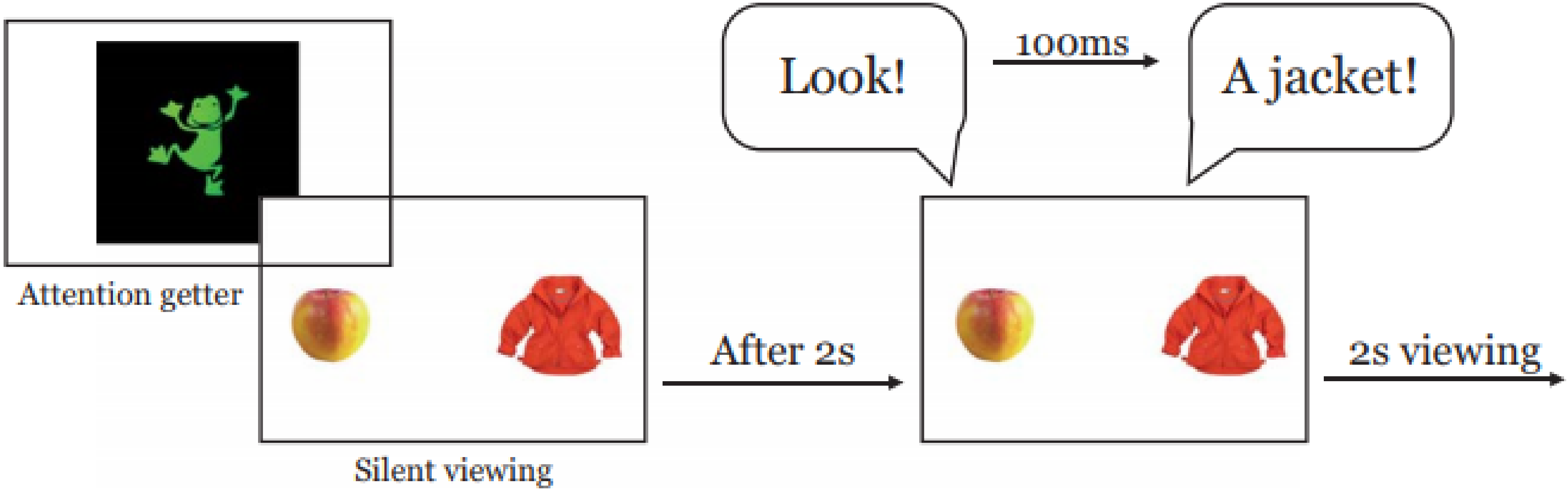
Technische
Universität
Braunschweig

Cluster-Based Permutation Tests in Eye-Tracking Data

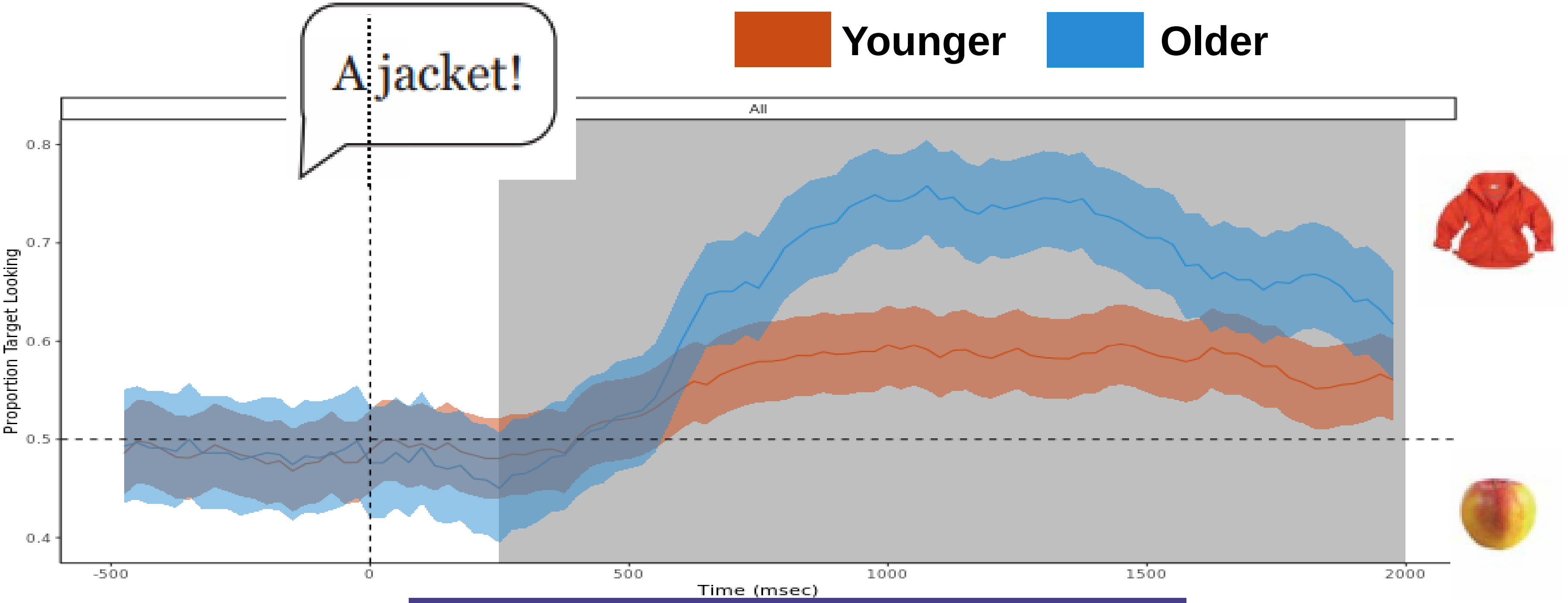
Katie Von Holzen

The Looking-While-Listening (LWL) Paradigm



Egger, Rowland, & Bergmann, 2020

Age in LWL studies

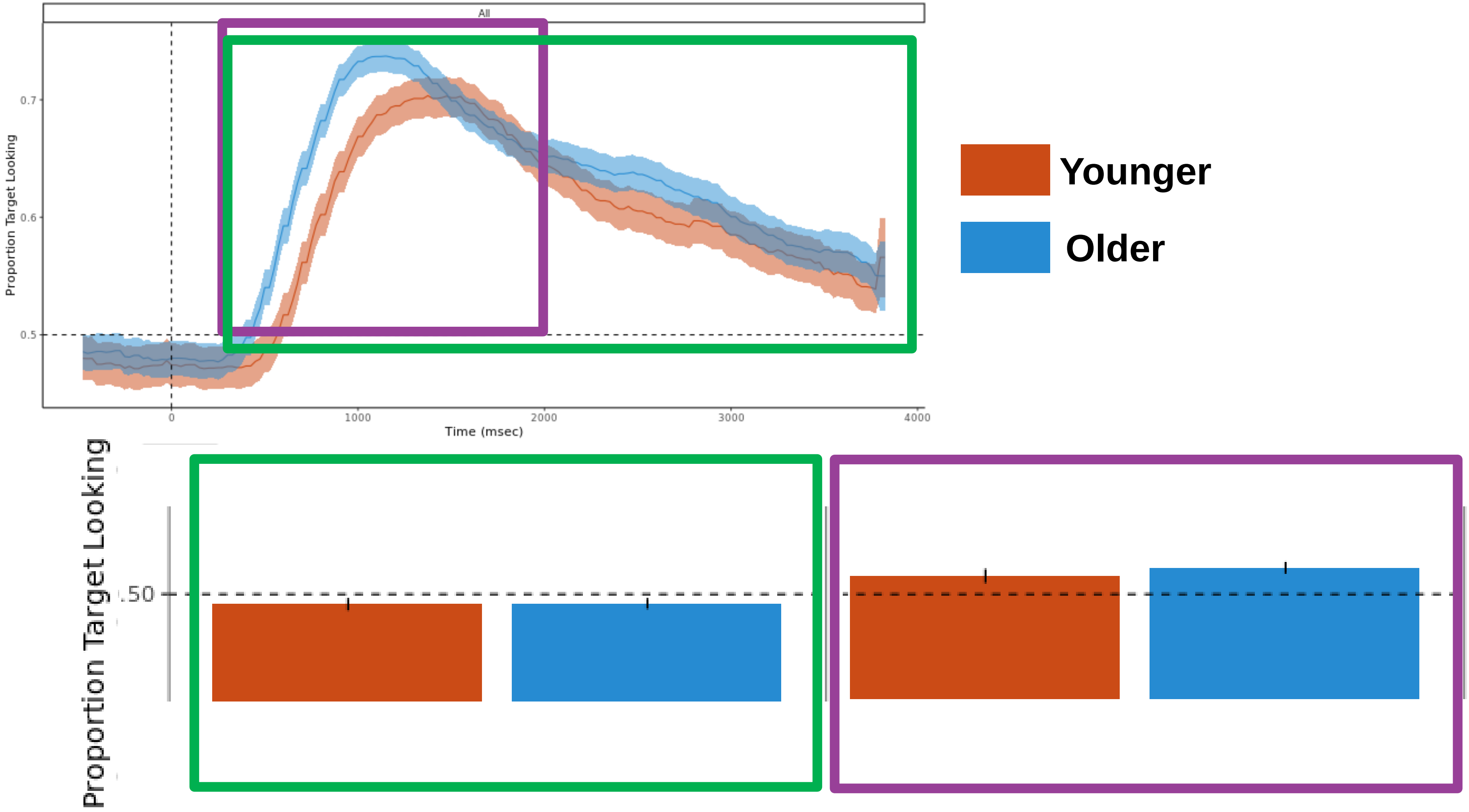


peekbank

A flexible and reproducible interface to developmental eyetracking datasets

Zettersten et al., 2022

Two Ways of Examining Eye-Tracking Data



Von Holzen & Bergmann, 2021; Zettersten et al., 2022

How to Examine the Time-Course

- T-test at every time point/bin
  - High family-wise error rate
    - “the more tests you do, the higher the likelihood that one of them is significant”
  - Bonferroni correction?
    - Assumes time points are independent
      - They’re not!
    - Overly conservative

- Focus: Cluster-Based Permutation Tests
  - Identify time-clusters of differences between conditions
  - Determine whether our found clusters are more likely than what we would expect from chance
    - randomly shuffle the data

