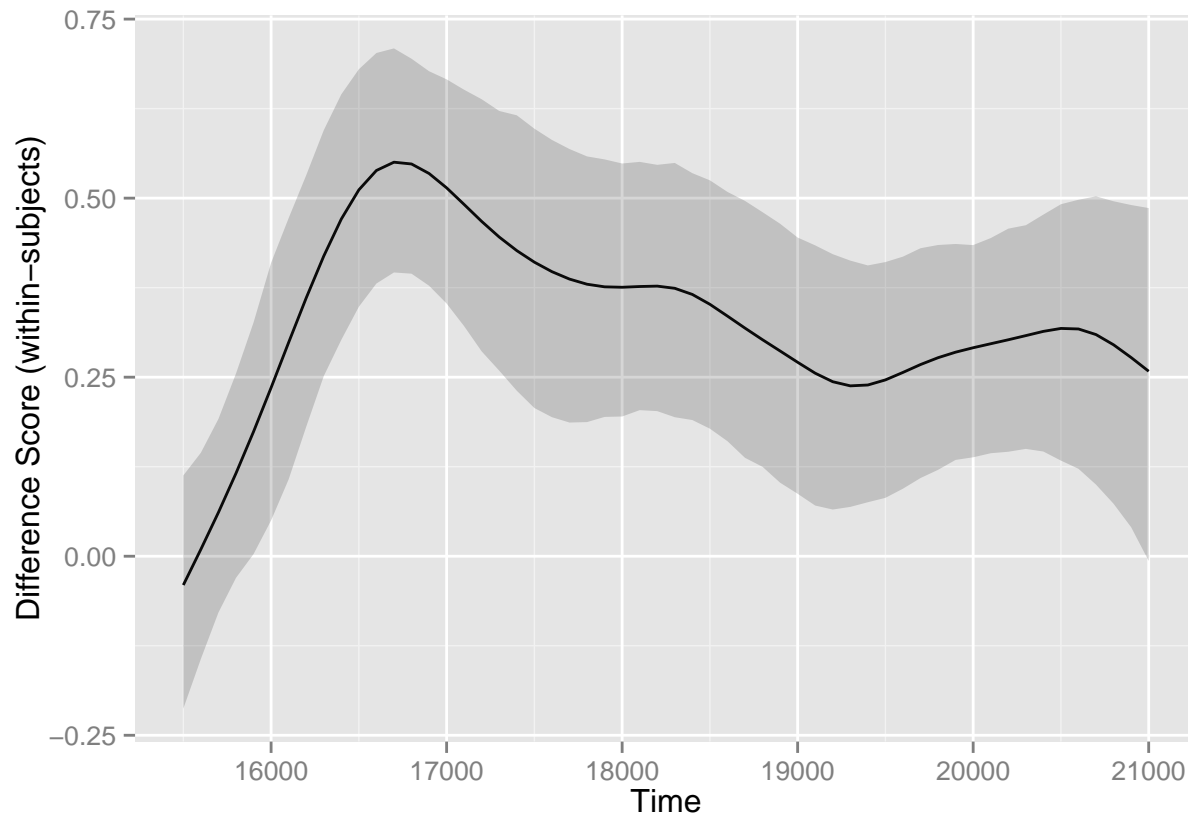
```
plot(bootstrapped_familiar)
```

```
## Plotting within-subjects differences...
```



```
bootstrap_analysis_familiar <- analyze_bootstraps(bootstrapped_familiar)
```

```
# same as above, because, for within-subjects data, it always plots the difference score estimates  
plot(bootstrap_analysis_familiar) # TO DO: this should tell you in which direction the difference was t
```



```
summary(bootstrap_analysis_familiar)
```

```
## Divergences:
## 1: 15900 - 21000
```

## Cluster Analysis

Perform a different type of bootstrapping analyses (Maris & Oostenveld, 2007), sometimes referred to as a cluster analysis. This analysis takes a summed statistic for each cluster of time bins that pass some level of significance, and compares each to the “null” distribution of sum statistics (obtained by bootstrap resampling data within the largest of the clusters).

