From Raw Eye-Tracking Data to Publishable Results – A Tutorial in R

Dr. Titus von der Malsburg [1,2] Anna Laurinavichyute [1,3]

[1]Dept. Linguistics, University of Potsdam

[2]Dept. Brain and Cognitive Sciences, Massachusetts Institute of Technology

[3]Center for Lanugage and Brain, Higher School of Economics, Moscow

Fri 18, 2018

Outline:

At the design stage of the experiment:

Statistical Power

1. What is power and why does it matter?
2. How to run a power analysis? This is how easy it is.
3. Interpreting the results.
4. Power for main effects vs interactions.

Outline:

After the data collection:

Data Analysis

1. A simple analysis from raw data to plots and inferential stats.
2. Transformation of the DV: Whether or not to transform and if yes how?
3. How to deal with measures that have zeros (e.g., second pass reading time)?
4. Issues with fitting "maximal" models and solutions.

Slides available for download at:

https://tmalsburg.github.io/hse_lecture1.pdf

Why use R?

Why use R?

- Replicability
- Recyclability
- Flexibility
- Efficiency

R packages `tidyverse`:

https://www.tidyverse.org/

- Powerful tools for manipulating and plotting data.
- Written by Hadley Wickham and many others.
- Highly recommended book (freely available online): Grolemund, G., & Wickham, H. (2017). R for data science. Sebastopol, CA 95472, USA: O'Reilly.

To install:

```
install.packages('tidyverse')
```

To load:

```
library(tidyverse)
```

Some tidyverse packages used in this tutorial:

- `readr` Tools for loading all kinds of data formats into R.
- `tidyr` Tools for whipping the data into a convenient shape for the analysis.
- `dplyr` Tools for manipulating data and calculating summary statistics.
- `ggplot2` Most powerful tool for plotting data on earth.

From raw data to dependent variables:

Data used in this tutorial from a German co-registration study (eye-tracking + event-related brain potentials):

- *Der verfallene Bauernhof braucht eine Renovierung.*
- * *Die verfallene Bauernhof masc braucht eine Renovierung.*
- * *Der neugierige Bauernhof masc braucht eine Renovierung.*

English:

- *The$_{masc}$ deteriorating farm$_{masc}$ needs a renovation.*
- * *The$_{fem}$ deteriorating farm$_{masc}$ needs a renovation.*
- * *The$_{masc}$ inquisitive farm$_{masc}$ needs a renovation.*

Results published in:

- Metzner, P., von der Malsburg, T., Vasishth, S., & Rösler, F. (2016). The importance of reading naturally: Evidence from combined recordings of eye movements and electric brain potentials. Cognitive Science, 41(S6), 1232–1263.

R package `edfR` for reading raw eye-tracking data produced by SR-Research trackers (here EyeLink 1000):

https://github.com/jashubbard/edfR

- Originally written by myself and my former student Tobias Günther.
- Maintained and improved by Jason Hubbard (U of Oregon).
- Requires Eyelink Developer's Kit (EDF API), can be found in SR-Research's support forum.

To install:

```
install.packages('devtools')
devtools::install_github('jashubbard/edfR')
```

To load:

```
library(edfR)
```

Read one participant's data:

```
d <- edf.trials("data/s022_1.edf")
```

Inspect the data:

```
fixations <- d$fixations
head(fixations)
```

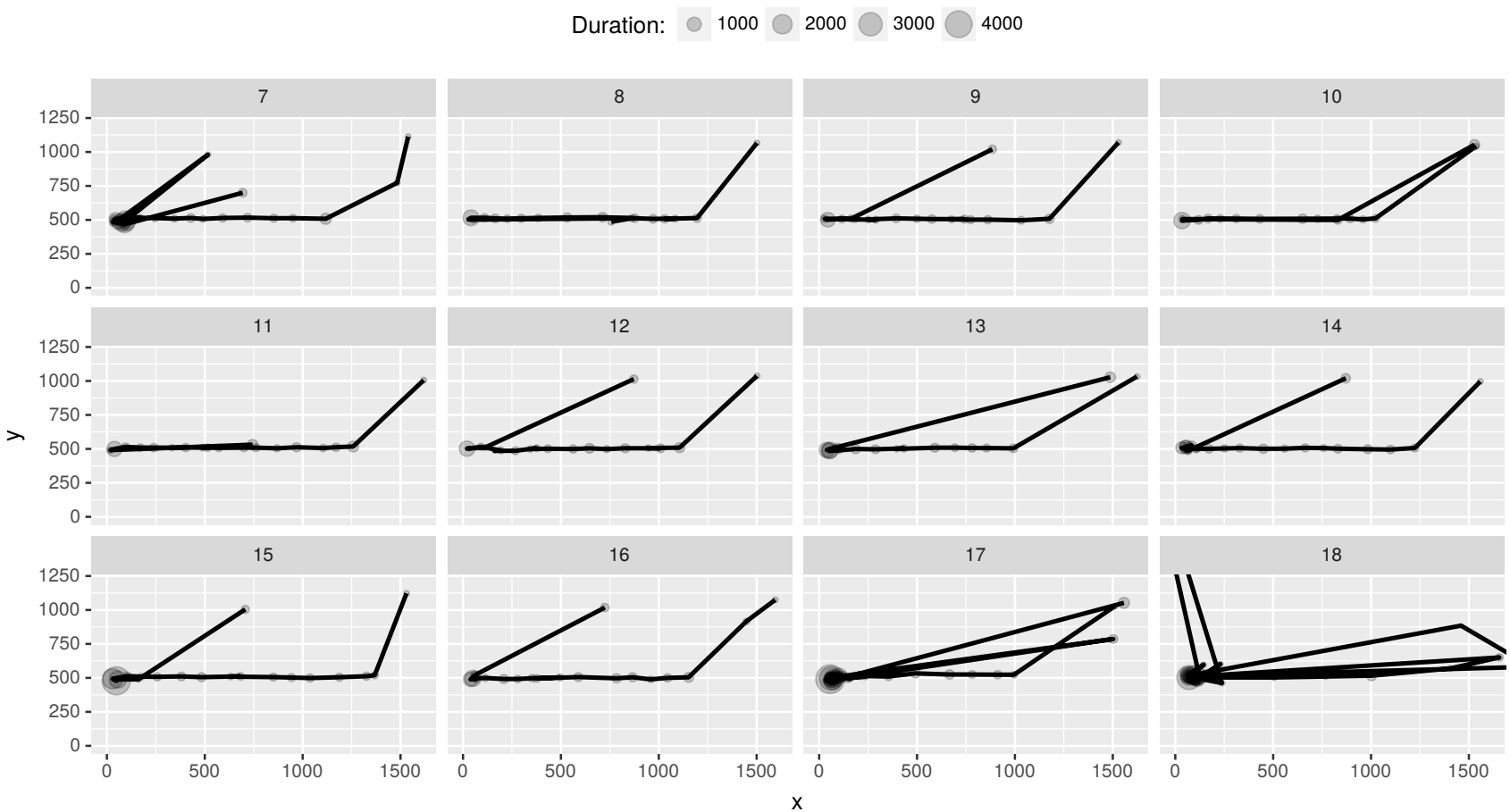
```
   eyetrial  sttime  entime  gavx  gavy
15        1 1871290 1871594 477.2 599.5
19        1 1871664 1871794  79.1 513.5
23        1 1871814 1876678  89.8 521.2
29        1 1876758 1877058 100.3 524.6
35        1 1877180 1880156  64.0 521.2
39        1 1880176 1880436  70.9 517.1
```

Calculate fixations durations:

```
fixations %>%
  mutate(dur = entime - sttime) %>%
  select(-sttime, -entime) -> fixations

head(fixations)
```

```
  eyetrial  gavx  gavy  dur
1        1 477.2 599.5  304
2        1  79.1 513.5  130
3        1  89.8 521.2 4864
4        1 100.3 524.6  300
5        1  64.0 521.2 2976
6        1  70.9 517.1  260
```

```
library(scanpath)

filter(fixations, eyetrial %in% 7:18) %>%
  plot_scanpaths(dur ~ gavx + gavy | eyetrial) +
  coord_cartesian(xlim=c(0, 1600), ylim=c(0, 1200))
```

