Data analysis Pilot 1

2024-11-22

Contents

Fit algorithm definition

```
fit_algo <- function(new_call,
                     newData = NULL,
                     type = "lmer")
{
  if (type != "lmer" && type != "glmer")
  {
    stop("type must be lmer or glmer")
  }
  new_call["start"] <- NULL
  new_call["control"] <-
    parse(text = paste0(
      type,
      'Control(optimizer ="bobyqa", optCtrl = list(maxfun=1000000))'
    ))
  error_msg <- tryCatch({
    rval <- eval(new_call)
    if (length(rval@optinfo$conv$lme4) != 0)
    {
      "convergence"
    }
    else
    {
      "ok"
    }
  }, error = function(e) {
    "error"
  })
  count <- 0
  while (error_msg != "ok" && count < 4)
  {
    if (error_msg == "convergence")
    {
      new_call["start"] <- parse(text = "list(fixef = fixef(rval))")
```

```
  }
  count <- count + 1
  error_msg <- tryCatch({
    rval <- eval(new_call)
    if (length(rval@optinfo$conv$lme4) != 0)
    {
      "convergence"
    }
    else
    {
      "ok"
    }
  }, error = function(e) {
    "error"
  })
}
new_call["start"] <- NULL
new_call["control"] <-
  parse(text = paste0(
    type,
    'Control(optimizer ="Nelder_Mead", optCtrl = list(maxfun=1000000))'
  ))
count <- 0
while (error_msg != "ok" && count < 5)
{
  count <- count + 1
  error_msg <- tryCatch({
    rval <- eval(new_call)
    if (length(rval@optinfo$conv$lme4) != 0)
    {
      "convergence"
    }
    else
    {
      "ok"
    }
  }, error = function(e) {
    "error"
  })
  if (error_msg == "convergence")
  {
    new_call["start"] <- parse(text = "list(fixef = fixef(rval))")
  }

}
new_call["start"] <- NULL
new_call["control"] <- NULL
count <- 0
while (error_msg != "ok" && count < 5)
{
  count <- count + 1
  error_msg <- tryCatch({
    rval <- eval(new_call)
```

```
      if (length(rval@optinfo$conv$lme4) != 0)
      {
        "convergence"
      }
      else
      {
        "ok"
      }
    }, error = function(e) {
      "error"
    })
    if (error_msg == "convergence")
    {
      new_call["start"] <- parse(text = "list(fixef = fixef(rval))")
    }

  }
  if (!exists("rval"))
  {
    eval(new_call)
  }
  return(rval)
}
```

Dataset loading

```
datasets <-
  list.files(file.path("final_data"),
             pattern = "*.csv",
             full.names = TRUE)
datasets       <- lapply(datasets, read_csv, show_col_types = FALSE)

d <- datasets[[1]]
for (i in seq(from = 2, to = length(datasets)))
{
  d <- rbind(d, datasets[[i]])
}
```

