# Statistical Analysis of Eye Tracking Data

Krzysztof Krejtz

LEAD-ME Summer Training School Warsaw 2021 (5-9 July 2021)
Eye tracking in media accessibility research - methods, technologies and data analyses

1

## Prepare data for the analyses

read / filter / select / mutate / join / write

2

## Read data frame from file

```
# read data frame
df <- read
```

read_builtin (readr)
read_csv (readr)
read_csv_chunked (readr)
read_csv2 (readr)
read_csv2_chunked (readr)
read_delim (readr)
read_delim_chunked (readr)

```
read_csv(file, col_names = TRUE, col_types = NULL,
    locale = default_locale(), na = c("", "NA"),
    quoted_na = TRUE, quote = "\"", comment = "",
    trim_ws = TRUE, skip = 0, n_max = Inf, guess_max =
    min(1000, n_max), progress = show_progress(),
    skip_empty_rows = TRUE)
```

read_csv() and read_tsv() are special cases of the general read_delim(). They're useful for reading the most common types of flat file data, comma separated values and tab separated values.

Press F1 for additional help

```
# read data frame
df <- read_csv(file = "data/")
```

all_data.csv data
all_data.Rda data
qualtrics.csv data
realeye.csv data
.DS_Store data

3

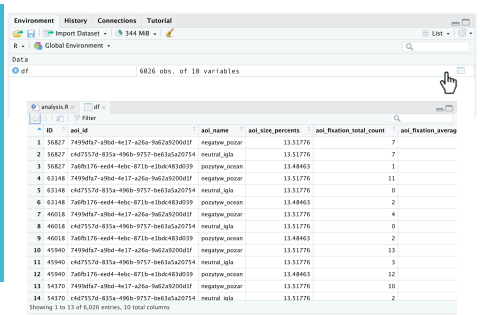
## Read data frame from file

```
> df <- read_csv(file = "data/realeye.csv")

── Column specification ────────────────────────────────────────
cols(
  ID = col_double(),
  aoi_id = col_character(),
  aoi_name = col_character(),
  aoi_size_percents = col_double(),
  aoi_fixation_total_count = col_double(),
  aoi_fixation_average_duration_ms = col_double(),
  aoi_fixation_ttff_ms = col_double(),
  aoi_fixation_average_total_time_spent_ms = col_double(),
  aoi_fixation_first_fixation_average_duration_ms = col_double(),
  notes = col_character()
)
```

4

## First look at data



5

## Create AOI type independent variable

- There were three different types of AOI (positive picture, negative and neutral)
- Currently data frame contains AOI names as follows:

```
> unique(df$aoi_name)
 [1] "negatyw_pozar"       "neutral_igla"        "pozytyw_ocean"       "negatyw_lodowiec"
 [5] "neutral_kosci"       "pozytyw_gory"        "negatyw_niedzwiedz"  "neutral_lyzeczki"
 [9] "pozytyw_las"         "negatyw_podtopienia" "neutral_muszla"      "pozytyw_morze"
[13] "negatyw_susza"       "neutral_zegar"       "pozytyw_rzeka"
```

- We want to make AOI type variable which will be used as a factor in further analysis
- We will use first element of unique AOI names as values of this new variable
- Technically we will split aoi_name variable into two columns by "_", creating new "aoi_type" and "picture" variables

```
df <- df %>%
  separate(aoi_name, sep = "_", into = c("aoi_type", "picture"))
```

6

## Select & filter variables

- Sometimes you want to delete from your data frame some variables (columns)
- This time we want to get rid off two variables "notes", "aoi_id", and "picture"
- Also we want to remove four subjects which were not following the instruction

```
df <- df %>%
  select(-notes, -aoi_id, -picture) %>%
  filter(ID != "61567" & ID!="64828" & ID!="66001" & ID != "54127")
```

