Reanalysis of Greco et al. (2022, *Sleep*)

Stephen Rhodes*

Last updated: 2023-03-10

In the "Statement of significance" section, Greco et al. (2022) claim that their findings "[. . .] suggests wearing an eye mask during sleep is an effective, economical, and noninvasive behavior that could benefit cognitive function and lead to measurable impacts on everyday life." Here we examine some of the analysis choices made by Greco et al. and critically assess this claim.

In Experiment 1, paired associates learning (PAL) was assessed on day 6 and 7 of the manipulation (eye mask or control), following 5 nights of habituation. On day 6 participants learned words paired to a particular cue word to a 60% correct criterion. The first test of PAL was then 10 minutes later with the number of correctly recalled words (out of 80) being the outcome. The second PAL assessment occurred on day 7 with no additional learning opportunities in between. The two PAL assessments were split up, for unknown reasons, and analyzed separately via the following mixed effects model (using syntax from the `lme4` package for `R`):

```
lmer(y ~ eye_mask + (1 | ID) + (1 | year), data = data, REML = FALSE)
```

where `y` is the outcome of interest, `year` was a factor with two levels representing changes in the nature of the control condition from 2018 to 2019, and `eye_mask` refers to whether or not participants had worn an eye mask (1 = yes, 0 = no) in the 5 or 6 nights prior to testing.

There was a significant difference in the average number of words recalled between mask and control on day 6. However, these conditions did not differ in terms of an "absolute consolidation" score, which was the difference in performance between days 6 and 7 (reported in the supplement).

It is not clear why the day 6 and 7 results were separated as performance on the PAL task could have been assessed via a single model. Indeed a single model analysis is preferable if the difference between immediate and delayed recall is of interest. Splitting the data to assess a 'mask effect' obviously inflates the type i error rate. A more appropriate model would have also allowed for an interaction between `year` and `eye_mask` as changing the nature of the control condition may have influenced the *difference* between the control and eye mask conditions, which is encoded in the fixed effect for `eye_mask`. Including `year` as a random effect only allows for overall performance to vary by `year` of study. Further, as year only has two levels, it seems inappropriate to try to estimate a variance component for this factor and, as we found when recreating the analysis, there were situations where this caused fit issues in the reported analysis (the 'singular fit' warning from `lme4`).

As participants were assessed under both the control and eye mask conditions assessing the PAL data as a whole allows one to model between participant variability in the size of the eye mask effect (i.e., a random slope term for `eye_mask`). Not accounting for this source of variability in the model is important as the standard errors for the fixed effect of `eye_mask` will be too small, leading to incorrect p-values, and, potentially, incorrect decisions as to whether to reject the null hypothesis of no difference (Barr et al., 2013).

We reanalyzed the PAL data with the following model, which allows for a interaction between condition and year, to account for the different control conditions, and an interaction between condition and day, which codes whether the assessment was on day 6 (immediate) or day 7 (delayed).[1]

*steverho89@gmail.com

[1]We have set `REML=FALSE` to fit the model via maximum likelihood, rather than restricted maximum likelihood (REML).

```
lmer(y ~ eye_mask + year + day + eye_mask:year + eye_mask:day +
            (1 + eye_mask + day | ID),
          data = pal1, REML = FALSE)
```

In this analysis the overall coefficient for `eye_mask` is not significantly different from zero (b = 0.99 [-0.01, 2.00], p = 0.053).[2] Further, the interactions between condition and year or day are not significant. There is also no significant eye mask vs control condition main effect in a mixed models ANOVA, which, unlike the above model, includes all two way and the three way interactions ($F(1, 81) = 3.77$, p = 0.056).

We also reanalyzed average reaction times from the psychomotor vigilance test via a mixed model that allowed for a fixed effect interaction between year and condition, as well as a participant-level random intercept and slope for the difference between eye mask and control. The difference between conditions is significantly different from zero, albeit with greater uncertainty in the magnitude of the difference, b = -6.239 [-12.363, -0.115], p = 0.047.

For the data from Experiment 2, it is not possible to estimate a random participant slope for eye mask as there was only one score per condition. Greco et al. also note that this experiment was not powered to detect differences in PAL or PVT scores, as the goal was to assess correlations with electrophysiological measures (see supplement to article). Nevertheless, we recreated the result for both tasks but found that the number of unique participant IDs in the data files was lower (N = 28) than that reported in the article (N = 33 following exclusions).