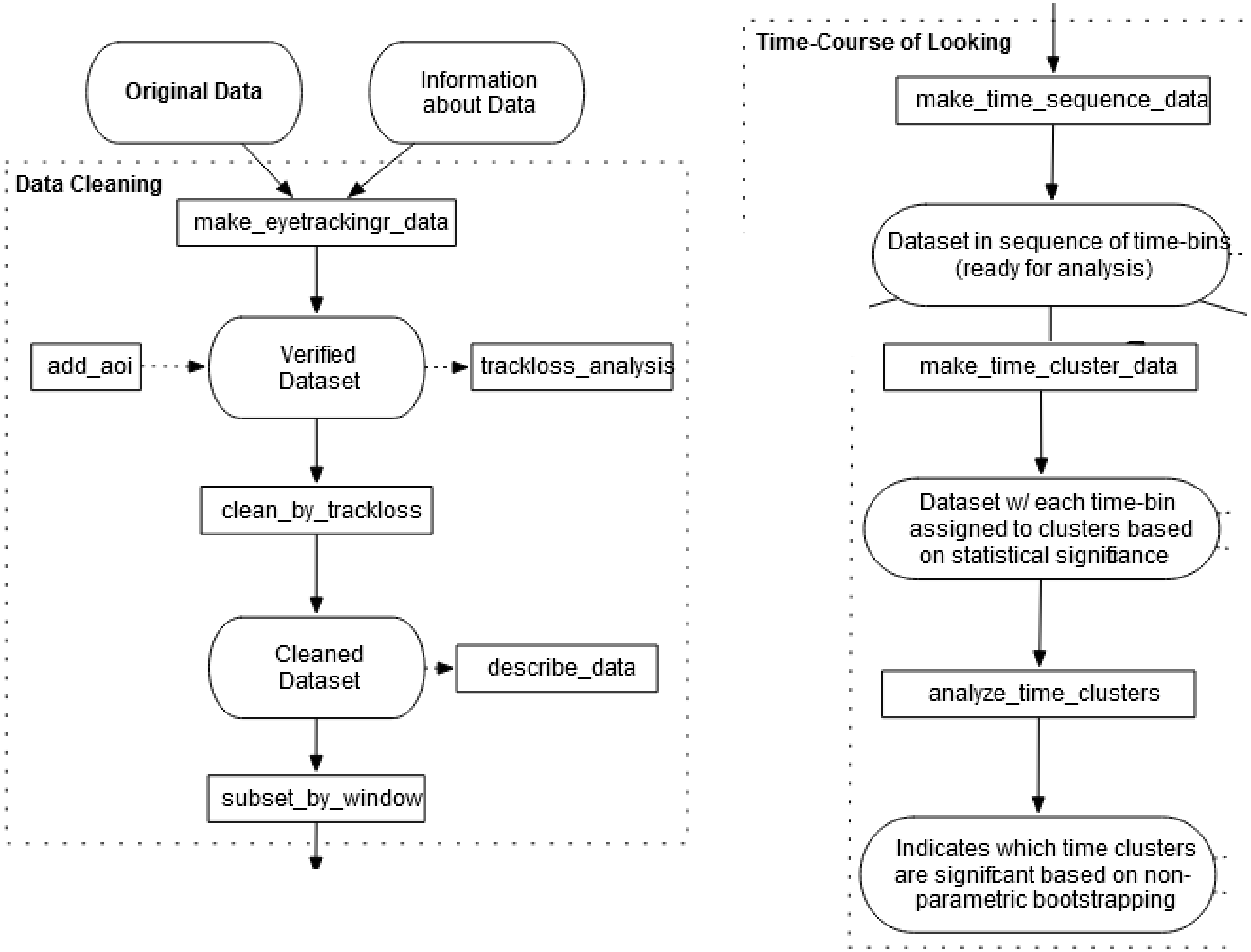
Maris & Oostenveld (2007)

Suggested Tools

eyetrackingR

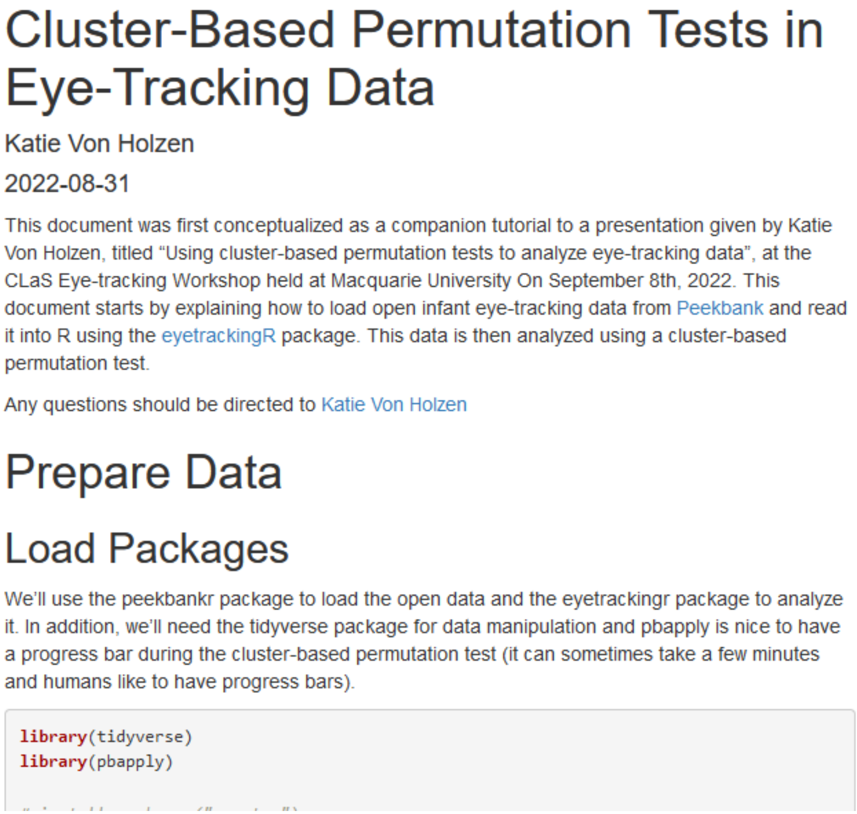
R package for analyzing eye-tracking data

http://www.eyetracking-r.com/



Companion Tutorial

https://kvonholzen.github.io/Macquarie_Presentation_buddy.html



Cluster-Based Permutation Tests in Eye-Tracking Data

Katie Von Holzen

2022-08-31

This document was first conceptualized as a companion tutorial to a presentation given by Katie Von Holzen, titled “Using cluster-based permutation tests to analyze eye-tracking data”, at the CLaS Eye-tracking Workshop held at Macquarie University On September 8th, 2022. This document starts by explaining how to load open infant eye-tracking data from Peekbank and read it into R using the eyetrackingR package. This data is then analyzed using a cluster-based permutation test.