- The list of new df variables names shows that the operation was successful

```
> names(df)
[1] "ID"                                    "aoi_type"
[3] "aoi_size_percents"                     "aoi_fixation_total_count"
[5] "aoi_fixation_average_duration_ms"      "aoi_fixation_ttff_ms"
[7] "aoi_fixation_average_total_time_spent_ms" "aoi_fixation_first_fixation_average_duration_ms"
```

7

## Missing values (NAs) a special case

- In R typically missing values are annotated with NA
- In statistical analysis NAs are not welcome
- Always try to find out why you have NAs in your data set
- This time NAs meaning is that the persons did not fixate an AOI
  - It make sense to replace NAs with o

```
df <- df %>%
  replace_na(list(aoi_fixation_ttff_ms = 0, aoi_fixation_average_duration_ms = 0,
                  aoi_fixation_average_total_time_spent_ms = 0,
                  aoi_fixation_first_fixation_average_duration_ms = 0))
```

8

## Read new data file with questionnaire answers

```
# read data file with additional variables
dq <- read_csv(file = "data/qualtrics.csv")
```

```
── Column specification ──
cols(
  Q1 = col_double(),
  Q3 = col_double(),
  Q5 = col_double(),
  Q7 = col_double(),
  Q9 = col_double(),
  Q11 = col_double(),
  Q13 = col_double(),
  Q15 = col_double(),
  ID = col_double()
)
```

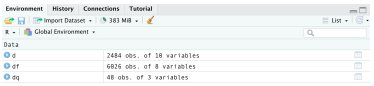
| | Q1 | Q3 | Q5 | Q7 | Q9 | Q11 | Q13 | Q15 | ID |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 4 | 5 | 4 | 5 | 5 | 5 | 4 | 4 | 41428 |
| 2 | 4 | 5 | 2 | 5 | 4 | 5 | 3 | 2 | 64666 |
| 3 | 4 | 4 | 4 | 5 | 4 | 4 | 2 | 4 | 32185 |
| 4 | 2 | 2 | 4 | 2 | 5 | 4 | 3 | 3 | 65176 |
| 5 | 4 | 4 | 4 | 5 | 3 | 5 | 5 | 5 | 45652 |
| 6 | 1 | 5 | 5 | 4 | 5 | 4 | 2 | 5 | 60466 |
| 7 | 3 | 5 | 4 | 5 | 4 | 4 | 4 | 4 | 63148 |
| 8 | 3 | 4 | 4 | 4 | 4 | 4 | 5 | 4 | 45940 |
| 9 | 2 | 4 | 5 | 5 | 2 | 4 | 3 | 5 | 14539 |
| 10 | 3 | 4 | 4 | 2 | 4 | 3 | 4 | 3 | 29236 |
| 11 | 4 | 2 | 3 | 3 | 5 | 3 | 4 | 4 | 54619 |
| 12 | 4 | 4 | 3 | 4 | 2 | 5 | 5 | 4 | 46417 |
| 13 | 4 | 4 | 4 | 4 | 5 | 4 | 2 | 4 | 65905 |

9

## Joining two data frames by subject ID

- We need to join two data frames (with eye tracking data and questionnaire data)
- We will join to data frames by subject ID variable

```
# join two data frames by ID (subject ID variable)
d <- inner_join(df,dq, by = "ID")
```




- Note that new data frame "d" has lower number of rows.
- It is due to the fact that not all subjects who participated in eye tracking study completed also the questionnaire.

10

## Calculate independent variables / factors

- Questionnaire data contains answers from the New Ecological Paradigm (NEP) questionnaire and **subject ID**
  - The answers were on 1-5 Likert-type scale (the higher value the higher sensitivity to climate change)
- We want to calculate one score of the sensitivity to climate change (NEP) which will be a mean of all given answers
- Next, we want to perform median split on NEP score (low vs high sensitivity to climate change) to use it as a factor in further analysis.
- We need also to set aoi_type as factor and make "neutral" value of it as a reference point (important for statistical analysis)
- Last, we do not need all row answers to each NEP question (they all starts with "Q").