When reassessing the data presented with more appropriate models we find less consistent evidence for cognitive effects of wearing an eye mask. However, this reanalysis should also be interpreted with caution. Information on task order was not available in the data files and is potentially important given that some participants were omitted from analysis, thereby possibly leading to an unbalanced design. Further, the participant-level data from the PVT are average reaction times from an unspecified number of trials where responses could be missing due to overly short or long response times (see supplement). Trial level data would allow one to take into account these additional sources of variation/uncertainty.

Even if we were to take these improved estimates at face value it is difficult to imagine how being able to, on average, remember 0 to 2 extra words out of 80 or being able to respond 0.1 to 12 milliseconds faster after wearing an eye mask could lead to measurable impacts on everyday life.

References

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.

Greco, V., Bergamo, D., Cuoccio, P., Konkoly, K. R., Lombardo, K. M., & Lewis, P. A. (2022). Wearing an eye mask during overnight sleep improves episodic learning and alertness. Sleep, online ahead of print. https://doi.org/10.1093/sleep/zsac305

This is to match Greco et al., who used ML to allow for a likelihood ratio test. However, `REML=TRUE` would likely lead to better estimates of fixed effect standard errors.

[2] `year` and `day` were 'sum-to-zero' coded so that the `eye_mask` fixed effect can be directly interpreted as an 'overall' eye mask effect.

Code for Reanalysis

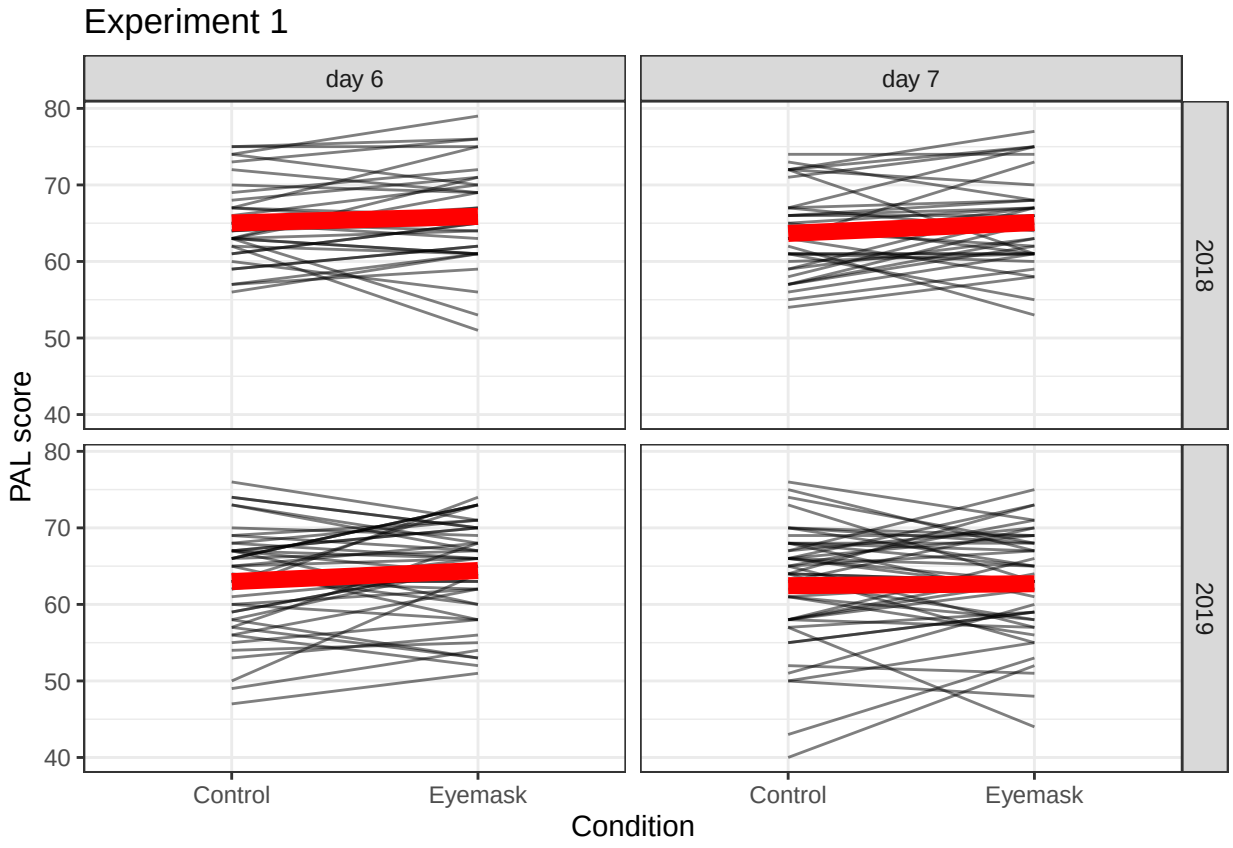
Additional code for results not presented here is available at https://github.com/stephenrho/sleep/blob/main/reanalysis.Rmd

