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

Workshop on Eye-Tracking Methods, Center for Language and Brain, Higher School of Economics, May 17–19, 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:

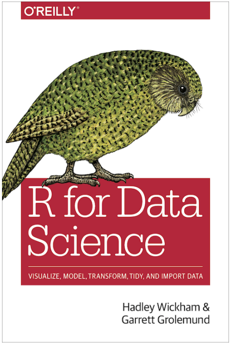
https://tmalsburg.github.io/hse/lecture4.pdf

Why use R?

- Replicability
- Transparency
- Recyclability
- Flexibility
- Efficiency

Learning R in two easy steps:

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



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 (concurrent recording of eye movements and event-related brain potentials):

(a) *Der verfallene Bauernhof braucht eine Renovierung.*
(b) * *Die verfallene Bauernhof masc braucht eine Renovierung.*
(c) * *Der neugierige Bauernhof masc braucht eine Renovierung.*

(a) *The$_{masc}$ deteriorating farm$_{masc}$ needs a renovation.*
(b) * *The$_{fem}$ deteriorating farm$_{masc}$ needs a renovation.*
(c) * *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:

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).

To install:

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

To load:

```
library(edfR)
```

Download eye-tracking data of participant 22:

https://tmalsburg.github.io/hse/s022_1.edf

Read data:

```
edf.data <- edf.trials("s022_1.edf")
names(edf.data)
```

`messages`	Various kinds of meta information
`header`	Start and end times for all trials
`saccades`	Start and end position, velocity, duration for all saccades
`blinks`	Start and end times for all blinks
`fixations`	Position and duration for all fixations

messages

```
head(edf.data$messages, 9)
```

```
  eyetrial  sttime                                                message
1        1 1871275                                              TRIALID 1
2        1 1871280     !V IMGLOAD FILL screenimages\\practice_01_-_s1.jpg
3        1 1871282      !V IAREA FILE interestareas\\practice_01_-_s1.ias
4        1 1871282 !V V_CRT MESSAGE sentence_1.SYNCTIME sentence_1.END_RT
5        1 1871285                                    RECCFG CR 500 2 1 R
6        1 1871285                                        ELCLCFG BTABLER
7        1 1871285                 GAZE_COORDS 0.00 0.00 1680.00 1050.00
8        1 1871285                                  THRESHOLDS R 114 255
9        1 1871285                                   ELCL_PROC ELLIPSE  (5)
```

header

```
head(edf.data$header, 10)
```

```
   eyetrial starttime endtime duration
1         1   1871275 1892201    20926
2         2   1897104 1909481    12377
3         3   1940295 1953411    13116
4         4   1957240 1964405     7165
5         5   1968778 1982479    13701
6         6   1987276 1995369     8093
7         7   2027386 2042035    14649
8         8   2046626 2052717     6091
9         9   2056328 2061981     5653
10       10   2062195 2068241     6046
```

saccades

```
head(edf.data$saccades, 10)
```

```
   eyetrial  sttime  entime  gstx  gsty  genx  geny  avel   pvel
17        1 1871596 1871662 478.5 594.2  96.5 517.9 149.0  307.2
21        1 1871796 1871812  81.4 512.5 104.1 501.7  45.5   73.9
27        1 1876680 1876756 117.5 533.1 106.2 521.2 596.5 1473.8
33        1 1877060 1877178 107.6 538.9  85.4 523.9 611.0 1485.1
37        1 1880158 1880174  59.4 517.5  70.2 517.2  33.7   49.1
41        1 1880438 1880516  87.4 514.7  62.6 503.0 473.3 1552.6
47        1 1880732 1880894  76.3 490.1  42.7 481.1 667.7 1539.5
54        1 1881578 1881606  50.2 502.4 237.2 505.9 136.0  290.9
58        1 1881876 1881894 219.7 517.0 152.5 515.9  70.1  114.9
62        1 1882200 1882216 146.6 501.3  99.4 502.9  55.4   72.3
```

`blinks`

```r
head(edf.data$blinks, 10)
```

```
    eyetrial  sttime  entime
26         1 1876718 1876722
32         1 1877102 1877140
46         1 1880780 1880852
87         1 1884154 1884178
189        2 1899956 1900012
280        3 1943702 1943770
294        3 1946062 1946190
336        3 1949944 1950012
417        4 1961334 1961434
530        5 1979092 1979208
```

fixations

```
head(edf.data$fixations, 10)
```

```
   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
43        1 1880518 1880730  72.0 495.4
52        1 1880896 1881576  45.2 496.4
56        1 1881608 1881874 225.7 511.1
60        1 1881896 1882198 151.6 505.6
```

Calculate fixations durations:

```
fixations <- edf.data$fixations

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
```

R package `scanpath`:

https://github.com/tmalsburg/scanpath

- Tools for analyzing and visualizing gaze trajectories (a.k.a. scanpaths).
- Details in the lecture on Saturday at 10:00.
- Here we only use this package for plotting the data.

