# Supplement 5

Experiment 1 – Modelling the SPARS stimulus-response relationship

*Peter Kamerman and Tory Madden*

*17 June 2018*

## Contents



---

This script is part 2 of our analysis of the stimulus-response characteristics of the SPARS. This script models the relationship between stimulus intensity and SPARS rating using linear mixed models and quantile mixed model regression.

Source URL: https://github.com/kamermanpr/SPARS/tree/supplementary_pdfs

Descriptive plots of the data are provided in *"outputs/supplement_4.pdf"*, the diagnostics on the final linear mixed model are described in *"outputs/supplement_6.pdf"*, the stability of the model is described in *"outputs/supplement_7.pdf"*, the sensitivity of the scale to changes in stimulus intensity are described in *"outputs/experiment_1_sensitivity.pdf"*, and the variance in ratings at each stimulus intensity is described in *"outputs/experiment_1_variance.pdf"*.

---

## Import and clean/transform data

```
#################################################################
#                                                               #
#                           Import                              #
#                                                               #
#################################################################
data <- read_rds('./data-cleaned/SPARS_A.rds')

#################################################################
#                                                               #
#                           Clean                               #
#                                                               #
#################################################################
data %<>%
  # Select required columns
  select(PID, block, block_order, trial_number, intensity, intensity_char, rating)

#################################################################
#                                                               #
```

```
#                Calculate 'Tukey trimean'                 #
#                                                          #
############################################################
# Define tri.mean function
tri.mean <- function(x) {
  # Calculate quantiles
  q1 <- quantile(x, probs = 0.25, na.rm = TRUE)[[1]]
  q2 <- median(x, na.rm = TRUE)
  q3 <- quantile(x, probs = 0.75, na.rm = TRUE)[[1]]
  # Calculate trimean
  tm <- (q2 + ((q1 + q3) / 2)) / 2
  # Convert to integer
  tm <- as.integer(round(tm))
  return(tm)
}

############################################################
#                                                          #
#                    Generate core data                    #
#                                                          #
############################################################
# Calculate the participant average
data_tm <- data %>%
  group_by(PID, intensity) %>%
  summarise(tri_mean = tri.mean(rating)) %>%
  ungroup()

# Calculate the group average
data_group <- data_tm %>%
  group_by(intensity) %>%
  summarise(median = median(tri_mean)) %>%
  ungroup()
```

---

## Linear mixed model regression

To allow for a curvilinear relationship between stimulus intensity and rating, we modelled the data using polynomial regression, with 1st (linear), 2nd (quadratic), and 3rd (cubic) order orthogonal polynomials. For each polynomial expression, we modelled the random effects as random intercept only, and as random intercept and slope.

The random intercept only and random intercept and slope models were compared using the likelihood test, and the better model taken forward.

#### 1st-order (linear) polynomial

```
# Intercept only
lmm1 <- lmer(tri_mean ~ intensity + (1 | PID),
             data = data_tm,
             REML = TRUE)
```

```r
# Intercept and slope
lmm1b <- lmer(tri_mean ~ intensity + (intensity | PID),
              data = data_tm,
              REML = TRUE)

# Better model?
anova(lmm1, lmm1b)
```

```
## Data: data_tm
## Models:
## lmm1: tri_mean ~ intensity + (1 | PID)
## lmm1b: tri_mean ~ intensity + (intensity | PID)
##       Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## lmm1   4 1814.7 1828.7 -903.37   1806.7
## lmm1b  6 1733.6 1754.6 -860.79   1721.6 85.146      2  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Anova of better model
Anova(lmm1b,
      type = 2,
      test.statistic = 'F')
```

```
## Analysis of Deviance Table (Type II Wald F tests with Kenward-Roger df)
##
## Response: tri_mean
##                 F Df Df.res    Pr(>F)
## intensity 94.707  1 17.998 1.356e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Print better model
summary(lmm1b)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: tri_mean ~ intensity + (intensity | PID)
##    Data: data_tm
##
## REML criterion at convergence: 1715.2
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.0493 -0.4430  0.0157  0.5165  3.6042
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  PID      (Intercept) 633.16   25.163
##           intensity    36.17    6.014   -0.89
##  Residual              42.54    6.522
## Number of obs: 244, groups:  PID, 19
##
## Fixed effects:
##             Estimate Std. Error t value
```

```
## (Intercept)  -39.764      5.895  -6.746
## intensity     14.126      1.451   9.732
##
## Correlation of Fixed Effects:
##           (Intr)
## intensity -0.885
```

```
# Doesn't work with LaTex
# sjt.lmer(lmm1b,
#          show.header = TRUE,
#          string.dv = "Response",
#          string.pred = "Coefficients",
#          depvar.labels = '',
#          pred.labels = 'intensity',
#          string.est = 'Estimate',
#          string.ci = '95% CI',
#          string.p = 'p-value',
#          show.icc = FALSE,
#          show.r2 = FALSE)
```

