Data analysis pilots

Nathanael Larigaldie

2024-01-19

Contents

Dataset loading 1

Dataset preparation 1

Variables plot (violin & boxplot + density) 1

Statistical model 2

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

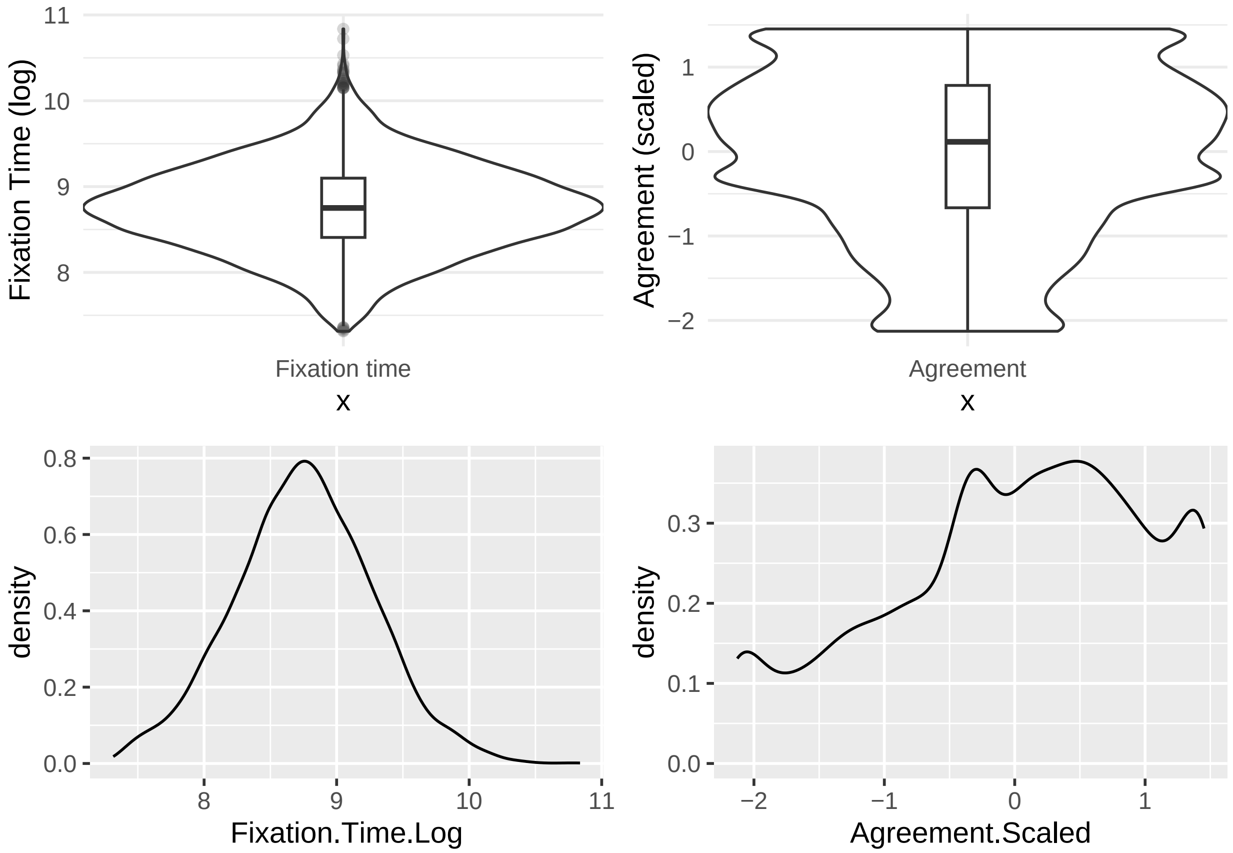
```
d <- d %>% mutate(Subject=rep(seq_len(length(datasets)), each=100))
```