Paired associates learning

```
### PAL (primary outcome) ----
length(unique(pal1$ID))
```

```
## [1] 83
```

```
ggplot(pal1, aes(x = condition, y=y, group=ID)) +
  geom_line(alpha=1/2) +
  stat_summary(aes(x = condition, y=y, group=1),
               fun="mean", geom="line", inherit.aes = F,
               lwd=3, col="red") +
  #facet_wrap(~day) +
  facet_grid(year ~ day) +
  labs(x="Condition", y="PAL score", title="Experiment 1")
```




```
aggregate(y ~ eye_mask, data = subset(pal1, day == "day 6"), FUN = mean)
```

```
##   eye_mask        y
## 1        0 63.86747
## 2        1 65.06024
```

```
aggregate(y ~ eye_mask, data = subset(pal1, day == "day 7"), FUN = mean)
```

```
##   eye_mask        y
## 1        0 62.97590
## 2        1 63.73494
```

```
aggregate(y ~ eye_mask, data = pal1, FUN = mean)
```

```
##   eye_mask        y
## 1        0 63.42169
## 2        1 64.39759
```

```
# original models
lmer(y ~ eye_mask + (1 | ID) + (1 | year), data = subset(pal1, day == "day 6"),
     REML = FALSE) |>
  summary()
```

```
## boundary (singular) fit: see help('isSingular')
```

```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
## Formula: y ~ eye_mask + (1 | ID) + (1 | year)
##    Data: subset(pal1, day == "day 6")
##
##      AIC      BIC   logLik deviance df.resid
##   1026.0   1041.6   -508.0   1016.0      161
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.31541 -0.47568 -0.01511  0.47858  1.81248
##
## Random effects:
##  Groups   Name        Variance  Std.Dev.
##  ID       (Intercept) 2.688e+01 5.184e+00
##  year     (Intercept) 9.193e-09 9.588e-05
##  Residual             1.097e+01 3.312e+00
## Number of obs: 166, groups:  ID, 83; year, 2
##
## Fixed effects:
##             Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)  63.8675     0.6752 110.3482   94.58   <2e-16 ***
## eye_mask      1.1928     0.5141  82.9998    2.32   0.0228 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr)
## eye_mask -0.381
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
```

```
lmer(y ~ eye_mask + (1 | ID) + (1 | year), data = subset(pal1, day == "day 7"),
     REML = FALSE) |>
  summary()
```

```
## boundary (singular) fit: see help('isSingular')
```

```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
```

```
## Formula: y ~ eye_mask + (1 | ID) + (1 | year)
##    Data: subset(pal1, day == "day 7")
##
##      AIC      BIC   logLik deviance df.resid
##   1059.7   1075.3   -524.9   1049.7      161
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.47284 -0.53857  0.03685  0.54086  1.70558
##
## Random effects:
##  Groups   Name        Variance  Std.Dev.
##  ID       (Intercept) 2.939e+01 5.421e+00
##  year     (Intercept) 9.660e-09 9.828e-05
##  Residual             1.454e+01 3.813e+00
## Number of obs: 166, groups:  ID, 83; year, 2
##
## Fixed effects:
##             Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)  62.9759     0.7275 114.6671  86.565   <2e-16 ***
## eye_mask      0.7590     0.5919  83.0002   1.282    0.203
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr)
## eye_mask -0.407
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
```

```
# note: in both cases there is a singular fit warning
# most likely trying to estimate random effect for 2 groups...

# more appropriate model
# (1) analyze the data from the PAL task as a whole.
# no need to separate days.
# (2) account for ID level variability in difference
# between eye mask and control
# (3) year = different control conditions, so should account
# for *interaction* between eye mask and year
# doesn't make sense to include year as random effect

# make contrast for year and day sum to zero so coefficient
# for mask is at 'average' year
contrasts(pal1$year) = c(-1,1)
contrasts(pal1$day) = c(-1,1)
mod = lmer(y ~ eye_mask + year + day + eye_mask:year + eye_mask:day +
             (1 + eye_mask + day | ID),
           data = pal1, REML = FALSE)

summary(mod)
```