#### 2nd-order (quadratic) polynomial

```
# Intercept only
lmm2 <- lmer(tri_mean ~ poly(intensity, 2) + (1 | PID),
             data = data_tm,
             REML = TRUE)

# Intercept and slope
lmm2b <- lmer(tri_mean ~ poly(intensity, 2) + (intensity | PID),
              data = data_tm,
              REML = TRUE)

# Better model?
anova(lmm2, lmm2b)
```

```
## Data: data_tm
## Models:
## lmm2: tri_mean ~ poly(intensity, 2) + (1 | PID)
## lmm2b: tri_mean ~ poly(intensity, 2) + (intensity | PID)
##       Df    AIC    BIC  logLik deviance Chisq Chi Df Pr(>Chisq)
## lmm2   5 1816.7 1834.2 -903.35   1806.7
## lmm2b  7 1735.5 1760.0 -860.74   1721.5 85.22      2  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Anova for better model
Anova(lmm2b,
      type = 2,
      test.statistic = 'F')
```

```
## Analysis of Deviance Table (Type II Wald F tests with Kenward-Roger df)
##
## Response: tri_mean
```

```
##                          F Df Df.res    Pr(>F)
## poly(intensity, 2) 46.667  2 43.413 1.526e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Print better model
summary(lmm2b)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: tri_mean ~ poly(intensity, 2) + (intensity | PID)
##    Data: data_tm
##
## REML criterion at convergence: 1704.1
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.0263 -0.4333  0.0007  0.5147  3.6042
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  PID      (Intercept) 633.22   25.164
##           intensity    36.17    6.014   -0.89
##  Residual              42.73    6.537
## Number of obs: 244, groups:  PID, 19
##
## Fixed effects:
##                     Estimate Std. Error t value
## (Intercept)           -4.666      3.184  -1.465
## poly(intensity, 2)1  205.327     21.102   9.730
## poly(intensity, 2)2    2.061      6.553   0.315
##
## Correlation of Fixed Effects:
##             (Intr) p(,2)1
## ply(ntn,2)1 -0.505
## ply(ntn,2)2  0.001  0.002
```

```r
# Doesn't work with LaTex
# sjt.lmer(lmm2b,
#          show.header = TRUE,
#          string.dv = "Response",
#          string.pred = "Coefficients",
#          depvar.labels = '',
#          pred.labels = 'intensity',
#          string.est = 'Estimate',
#          string.ci = '95% CI',
#          string.p = 'p-value',
#          show.icc = FALSE,
#          show.r2 = FALSE)
```