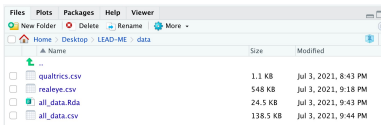
```
d <- d %>%
  mutate(NEP = rowSums(select(.,starts_with("Q")))) %>%
  mutate(NEPsplit = cut(NEP, 2, labels = c("low", "high"))) %>%
  mutate(aoi_type = factor(aoi_type)) %>%
  mutate(aoi_type = relevel(aoi_type, ref = "neutral")) %>%
  select(-starts_with("Q"))
```

11

## Write data files

- Save the entire data frame into a new file
- We can do it in several formats. The most useful are:
  - .csv files (great for sharing even with those who do not use R)
  - Data format (.Rda) - great for further use within R and hard drive space saver

```
# write the entire data base into RData file
save(d, file = "data/all_data.Rda")
# or csv file
write_csv(d, file = "data/all_data.csv")
```




12

# Descriptive statistics

With visualisations

13

## Create new R script "analysis.R"

- It is good to have several R scripts for different purposes and ease-of-reading.
- We will create new R script named "analysis.R"
- Start the script with useful libraries

analysis.R

```
# load useful packages
library(psych)
library(tidyverse)
library(afex)
library(emmeans)
```

14

## Read data frame of RData format

- Reading RData format will load automatically the data frame name

```
# read data frame from RData format
load("data/all_data.Rda")
```

Environment | History | Connections | Tutorial

Import Dataset | 113 MiB

R | Global Environment

Data

d — 2430 obs. of 10 variables

15

## Describe data frame

- Summary is a generic function used to produce result summaries of the results of various model fitting functions.
- The function invokes particular methods which depend on the class of the first argument.
- When applied to data frame it returns basic descriptive statistics for all numerical variables

```
> summary(d)
       ID           aoi_type         aoi_size_percents aoi_fixation_total_count aoi_fixation_average_duration_ms aoi_fixation_ttff_ms
Min.   :10564   Length:2430        Min.   :11.97     Min.   : 0.000           Min.   :100.0                    Min.   :  500
1st Qu.:41710   Class :character   1st Qu.:12.18     1st Qu.: 1.000           1st Qu.:144.0                    1st Qu.: 1044
Median :55152   Mode  :character   Median :12.30     Median : 4.000           Median :162.0                    Median : 1918
Mean   :51814                      Mean   :12.46     Mean   : 6.779           Mean   :169.1                    Mean   : 3119
3rd Qu.:63148                      3rd Qu.:12.80     3rd Qu.:10.000           3rd Qu.:183.0                    3rd Qu.: 3998
Max.   :67201                      Max.   :13.52     Max.   :44.000           Max.   :583.0                    Max.   :14882
                                                                              NA's   :486                      NA's   :486
aoi_fixation_average_total_time_spent_ms aoi_fixation_first_fixation_average_duration_ms      NEP        NEPsplit
Min.   :  100                            Min.   : 100.0                                    Min.   :3.125   low :1125
1st Qu.:  450                            1st Qu.: 123.0                                    1st Qu.:3.750   high:1305
Median : 1048                            Median : 143.0                                    Median :4.125
Mean   : 1492                            Mean   : 163.9                                    Mean   :4.113
3rd Qu.: 2020                            3rd Qu.: 179.0                                    3rd Qu.:4.500
Max.   :13419                            Max.   :2051.0                                    Max.   :5.000
NA's   :486                              NA's   :486
```