Any questions should be directed to Katie Von Holzen

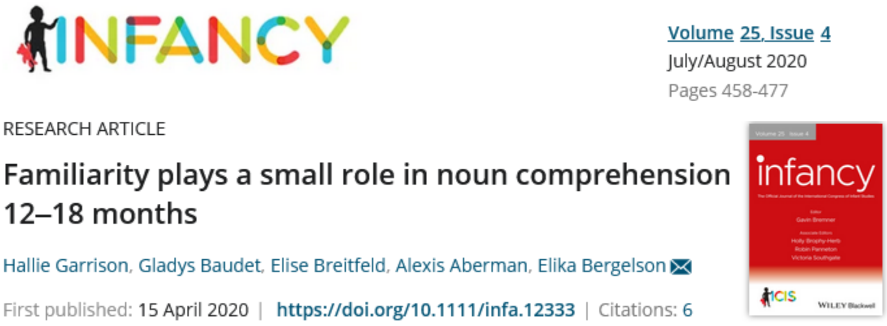
Prepare Data

Load Packages

We’ll use the peekbankr package to load the open data and the eyetrackingr package to analyze it. In addition, we’ll need the tidyverse package for data manipulation and pbapply is nice to have a progress bar during the cluster-based permutation test (it can sometimes take a few minutes and humans like to have progress bars).

```
library(tidyverse)
library(pbapply)
```

HowTo: Example Experiment



INFANCY

Volume 25, Issue 4
July/August 2020
Pages 458-477

RESEARCH ARTICLE

Familiarity plays a small role in noun comprehension 12–18 months

Hallie Garrison, Gladys Baudet, Elise Breitfeld, Alexis Aberman, Elika Bergelson

First published: 15 April 2020 | https://doi.org/10.1111/infa.12333 | Citations: 6

peekbank

A flexible and reproducible interface to developmental eyetracking datasets

Zettersten et al., 2022

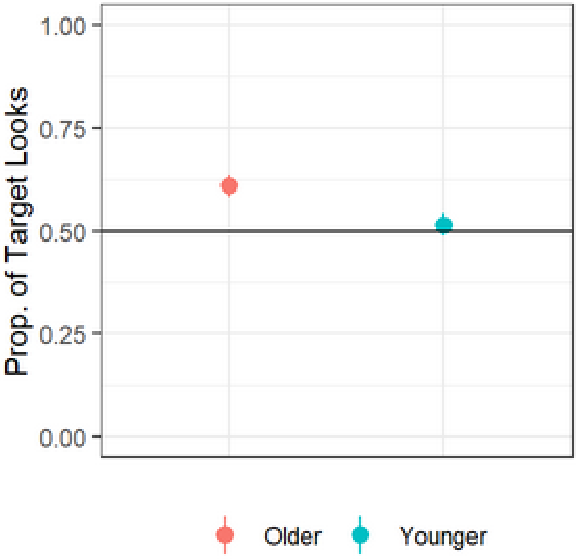
Younger

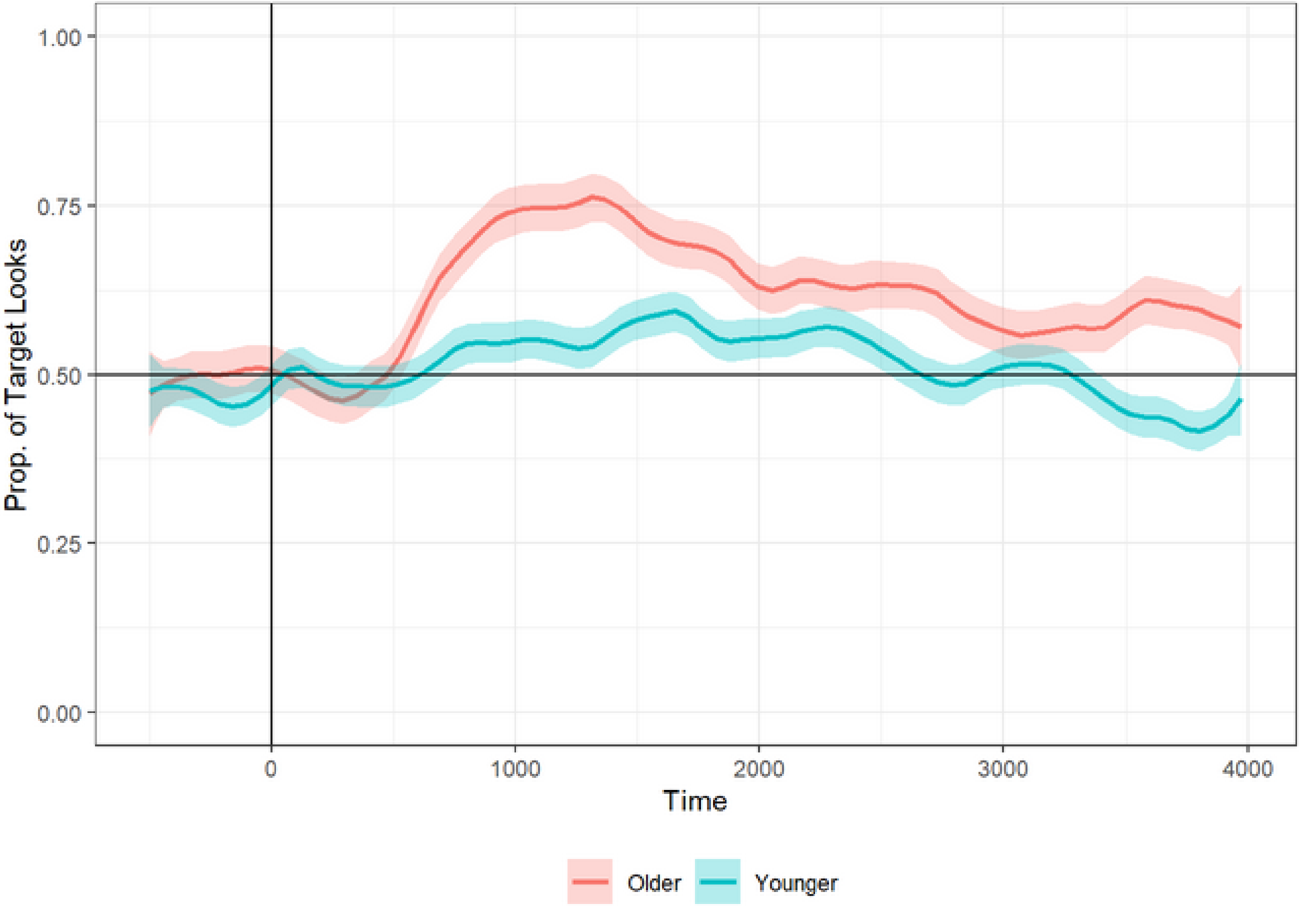
12-13 months

Older

17-18 months

HowTo: Visualize the Data





Cluster-Based Permutation Tests

1. Compare conditions using a statistical test at each time point/bin

2. Identify "time-clusters" of time points that are significant and calculate the sum of the statistics for the time points inside it.