Dataset preparation

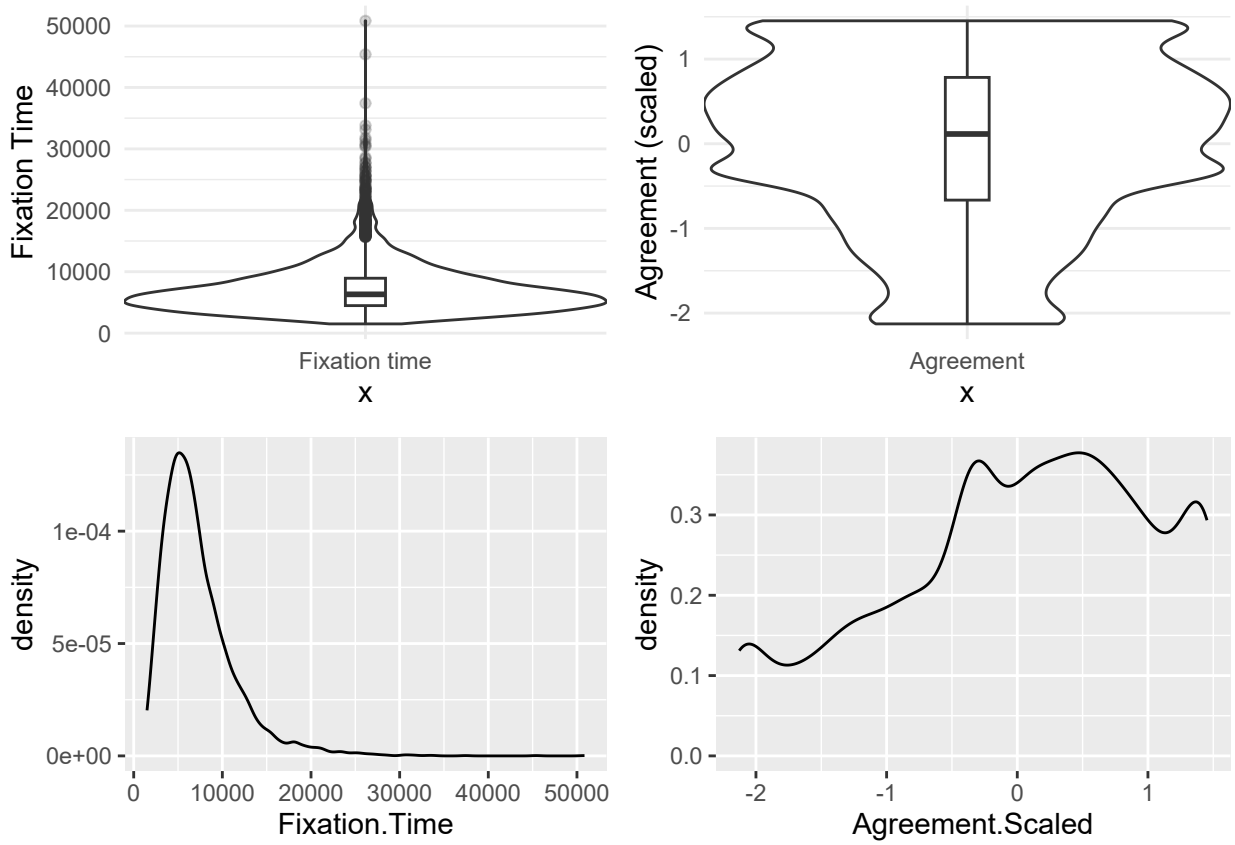
```
d <- d %>% mutate(Subject = rep(seq_len(length(datasets)), each = 100))
```

Then we remove fixation times <=1500ms, scale the Agreement feature and take its square, and log-transform Fixation.Time

```
d_final <- d %>% filter(Fixation.Time > 1500) %>%
  mutate(
    Fixation.Time.Log = log(Fixation.Time),
    Agreement.Scaled = (Agreement - mean(Agreement)) / sd(Agreement),
    Squared.Agreement.Scaled = Agreement.Scaled ^ 2,
    Index = as.factor(Index),
    Subject = as.factor(Subject)
  )
```

Variables plot (violin & boxplot + density)

```
violin_fixation <-
  ggplot(d_final, aes(x = "Fixation time", y = Fixation.Time)) +
  ylab("Fixation Time") + theme_minimal() +
  theme(legend.position = "none") +
  geom_violin(width = 1.2) + geom_boxplot(width = 0.1, alpha = 0.2)


violin_agreement <-
  ggplot(d_final, aes(x = "Agreement", y = Agreement.Scaled)) +
  ylab("Agreement (scaled)") + theme_minimal() +
  theme(legend.position = "none") +
  geom_violin(width = 1.2) + geom_boxplot(width = 0.1, alpha = 0.2)

density_fixation <-
  ggplot(d_final, aes(x = Fixation.Time)) + geom_density()

density_agreement <-
  ggplot(d_final, aes(x = Agreement.Scaled)) + geom_density()

grid.arrange(violin_fixation,
             violin_agreement,
             density_fixation,
             density_agreement,
             nrow = 2)
```