**3rd-order (cubic) polynomial**

```r
# Intercept only
lmm3 <- lmer(tri_mean ~ poly(intensity, 3) + (1 | PID),
             data = data_tm,
             REML = TRUE)

# Intercept and slope
lmm3b <- lmer(tri_mean ~ poly(intensity, 3) + (intensity | PID),
              data = data_tm,
              REML = TRUE)

# Better model?
anova(lmm3, lmm3b)
```

```
## Data: data_tm
## Models:
## lmm3: tri_mean ~ poly(intensity, 3) + (1 | PID)
## lmm3b: tri_mean ~ poly(intensity, 3) + (intensity | PID)
##       Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## lmm3   6 1813.8 1834.8 -900.90   1801.8
## lmm3b  8 1727.0 1754.9 -855.48   1711.0 90.841      2  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Anova for better model
Anova(lmm3b,
      type = 2,
      test.statistic = 'F')
```

```
## Analysis of Deviance Table (Type II Wald F tests with Kenward-Roger df)
##
## Response: tri_mean
##                          F Df Df.res    Pr(>F)
## poly(intensity, 3) 34.148  3 71.491 8.318e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Print better model
summary(lmm3b)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: tri_mean ~ poly(intensity, 3) + (intensity | PID)
##    Data: data_tm
##
## REML criterion at convergence: 1688.1
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.0170 -0.4757  0.0340  0.4967  3.4425
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  PID      (Intercept) 639.31   25.285
##           intensity    36.93    6.077   -0.89
##  Residual              40.77    6.385
```

```
## Number of obs: 244, groups:  PID, 19
##
## Fixed effects:
##                     Estimate Std. Error t value
## (Intercept)           -4.666      3.178  -1.468
## poly(intensity, 3)1  205.350     21.255   9.661
## poly(intensity, 3)2    2.125      6.401   0.332
## poly(intensity, 3)3   20.946      6.399   3.273
##
## Correlation of Fixed Effects:
##             (Intr) p(,3)1 p(,3)2
## ply(ntn,3)1 -0.507
## ply(ntn,3)2  0.001  0.002
## ply(ntn,3)3  0.000  0.000  0.003
```

```
# Doeasn't work with LaTex
# sjt.lmer(lmm3b,
#          show.header = TRUE,
#          string.dv = "Response",
#          string.pred = "Coefficients",
#          depvar.labels = '',
#          pred.labels = 'intensity',
#          string.est = 'Estimate',
#          string.ci = '95% CI',
#          string.p = 'p-value',
#          show.icc = FALSE,
#          show.r2 = FALSE)
```

**Compare models**

```
knitr::kable(broom::tidy(anova(lmm1b, lmm2b, lmm3b)),
             caption = 'Linear model vs quadratic model and cubic model')
```

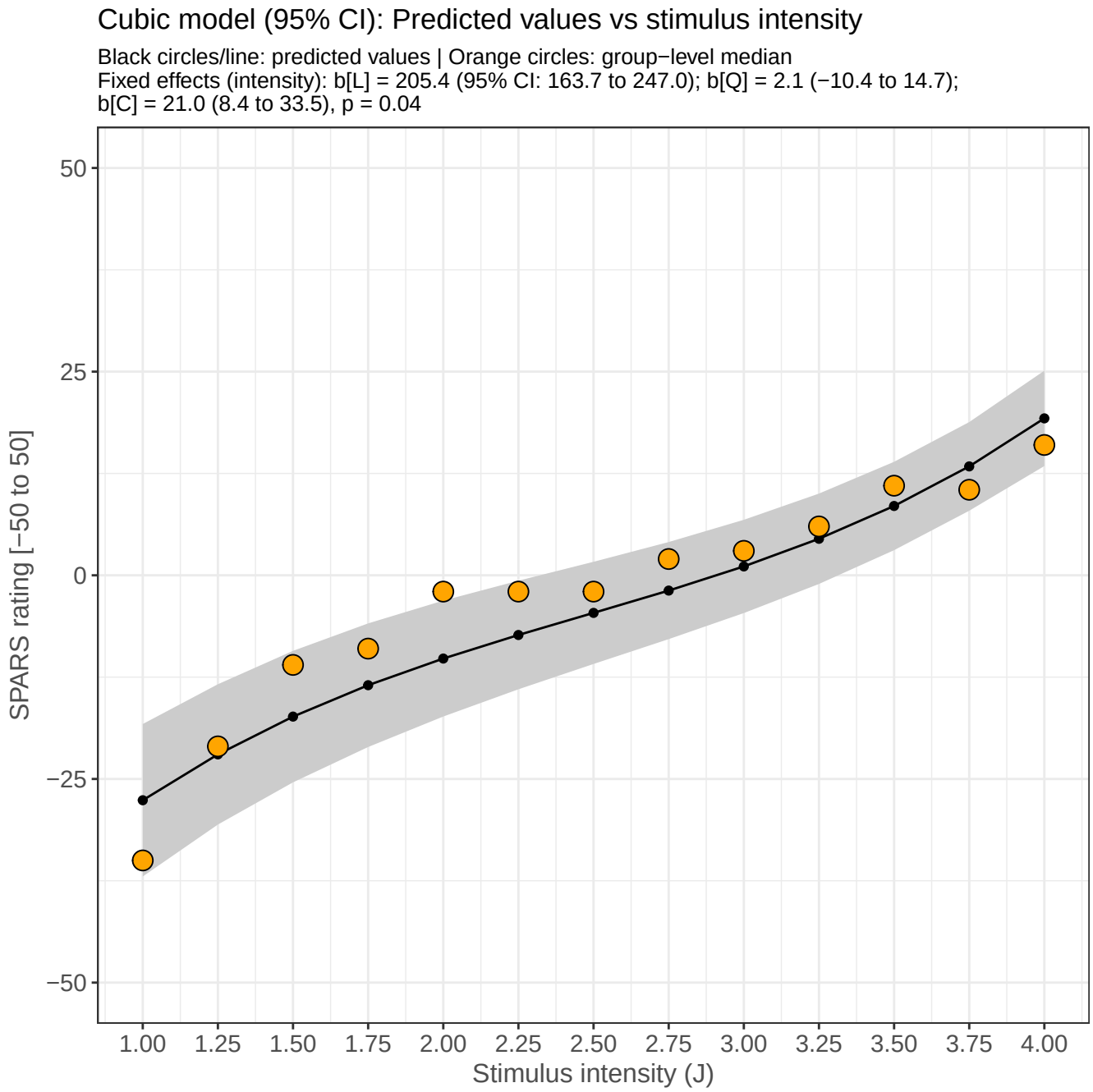
Table 1: Linear model vs quadratic model and cubic model