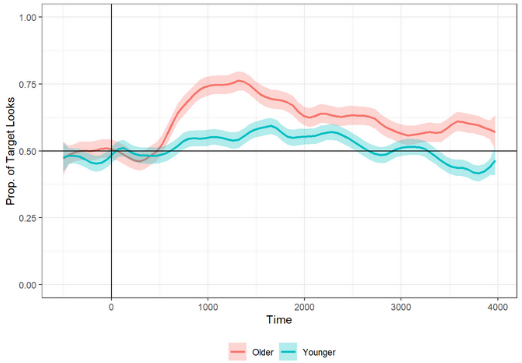
3. Shuffle the data and repeat 1-3 and identify the biggest time-cluster many times (at least 1,000)

4. Compare the cluster statistics from the original data (step 3) with that of the shuffled data (step 4) to calculate a Monte Carlo *p*-value, or the probability that our original cluster occurred by chance.

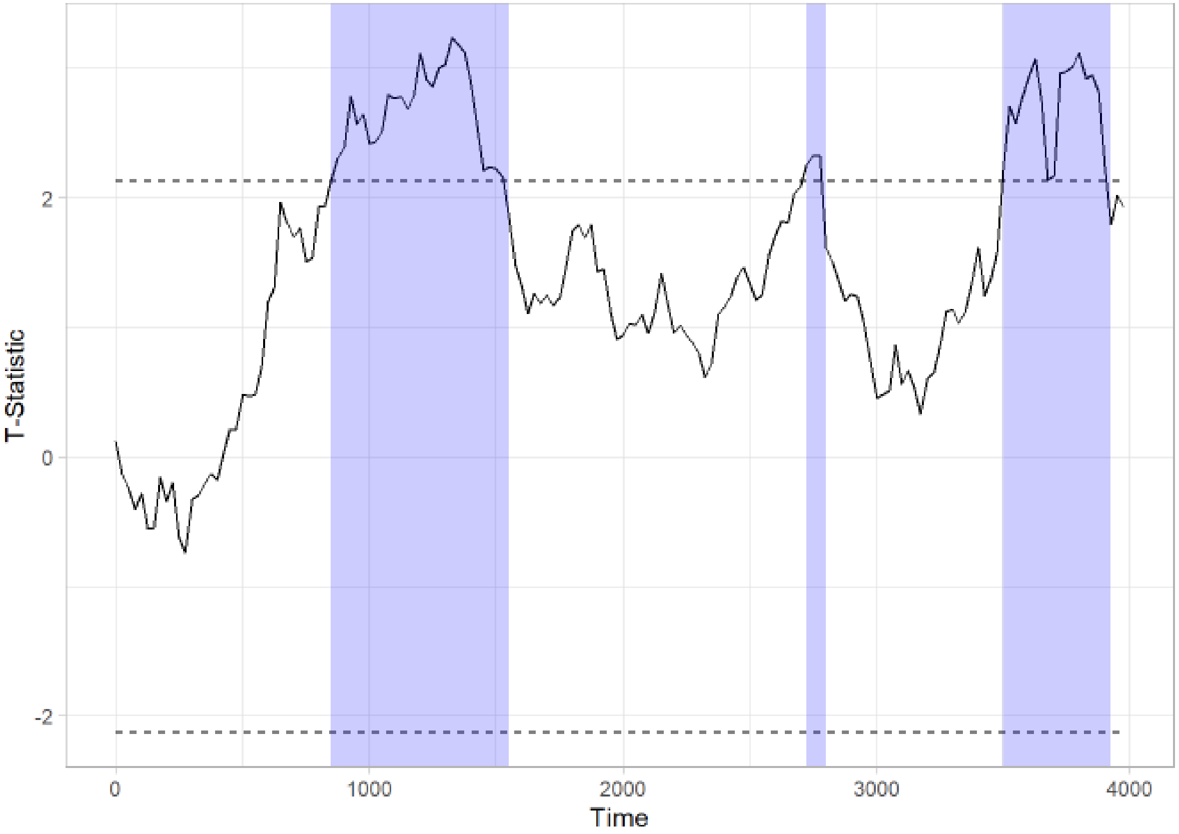
5. Summarize the results

Maris & Oostenveld (2007)