Statistical models (log-tranformed, GLMM with Gamma-distributed errors and LMM with normal-distributed errors)

```
log_call <- parse(
  text = 'lmer(Fixation.Time.Log ~ Agreement.Scaled +
                    Squared.Agreement.Scaled +
                    (1 | Index) +
                    (Agreement.Scaled + Squared.Agreement.Scaled | Subject),
                  data=d_final)'
)[[1]]

fit_log <- fit_algo(log_call)

summary(fit_log)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: Fixation.Time.Log ~ Agreement.Scaled + Squared.Agreement.Scaled +
##     (1 | Index) + (Agreement.Scaled + Squared.Agreement.Scaled |      Subject)
##    Data: d_final
## Control: lmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 1e+06))
##
## REML criterion at convergence: 5400.9
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.3943 -0.6265 -0.0511  0.5946  4.7118
##
## Random effects:
##  Groups   Name                     Variance  Std.Dev. Corr
##  Index    (Intercept)              0.0306873 0.17518
##  Subject  (Intercept)              0.1167606 0.34170
##           Agreement.Scaled         0.0006924 0.02631  -0.32
##           Squared.Agreement.Scaled 0.0009278 0.03046  -0.84  0.28
##  Residual                          0.1488998 0.38588
## Number of obs: 5233, groups:  Index, 100; Subject, 54
##
## Fixed effects:
##                            Estimate Std. Error        df t value Pr(>|t|)
## (Intercept)                8.753405   0.050296 67.875475 174.039  < 2e-16 ***
## Agreement.Scaled          -0.022933   0.007933 83.549180  -2.891  0.00489 **
## Squared.Agreement.Scaled -0.013453   0.007030 43.251784  -1.914  0.06231 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) Agrm.S
## Agrmnt.Scld -0.166
## Sqrd.Agrm.S -0.549  0.317
```

```
fit_call <- parse(text = 'lmer(Fixation.Time ~ Agreement.Scaled +
                    Squared.Agreement.Scaled +
                    (1 | Index) +
                    (1 | Subject),
```