| term | df | AIC | BIC | logLik | deviance | statistic | Chi.Df | p.value |
|---|---|---|---|---|---|---|---|---|
| lmm1b | 6 | 1733.586 | 1754.569 | -860.7930 | 1721.586 | NA | NA | NA |
| lmm2b | 7 | 1735.487 | 1759.967 | -860.7434 | 1721.487 | 0.0991866 | 1 | 0.7528079 |
| lmm3b | 8 | 1726.958 | 1754.936 | -855.4791 | 1710.958 | 10.5285980 | 1 | 0.0011754 |

**PLot the model**

```
predicted <- ggeffects::ggpredict(model = lmm3b,
                                  terms = 'intensity',
                                  ci.lvl = 0.95)
ggplot(data = predicted) +
    geom_ribbon(aes(x = x,
                    ymin = conf.low,
                    ymax = conf.high),
```

```
              fill = '#cccccc') +
  geom_line(aes(x = x,
                y = predicted)) +
  geom_point(aes(x = x,
                y = predicted)) +
  geom_point(data = data_group,
             aes(x = intensity,
                 y = median),
             shape = 21,
             size = 4,
             fill = '#FFA500') +
labs(title = 'Cubic model (95% CI): Predicted values vs stimulus intensity',
     subtitle = 'Black circles/line: predicted values | Orange circles: group-level medi
     x = 'Stimulus intensity (J)',
     y = 'SPARS rating [-50 to 50]') +
scale_y_continuous(limits = c(-50, 50)) +
scale_x_continuous(breaks = seq(from = 1, to = 4, by = 0.25))
```




The cubic model has the best fit. The resulting curvilinear response function is *steepest* at the extremes and *flattens out* in the mid-ranges of stimulus intensity. We performed diagnostics on this model to confirm that the model was properly specified.