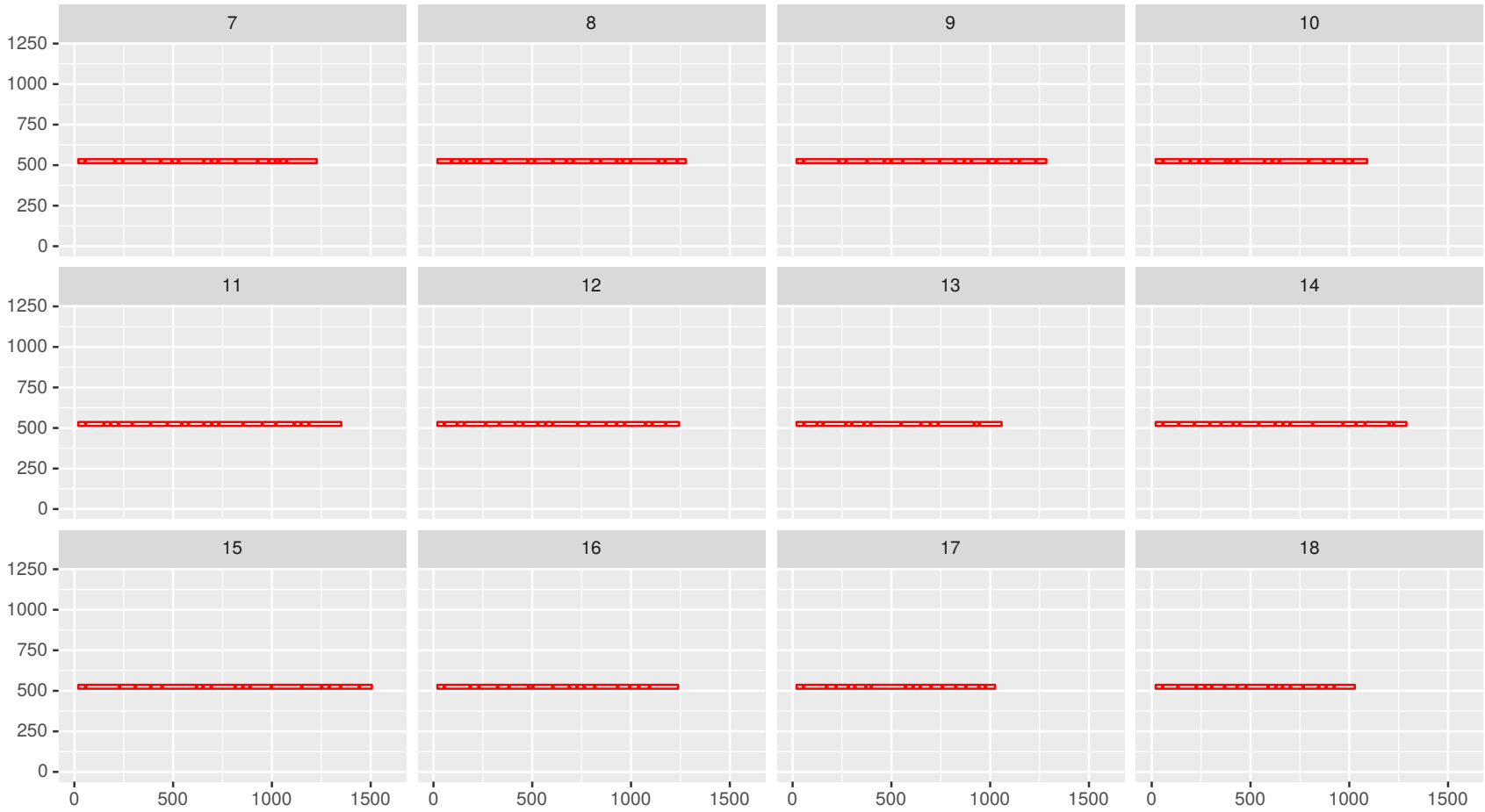


Load regions of interest (ROIs) generated by presentation software (e.g. OpenSesame):

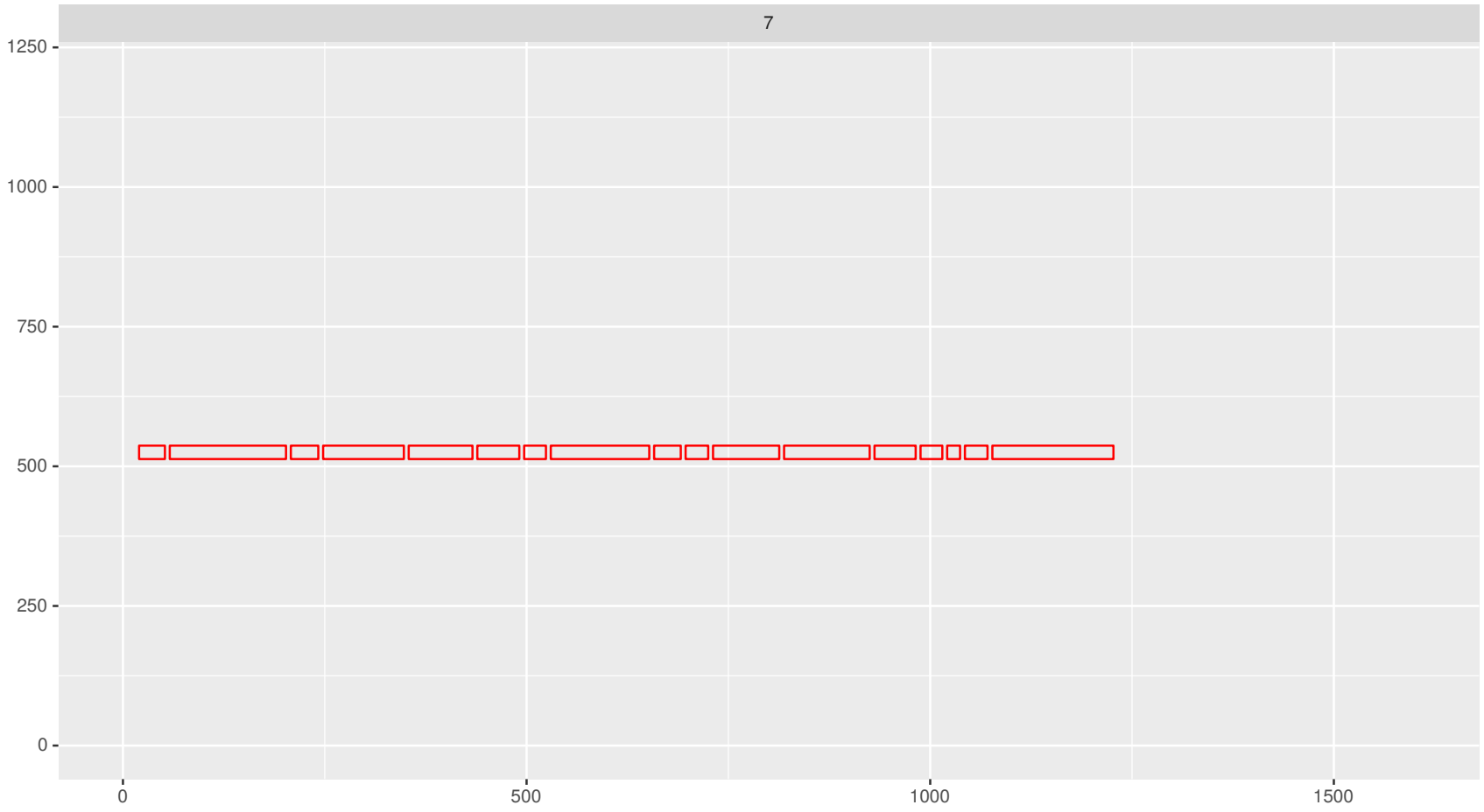
```
head(rois)
```

```
  eyetrial     expt item cond      geom wn  x1  y1  x2  y2      word
1        1 practice    1    - RECTANGLE  1  20 513  63 537      Dies
2        1 practice    1    - RECTANGLE  2  69 513  91 537       ist
3        1 practice    1    - RECTANGLE  3  97 513 125 537       ein
4        1 practice    1    - RECTANGLE  4 131 513 216 537 Testsatz,
5        1 practice    1    - RECTANGLE  5 222 513 241 537        er
6        1 practice    1    - RECTANGLE  6 247 513 285 537      wird
```

```
filter(rois, eyetrial%in% 7:18) %>%
  ggplot(aes(xmin=x1, xmax=x2, ymin=y1, ymax=y2)) +
  geom_rect(color="red", fill=NA) +
  coord_cartesian(xlim=c(0, 1600), ylim=c(0, 1200)) +
  facet_wrap(~eyetrial)
```



```
filter(rois, eyetrial == 7) %>%
  ggplot(aes(xmin=x1, xmax=x2, ymin=y1, ymax=y2)) +
  geom_rect(color="red", fill=NA) +
  coord_cartesian(xlim=c(0, 1600), ylim=c(0, 1200)) +
  facet_wrap(~eyetrial)
```



Mapping fixations to ROIs:

Install `intervals` package:

```
install.packages("intervals")
```

Load helper function for mapping fixations to ROIs:

```
source("R/map_fixations.function.R")
```

