

Data Analysis Component

Paul Harmon

3/10/2019

Read Liang, Hiley, and Kanosue (2019) available at <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0212334>. The data set related to AOS (or just “Area” in the spreadsheet) is provided in a similar format to what they posted with some useful metadata in the second sheet of the spreadsheet.

Answer the first seven questions inline with each question (including code and output with each question), then attach a written report that puts some of these results together based on the instructions in the last question. Include an appendix to the written report with any additional work (code and output) not completed in earlier questions. This should take approximately two days to complete.

- Resources: For this exam you may use any textbooks or course materials. You are also free to use any publicly available materials that you might come across in a research setting; however, all writing must be original and all reference materials must be cited.
- Computer Code and Reproducibility: Please turn in all relevant computer code to reproduce your results; a reproducible document is a requirement for this section of the PhD comp. A PDF generated from R Markdown is the preferred form.
- Time: This part of the exam is meant to take two days - approximately 16 hours to complete.
- Advice: Be sure to adequately justify your answers and choices and appropriately reference any sources used. If you have questions about the exam, please email Mark Greenwood (greenwood@montana.edu) for clarifications.

Introduction

This document outlines the questions asked in the data analysis component of the PhD Comprehensive Exam. As question 8 requires writing a separate report, that is submitted as a separate document. In line with the requirements for this examination, this document was created in Rmarkdown with reproducible R-code in-line.

1) Modify the data set for the AOS (or just “Area”) to be prepared for analysis. Preferably perform data modifications in a reproducible fashion and report the code you used to perform this. If not, clearly document the steps you took to prepare the data set for analysis.

The code below modifies the data so that they can be analyzed with a single response variable, Area of the Stabiliogram. The data are saved in an Excel file so it makes sense to utilize the readxl package for this. Within the dataset, the data were recorded in a non-standard manner - there are 3 data matrices of dimension 30x8 that need to be parsed out - the easiest way to do this was to read the data into R and wrangle things as shown in the code chunk below.

The two data matrices that did not pertain to the Area response were removed from the data and the column names were re-written and replaced by the first row of the read-in file. Then, data were saved as a tibble and coerced into the proper forms (i.e. Sports is a factor, Y is a double).

```
## Requires the relevant packages
## Reads in the data
bal <- read_excel('Data/LiangData.xlsx', sheet = "Liang_etal")
```

```
## New names:
## * `` -> ...2
## * `` -> ...3
## * `` -> ...4
```

```
## * `` -> ...5
## * `` -> ...6
## * ... and 18 more problems
```

```
head(bal)
```

```
## # A tibble: 6 x 26
##   Area ...2 ...3 ...4 ...5 ...6 ...7 ...8 ...9 Speed...10 ML
##   <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <lgl> <chr>      <chr>
## 1 Sport OQ    OOL  OOL_F CQ    COL  COL_F OT    NA    Sport2    CQ
## 2 1      141.~ 517.~ 505.~ 348.~ 2488~ 2404~ 614.~ NA    1          12.5~
## 3 1      54.7~ 437.~ 371.~ 49.5~ 1586~ 1235~ 512.~ NA    1          6.29~
## 4 1      300.~ 793.~ 771.~ 318.~ 2451~ 1866~ 958.~ NA    1          9.62~
## 5 1      305.~ 357.~ 373.~ 191.~ 915.~ 2042~ 421.~ NA    1          10.2~
## 6 1      115.~ 372.~ 368.~ 348.~ 2007~ 2146~ 496.~ NA    1          12.7~
## # ... with 15 more variables: ...12 <chr>, ...13 <chr>, ...14 <chr>,
## #   ...15 <chr>, ...16 <chr>, ...17 <chr>, ...18 <lgl>, Speed...19 <chr>,
## #   AP <chr>, ...21 <chr>, ...22 <chr>, ...23 <chr>, ...24 <chr>,
## #   ...25 <chr>, ...26 <chr>
```

#nice to see these guys are real data gurus. What an awful format

#goal: modify the data for AOS to be prepped for analysis

```
bal_area <- bal[,1:8] #pulls the first 8 columns before column 9 (the empty one)
column_names <- bal_area[1,]
bal_tibble <- as_tibble(bal_area[-1,]);names(bal_tibble) <- as.character(column_names)
dim(bal_tibble)
```

```
## [1] 30 8
```

```
bal_tibble$Participant <- 1:30
```

#the above code outputs a dataset with 30 rows and 8 measurements

#still a weird format - we would prefer to have 3 columns: Sport, Eyes, Position

eyenames <- grep('C',column_names) #want C because O would be conflated with OL

#recall that we don't have a column for open toes here so only 7 columns instead of 8

```
bal_tibble_open <- select(bal_tibble, -eyenames)
```

```
bal_tibble_closed <- select(bal_tibble, c(eyenames,Sport,Participant))
```

#adds a feature to the dataset for eyes open or closed

```
bal_tibble_open$Eyes <- rep("Open", nrow(bal_tibble_open))
```

```
bal_tibble_closed$Eyes <- rep("Closed", nrow(bal_tibble_closed))
```

#these will be our position variables to append to the 'long'-form dataset

```
open_position <- rep(c('Q','OL','OL_F','T'), each = nrow(bal_tibble_open))
```

```
closed_position <- rep(c('Q','OL','OL_F'), each = nrow(bal_tibble_closed))
```