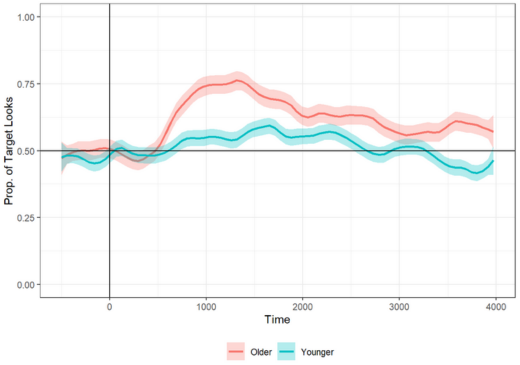
HowTo: Step 1



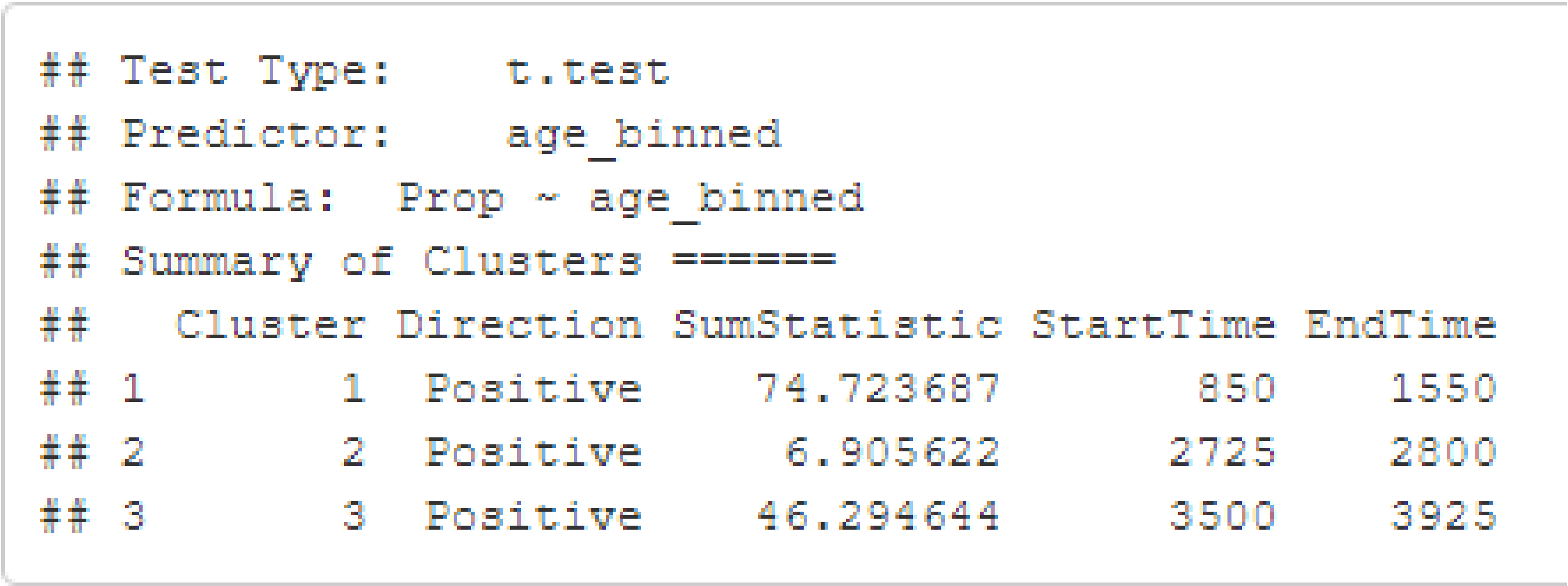
Compare conditions using a statistical test at each time point/bin



HowTo: Step 2

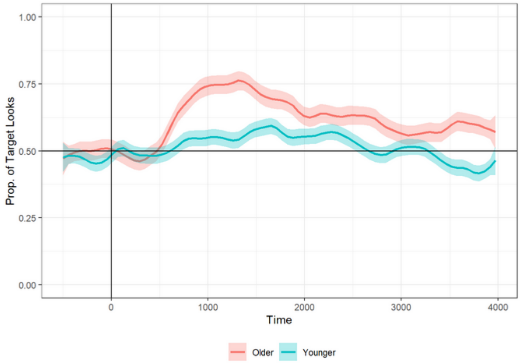


Identify “time-clusters” of time points that are significant and calculate the sum of the statistics for the time points inside it.

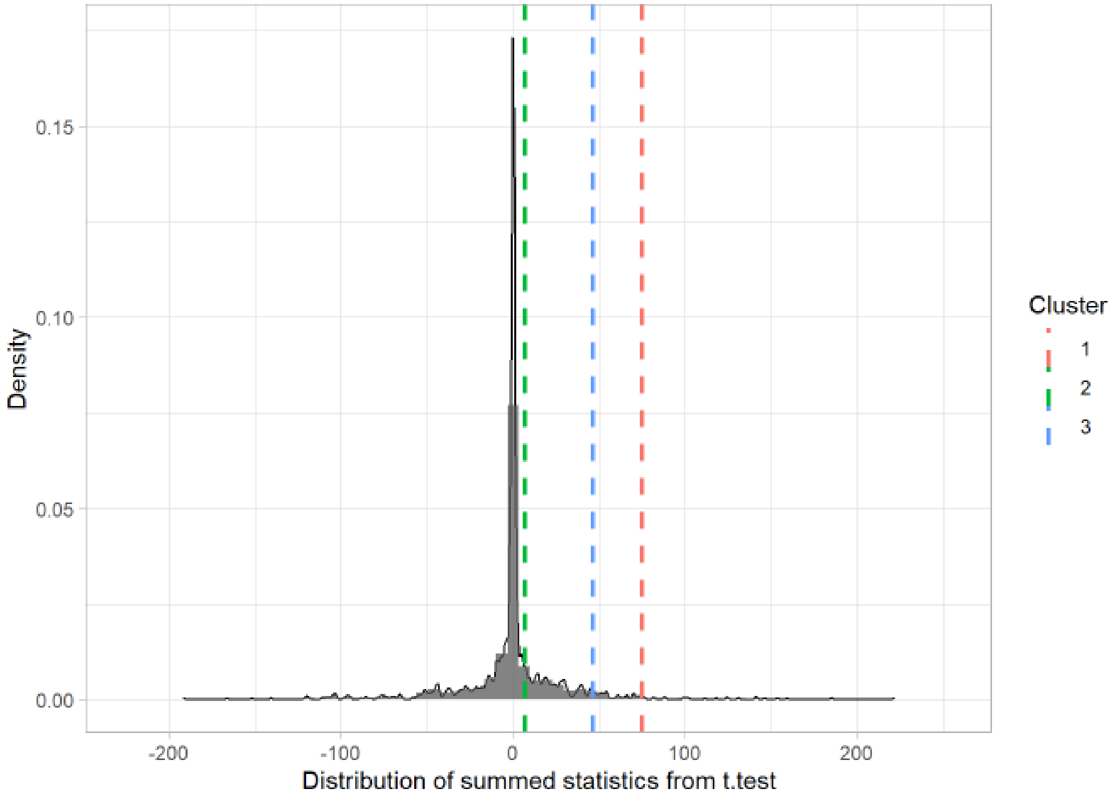


```
## Test Type:     t.test
## Predictor:     age_binned
## Formula:   Prop ~ age_binned
## Summary of Clusters ======
##   Cluster Direction SumStatistic StartTime EndTime
## 1       1  Positive    74.723687       850    1550
## 2       2  Positive     6.905622      2725    2800
## 3       3  Positive    46.294644      3500    3925
```