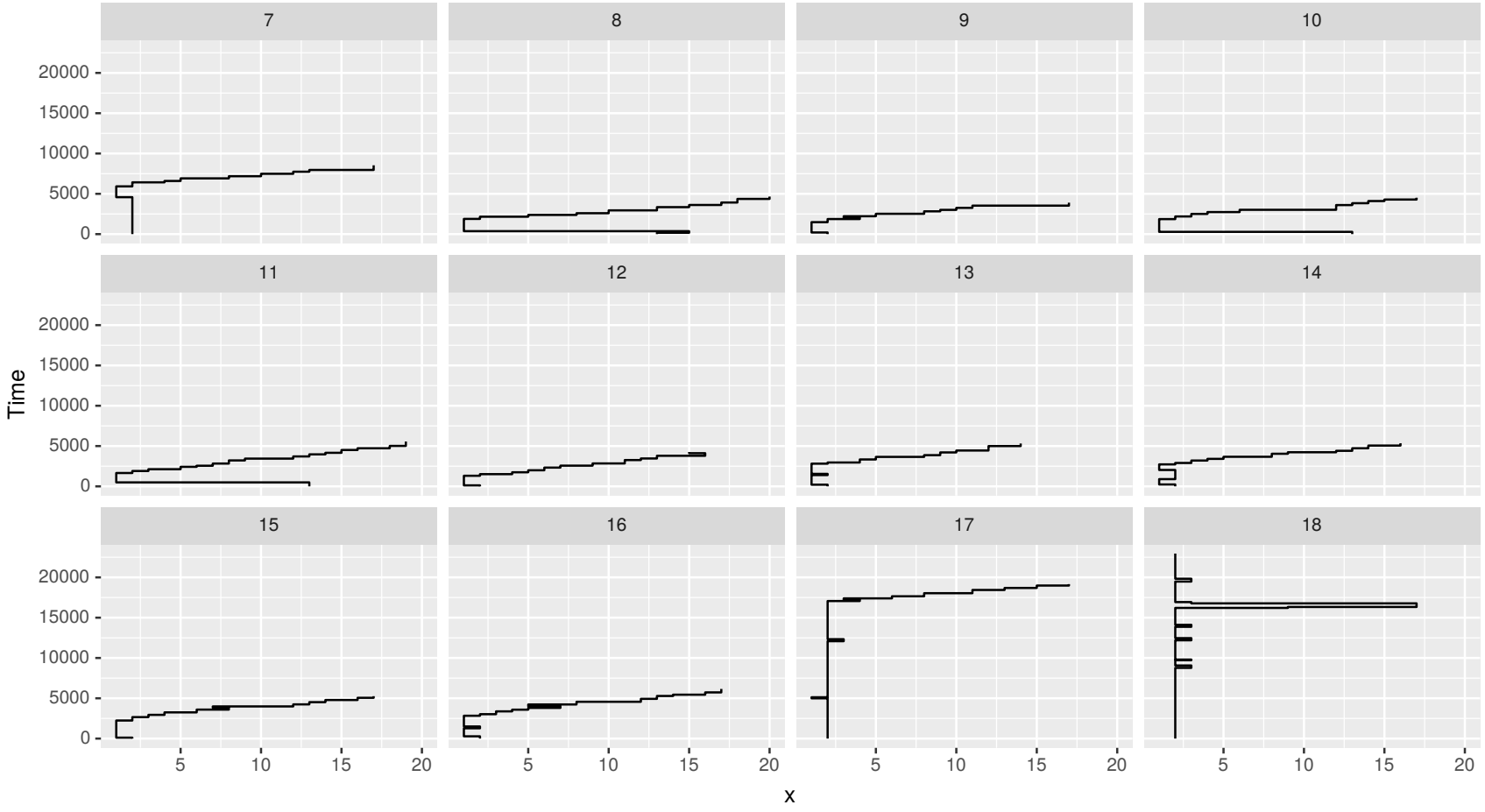
```
fixations <- cbind(fixations,
                   map_fixations(fixations, rois))
head(fixations, 10)
```

```
   eyetrial  gavx  gavy  dur wn word
1         1 477.2 599.5  304 NA <NA>
2         1  79.1 513.5  130  2  ist
3         1  89.8 521.2 4864  2  ist
4         1 100.3 524.6  300  3  ein
5         1  64.0 521.2 2976 NA <NA>
6         1  70.9 517.1  260  2  ist
7         1  72.0 495.4  212 NA <NA>
8         1  45.2 496.4  680 NA <NA>
9         1 225.7 511.1  266 NA <NA>
10        1 151.6 505.6  302 NA <NA>
```

```
fixations <- cbind(fixations,
                   map_fixations(fixations, rois, ybuffer=30))
head(fixations, 10)
```

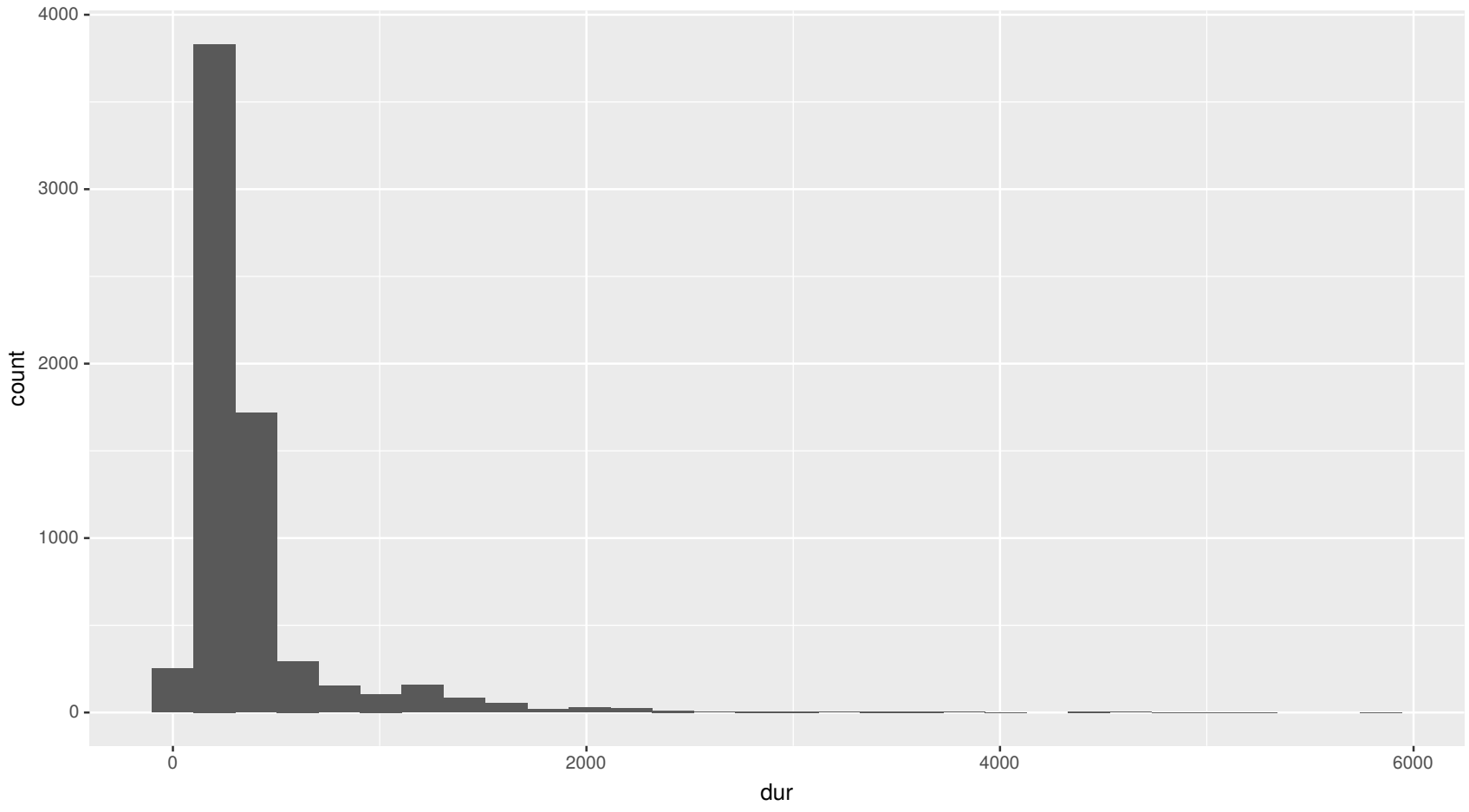
```
   eyetrial  gavx  gavy  dur wn       word
1         1 477.2 599.5  304 NA       <NA>
2         1  79.1 513.5  130  2        ist
3         1  89.8 521.2 4864  2        ist
4         1 100.3 524.6  300  3        ein
5         1  64.0 521.2 2976 NA       <NA>
6         1  70.9 517.1  260  2        ist
7         1  72.0 495.4  212  2        ist
8         1  45.2 496.4  680  1       Dies
9         1 225.7 511.1  266  5         er
10        1 151.6 505.6  302  4 Testsatz,
```

```
fixations %>%
  filter(eyetrial %in% 7:18, !is.na(wn)) %>%
  plot_scanpaths(dur ~ wn | eyetrial)
```