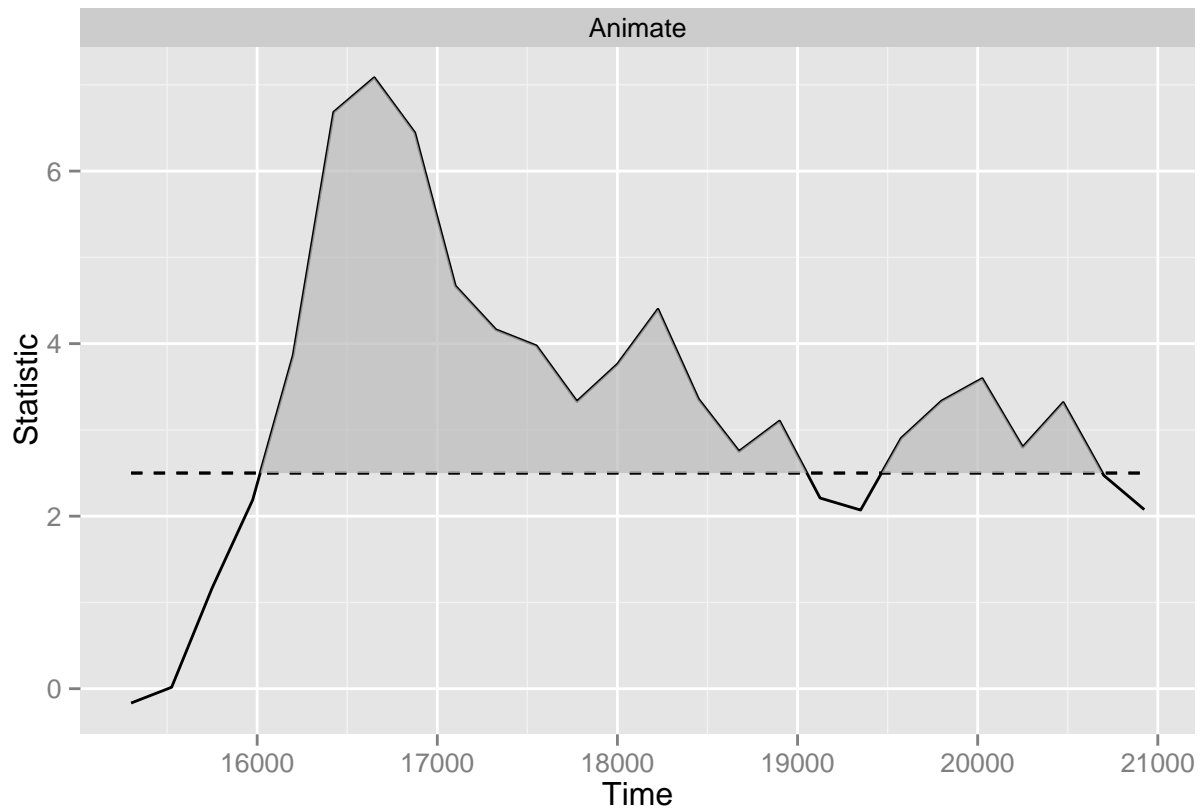
This type of analysis should often give similar results to the above, with the main advantage is that it can be used with several types of statistical techniques (`t.test`, `wilcox.test`, `lm`, and `lmer`), so that continuous predictors, covariates, etc. can be included in the model being tested; and the main disadvantage is that it is slower.

```
response_time_by_subj <- make_time_sequence_data(response_window, data_options, time_bin_size = 225,
  predictor_columns = c("Target"),
  aois = c("Animate"),
  summarize_by = "ParticipantName"
)

df_timeclust = make_time_cluster_data(response_time_by_subj,
  test = "t.test", paired = TRUE,
  predictor_column = "Target",
  threshold = 2.5) # <--- just to make things more challenging
```

```
## Computing t.test for each time bin...
```

```
plot(df_timeclust)
```



```
clust_analysis = analyze_time_clusters(df_timeclust, within_subj = TRUE, paired=TRUE, samples=500)
clust_analysis
```

```
## Test Type:      t.test
## Predictor:      Target
## Formula:  Prop ~ Target
## Null Distribution =====
## Mean:      -0.05562
## SD:  10.34628
## Cluster 1 =====
## Time:      16200 - 19125
## Sum Statistic:  57.57535
## Probability:    0
## Cluster 2 =====
## Time:      19575 - 20700
## Sum Statistic:  15.94725
## Probability:    0.066
```

```
plot(clust_analysis)
```