```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
## Formula: y ~ eye_mask + year + day + eye_mask:year + eye_mask:day + (1 +
```

```
##     eye_mask + day | ID)
##    Data: pal1
##
##      AIC      BIC   logLik deviance df.resid
##   1934.7   1984.1   -954.3   1908.7      319
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.4422 -0.3903  0.0354  0.4336  1.8659
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  ID       (Intercept) 34.929   5.910
##           eye_mask    16.315   4.039    -0.37
##           day1         1.268   1.126     0.15  0.04
##  Residual              4.585   2.141
## Number of obs: 332, groups:  ID, 83
##
## Fixed effects:
##                Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)     63.5294     0.6756  83.0275  94.040   <2e-16 ***
## eye_mask         0.9949     0.5063  83.0007   1.965   0.0527 .
## year1           -0.8127     0.6720  82.9971  -1.209   0.2299
## day1            -0.4458     0.2071 147.3204  -2.152   0.0330 *
## eye_mask:year1  -0.1432     0.5062  82.9992  -0.283   0.7780
## eye_mask:day1   -0.2169     0.2350  83.0008  -0.923   0.3589
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) ey_msk year1  day1   ey_msk:y1
## eye_mask    -0.399
## year1       -0.132  0.053
## day1         0.085  0.018  0.000
## eye_msk:yr1  0.053 -0.132 -0.404  0.000
## eye_msk:dy1  0.000  0.000  0.000 -0.567  0.000
```

```
confint(mod)
```

```
## Computing profile confidence intervals ...
## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep
```

```
## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep
```

```
## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep
```

```
## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep
```

```
## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep
```

```
## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in nextpar(mat, cc, i, delta, lowcut, upcut): Last two rows have
## identical or NA .zeta values: using minstep

## Warning in FUN(X[[i]], ...): non-monotonic profile for .sig03

## Warning in confint.thpr(pp, level = level, zeta = zeta): bad spline fit
## for .sig03: falling back to linear interpolation

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

##                   2.5 %      97.5 %
## .sig01         5.053420664  7.00022554
## .sig02        -0.562801918 -0.13689757
## .sig03        -0.158218807  0.44300848
## .sig04         3.288056165  4.93011533
## .sig05        -0.298900327  0.36696218
## .sig06         0.725117612  1.49258991
## .sigma         1.852737338  2.51354342
```

```
## (Intercept)    62.189849761 64.86888442
## eye_mask       -0.009002654  1.99875178
## year1          -2.145130099  0.51972010
## day1           -0.854400642 -0.03716564
## eye_mask:year1 -1.146831500  0.86047341
## eye_mask:day1  -0.682932725  0.24919783
```

```
# some warnings but profile plots/zeta diagrams look ok
# https://stackoverflow.com/questions/74018300/warnings-when-computing-confidence-intervals-using-confi
# pp = profile(mod)
# lattice::xyplot(pp)

# mixed ANOVA (note: tests extra interactions)
aov_car(y ~ condition*year*day + Error(ID/condition*day),
        data = pal1, fun_aggregate = mean)
```

```
## Contrasts set to contr.sum for the following variables: year
```

```
## Anova Table (Type 3 tests)
##
## Response: y
##                Effect    df    MSE        F   ges p.value
## 1                year 1, 81 128.37     2.01  .019    .160
## 2           condition 1, 81  21.42   3.77 +  .006    .056
## 3      year:condition 1, 81  21.42     0.08 <.001    .780
## 4                 day 1, 81   9.89 10.07 **  .007    .002
## 5            year:day 1, 81   9.89     0.00 <.001    .949
## 6       condition:day 1, 81   4.51     0.46 <.001    .501
## 7 year:condition:day 1, 81   4.51   3.45 +  .001    .067
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '+' 0.1 ' ' 1
```

```
# Experiment 2
# n unique IDs don't match N = 33 included in analysis
length(unique(pal2$ID))
```