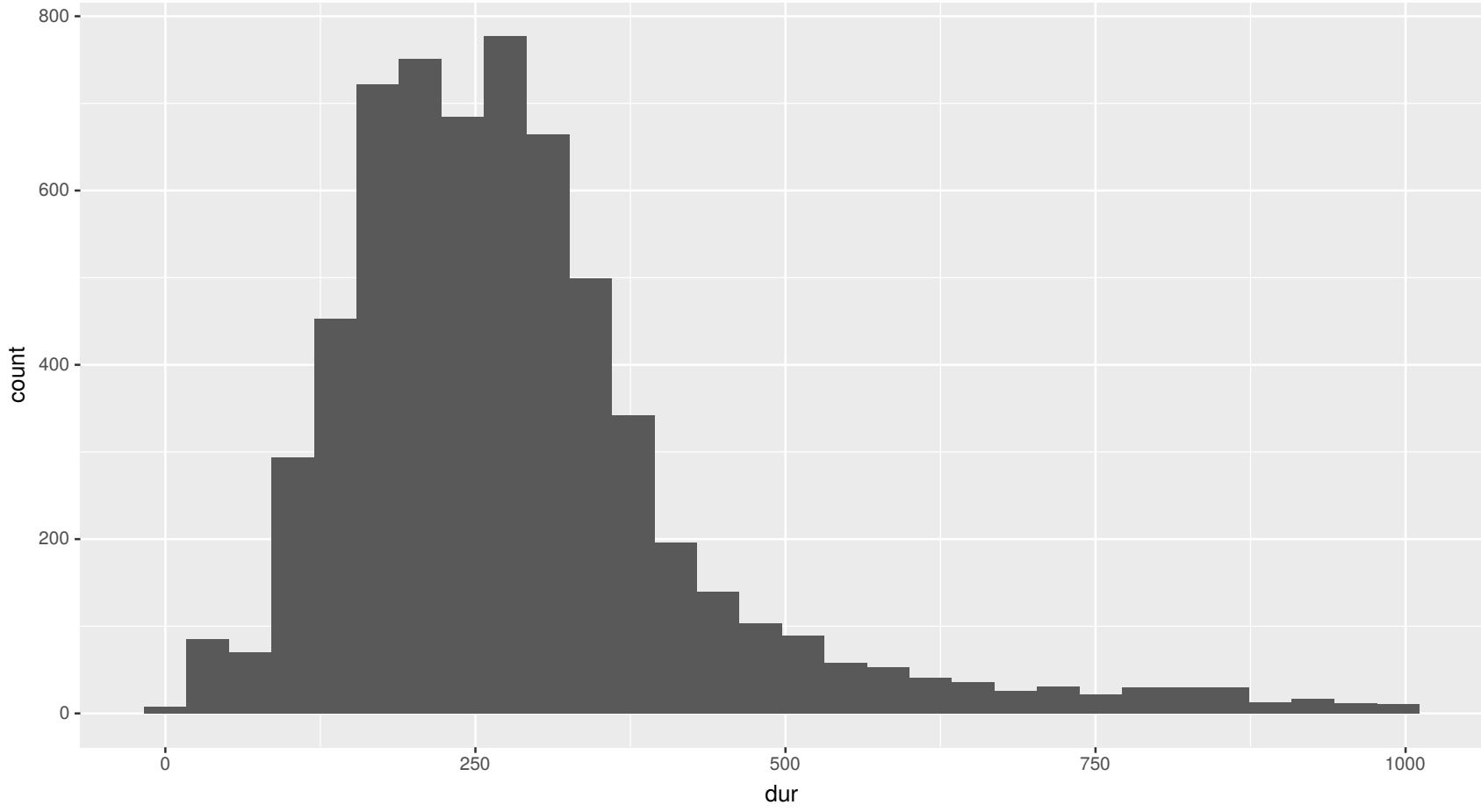


Removing outliers:

```
ggplot(fixations, aes(x=dur)) + geom_histogram()
```



```
fixations %>%
  filter(dur<1000) -> fixations

ggplot(fixations, aes(x=dur)) + geom_histogram()
```



Write this participant's preprocessed data to disk:

```
write_tsv(fixations, "data/s022_1.edf.fix")
```

Trial-level information (generate by presentation software):

```
# A tibble: 10 x 4
   expt       item cond  qacc
   <chr>     <int> <chr> <chr>
 1 practice      1 -     1
 2 practice      4 -     .
 3 practice      4 -     1
 4 practice      4 -     1
 5 practice      4 -     1
 6 practice      4 -     1
 7 judith       70 a     1
 8 filler      180 -     1
 9 filler      147 -     .
10 judith        3 f     1
```

Loading data from all participants:

```
fix.files <- list.files("data/", ".edf.fix", full.names=TR
txt.files <- list.files("data/", ".txt", full.names=TRUE)

l <- list()
for (i in 1:55) {
  fixations   <- read_tsv(fix.files[[i]])
  trial.infos <- read_tsv(txt.files[[i]])

  l[[i]] <- inner_join(fixations, trial.infos)
}

# Combine all data frames (one for each participant):
all.fixations <- do.call(rbind, l)
```

```
head(all.fixations)
```

```
# A tibble: 6 x 11
  eyetrial    wn word          x     y   dur subj   expt       item c
     <int> <int> <chr>     <int> <int> <int> <chr>  <chr>     <int> <
1        1    NA <NA>         NA   508   168 s001_3 practice      1 -
2        1     1 Dies         37   508   178 s001_3 practice      1 -
3        1     2 ist          86   509   164 s001_3 practice      1 -
4        1     1 Dies         53   511   252 s001_3 practice      1 -
5        1     4 Testsatz,   205   517   318 s001_3 practice      1 -
6        1     6 wird        277   516   178 s001_3 practice      1 -
```

Calculating the canonical eye-tracking measures:

R package `em2` for calculating eye-tracking measures common in reading research:

https://tmalsburg.github.io/downloads/em2_0.9.tar.gz

- Written by Pavel Logačev (Bogazici University, Turkey).

To install:

```
install.packages(
  "https://tmalsburg.github.io/downloads/em2_0.9.tar.gz",
  repos=NULL, method="libcurl")
```

To load:

```
library(em2)
```

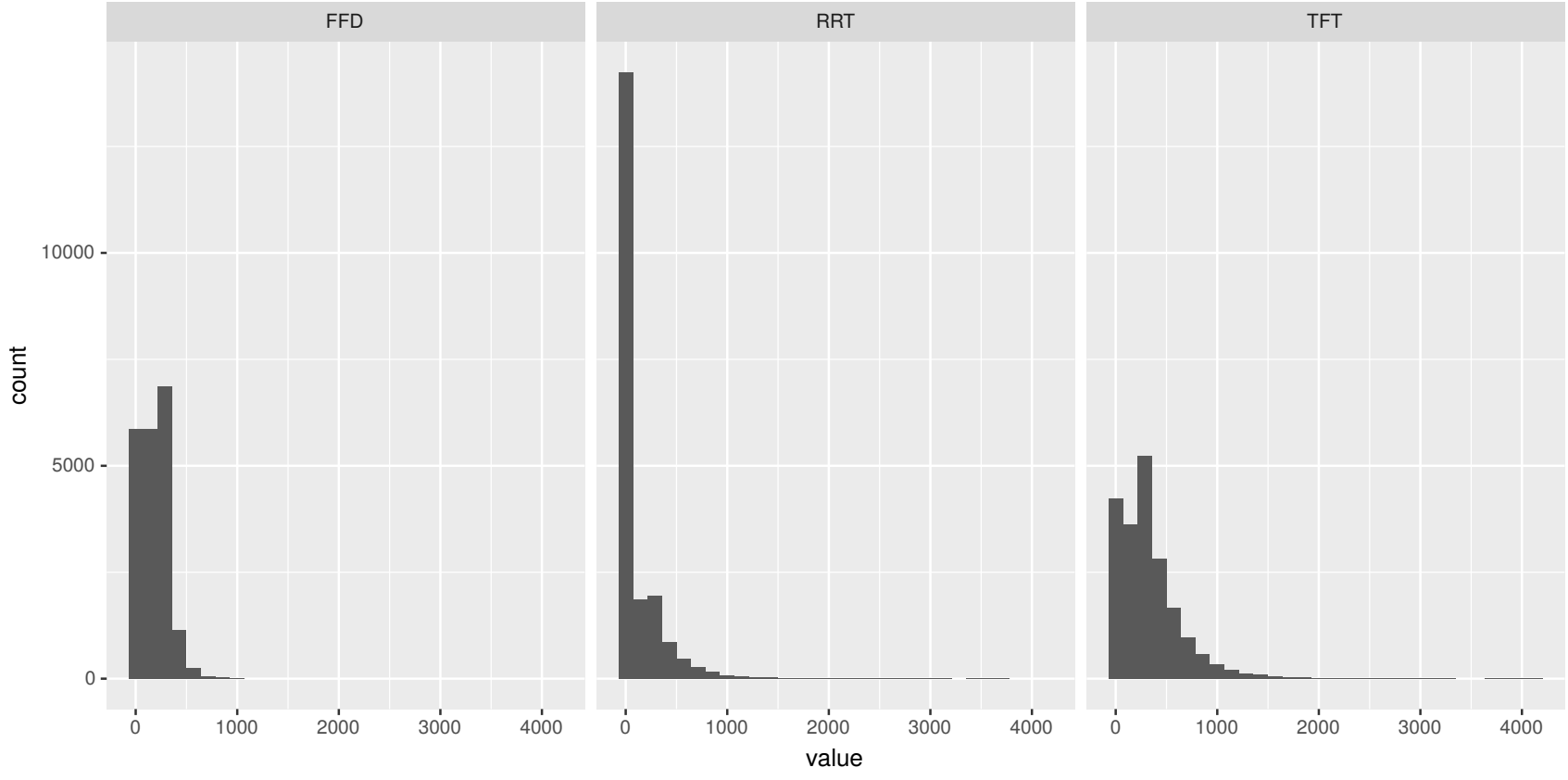
```
et.measures <- em2(all.fixations$wn, all.fixations$dur,
                   select(all.fixations, eyetrial, subj,
                          expt, item, cond))

head(et.measures)
```

```
  eyetrial   subj     expt item cond roi FFD FFP SFD FPRT
1        1 s001_3 practice    1    -   1 178   1   0  178
2        1 s001_3 practice    1    -   2 164   1 164  164
3        1 s001_3 practice    1    -   3   0   0   0    0
4        1 s001_3 practice    1    -   4 318   1 318  318
5        1 s001_3 practice    1    -   5   0   0   0    0
6        1 s001_3 practice    1    -   6 178   1 178  178
```