---

## Quantile mixed model regression

```
# Quantile model with 2.5, 25, 50, 75, and 97.5% quantiles
qmm <- lqmm(fixed = tri_mean ~ poly(intensity, 3),
            random = ~ intensity,
            group = PID,
            data = data_tm,
            tau = c(0.025, 0.25, 0.5, 0.75, 0.975))
```

```r
# Summary
summary(qmm)
```

```
## Call: lqmm(fixed = tri_mean ~ poly(intensity, 3), random = ~intensity,
##     group = PID, tau = c(0.025, 0.25, 0.5, 0.75, 0.975), data = data_tm)
##
## tau = 0.025
##
## Fixed effects:
##                         Value Std. Error lower bound upper bound  Pr(>|t|)
## (Intercept)          -36.3724     5.7321    -47.8914     -24.853 6.854e-08
## poly(intensity, 3)1 204.7079    24.5270    155.4191     253.997 5.674e-11
## poly(intensity, 3)2  11.5495    22.7117    -34.0914      57.190   0.61337
## poly(intensity, 3)3  26.7629    13.6219     -0.6113      54.137   0.05513
##
## (Intercept)          ***
## poly(intensity, 3)1 ***
## poly(intensity, 3)2
## poly(intensity, 3)3 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## tau = 0.25
##
## Fixed effects:
##                         Value Std. Error lower bound upper bound  Pr(>|t|)
## (Intercept)          -16.06242    7.78308   -31.70310     -0.4217  0.044357
## poly(intensity, 3)1 205.06628   22.64024   159.56902    250.5636 4.839e-12
## poly(intensity, 3)2   0.84314   12.99527   -25.27184     26.9581  0.948533
## poly(intensity, 3)3  21.92427    8.14480     5.55668     38.2919  0.009696
##
## (Intercept)          *
## poly(intensity, 3)1 ***
## poly(intensity, 3)2
## poly(intensity, 3)3 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## tau = 0.5
##
## Fixed effects:
##                         Value Std. Error lower bound upper bound  Pr(>|t|)
## (Intercept)           3.2873     6.4208     -9.6158      16.190  0.610969
## poly(intensity, 3)1 204.0394    22.8976    158.0249     250.054 8.006e-12
## poly(intensity, 3)2   2.2389    12.0622    -22.0011      26.479  0.853518
## poly(intensity, 3)3  22.1176     7.7338      6.5761      37.659  0.006211
##
## (Intercept)
## poly(intensity, 3)1 ***
## poly(intensity, 3)2
## poly(intensity, 3)3 **
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## tau = 0.75
##
## Fixed effects:
##                          Value Std. Error lower bound upper bound  Pr(>|t|)
## (Intercept)           19.0218     7.1574      4.6385      33.405  0.010595
## poly(intensity, 3)1 203.2674    23.8261    155.3870     251.148 2.977e-11
## poly(intensity, 3)2   5.9630    13.0829    -20.3280      32.254  0.650556
## poly(intensity, 3)3  22.6834     7.8318      6.9448      38.422  0.005628
##
## (Intercept)          *
## poly(intensity, 3)1 ***
## poly(intensity, 3)2
## poly(intensity, 3)3 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## tau = 0.975
##
## Fixed effects:
##                          Value Std. Error lower bound upper bound  Pr(>|t|)
## (Intercept)           22.0604    15.0078     -8.0989      52.220    0.1480
## poly(intensity, 3)1 188.9824    27.6871    133.3432     244.622 1.236e-08
## poly(intensity, 3)2  22.3598    14.1635     -6.1029      50.822    0.1208
## poly(intensity, 3)3  12.1005    10.7907     -9.5843      33.785    0.2676
##
## (Intercept)
## poly(intensity, 3)1 ***
## poly(intensity, 3)2
## poly(intensity, 3)3
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## AIC:
## [1] 2304 (df = 7) 1892 (df = 7) 1858 (df = 7) 1913 (df = 7) 2212 (df = 7)
```