Can be hard to read however

16

## Descriptive statistics of selected variables

The **describe** function in the **psych** package is meant to produce the most frequently requested stats in psychometric and psychology studies, and to produce them in an easy to read data.frame.

```
> describe(d[,4:8])
                                        vars    n    mean      sd median trimmed     mad min   max range skew kurtosis    se
aoi_fixation_total_count                   1 2430    6.78    7.42    4.0    5.58    5.93   0    44    44 1.52     2.34  0.15
aoi_fixation_average_duration_ms           2 1944  169.09   42.01  162.0  163.86   28.17 100   583   483 3.16    18.95  0.97
aoi_fixation_ttff_ms                       3 1944 3119.15 3063.71 1918.5 2495.53 1650.88 500 14882 14382 1.80     3.84 69.49
aoi_fixation_average_total_time_spent_ms   4 1944 1492.11 1536.47 1048.5 1230.19 1027.44 100 13419 13319 2.86    14.39 34.85
aoi_fixation_first_fixation_average_duration_ms 5 1944  163.92   79.41  143.0  151.01   35.58 100  2051  1951 8.71   169.99  1.80
```

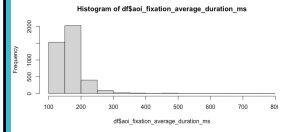
The **fast=TRUE** option will lead to a speed up of about 50% for larger data sets

It is also easy to read!

```
> describe(d[,4:8], fast = TRUE)
                                        vars    n    mean      sd min   max range    se
aoi_fixation_total_count                   1 2430    6.78    7.42   0    44    44  0.15
aoi_fixation_average_duration_ms           2 1944  169.09   42.81 100   583   483  0.97
aoi_fixation_ttff_ms                       3 1944 3119.15 3063.71 500 14882 14382 69.49
aoi_fixation_average_total_time_spent_ms   4 1944 1492.11 1536.47 100 13419 13319 34.85
aoi_fixation_first_fixation_average_duration_ms 5 1944  163.92   79.41 100  2051  1951  1.80
```

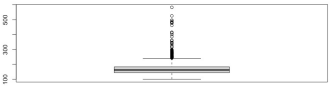
17

```
# plot distribution of variable values with histogram
hist(d$aoi_fixation_average_duration_ms)
```




**Useful to visually inspect normality of distribution**

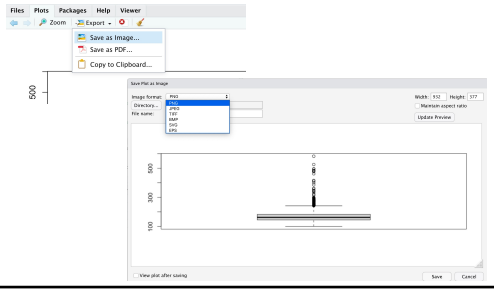
```
# plot distribution of variable values with boxplot
boxplot(d$aoi_fixation_average_duration_ms)
```




**Useful to identify outlying values**

18

## Saving graphs in RStudio graphical interface




19

# Moderation analysis with linear regression

Mixed-design linear regression with continuous and nominal predictors and interaction term

20

## Linear regression

- In regression we are fitting a model to our data
  - To not loose variance if predictor is continuos and does not need to be spliced into categorical variable
  - To predict values of the dependent variable from one or more independent variable
  - To understand the relationship
    - How does one variable change as the other changes?
    - How much ...does ..... rise with a one unit increase in .....?

21

## Linear regression

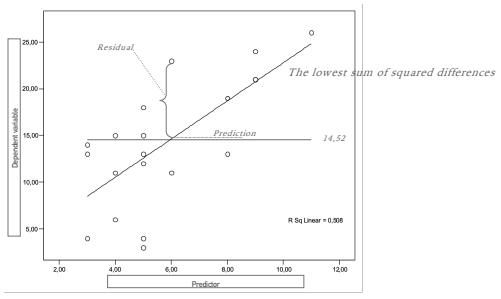
- The outcome variable is predicted using the equation of a straight line for which the squared differences between line and the actual data is minimised.

$$\hat{Y} = a + bX$$

- $\hat{Y}$= the predicted value of ***Y*** (.....)
- ***X*** = ..... (predictor, independent variable)
- Indices, coefficients are computed to asses how accurately Y scores are predicted by the linear equation:
  - ***b*** (slope) - the amount of change in predicted *Y* for one unit change in *X*
  - ***a*** (constant) - an intercept, value of $\hat{Y}$ when *X* = 0