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.Log)) +
  ylab("Fixation Time (log)") + 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.Log)) + 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)
```

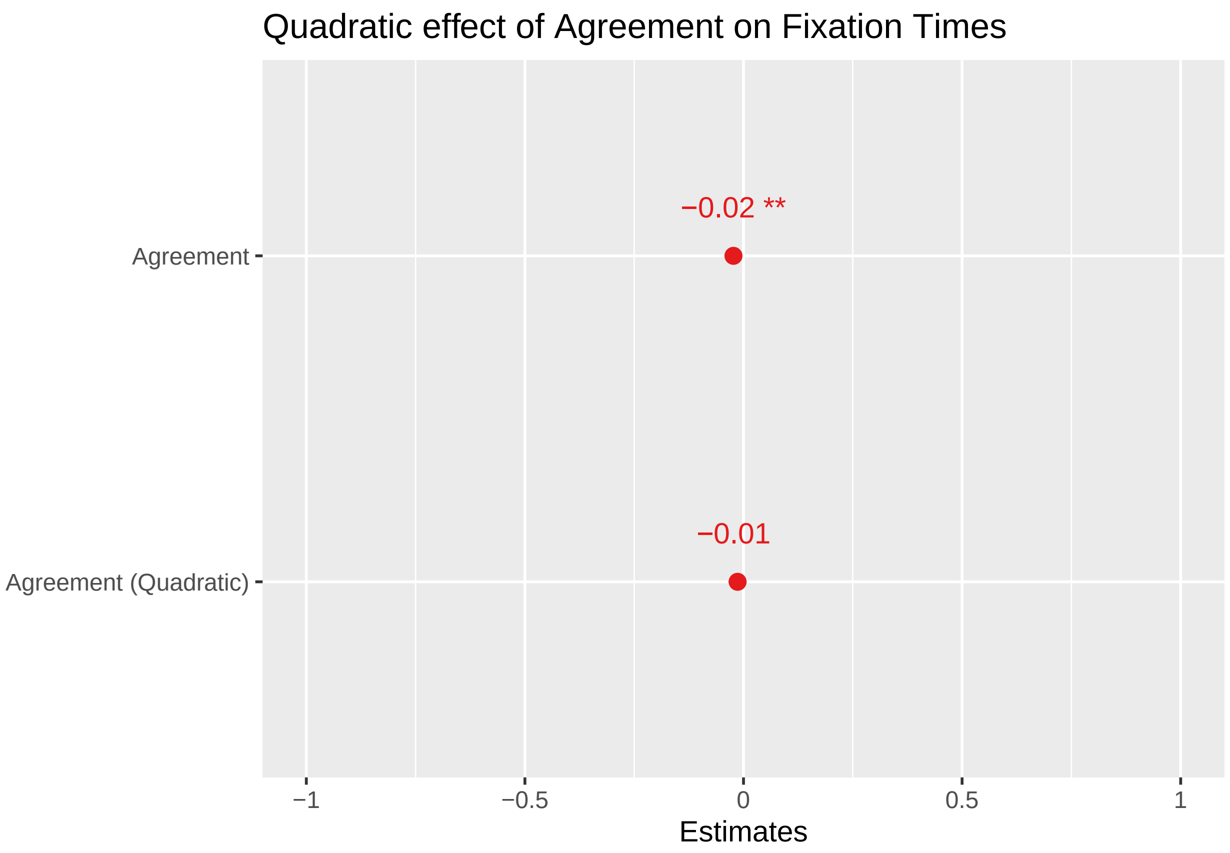


Statistical model

```
res_mixed <- lmer(Fixation.Time.Log ~ Agreement.Scaled +
                    Squared.Agreement.Scaled +
                    (1 | Index) +
                    (Agreement.Scaled + Squared.Agreement.Scaled | Subject),
                  data=d_final, control = lmerControl(optimizer ="bobyqa"))