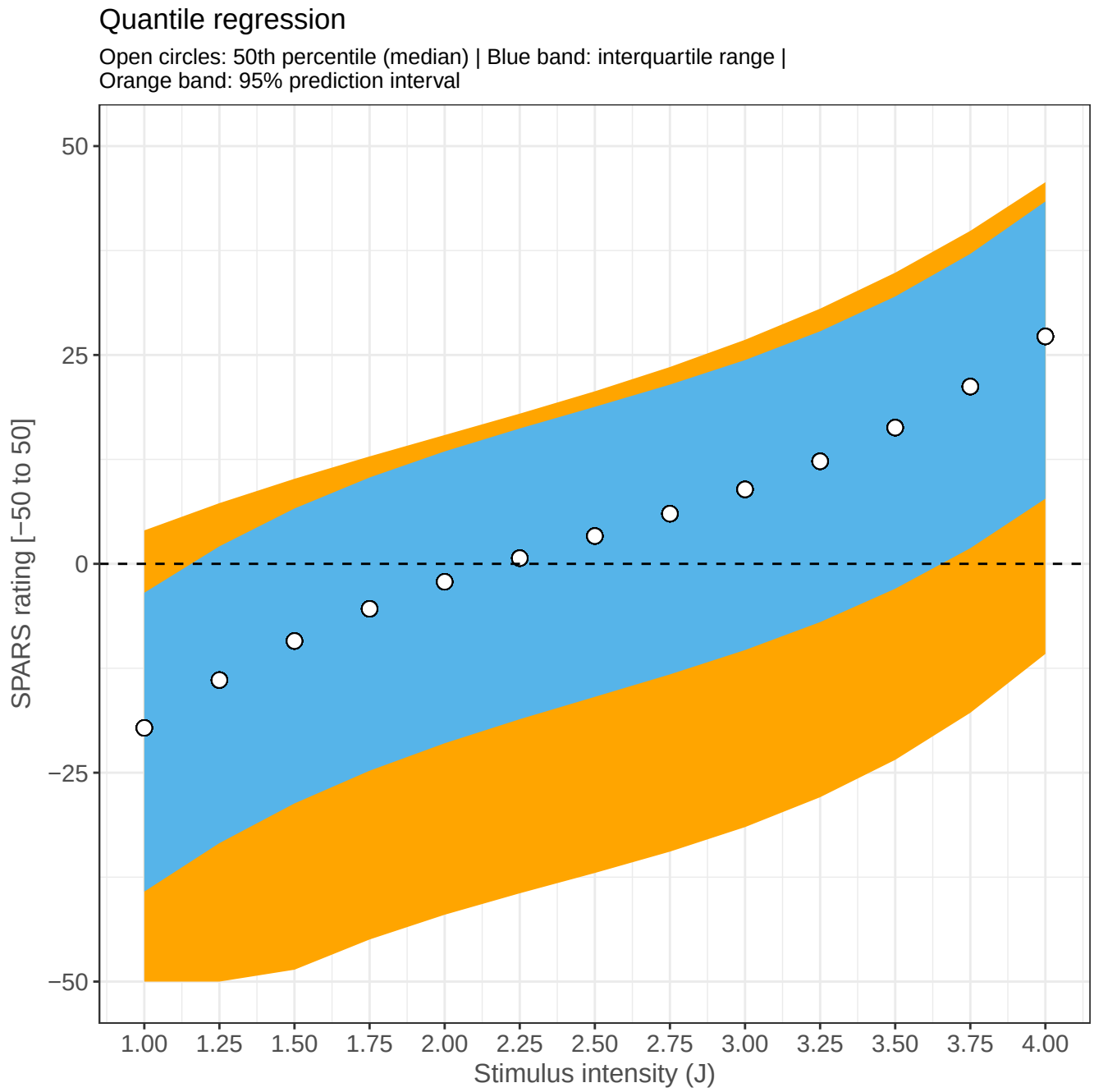
```
# Get predicted values
## Level 0 (conditional, note difference to the lmer diagnostics)
quant_predict <- as.data.frame(predict(qmm, level = 0))
names(quant_predict) <- paste0('Q', c(2.5, 25, 50, 75, 97.5))

# Join with 'central_lmm'
data_lqmm <- data_tm %>%
  bind_cols(quant_predict)

# Trim prediction to upper and lower limits of the scale
data_lqmm %<>%
  mutate_if(is.numeric,
            funs(ifelse(. > 50,
                        yes = 50,
                        no = ifelse(. < -50,
                                    yes = -50,
```