22

## least squares Method




23

## Hypotheses testing

- Null hypotheses
  - **b** slope = 0
  - **a** constant = 0

24

## Simple vs. Multiple regression

- In simple regression
  - The outcome variable *Y* is predicted using the equation of a straight line
- Multiple regression
  - A logical extension of the principles to situation in which there are several predictors.
  - Outcome = (Model)+ error

25

## Multiple regression without and with interaction

- Each predictor has its own regression coefficient
  - For every extra predictor you include, another coefficient need to be estimated
- We are looking for linear combination of predictors that correlate maximally with the outcome variable
  - multiple regression formula with two predictors (no interaction term)

$$\hat{Y} = a + \beta_1 X_1 + \beta_2 X_2 + \epsilon$$

  - multiple regression formula with two predictors and interaction term

$$\hat{Y} = a + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2 + \epsilon$$

26

## Usefull libraries

```
library(emmeans)
library(lme4)
```

**lme4** provides functions for fitting and analyzing mixed models:
linear (lmer),
generalized linear (glmer), and
nonlinear (nlmer.)

27

## Hypotheses

- We want to test the following hypotheses:
  - The more sensitive to climate change people are the longer fixation duration on environment pictures
  - Positive and negative pictures of environment will evoke longer fixation duration
  - Sensitivity to climate change will predict fixation duration differently while looking on environment pictures of different emotional valence (interaction hypothesis).

28

## Multiple regression. Model definition

```
fit <- lmer(aoi_fixation_average_duration_ms ~ NEP + aoi_type + NEP:aoi_type + (1 | ID),
            data = d)
```

Data frame — Dependent variable — Predictor 1 — Predictor 2 — Interaction of predictors — Random intercept for subjects

**Shorter form of full model definition**

```
fit <- lmer(aoi_fixation_average_duration_ms ~ NEP*aoi_type + (1 | ID),
            data = d)
```

29

## Model results basic table

```
> anova(fit)
Type III Analysis of Variance Table with Satterthwaite's method
             Sum Sq Mean Sq NumDF DenDF F value  Pr(>F)
NEP            2269  2269.4     1    42  0.3989 0.53107
aoi_type      35625 17812.5     2  1932  3.1312 0.04389 *
NEP:aoi_type  51223 25611.7     2  1932  4.5022 0.01120 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

30

## Model results.

## Detailed table

```
> summary(fit)
Linear mixed model fit by REML. t-tests use Satterthwaite's method ['lmerModLmerTest']
Formula: aoi_fixation_average_duration_ms ~ NEP + aoi_type + NEP:aoi_type +      (1 | ID)
   Data: d

REML criterion at convergence: 22769

Scaled residuals:
    Min      1Q  Median      3Q     Max
-2.2882 -0.1971  0.2103  0.5200  5.7394

Random effects:
 Groups   Name        Variance Std.Dev.
 ID       (Intercept)  307.1   17.52
 Residual             5688.8   75.42
Number of obs: 1980, groups:  ID, 44

Fixed effects:
                    Estimate Std. Error         df t value Pr(>|t|)
(Intercept)          161.0240    31.1653  104.5102   5.167 1.15e-06 ***
NEP                    -1.0629     0.9374  104.5102  -1.134  0.25946
aoi_typenegatyw       -81.3550    32.7634 1932.0000  -2.483  0.01311 *
aoi_typepozytyw       -31.8670    32.7634 1932.0000  -0.973  0.33085
NEP:aoi_typenegatyw     2.9503     0.9855 1932.0000   2.994  0.00279 **
NEP:aoi_typepozytyw     1.6580     0.9855 1932.0000   1.674  0.09414 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
            (Intr) NEP    a_typn a_typp NEP:_typn
NEP         -0.992
ao_typngtyw -0.526  0.521
a_typpoztyw -0.526  0.521  0.500
NEP:_typngt  0.521 -0.526 -0.992 -0.496
NEP:_typpzy  0.521 -0.526 -0.496 -0.992  0.500
```

NOTE: In the report remember to provide all the values from the coefficients table: coefficient value, standard error, t-test value, degrees of freedom and p-value.

31

## Trends analysis for interaction

Code