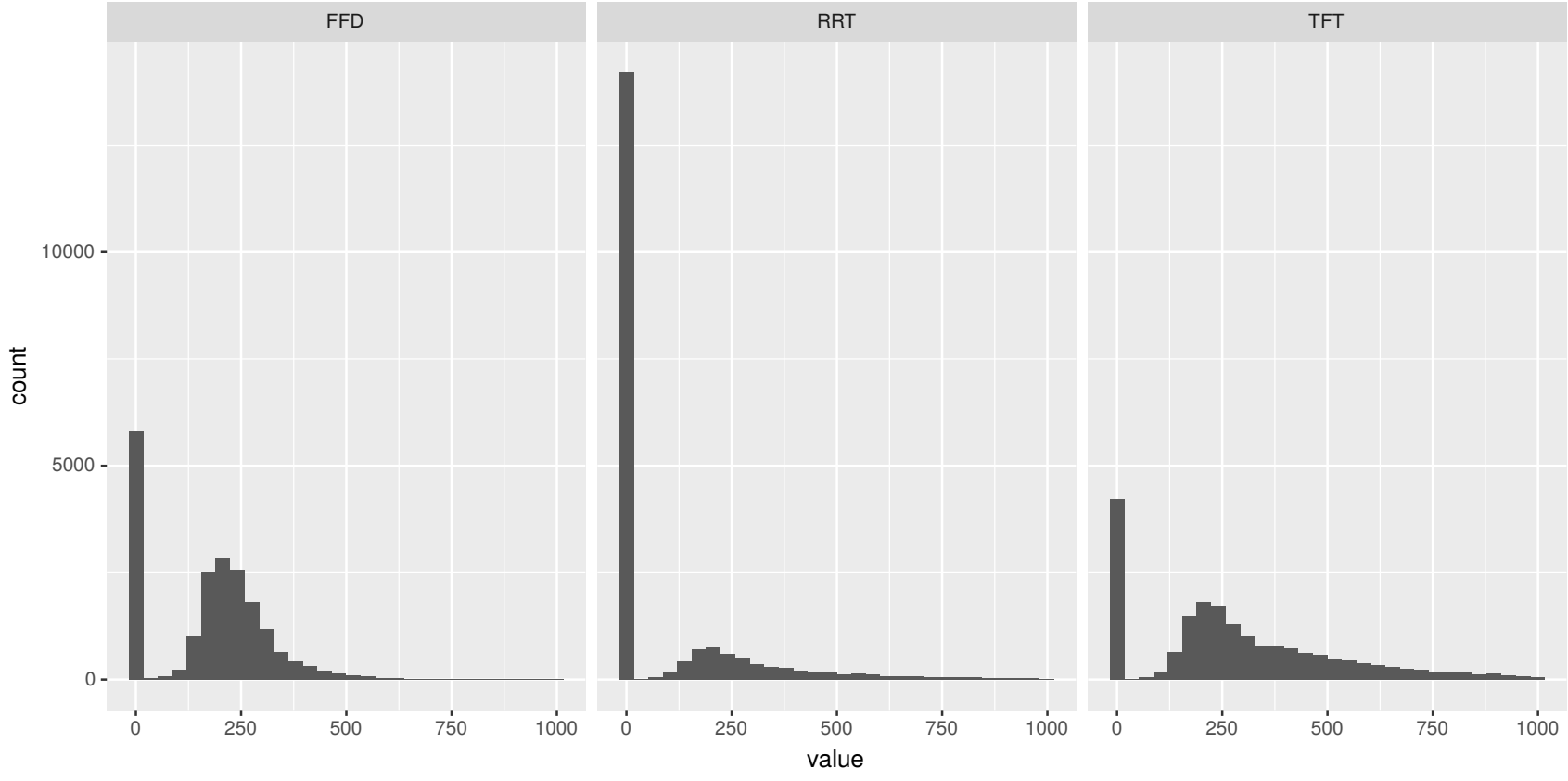
Cleaning up after using `em2`:

```
detach("package:em2", unload=TRUE)
detach("package:dplyr", unload=TRUE)
library(dplyr)
```

```
gather(et.measures, measure, value, 7:20) %>%
  filter(roi == 3,
         measure %in% c("FFD", "TFT", "RRT")) %>%
  ggplot(aes(x=value)) +
  geom_histogram() + facet_wrap(~measure)
```



```
gather(et.measures, measure, value, 7:20) %>%
  filter(roi == 3, value < 1000,
         measure %in% c("FFD", "TFT", "RRT")) %>%
  ggplot(aes(x=value)) +
  geom_histogram() + facet_wrap(~measure)
```



Descriptive stats, tables and plots:

First gather, then summarize. (Don't gather in next step.)

Calculating means for three measures:

```
et.measures %>%
  filter(expt == "judith",
         roi  == 3,
         cond %in% c("a", "b", "c")) %>%
  group_by(cond) %>%
  summarize(
    mean.FPRT = mean(FPRT),
    mean.RRT  = mean(RRT),
    mean.TFT  = mean(TFT)) -> means
```

Producing tables in LaTeX format:

```
library(xtable)

tab <- xtable(means, digits=0)
print(tab, include.rownames=FALSE)
```

cond	mean.FPRT	mean.RRT	mean.TFT
a	121	59	180
b	312	176	488
c	115	72	187

Calculating means and confidence intervals:

```
et.measures %>%
  filter(expt=="judith",
         roi==3,
         cond %in% c("a", "b", "c")) %>%
  group_by(cond, subj) %>%
  summarize(
    FPRT.m = mean(FPRT),
    RRT.m  = mean(RRT),
    TFT.m  = mean(TFT)) -> by.subject.means
```

```
head(by.subject.means)
```

```
# A tibble: 6 x 5
# Groups:   cond [1]
  cond  subj   FPRT.m RRT.m TFT.m
  <fct> <fct>   <dbl> <dbl> <dbl>
1 a     s001_3  122    30.8   153
2 a     s002_1   99.5  12.7   112
3 a     s003_1   89.7  66.6   156
4 a     s004_1   29.1  93.6   123
5 a     s005_1  196    20.1   216
6 a     s006_1  127    19.0   146
```

```
by.subject.means %>%
  gather(measure, value, 3:5) %>%
  group_by(measure, cond) %>%
  summarize(
    gm = mean(value),
    ci.lower = gm - 2*sd(value)/sqrt(n()),
    ci.upper = gm + 2*sd(value)/sqrt(n())) ->
  grand.means
```