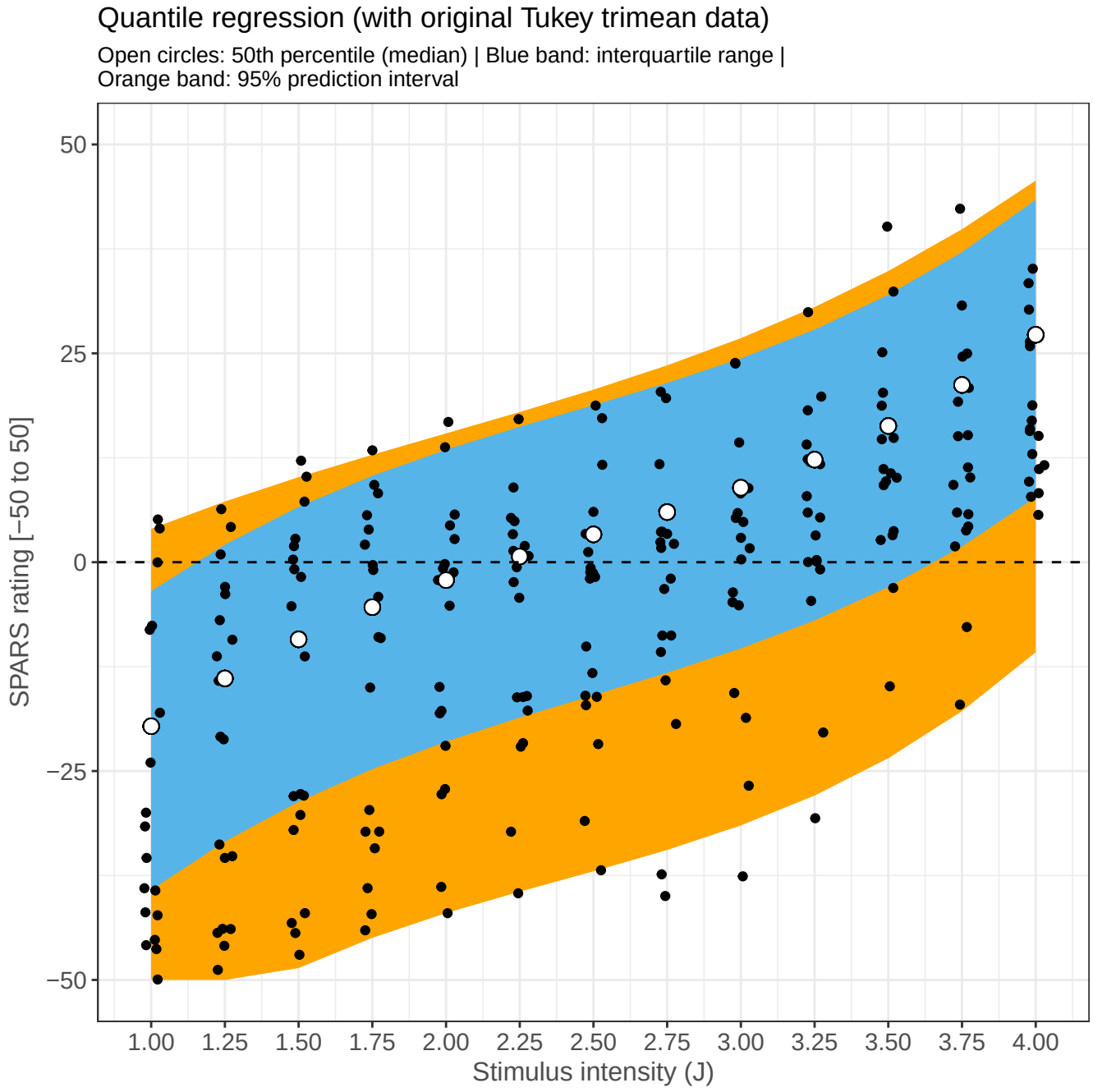
```
                                    no = .))))

# Plot
ggplot(data = data_lqmm) +
  aes(x = intensity,
      y = Q50) +
  geom_ribbon(aes(ymin = `Q2.5`,
                  ymax = `Q97.5`),
              fill = '#FFA500') +
  geom_ribbon(aes(ymin = `Q25`,
                  ymax = `Q75`),
              fill = '#56B4E9') +
  geom_point(size = 3,
             shape = 21,
             fill = '#FFFFFF',
             colour = '#000000') +
  geom_hline(yintercept = 0,
             linetype = 2) +
  labs(title = paste('Quantile regression'),
       subtitle = 'Open circles: 50th percentile (median) | Blue band: interquartile range
       x = 'Stimulus intensity (J)',
       y = 'SPARS rating [-50 to 50]') +
  scale_y_continuous(limits = c(-50, 50)) +
  scale_x_continuous(breaks = unique(data_lqmm$intensity))
```




```
## With original data
ggplot(data = data_lqmm) +
  aes(x = intensity,
      y = Q50) +
  geom_ribbon(aes(ymin = `Q2.5`,
                  ymax = `Q97.5`),
              fill = '#FFA500') +
  geom_ribbon(aes(ymin = `Q25`,
                  ymax = `Q75`),
              fill = '#56B4E9') +
  geom_point(data = data_tm,
             aes(y = tri_mean),
             position = position_jitter(width = 0.03)) +
  geom_point(size = 3,
             shape = 21,
             fill = '#FFFFFF',
             colour = '#000000') +
```

```
  geom_hline(yintercept = 0,
             linetype = 2) +
  labs(title = paste('Quantile regression (with original Tukey trimean data)'),
       subtitle = 'Open circles: 50th percentile (median) | Blue band: interquartile range
       x = 'Stimulus intensity (J)',
       y = 'SPARS rating [-50 to 50]') +
  scale_y_continuous(limits = c(-50, 50)) +
  scale_x_continuous(breaks = unique(data_lqmm$intensity))
```




There is good stability in the shape of the response characteristics across the quantiles. For all stimulus intensities, the distribution is left skewed (long tail towards lower ratings).