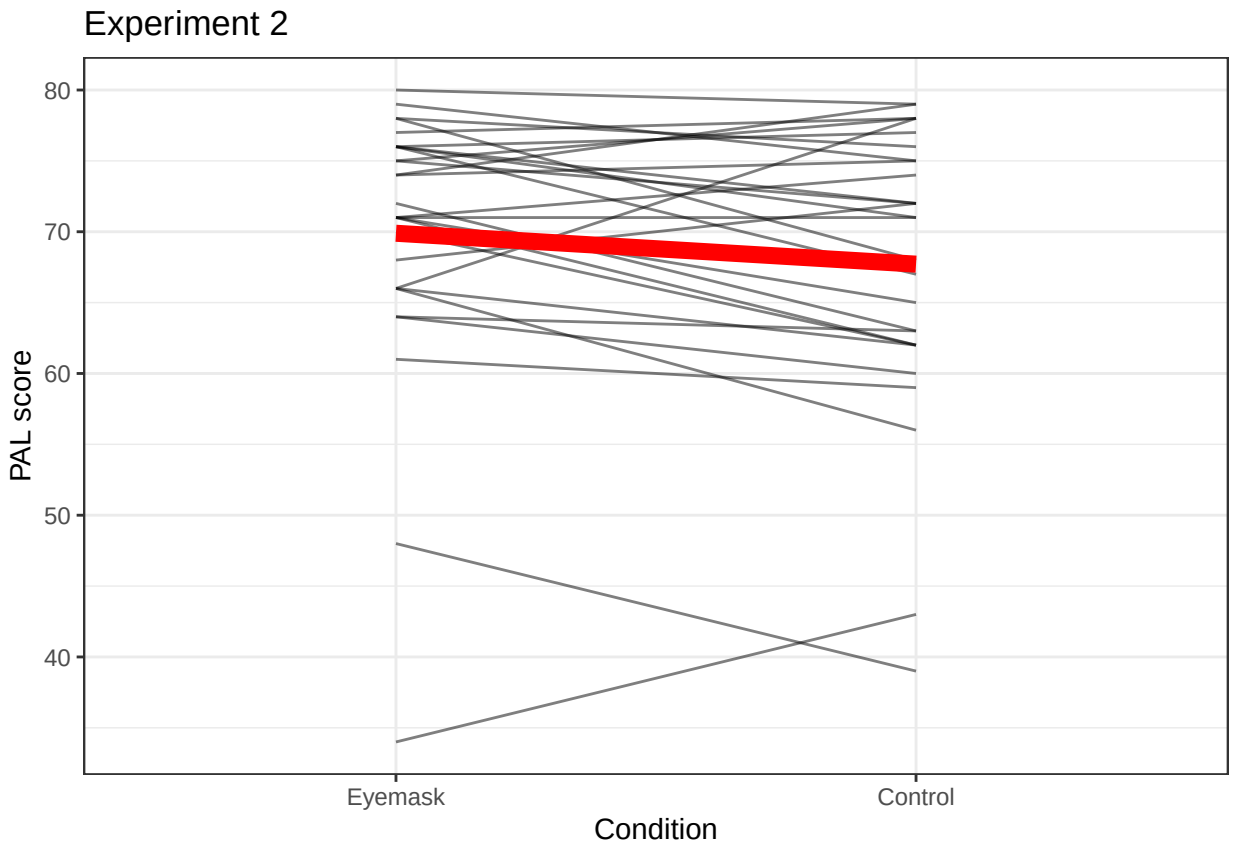
```
## [1] 28
```

```
aggregate(y ~ condition, pal2, mean)
```

```
##   condition        y
## 1   Eyemask 69.89286
## 2   Control 67.71429
```

```
# the means match though...

ggplot(pal2, aes(x = condition, y=y, group=ID)) +
  geom_line(alpha=1/2) +
  stat_summary(aes(x = condition, y=y, group=1),
               fun="mean", geom="line", inherit.aes = F,
               lwd=3, col="red") +
  labs(x="Condition", y="PAL score", title="Experiment 2")
```




```
# recreate original analysis
mod = lmer(y ~ eye_mask + (1  | ID), data = pal2, REML = F)

summary(mod)
```

```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
## Formula: y ~ eye_mask + (1 | ID)
##    Data: pal2
##
##      AIC      BIC   logLik deviance df.resid
##    389.1    397.2   -190.5    381.1       52
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.08816 -0.40609  0.05209  0.53920  1.85424
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  ID       (Intercept) 80.33    8.963
##  Residual             15.82    3.978
## Number of obs: 56, groups:  ID, 28
##
## Fixed effects:
##             Estimate Std. Error     df t value Pr(>|t|)
## (Intercept)   67.714      1.853 32.981  36.541   <2e-16 ***
## eye_mask       2.179      1.063 28.000   2.049   0.0499 *
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr)
## eye_mask -0.287
```

```
confint(mod)
```

```
## Computing profile confidence intervals ...
```

```
##                   2.5 %    97.5 %
## .sig01       6.79628773 12.198409
## .sigma       3.12625783  5.299753
## (Intercept) 63.97268304 71.455888
## eye_mask     0.02132378  4.335819
```

```
aov_car(y ~ condition + Error(ID/condition), data = pal2, fun_aggregate = mean)
```