summary(res_mixed)
```

```
## 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")
##
## 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)              0.1167607 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 67.875373 174.039  < 2e-16 ***
## Agreement.Scaled         -0.022933   0.007933 83.551563  -2.891  0.00489 **
## Squared.Agreement.Scaled -0.013453   0.007030 43.252224  -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
```

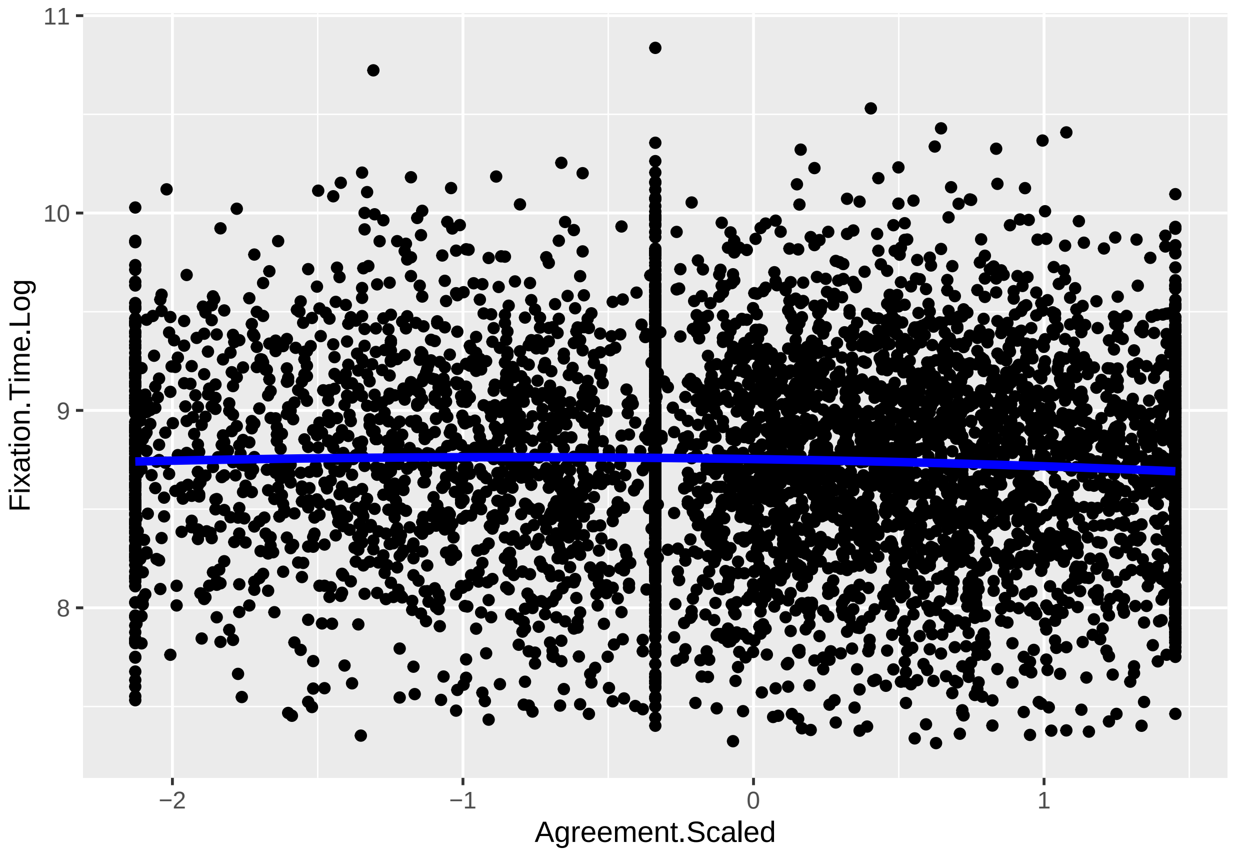
```r
plot_model(res_mixed,
           axis.labels=c("Agreement (Quadratic)", "Agreement"),
           show.values=TRUE, show.p=TRUE,
           title="Quadratic effect of Agreement on Fixation Times")
```



```
quad_function <- function(x, alpha, beta1, beta2)
{
  return(alpha + beta1*x + beta2*x^2)
}
```

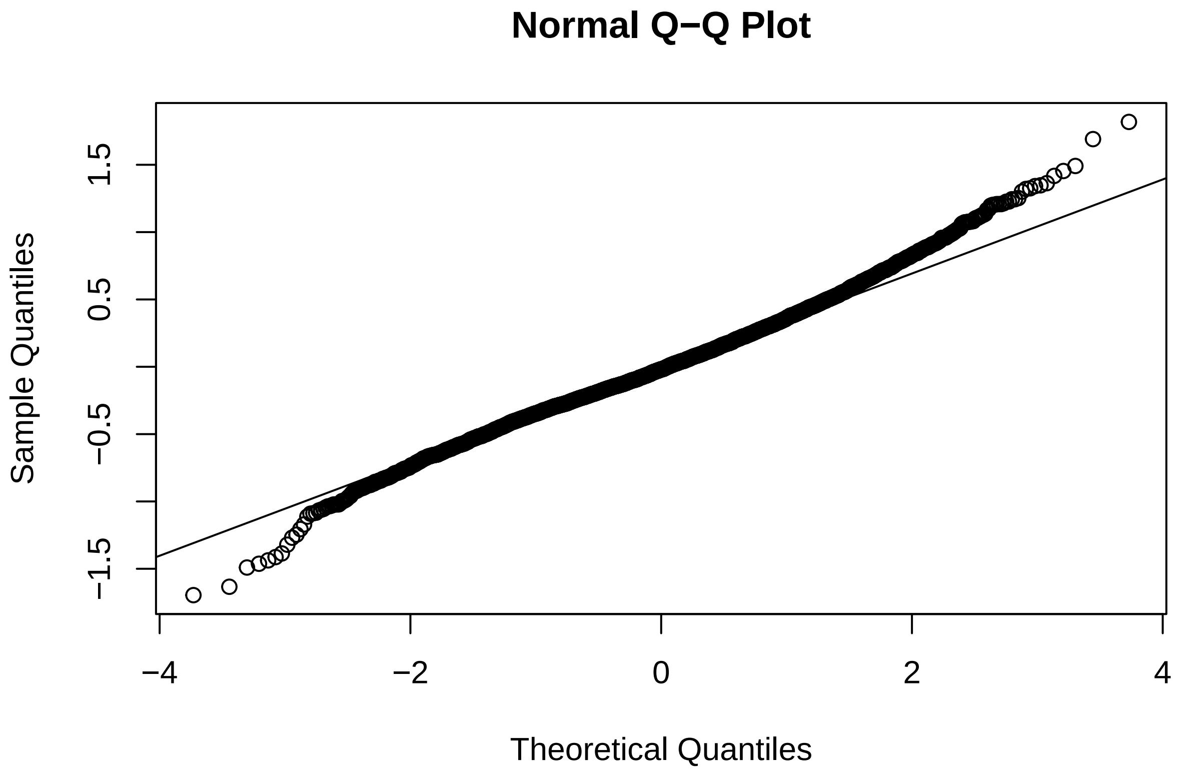
Random effect plot