## Session information

```
sessionInfo()
```

```
## R version 3.5.0 (2018-04-23)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS High Sierra 10.13.5
##
## Matrix products: default
## BLAS: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_GB.UTF-8/en_GB.UTF-8/en_GB.UTF-8/C/en_GB.UTF-8/en_GB.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] bindrcpp_0.2.2     car_3.0-0          carData_3.0-1
##  [4] sjPlot_2.4.1       HLMdiag_0.3.1      lqmm_1.5.4
##  [7] lme4_1.1-17        Matrix_1.2-14      forcats_0.3.0
## [10] stringr_1.3.1      dplyr_0.7.5        purrr_0.2.5
## [13] readr_1.1.1        tidyr_0.8.1        tibble_1.4.2
## [16] ggplot2_2.2.1.9000 tidyverse_1.2.1    magrittr_1.5
##
## loaded via a namespace (and not attached):
##   [1] TH.data_1.0-8      minqa_1.2.4        colorspace_1.3-2
##   [4] rio_0.5.10         modeltools_0.2-21  ggridges_0.5.0
##   [7] sjlabelled_1.0.11  rprojroot_1.3-2    estimability_1.3
##  [10] snakecase_0.9.1    rstudioapi_0.7     glmmTMB_0.2.1.0
##  [13] DT_0.4             mvtnorm_1.0-8      lubridate_1.7.4
##  [16] coin_1.2-2         xml2_1.2.0         codetools_0.2-15
##  [19] splines_3.5.0      mnormt_1.5-5       knitr_1.20
##  [22] sjmisc_2.7.2       effects_4.0-1      bayesplot_1.5.0
##  [25] jsonlite_1.5       nloptr_1.0.4       ggeffects_0.3.4
##  [28] pbkrtest_0.4-7     broom_0.4.4        shiny_1.1.0
##  [31] compiler_3.5.0     httr_1.3.1         sjstats_0.15.0
##  [34] emmeans_1.2.1      backports_1.1.2    assertthat_0.2.0
##  [37] lazyeval_0.2.1     survey_3.33-2      cli_1.0.0
##  [40] later_0.7.3        htmltools_0.3.6    tools_3.5.0
##  [43] SparseGrid_0.8.2   coda_0.19-1        gtable_0.2.0
##  [46] glue_1.2.0         reshape2_1.4.3     merTools_0.4.1
##  [49] Rcpp_0.12.17       cellranger_1.1.0   nlme_3.1-137
##  [52] psych_1.8.4        lmtest_0.9-36      openxlsx_4.1.0
##  [55] rvest_0.3.2        mime_0.5           stringdist_0.9.5.1
##  [58] MASS_7.3-50        zoo_1.8-1          scales_0.5.0.9000
##  [61] promises_1.0.1     hms_0.4.2          parallel_3.5.0
##  [64] sandwich_2.4-0     pwr_1.2-2          TMB_1.7.13
##  [67] curl_3.2           yaml_2.1.19        stringi_1.2.2
##  [70] highr_0.6          blme_1.0-4         zip_1.0.0
##  [73] rlang_0.2.1        pkgconfig_2.0.1    arm_1.10-1
```

```
##  [76] evaluate_0.10.1    lattice_0.20-35    prediction_0.3.6
##  [79] bindr_0.1.1        labeling_0.3       htmlwidgets_1.2
##  [82] tidyselect_0.2.4   plyr_1.8.4         R6_2.2.2
##  [85] multcomp_1.4-8     RLRsim_3.1-3       pillar_1.2.3
##  [88] haven_1.1.1        foreign_0.8-70     withr_2.1.2
##  [91] mgcv_1.8-23        survival_2.42-3    abind_1.4-5
##  [94] nnet_7.3-12        modelr_0.1.2       crayon_1.3.4
##  [97] rmarkdown_1.9      grid_3.5.0         readxl_1.1.0
## [100] data.table_1.11.4  digest_0.6.15      xtable_1.8-2
## [103] httpuv_1.4.3       stats4_3.5.0       munsell_0.4.3
```