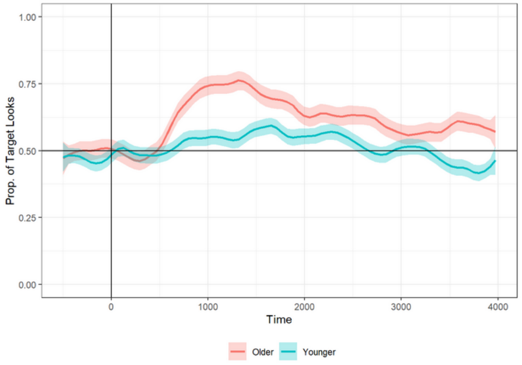
HowTo: Step 3



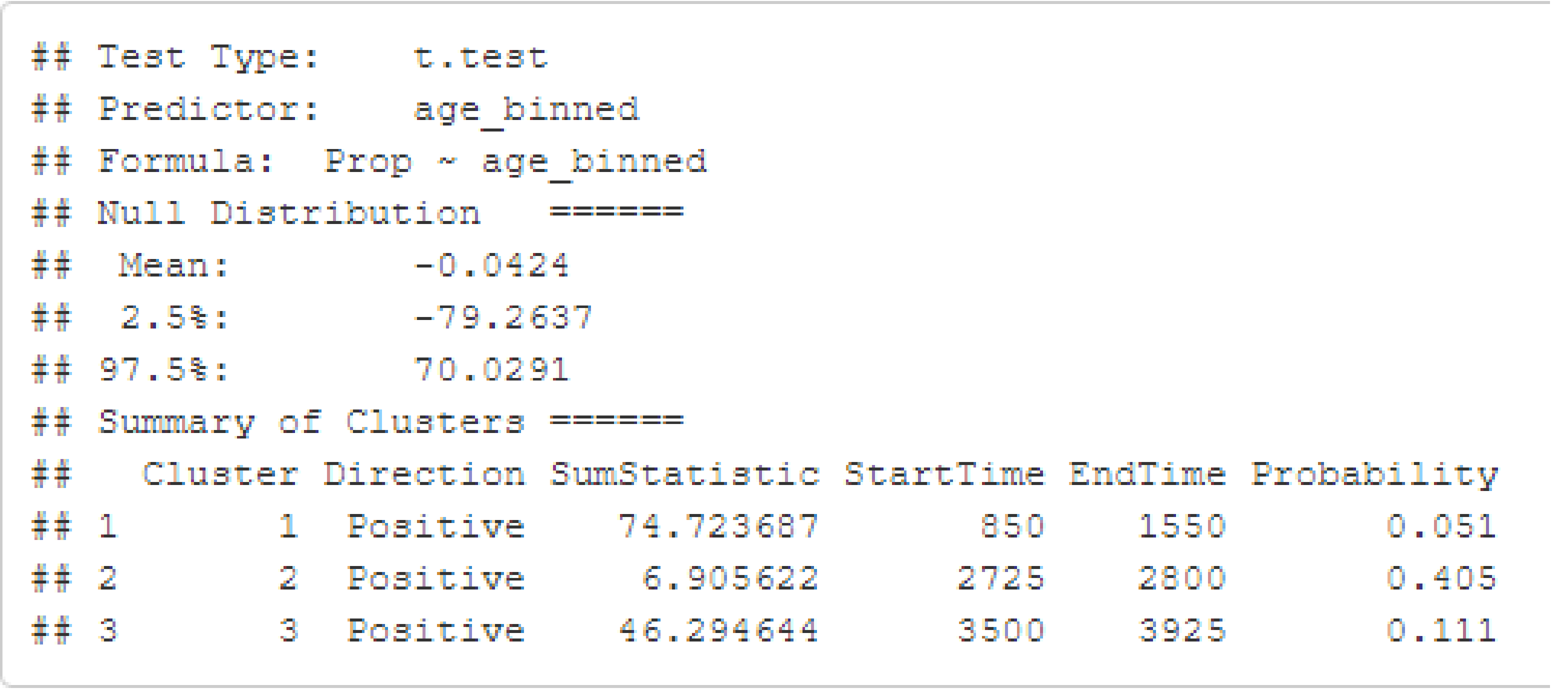
Shuffle the data and repeat 1-3 and identify the biggest time-cluster many times (at least 1,000)



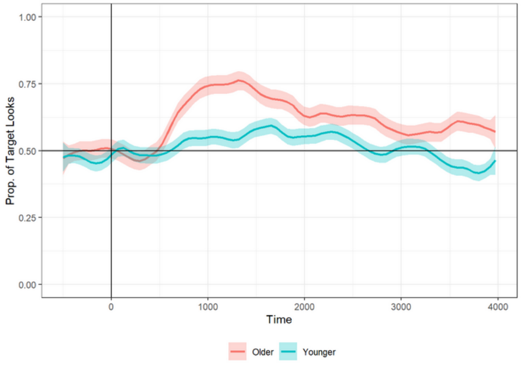
HowTo: Step 4



Compare the cluster statistics from the original data (step 3) with that of the shuffled data (step 4) to calculate a Monte Carlo *p*-value, or the probability that our original cluster occurred by chance.