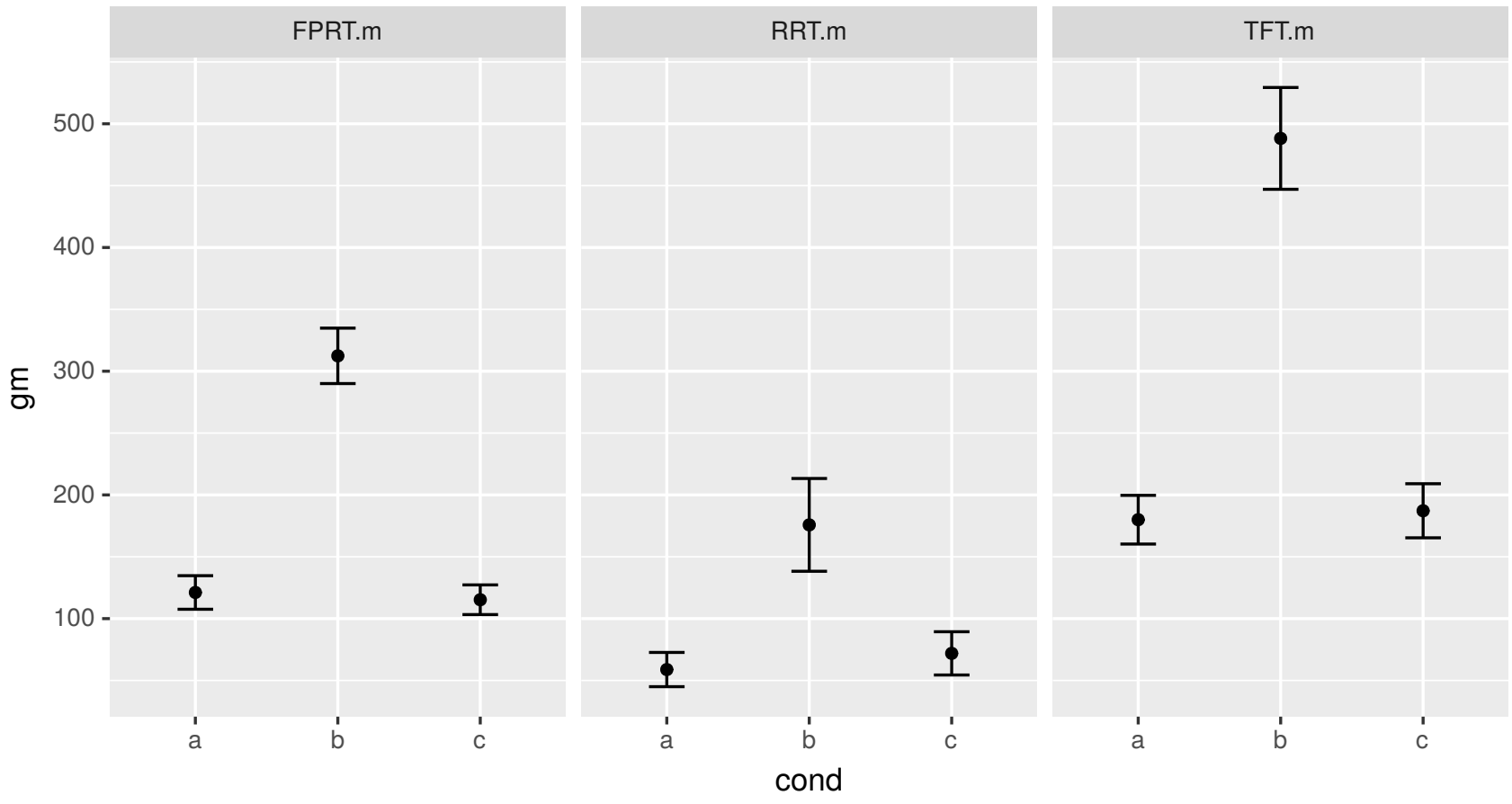
```
grand.means
```

```
# A tibble: 9 x 5
# Groups:   measure [?]
  measure cond     gm ci.lower ci.upper
  <chr>   <fct> <dbl>    <dbl>    <dbl>
1 FPRT.m  a       121      108      135
2 FPRT.m  b       312      290      335
3 FPRT.m  c       115      103      127
4 RRT.m   a        58.8     45.0     72.7
5 RRT.m   b       176      138      213
6 RRT.m   c        71.9     54.4     89.4
7 TFT.m   a       180      160      200
8 TFT.m   b       488      447      529
9 TFT.m   c       187      165      209
```

```
tab <- xtable(grand.means, digits=0)
print(tab, include.rownames=FALSE)
```

measure	cond	gm	ci.lower	ci.upper
FPRT.m	a	121	108	135
FPRT.m	b	312	290	335
FPRT.m	c	115	103	127
RRT.m	a	59	45	73
RRT.m	b	176	138	213
RRT.m	c	72	54	89
TFT.m	a	180	160	200
TFT.m	b	488	447	529
TFT.m	c	187	165	209

```
ggplot(grand.means, aes(x=cond, y=gm)) +
  geom_point() +
  geom_errorbar(aes(ymin=ci.lower, ymax=ci.upper),
                width=0.25) +
  facet_wrap(~measure)
```



Inferential stats:

Comparison of conditions a and b:

```
et.measures %>%
  filter(expt == "judith",
         roi  == 3,
         cond %in% c("a", "b")) %>%
  select(subj, item, cond, FPRT) %>%
  droplevels() -> d
head(d)
```

```
    subj item cond FPRT
1 s001_3  158    b  474
2 s001_3  194    b  680
3 s001_3   38    b  228
4 s001_3    8    b  262
5 s001_3   68    b  202
6 s001_3  122    b  196
```

R package `brms`:

https://github.com/paul-buerkner/brms

- Linear mixed effects model, similar to `lme4` but Bayesian and much more powerful.
- Uses the Stan system for Bayesian inference behind the scenes.
- Developed by Paul Bürkner.

To install:

```
devtools::install_github("paul-buerkner/brms",
                         dependencies = TRUE)
```

To load:

```
library(brms)
```

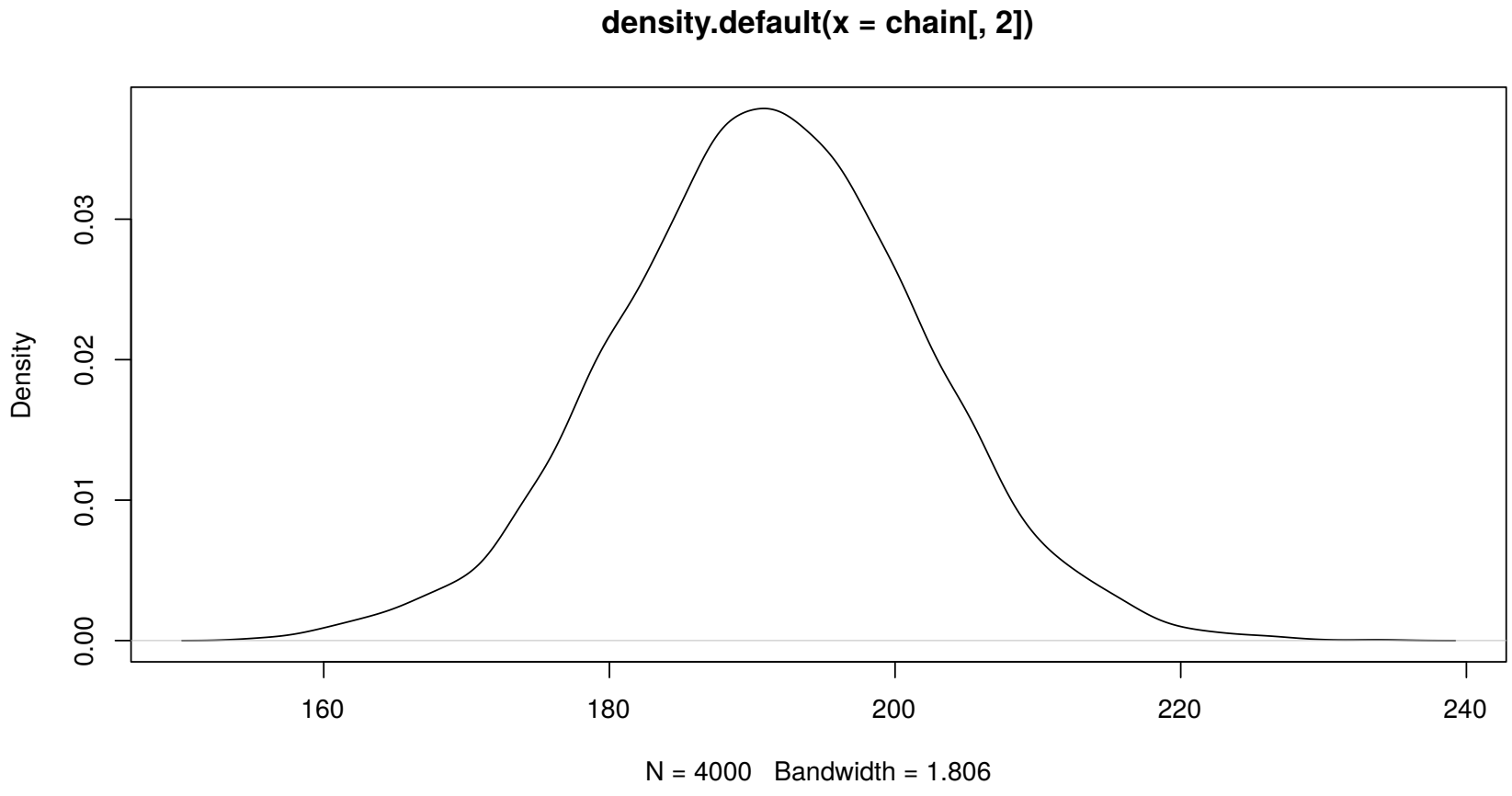
```
options("mc.cores" = 4)
m1 <- brm(FPRT ~ cond + (cond|subj) + (cond|item), d)
```

```
summary(m1)
```