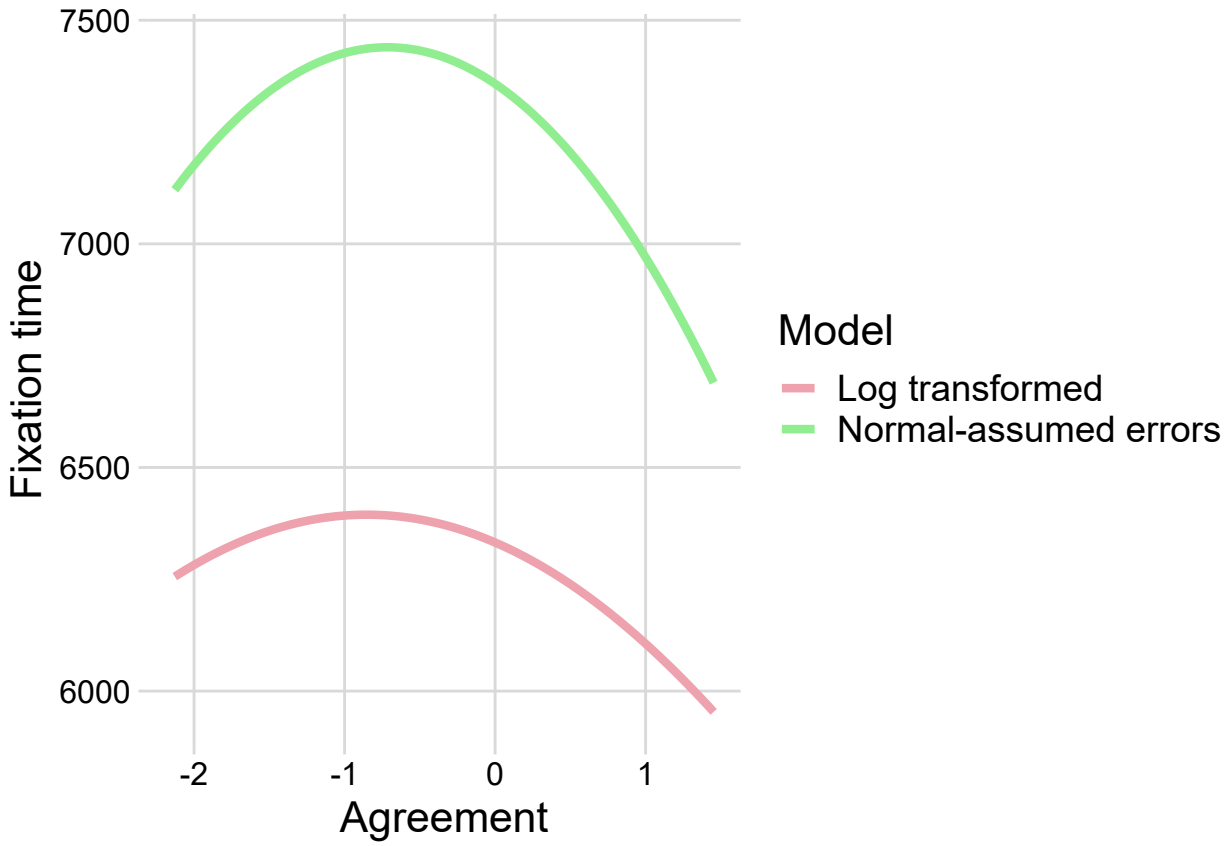
```
                  data=d_final)')[[1]]

fit <- fit_algo(fit_call)

summary(fit)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: Fixation.Time ~ Agreement.Scaled + Squared.Agreement.Scaled +
##     (1 | Index) + (1 | Subject)
##    Data: d_final
## Control: lmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 1e+06))
##
## REML criterion at convergence: 99935.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.0838 -0.5725 -0.1670  0.3644 11.3793
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Index    (Intercept)  1631162 1277
##  Subject  (Intercept)  5062554 2250
##  Residual             10691830 3270
## Number of obs: 5233, groups:  Index, 100; Subject, 54
##
## Fixed effects:
##                            Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)                 7357.90     338.16   73.88  21.759  < 2e-16 ***
## Agreement.Scaled            -228.41      58.69 4936.58  -3.892 0.000101 ***
## Squared.Agreement.Scaled    -159.69      47.28 5196.80  -3.378 0.000736 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) Agrm.S
## Agrmnt.Scld -0.049
## Sqrd.Agrm.S -0.139  0.353
```