```
ggplot() +
  geom_point(data=d_final, aes(Agreement.Scaled, Fixation.Time.Log)) +
  geom_function(fun=quad_function,
                args=list(alpha=summary(res_mixed)[["coefficients"]][1],
                                        beta1=summary(res_mixed)[["coefficients"]][2],
                                        beta2=summary(res_mixed)[["coefficients"]][3]),
                color="blue",
                linewidth=1.5)
```

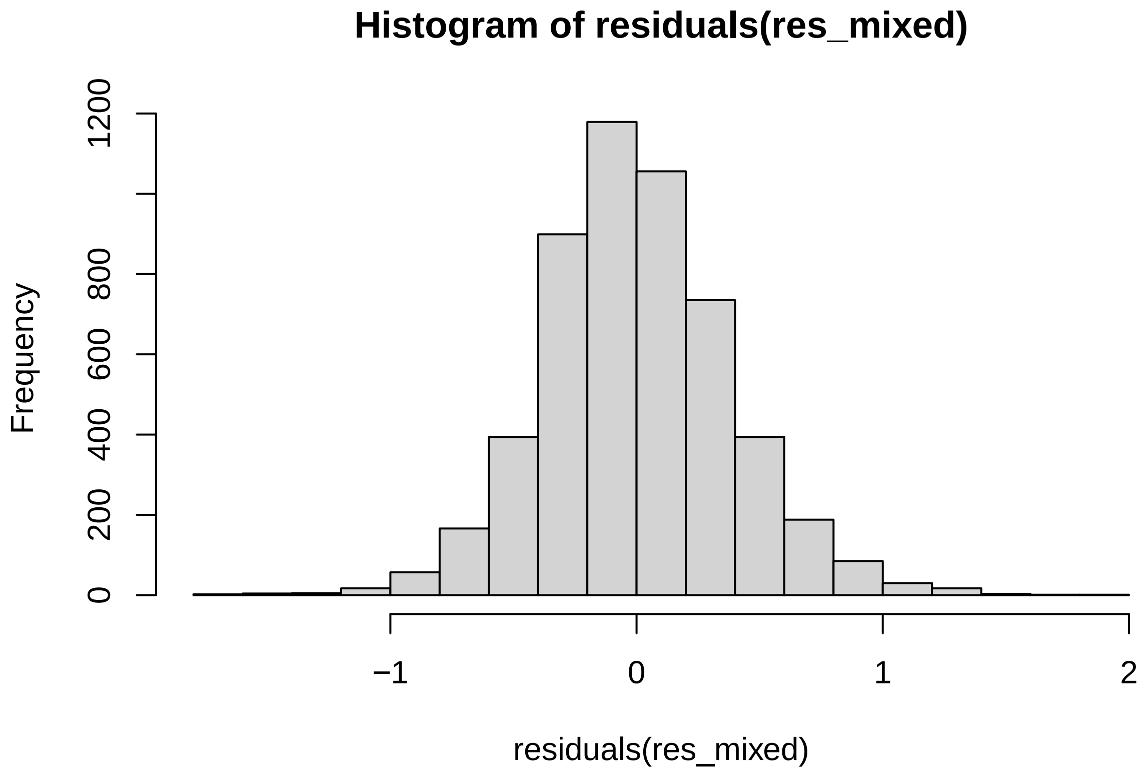


Residuals analysis

```
qqnorm(residuals(res_mixed))
qqline(residuals(res_mixed))
```



```
hist(residuals(res_mixed))
```



```
ks.test(x=residuals(res_mixed), y='pnorm')
```

```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  residuals(res_mixed)
## D = 0.23004, p-value < 2.2e-16
## alternative hypothesis: two-sided
```