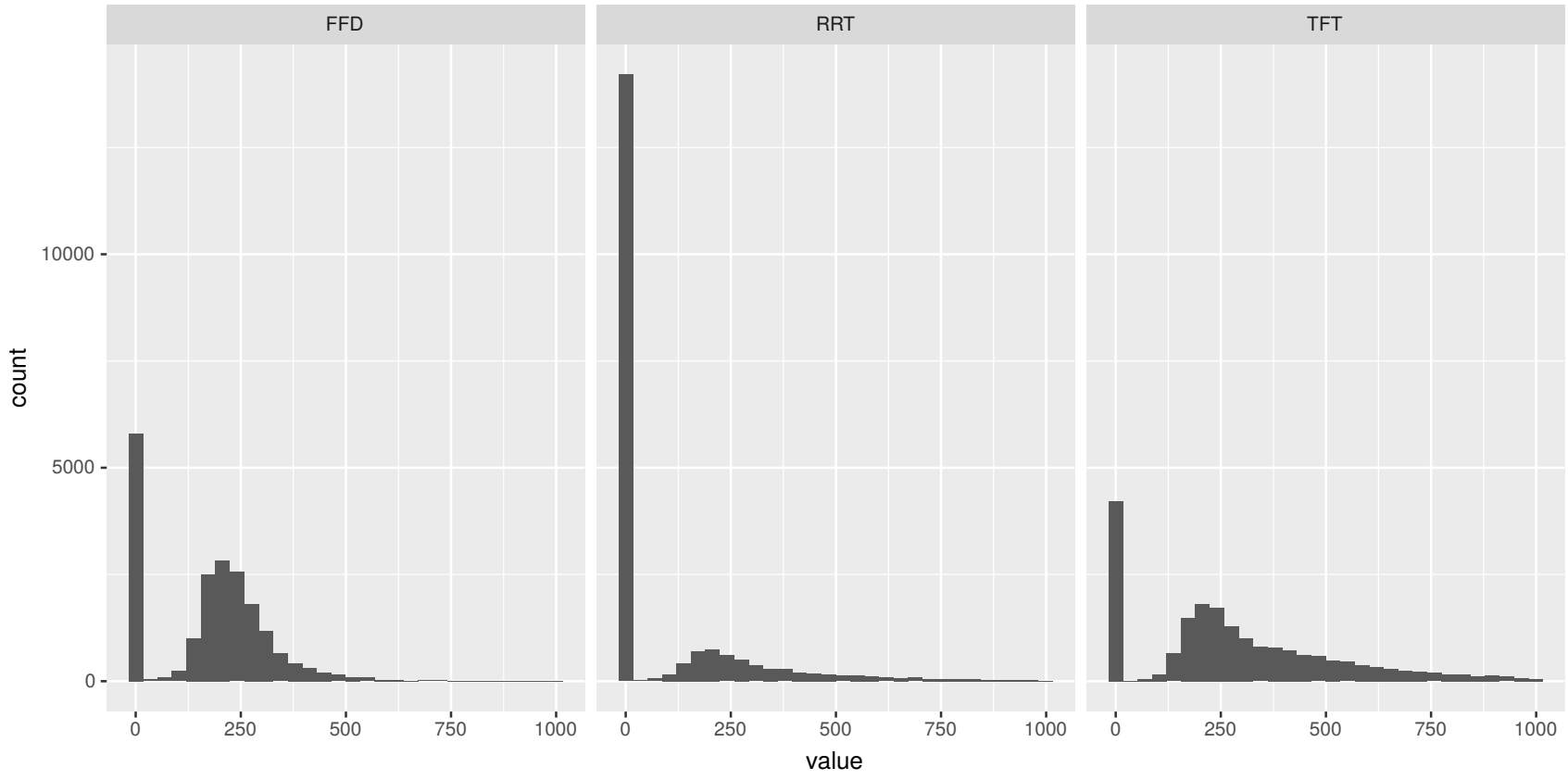
```
Population-Level Effects:
          Estimate Est.Error l-95% CI u-95% CI Eff.Sample
Intercept   216.76      8.83   198.84   233.70        1019
cond2M1     191.19     10.96   169.81   212.46        1782
```

Posterior density of the paramter capturing the difference between conditions:

```
chain <- as.mcmc(m1, combine_chains=TRUE)
plot(density(chain[,2]))
```



Issues and potential pitfalls:



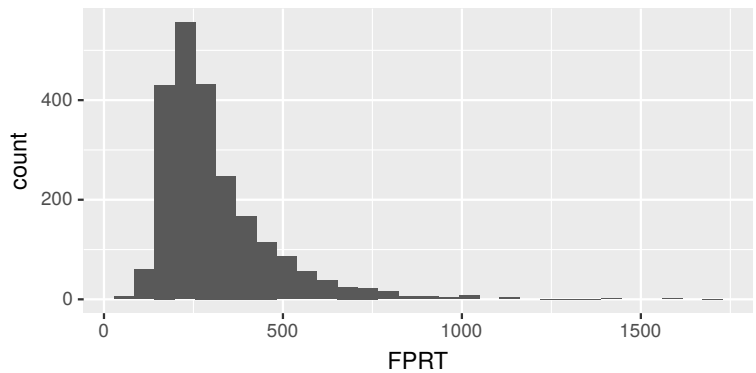
- Zeroes
- Non-normal distribution

How to deal with measures that contain zeroes?

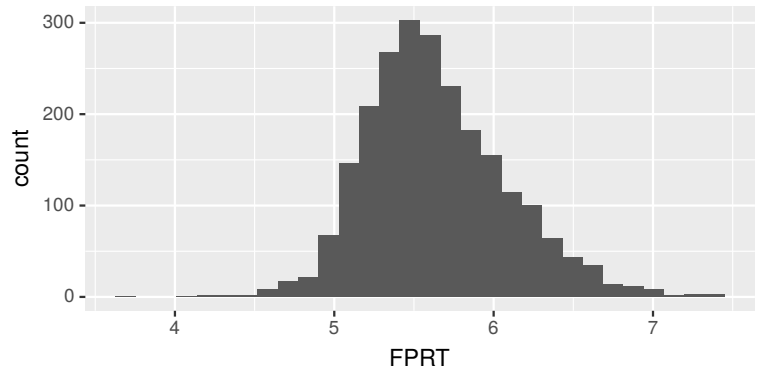
Fit two models:

1. Like before but only for non-zero values.
2. Additional models testing whether the value was more often zero in one condition than in the other.

Transformation of the dependent variables: Why, when, and how?



Raw FPRTs on the *ms* scale



Log-transformed FPRTs

```
m2 <- brm(log(FPRT) ~ cond + (cond|subj) + (cond|item), d)
save(m2, file="data/m2.Rda")
```

Maximal random effects structures: How to deal with non-converging models?

Other useful packages:

R package `saccades`:

https://github.com/tmalsburg/saccades

- Can be used if you don't want to rely on black-box algo offered my eye-tracker manufacturer.
- Algorithm for detecting saccades and fixations proposed by: Engbert, R., & Kliegl, R. (2003). Microsaccades uncover the orientation of covert attention. Vision Research, 43(9), 1035–1045.