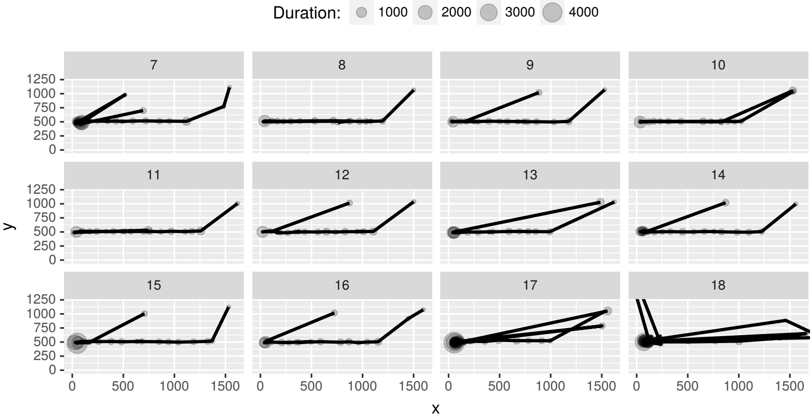
To install:

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

To load:

```
library(scanpath)
```

```
library(scanpath)

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

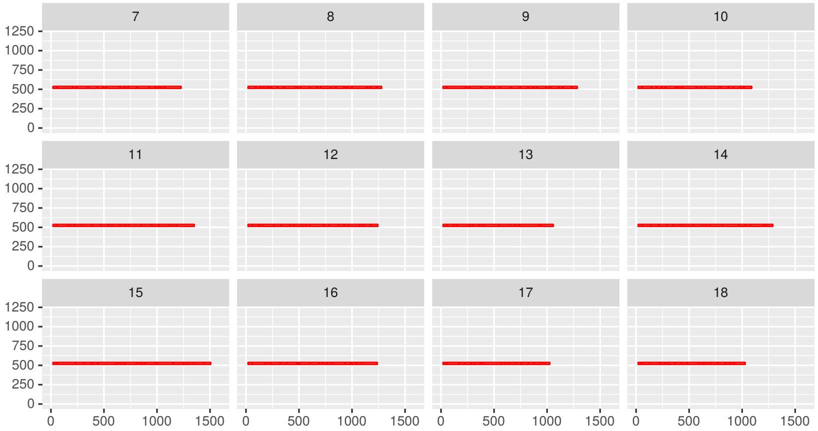


Load regions of interest (ROIs) generated by presentation software (e.g. OpenSesame):

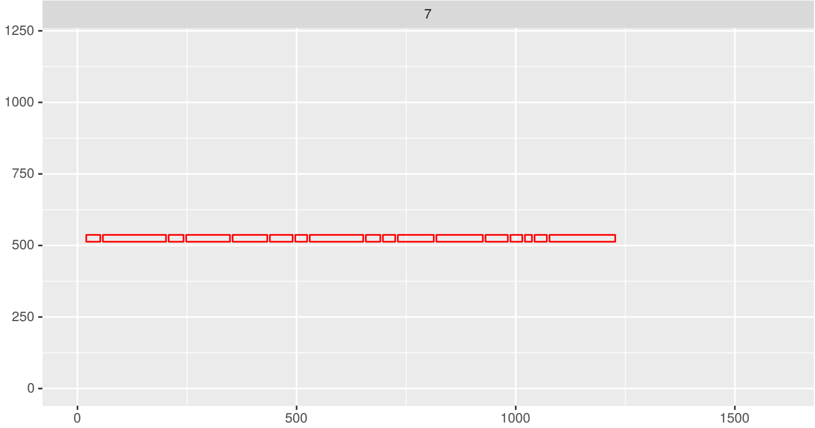
```
head(rois)
```