```
em <- emtrends(fit, ~ NEP:aoi_type, var = "NEP")
```

Results of trends analysis

```
> print(em)
 NEP aoi_type NEP.trend    SE  df lower.CL upper.CL
  33 neutral     -1.863 0.937 105  -3.9218    0.796
  33 negatyw      1.087 0.937 105   0.0286    3.746
  33 pozytyw      0.587 0.937 105  -1.2718    2.446

Degrees-of-freedom method: kenward-roger
Confidence level used: 0.95
```

Statistical comparison of slopes

```
> pairs(em)
 contrast                                                estimate    SE   df t.ratio p.value
 32.9772727272727 neutral - 32.9772727272727 negatyw     -2.95 0.986 1932 -2.994  0.0079
 32.9772727272727 neutral - 32.9772727272727 pozytyw     -1.65 0.986 1932 -1.674  0.2114
 32.9772727272727 negatyw - 32.9772727272727 pozytyw      1.30 0.986 1932  1.319  0.3845

Degrees-of-freedom method: kenward-roger
P value adjustment: tukey method for comparing a family of 3 estimates
```

32

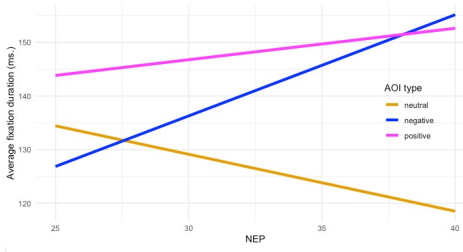
Interaction visualization.

Publication ready

```
plo <- emmip(fit, aoi_type ~ NEP, cov.reduce = range) +
        geom_line(size=1.5) +
        scale_y_continuous(name = "Average fixation duration (ms.)") +
        scale_x_continuous(name = "NEP") +
        scale_color_manual(name = "AOI type",
                           labels = c("neutral", "negative", "positive"),
                           values = c("#E69F00", "#0000FF", "#FF00FF")) +
        theme_minimal() +
        theme(legend.position = c(.85, .5))
```

33

Save the graph




```
print(plo)
ggsave(plot = plo, filename = "graphs/fixationDuration_regression.jpg")
dev.off()
```

34

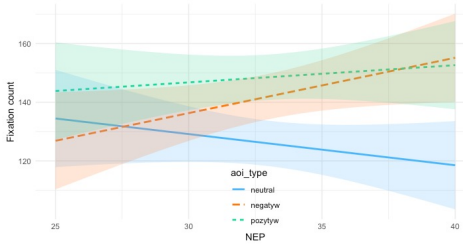
Alternative plot of interaction

```
# alternatively
require(interactions)
interact_plot(fit, pred = NEP, modx = aoi_type, interval = TRUE) +
  scale_y_continuous(name = "Fixation count") +
  scale_x_continuous(name = "NEP") +
  theme_minimal() +
  theme(legend.position = c(.5, .1))
```

35

Alternative plot of interaction

(with confidence intervals for slopes and different predefined line types)




36

After this workshop … you are able to perform mixed-designed tests of hypotheses for main effects and interaction

(you should be able to run tests of moderation)

- **Prepare the data frames for the analysis**
  - Read/load data into R
  - Calculate new variables
  - Prepare dichotomous factor with median-split
  - Select variables and filter observations
  - Merge two data frames
  - Write data frame to a file
- **Perform mixed-design ANOVA**
  - Read and interpret the results
  - Estimate means
  - Run pairwise comparisons (post hoc tests) for significant effects
  - Prepare publication ready bar graphs
- **Perform mixed-design linear multiple regression analysis**
  - Read and interpret the results
  - Perform trends (simple slopes) analysis
  - Compare statistically simple slopes
  - Prepare publication ready line graphs with confidence intervals

37

Thank you!

Any questions?

38