To install:

```
install.package("saccades")
```

To load:

```
library(saccades)
```

Usage:

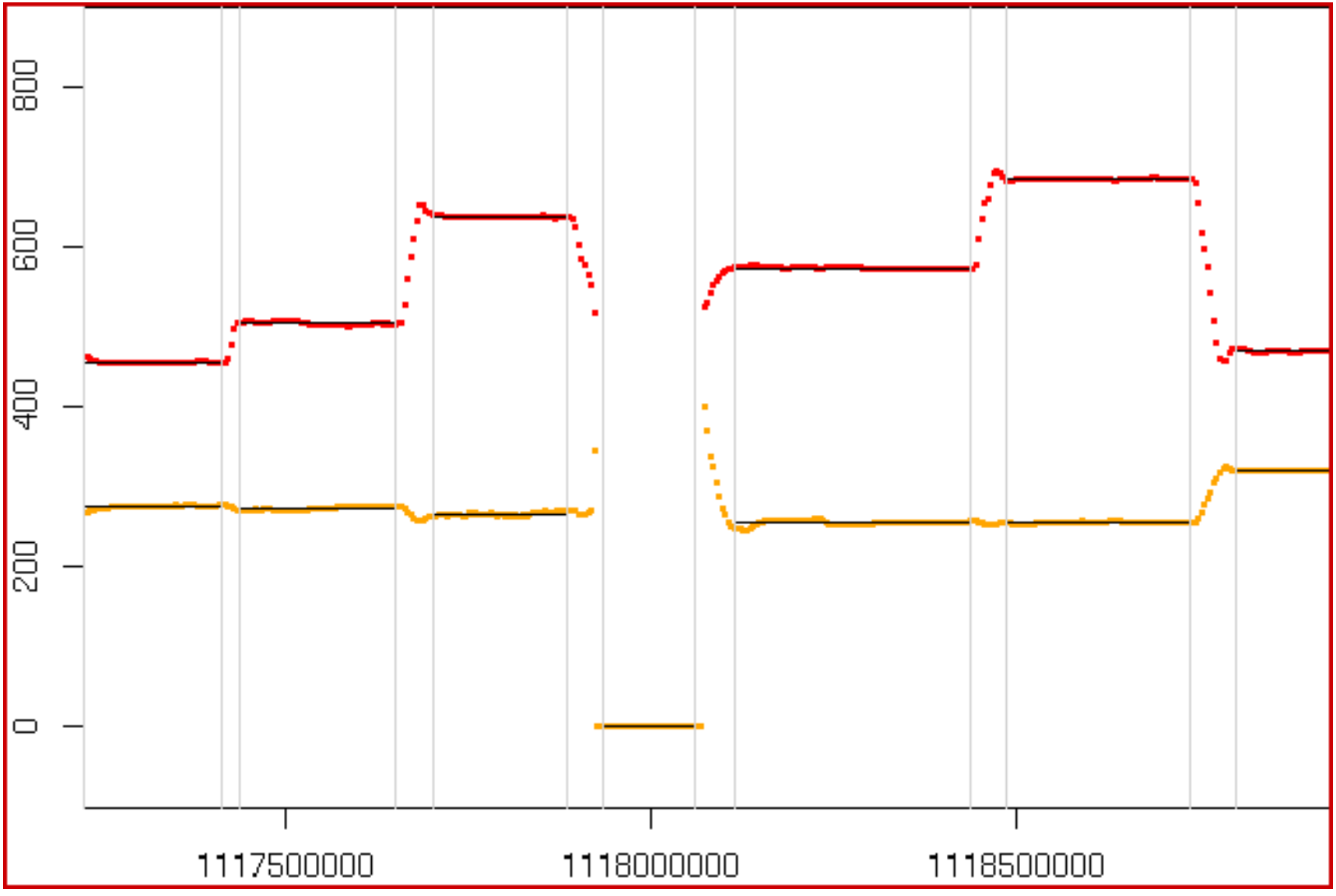
```
data(samples)
head(samples)
```

```
  time      x      y trial
1    0 53.18 375.73     1
2    4 53.20 375.79     1
3    8 53.35 376.14     1
4   12 53.92 376.39     1
5   16 54.14 376.52     1
6   20 54.46 376.74     1
```

```
fixations <- detect.fixations(samples)
head(fixations[c(1,4,5,10)])
```

```
  trial        x         y  dur
0     1 53.81296 377.40741   71
1     1 39.68156 379.58711  184
2     1 59.99267 379.92467   79
3     1 18.97898  56.94046  147
4     1 40.28365  39.03599  980
5     1 47.36547  35.39441 1310
```

```
diagnostic.plot(samples, fixations)
```



R package `scanpath`:

https://github.com/tmalsburg/scanpath

- Tools for analyzing gaze trajectories (a.k.a. scanpaths).
- Easy-going introductory and review paper: von der Malsburg, T., Vasishth, S., & Kliegl, R. (2012). Scanpaths in reading are informative about sentence processing. Proceedings of the First Workshop on Eye-tracking and Natural Language Processing (pp. 37–53). Mumbai, India.
- Details in the lecture on Saturday at 10:00.

To install:

```
devtools::install_github("tmalsburg/scanpath/scanpath",
                         dependencies=TRUE)
```

To load:

```
library(scanpath)
```

Bibliography:

R and Tidyverse

- Introduction to R, an interactive tutorial: https://www.datacamp.com/courses/free-introduction-to-r
- Grolemund, G., & Wickham, H. (2017). R for data science. Sebastopol, CA 95472, USA: O'Reilly. http://r4ds.had.co.nz/

Maximal random-effects structures:

- Barr, D. J., Levy, R., Scheepers, C., & Tily, H. J. (2013). Random effects structure for confirmatory hypothesis testing: Keep it maximal. Journal of Memory and Language, 68(3), 255–278. http://dx.doi.org/10.1016/j.jml.2012.11.001
- Matuschek, H., Bates, D., Kliegl, R., Vasishth, S., & Baayen, H. (2015). Balancing type i error and power in linear mixed models. Unpublished manuscript.
- Bates, D., Kliegl, R., Vasishth, S., & Baayen, H. (2015). Parsimonious mixed models. Unpublished manuscript.

Saccade detection

- Engbert, R., & Kliegl, R. (2003). Microsaccades uncover the orientation of covert attention. Vision Research, 43(9), 1035–1045. http://dx.doi.org/10.1016/S0042-6989(03)00084-1

Scanpaths

- von der Malsburg, T., & Vasishth, S. (2011). What is the scanpath signature of syntactic reanalysis? Journal of Memory and Language, 65(2), 109–127. http://dx.doi.org/10.1016/j.jml.2011.02.004
- von der Malsburg, T., & Vasishth, S. (2013). Scanpaths reveal syntactic underspecification and reanalysis strategies. Language and Cognitive Processes, 28(10), 1545–1578. http://dx.doi.org/10.1080/01690965.2012.728232
- von der Malsburg, T., Kliegl, R., & Vasishth, S. (2015). Determinants of scanpath regularity in reading. Cognitive Science, 39(7), 1675–1703. http://dx.doi.org/10.1111/cogs.12208
- von der Malsburg, T., Vasishth, S., & Kliegl, R. (2012). Scanpaths in reading are informative about sentence processing. In P. B. Michael Carl, & K. K. Choudhary, Proceedings of the First Workshop on Eye-tracking and Natural Language Processing (pp. 37–53). Mumbai, India: The COLING 2012 organizing committee.

Linear mixed models

- McElreath, R. (2016). Statistical rethinking: A Bayesian course with examples in R and Stan. Boca Ranton, Florida, USA: CRC Press.
- Shravan Vasishth and Bruno Nicenboim. Statistical Methods for Linguistic Research: Foundational Ideas – Part I. Language and Linguistics Compass, 10(8):349-369, 2016.
- Bruno Nicenboim and Shravan Vasishth. Statistical methods for linguistic research: Foundational Ideas - Part II. Language and Linguistics Compass, 10:591-613, 2016.
- Gelman, A., & Hill, J. (2007). Data analysis using regression and multilevel/hierarchical models. : Cambridge University Press.

End of file

:noexport:

R package `lme4`:

https://github.com/lme4/lme4/

- Package for fitting (frequentist) linear mixed effects models.
- Originaly developed by Doug Bates, now maintained by Ben Bolker.

To install:

```
install.packages("lme4")
```

To load:

```
library(lme4)
```

:noexport:

```
contrasts(d$cond) <- MASS::contr.sdif(2)

m1 <- lmer(FPRT ~ cond + (cond|subj) + (cond|item), d)
summary(m1)
```

```
Fixed effects:
            Estimate Std. Error t value
(Intercept)  216.787      8.517   25.45
cond2-1      191.237     10.570   18.09
```

:noexport:

Calculating a p-value:

```
m0 <- lmer(FPRT ~ 1    + (cond|subj) + (cond|item), d, REML=FAL
m1 <- lmer(FPRT ~ cond + (cond|subj) + (cond|item), d, REML=FAL
anova(m0, m1)
```

```
Data: d
Models:
m0: FPRT ~ 1 + (cond | subj) + (cond | item)
m1: FPRT ~ cond + (cond | subj) + (cond | item)
   Df   AIC   BIC logLik deviance  Chisq Chi Df Pr(>Chisq)
m0  8 43233 43282 -21609    43217
m1  9 43113 43168 -21547    43095 122.62      1  < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```