```
  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)
```



```r
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 using helper function `map_fixations`:

Install `intervals` package:

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

Download helper function:

https://tmalsburg.github.io/hse/map_fixations.function.R

To load:

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

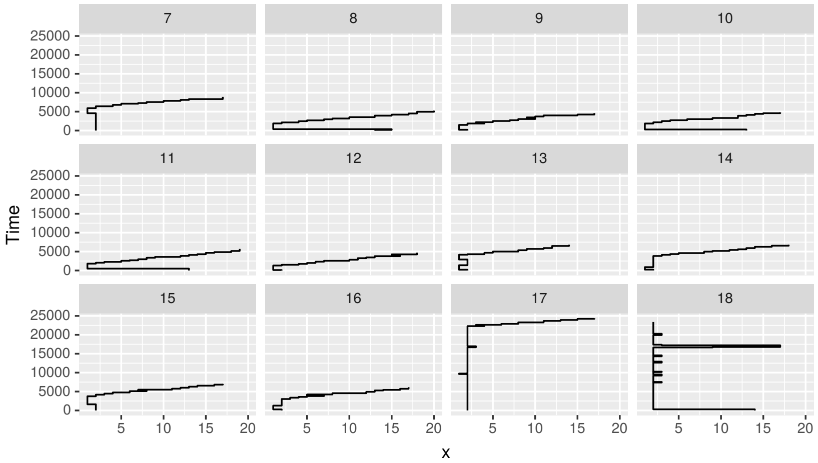
```
fixations <- cbind(fixations,
                   map_fixations(fixations, rois))
head(fixations, 10)
```

```
   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,
                                 xbuffer=8,
                                 ybuffer=30))
head(fixations, 10)
```

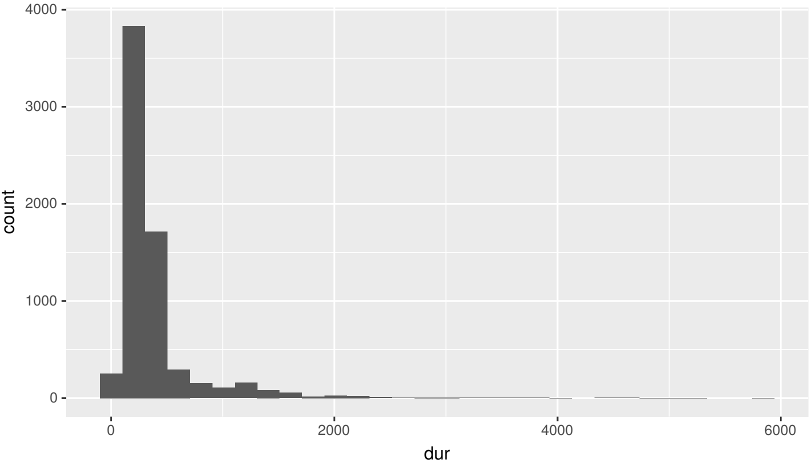
```
   eyetrial  gavx  gavy  dur wn      word
1         1 477.2 599.5  304 NA      <NA>
2         1  79.1 513.5  130  2       ist
3         1  89.8 521.2 4864  3       ein
4         1 100.3 524.6  300  3       ein
5         1  64.0 521.2 2976  2       ist
6         1  70.9 517.1  260  2       ist
7         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,
```

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

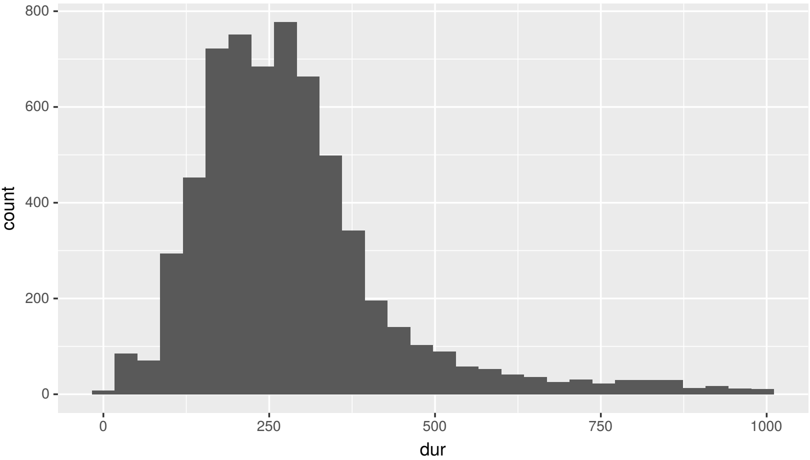


Removing outliers:

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



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

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



Add subject ID and write preprocessed data to disk:

```
fixations %>%
  mutate(subj = "s022_1") %>%
  write_tsv("s022_1.edf.fix")
```

Trial-level information (generate by presentation software):

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

Loading data from one participant:

```
fixations   <- read_tsv("s022_1.edf.fix")
trial.infos <- read_tsv("s022_1.txt")

d <- inner_join(fixations, trial.infos)
```

```
# A tibble: 6 x 11
  eyetrial    wn word            x     y   dur subj    expt
     <int> <int> <chr>       <int> <int> <int> <chr>   <chr>
1        1     1 Dies           46   500   222 s022_1  practi
2        1     5 er            226   511   248 s022_1  practi
3        1     4 Testsatz,     151   505   300 s022_1  practi
4        1     3 ein           106   503   270 s022_1  practi
5        1     7 von           303   510   188 s022_1  practi
6        1     9 Frage         392   510   148 s022_1  practi
```

Loading data from all participants and merge in one data frame:

```
fix.files <- list.files(".", ".edf.fix")
txt.files <- list.files(".", ".txt")

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 canonical eye-tracking measures:

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

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

To install:

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

To load:

```
library(em2)
```

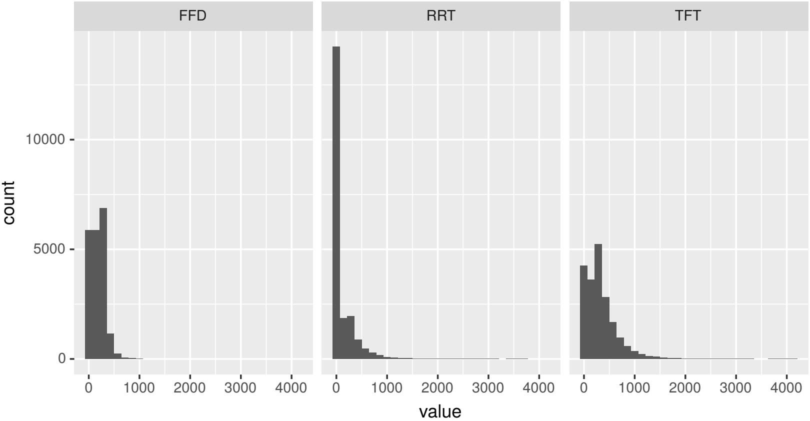
```r
et.measures <- em2(all.fixations$wn, all.fixations$dur,
                   select(all.fixations, eyetrial, subj,
                          expt, item, cond))

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
```

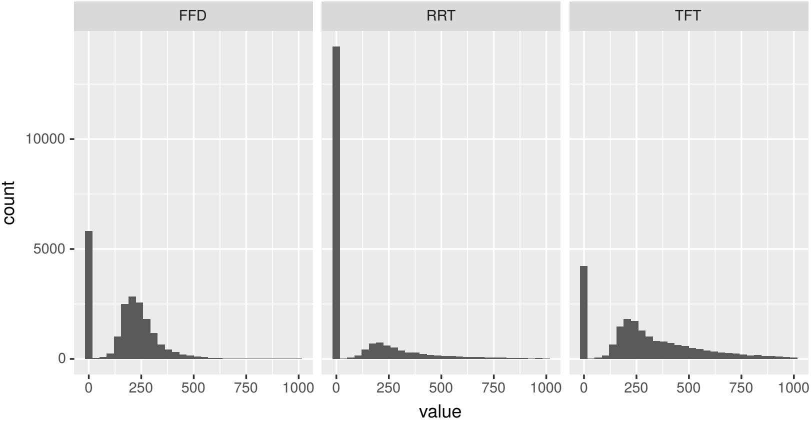
Cleaning up after using `em2`:

```
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:

By-condition means for FPRT, RRT, TFT

Step 1: Remove unnecessary information from the data frame:

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

head(x)
```

```
  cond FPRT RRT TFT
1    b  474   0 474
2    c  240   0 240
3    c  204   0 204
4    c  242   0 242
5    b  680 146 826
6    b  228   0 228
```

Step 2: For convenience, rearrange the data slightly:

```
x %>% gather(measure, value, 2:4) -> x

head(x)
```

```
  cond measure value
1    b    FPRT   474
2    c    FPRT   240
3    c    FPRT   204
4    c    FPRT   242
5    b    FPRT   680
6    b    FPRT   228
```

Step 3: Calculate the by-condition means:

```
x %>%
  group_by(measure, cond) %>%
  summarize(mean = mean(value)) %>%
  spread(measure, mean) -> means

head(means)
```

```
# A tibble: 3 x 4
  cond   FPRT   RRT   TFT
  <fct> <dbl> <dbl> <dbl>
1 a       121  58.8   180
2 b       312 176     488
3 c       115  71.9   187
```

Same but all steps in one go:

```
et.measures %>%
  filter(expt == "judith",
         roi  == 3,
         cond %in% c("a", "b", "c")) %>%
  select(cond, FPRT, RRT, TFT) %>%
  gather(measure, value, 2:4) %>%
  group_by(measure, cond) %>%
  summarize(mean = mean(value)) %>%
  spread(measure, mean) -> means
```

Table in $\LaTeX$ format:

```
library(xtable)

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

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

Calculating means and 95% confidence intervals:

Step 1: Calculate by-subject, by-condition means:

```
et.measures %>%
  filter(expt == "judith",
         roi  == 3,
         cond %in% c("a", "b", "c")) %>%
  select(subj, cond, FPRT, RRT, TFT) %>%
  gather(measure, value, 3:5) %>%
  group_by(measure, cond, subj) %>%
  summarize(s.mean = mean(value)) ->
  by.subject.means
```

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

```
# A tibble: 6 x 4
# Groups:   measure, cond [1]
  measure cond  subj   s.mean
  <chr>   <fct> <fct>   <dbl>
1 FPRT    a     s001_3  122
2 FPRT    a     s002_1   99.5
3 FPRT    a     s003_1   89.7
4 FPRT    a     s004_1   29.1
5 FPRT    a     s005_1  196
6 FPRT    a     s006_1  127
```

Step 2: Calculate by-condition means and 95% confidence intervals:

```
by.subject.means %>%
  summarize(
    g.mean   = mean(s.mean),
    ci.lower = g.mean - 1.96 * sd(s.mean) / sqrt(n()),
    ci.upper = g.mean + 1.96 * sd(s.mean) / sqrt(n())) ->
  grand.means
```

```
grand.means
```

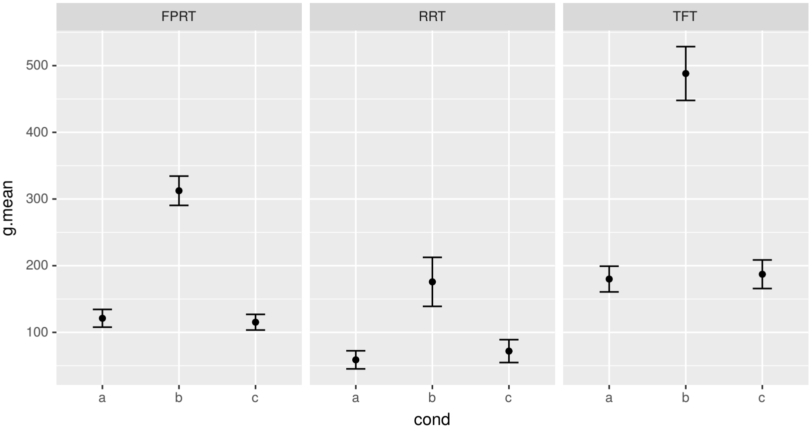
```
# A tibble: 9 x 5
# Groups:   measure [?]
  measure cond  g.mean ci.lower ci.upper
  <chr>   <fct>  <dbl>    <dbl>    <dbl>
1 FPRT    a      121      108      134
2 FPRT    b      312      290      334
3 FPRT    c      115      104      127
4 RRT     a       58.8     45.3     72.4
5 RRT     b      176      139      213
6 RRT     c       71.9     54.8     89.1
7 TFT     a      180      161      199
8 TFT     b      488      448      529
9 TFT     c      187      166      209
```

Table in LaTeX format:

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

measure	cond	g.mean	ci.lower	ci.upper
FPRT	a	121	108	134
FPRT	b	312	290	334
FPRT	c	115	104	127
RRT	a	59	45	72
RRT	b	176	139	213
RRT	c	72	55	89
TFT	a	180	161	199
TFT	b	488	448	529
TFT	c	187	166	209

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



Inferential stats:

Goal: Test whether the difference between conditions a (baseline) and b (syntactic violaion) is statistically robust

Select data for comparison of FPRTs between conditions:

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

```
    subj item cond FPRT
1 s001_3  158    b  474
2 s001_3   33    c  240
3 s001_3  171    c  204
4 s001_3  123    c  242
5 s001_3  194    b  680
6 s001_3   38    b  228
```

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)
```

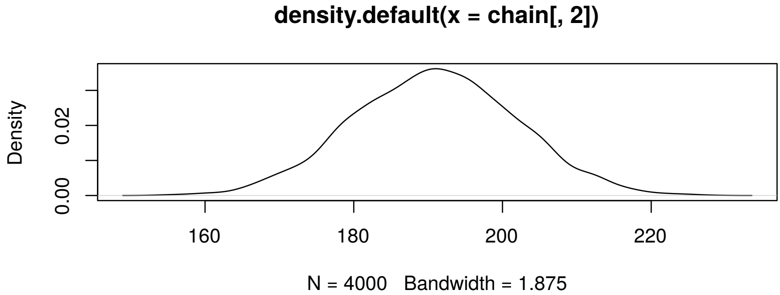
```
options("mc.cores" = 4)
m1 <- brm(FPRT ~ cond + (cond|subj) + (cond|item), d)
```

```
summary(m1)
```

```
Population-Level Effects:
          Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
Intercept   121.14      6.82   107.23   134.15       1185 1.00
condb       191.05     10.96   169.75   212.55       1423 1.00
condc        -5.94      5.41   -16.71     4.38       4000 1.00
```

Posterior density of the paramter capturing the difference between conditions a and b:

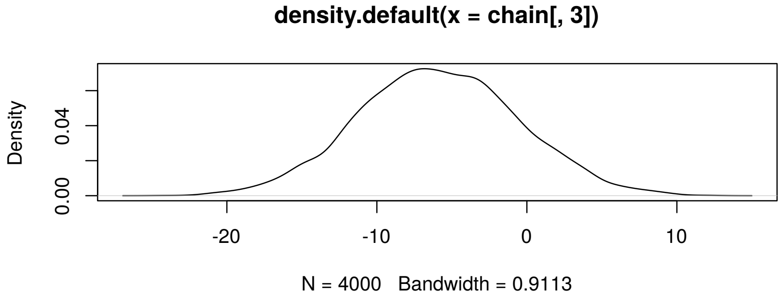
```
chain <- as.mcmc(m1, combine_chains=TRUE)
plot(density(chain[,2]))
```



Conclusion: First pass reading substantially slowed in response to syntactic violations (192 ms). The evidence is really strong ($P(\beta > 0) = 1$)

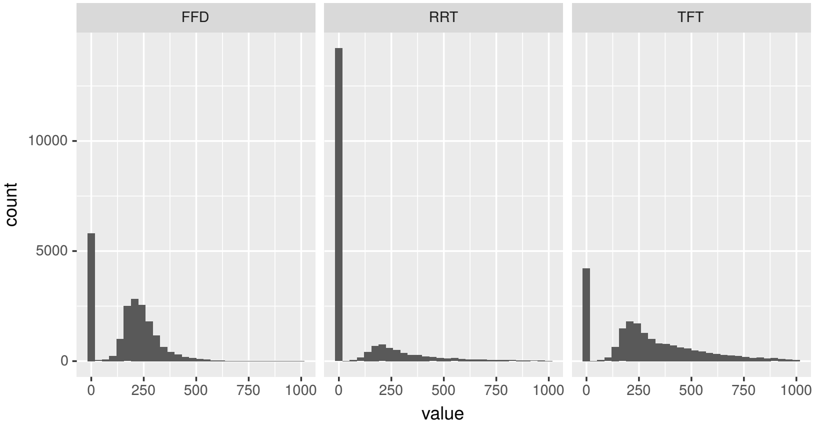
Posterior density of the paramter capturing the difference between conditions a and c:

```
chain <- as.mcmc(m1, combine_chains=TRUE)
plot(density(chain[,3]))
```



Conclusion: No evidence that semantic violations did affect first pass reading times in any meaningful way.

Issues and potential pitfalls:



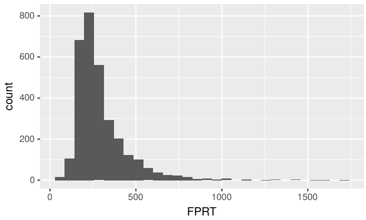
- Zeroes
- Non-normal distribution

How to deal with measures that contain zeroes?

Fit two models:

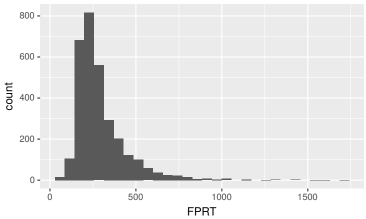
1. Like before but only for non-zero values.
2. Additional model testing whether the value was more often zero in one condition than in the other.

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

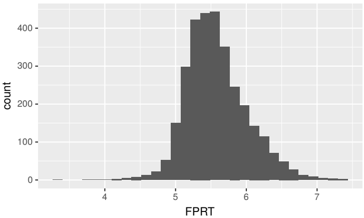


Raw FPRTs on the *ms* scale

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



Raw FPRTs on the *ms* scale



Log-transformed FPRTs

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

No clear answer, let's discuss.

Another useful package:

R package `saccades`:

https://github.com/tmalsburg/saccades

- Offers algorithms for detecting saccades and fixations.
- Can be used if you don't want to rely on black-box algo offered my eye-tracker manufacturer (some are fairly bad).
- Velocity-based algorithm 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:

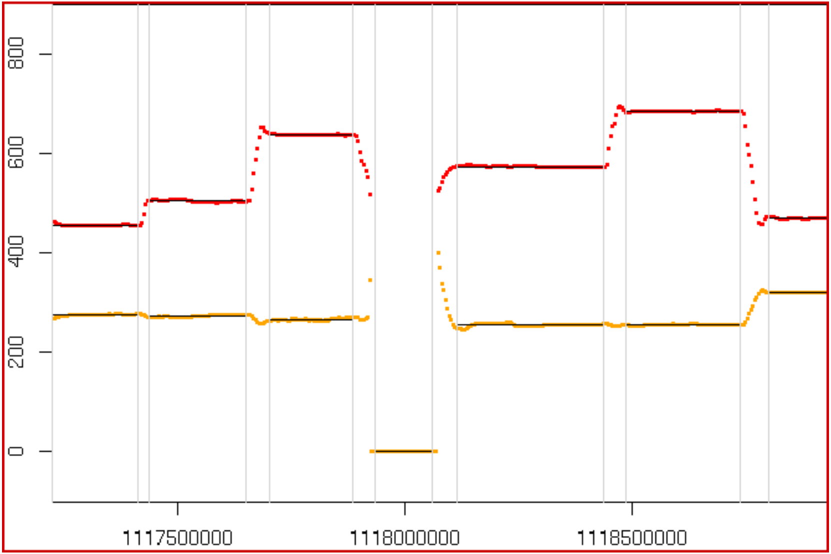
```
data(samples)
head(samples)
```

```
  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
```

```r
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)
```



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.

Appendix

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)
```

```
library(lme4)
m2 <- lmer(FPRT ~ cond + (cond|subj) + (cond|item), d)
summary(m2)
```

```
Fixed effects:
            Estimate Std. Error t value
(Intercept)  121.238      7.003  17.311
condb        191.289     10.773  17.757
condc         -5.966      5.216  -1.144
```

Calculating a p-value:

```
m0 <- lmer(FPRT ~ 1    + (cond|subj) + (cond|item), d,
           REML=FALSE)
m2 <- lmer(FPRT ~ cond + (cond|subj) + (cond|item), d,
           REML=FALSE)
anova(m0, m2)
```

```
Data: d
Models:
m0: FPRT ~ 1 + (cond | subj) + (cond | item)
m2: FPRT ~ cond + (cond | subj) + (cond | item)
   Df   AIC   BIC logLik deviance  Chisq Chi Df Pr(>Chisq)
m0 14 64104 64195 -32038    64076
m2 16 63980 64084 -31974    63948 128.17      2  < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

End of file