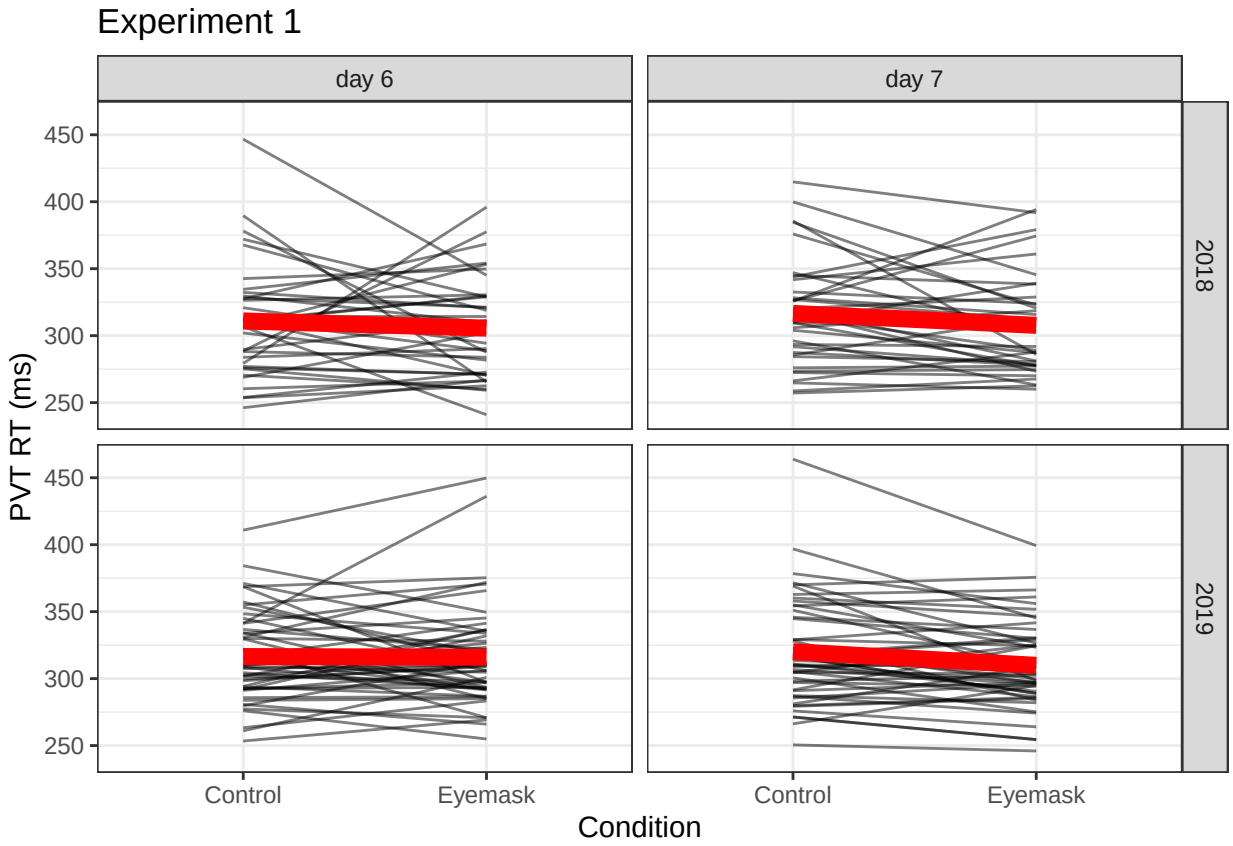
```
## Anova Table (Type 3 tests)
##
## Response: y
##      Effect    df   MSE      F  ges p.value
## 1 condition 1, 27 16.41 4.05 + .012    .054
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '+' 0.1 ' ' 1
```