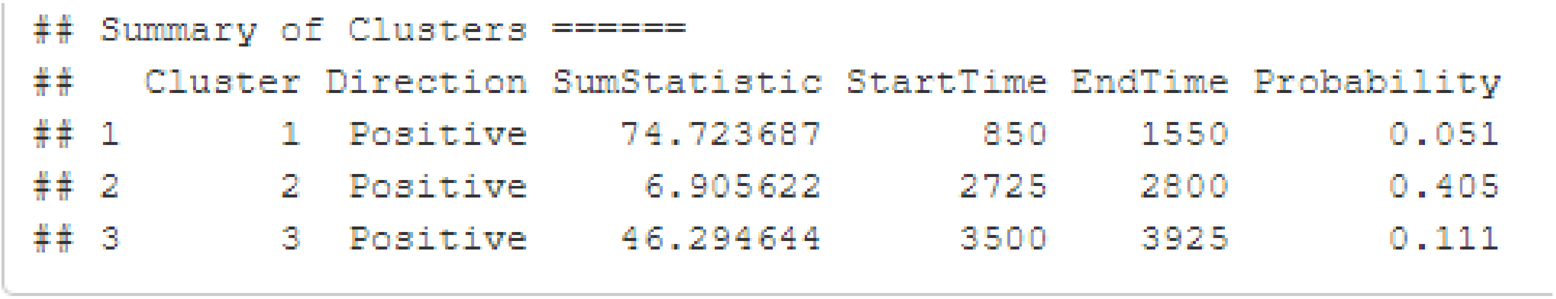
```
## Test Type:     t.test
## Predictor:     age_binned
## Formula:   Prop ~ age_binned
## Null Distribution   ======
##  Mean:            -0.0424
##  2.5%:            -79.2637
## 97.5%:            70.0291
## Summary of Clusters ======
##   Cluster Direction SumStatistic StartTime EndTime Probability
## 1       1  Positive    74.723687       850    1550       0.051
## 2       2  Positive     6.905622      2725    2800       0.405
## 3       3  Positive    46.294644      3500    3925       0.111
```



HowTo: Step 5

Summarize the results

Comparisons using cluster-based permutation analysis revealed that target looking behavior for Older and Younger children significantly deviated from each other between 850 and 1550 ms (cluster t statistic = 74.72, Monte Carlo p *= .05), with reduced target looking for Younger children*



```
## Summary of Clusters ======
##   Cluster Direction SumStatistic StartTime EndTime Probability
## 1       1  Positive    74.723687       850    1550       0.051
## 2       2  Positive     6.905622      2725    2800       0.405
## 3       3  Positive    46.294644      3500    3925       0.111
```

Cluster-Based Permutation Tests

1. Compare conditions using a statistical test at each time point/bin

2. Identify "time-clusters" of time points that are significant and calculate the sum of the statistics for the time points inside it.

3. Shuffle the data and repeat 1-3 and identify the biggest time-cluster many times (at least 1,000)

4. Compare the cluster statistics from the original data (step 3) with that of the shuffled data (step 4) to calculate a Monte Carlo *p*-value, or the probability that our original cluster occurred by chance.

5. Summarize the results

Maris & Oostenveld (2007)

Application

- Use when you want to know where two conditions differ in time

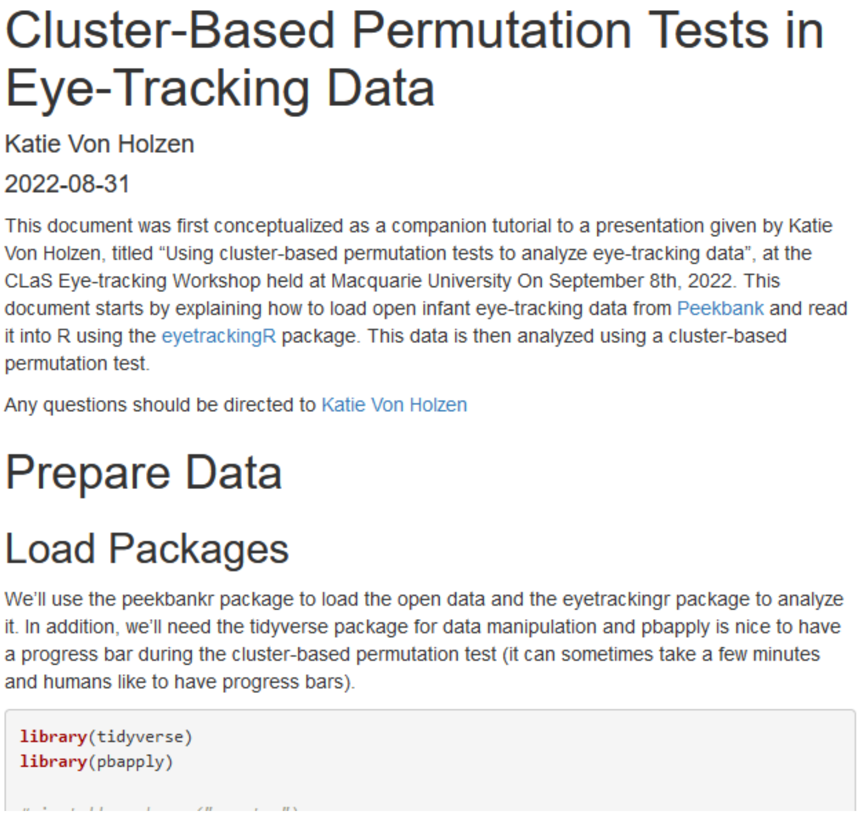
- Best if you have a clear baseline
  - e.g. “unrelated” condition in a priming experiment

- Or a good hypothesis for why two conditions of interest should differ in time

This Photo by Unknown Author is licensed under CC BY-ND

Companion Tutorial

https://kvonholzen.github.io/Macquarie_Presentation_buddy.html



Cluster-Based Permutation Tests in Eye-Tracking Data

Katie Von Holzen

2022-08-31

This document was first conceptualized as a companion tutorial to a presentation given by Katie Von Holzen, titled “Using cluster-based permutation tests to analyze eye-tracking data”, at the CLaS Eye-tracking Workshop held at Macquarie University On September 8th, 2022. This document starts by explaining how to load open infant eye-tracking data from Peekbank and read it into R using the eyetrackingR package. This data is then analyzed using a cluster-based permutation test.

Any questions should be directed to Katie Von Holzen

Prepare Data

Load Packages

We’ll use the peekbankr package to load the open data and the eyetrackingr package to analyze it. In addition, we’ll need the tidyverse package for data manipulation and pbapply is nice to have a progress bar during the cluster-based permutation test (it can sometimes take a few minutes and humans like to have progress bars).

```
library(tidyverse)
library(pbapply)
```

Contact
k.von-holzen@tu-braunschweig.de
katie.m.vonholzen@gmail.com

@KatieVonHolzen