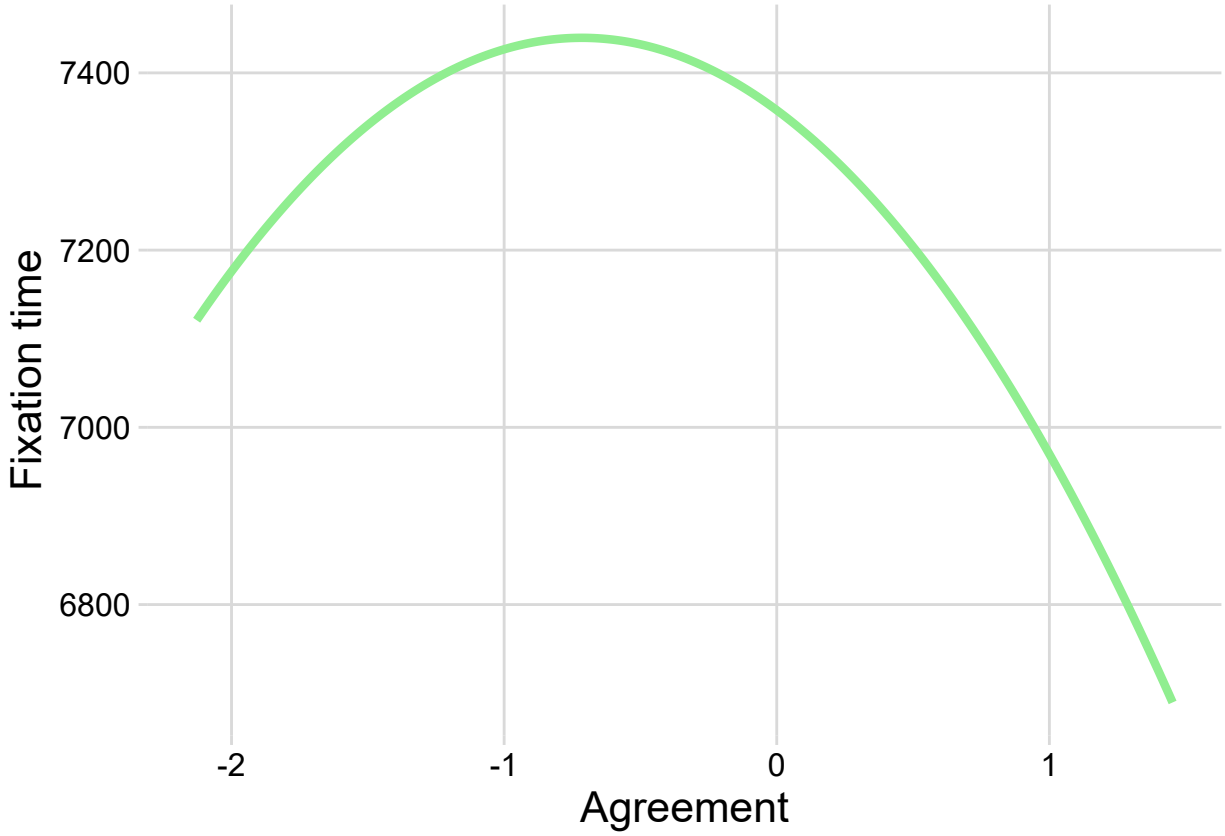
Fixed effects plots

```
d_final$log_prediction <- exp(predict(fit_log, re.form = NA))
d_final$prediction <- predict(fit, re.form = NA)

colors <-
  c(
    "Log transformed" = "lightpink2",
    "Normal-assumed errors" = "lightgreen"
  )

ggplot(d_final) +
  geom_line(aes(x = Agreement.Scaled, log_prediction, color = "Log transformed"),
            linewidth = 1.5) +
```

```
  geom_line(aes(x = Agreement.Scaled, prediction, color = "Normal-assumed errors"),
            linewidth = 1.5) +
  theme_minimal_grid() +
  theme(text = element_text(size = 16)) +
  labs(y = "Fixation time", x = "Agreement", color = "Model") +
  scale_color_manual(values = colors)
```




```
plot <- ggplot(d_final) +
  geom_line(
    aes(x = Agreement.Scaled, prediction),
    color = "lightgreen",
    linewidth = 1.5
  ) +
  theme_minimal_grid() +
  theme(text = element_text(size = 16)) +
  labs(y = "Fixation time", x = "Agreement")
plot
```




```
ggsave(plot, filename = "study1_plot.png", type = "cairo")
```

```
## Saving 6.5 x 4.5 in image
```

```
## Warning: Using ragg device as default. Ignoring `type` and `antialias`
## arguments
```

```
vertex <-
  (-summary(fit)[["coefficients"]][2]) / (2 * summary(fit)[["coefficients"]][3])

pred <-
  predict(
    fit,
    re.form = NA,
    newdata = tibble(
      Agreement.Scaled = c(
        min(d_final$Agreement.Scaled),
        vertex,
        max(d_final$Agreement.Scaled)
      ),
      Squared.Agreement.Scaled = Agreement.Scaled ** 2
    )
  )

table <- round(matrix(
  c(
    vertex,
```

```
    min(d_final$Agreement.Scaled),
    max(d_final$Agreement.Scaled),
    pred[2] - pred[1],
    pred[2] - pred[3]
  ),
  ncol = 1,
  byrow = TRUE
), 3)

rownames(table) <-
  c(
    'Vertex position',
    'min agreement',
    'max agreement',
    'diff min vs. vertex',
    'diff max vs. vertex'
  )
colnames(table) <- c('Results')
table <- as.table(table)
table
```

```
##                     Results
## Vertex position      -0.715
## min agreement        -2.128
## max agreement         1.452
## diff min vs. vertex 318.699
## diff max vs. vertex 750.026
```