Psychomotor vigilance test

```
### PVT ----
length(unique(pvt1$ID))
```

```
## [1] 85
```

```
ggplot(pvt1, aes(x = condition, y=y, group=ID)) +
  geom_line(alpha=1/2) +
  stat_summary(aes(x = condition, y=y, group=1),
               fun="mean", geom="line", inherit.aes = F,
               lwd=3, col="red") +
  #facet_wrap(~day) +
  facet_grid(year ~ day) +
  labs(x="Condition", y="PVT RT (ms)", title="Experiment 1")
```




```
# this matches reported means (but figure 2b means look different?)
aggregate(y ~ eye_mask, data = pvt1, FUN = mean)
```

```
##   eye_mask        y
## 1        0 316.3667
## 2        1 310.2642
```

```
# original model
lmer(y ~ eye_mask + (1 | ID) + (1 | year), data = pvt1, REML = FALSE) |>
  summary()
```

```
## boundary (singular) fit: see help('isSingular')
```

```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
## Formula: y ~ eye_mask + (1 | ID) + (1 | year)
##    Data: pvt1
##
##      AIC      BIC   logLik deviance df.resid
##   3229.9   3249.0  -1609.9   3219.9      335
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.1558 -0.5352 -0.0691  0.4103  3.9252
##
## Random effects:
##  Groups   Name        Variance  Std.Dev.
##  ID       (Intercept) 9.650e+02 3.106e+01
##  year     (Intercept) 1.661e-13 4.076e-07
```

```
##  Residual             4.264e+02 2.065e+01
## Number of obs: 340, groups:  ID, 85; year, 2
##
## Fixed effects:
##             Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)  316.367      3.723 102.416  84.975  < 2e-16 ***
## eye_mask      -6.103      2.240 255.000  -2.725  0.00688 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr)
## eye_mask -0.301
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
```

```
# note: boundary (singular) fit: see help('isSingular')

# more appropriate model
contrasts(pvt1$year) = c(-1,1)
mod = lmer(y ~ eye_mask*year + (1 + eye_mask | ID), data = pvt1, REML = FALSE)

summary(mod)
```

```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
## Formula: y ~ eye_mask * year + (1 + eye_mask | ID)
##    Data: pvt1
##
##      AIC      BIC   logLik deviance df.resid
##   3191.8   3222.4  -1587.9   3175.8      332
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.7233 -0.4018 -0.0342  0.3265  4.0790
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  ID       (Intercept) 1221.5   34.95
##           eye_mask     549.2   23.44    -0.43
##  Residual              243.1   15.59
## Number of obs: 340, groups:  ID, 85
##
## Fixed effects:
##                Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)    316.0236     4.0223  84.9998  78.568   <2e-16 ***
## eye_mask        -6.2391     3.0894  85.0002  -2.019   0.0466 *
## year1            2.2433     4.0223  84.9998   0.558   0.5785
## eye_mask:year1   0.8929     3.0894  85.0002   0.289   0.7733
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) ey_msk year1
## eye_mask    -0.462
```

```
## year1       -0.153  0.071
## eye_msk:yr1  0.071 -0.153 -0.462
```

```
confint(mod)
```

```
## Computing profile confidence intervals ...
```

```
##                      2.5 %      97.5 %
## .sig01          29.7633896  41.4972967
## .sig02          -0.6217213  -0.1952631
## .sig03          18.5268658  29.1157177
## .sigma          14.0705657  17.4081695
## (Intercept)    308.0501885 323.9970280
## eye_mask       -12.3633000  -0.1148689
## year1           -5.7300893  10.2167502
## eye_mask:year1  -5.2313556   7.0170756
```

```
# mixed ANOVA (averages day 6 and 7)
aov_car(y ~ condition*year + Error(ID/condition),
        data = pvt1, fun_aggregate = mean)
```

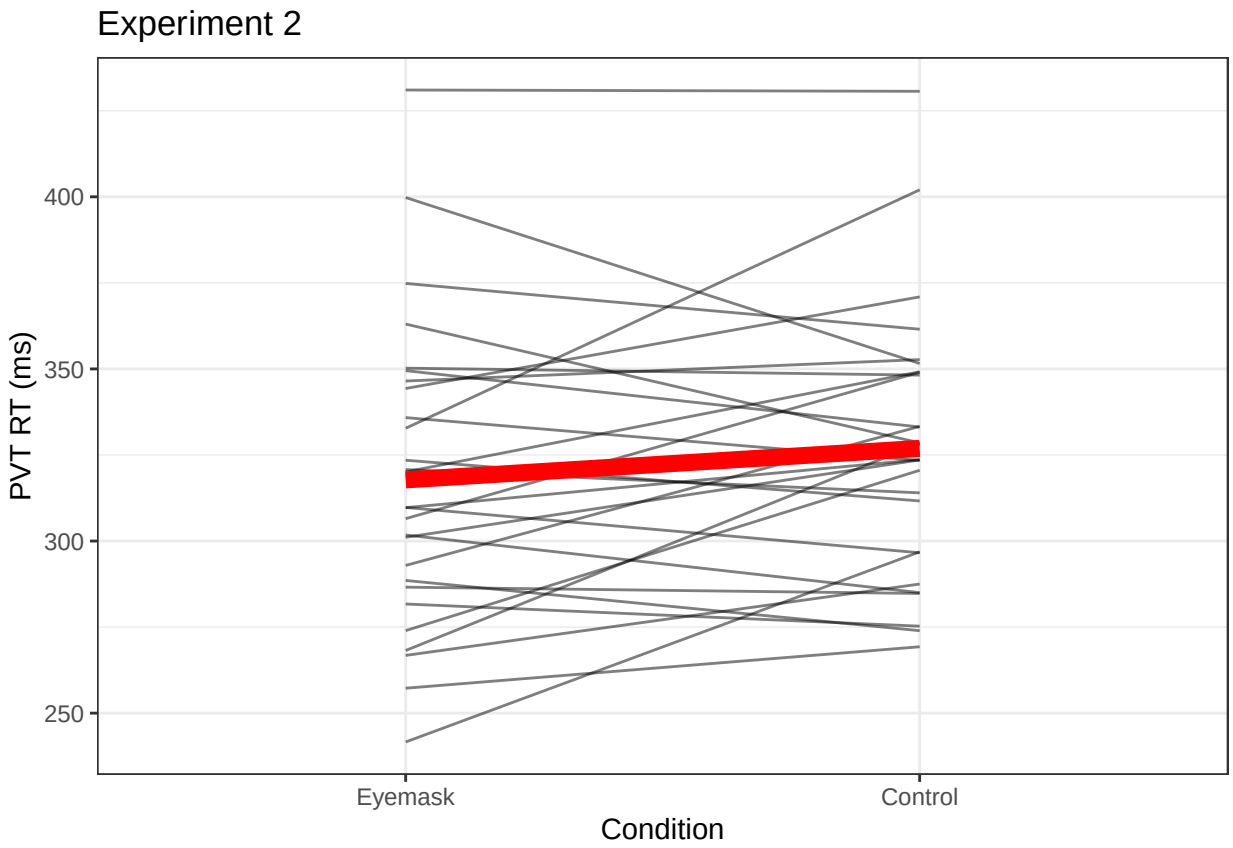
```
## Contrasts set to contr.sum for the following variables: year
```

```
## Anova Table (Type 3 tests)
##
## Response: y
##           Effect    df     MSE      F   ges p.value
## 1           year 1, 83 2180.32   0.55  .006    .460
## 2      condition 1, 83  405.70 3.98 *  .007    .049
## 3 year:condition 1, 83  405.70   0.08 <.001    .776
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '+' 0.1 ' ' 1
```

```
# Experiment 2
# also 28, not 33
length(unique(pvt2$ID))
```

```
## [1] 28
```

```
ggplot(pvt2, aes(x = condition, y=y, group=ID)) +
  geom_line(alpha=1/2) +
  stat_summary(aes(x = condition, y=y, group=1),
               fun="mean", geom="line", inherit.aes = F,
               lwd=3, col="red") +
  labs(x="Condition", y="PVT RT (ms)", title="Experiment 2")
```




```
aggregate(y ~ condition, pvt2, mean)
```

```
##   condition        y
## 1   Eyemask 317.7671
## 2   Control 326.8975
```

```
aov_car(y ~ condition + Error(ID/condition), data = pvt2, fun_aggregate = mean)
```

```
## Anova Table (Type 3 tests)
##
## Response: y
##      Effect    df    MSE    F  ges p.value
## 1 condition 1, 27 420.39 2.78 .013    .107
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '+' 0.1 ' ' 1
```

```
lmer(y ~ eye_mask + (1  | ID), data = pvt2, REML = F) |>
  summary()
```

```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
## Formula: y ~ eye_mask + (1 | ID)
##    Data: pvt2
##
##      AIC      BIC   logLik deviance df.resid
##    556.9    565.0   -274.4    548.9       52
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -1.53243 -0.54374  0.00569  0.56357  1.82262
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  ID       (Intercept) 1175.6   34.29
##  Residual              405.4   20.13
## Number of obs: 56, groups:  ID, 28
##
## Fixed effects:
##             Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)  326.898      7.514  36.061  43.503   <2e-16 ***
## eye_mask      -9.130      5.381  28.000  -1.697    0.101
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr)
## eye_mask -0.358
```