#now we want to actually do the conversion on bal_tibbles to long form

```
library(tidyr)
```

```
open_long <- bal_tibble_open %>% gather(key = Position, value = Y,OQ,OOL, OOL_F,OT)
```

```
closed_long <- bal_tibble_closed %>% gather(key = Position, value = Y, CQ,COL,COL_F)
```

```
open_long$POS <- open_position
closed_long$POS <- closed_position

#Whew! Now we put it all together by rbinding the two tibbles
full_dat <- rbind(open_long, closed_long)
full_dat$Y <- as.numeric(as.character(full_dat$Y))

fdat <- full_dat %>% filter(!POS %in% 'T')
filter_dat <- fdat %>% select(-Y) %>% lapply(as.factor)%>% as_tibble()
filter_dat$Y <- fdat$Y

pander(head(filter_dat,3)) #prints just the first three observations
```

Sport	Participant	Eyes	Position	POS	Y
1	1	Open	OQ	Q	141.6
1	2	Open	OQ	Q	54.73
1	3	Open	OQ	Q	300.4

2) Make plot that resembles their Figure 2 that incorporates some sort of uncertainty estimates with the means. Note that this will show that the units in the data set provided appear to be in square millimeters instead of square centimeters in their plot.

Figure 2 in the document includes a 3-way interaction plot, with error bars for uncertainty. I took a slightly different approach in replicating this plot with shaded areas for the uncertainty in each mean estimate. The plot gives the mean area (in square mm, compared with the plot in Figure 2 shown in square cm) associated with each group of responses. Uncertainty in the mean estimates was incorporated by fitting a `geom_ribbon()` with upper and lower boundaries that match a 95% Confidence Interval around the means.

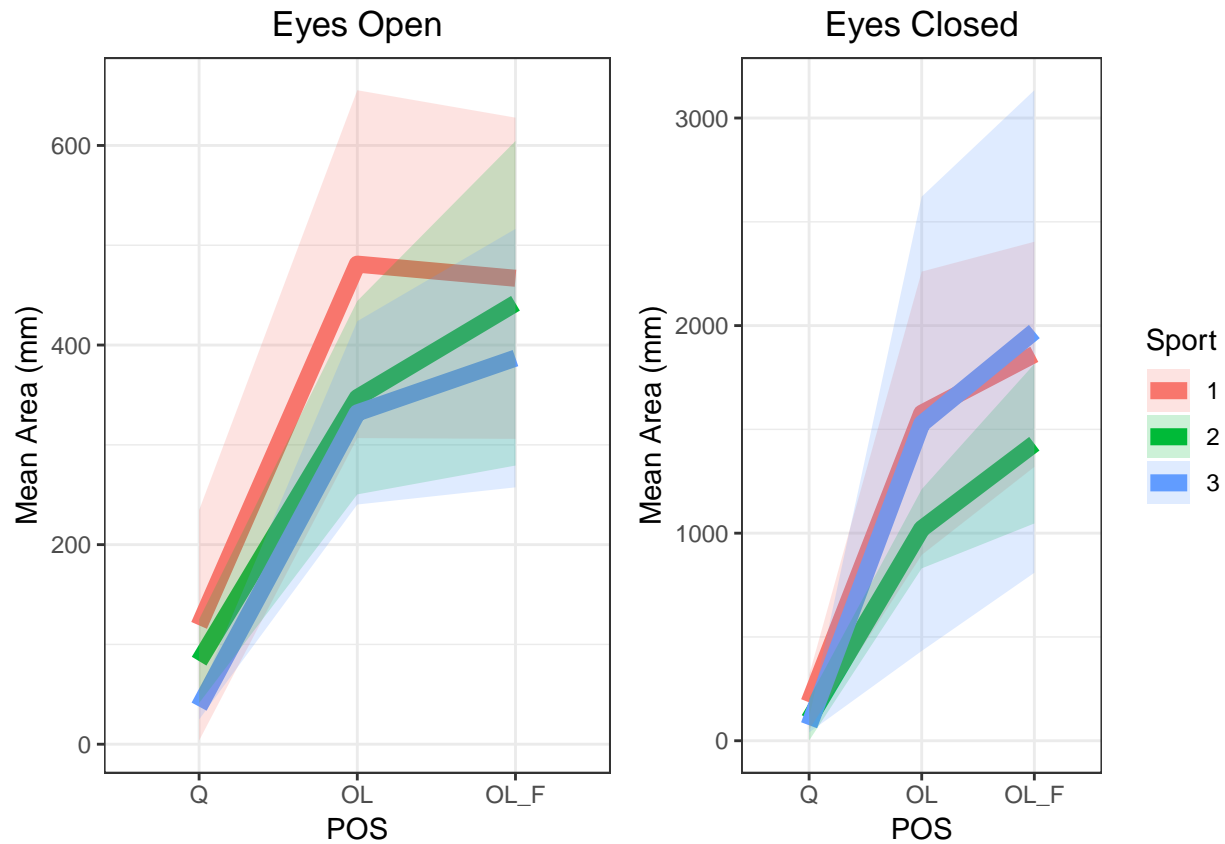
```
means <- filter_dat %>% group_by(Sport,Eyes,POS) %>% summarise(mean(Y))
names(means)[4] <- 'Agmean'
means$POS <- factor(means$POS, levels = c("Q","OL","OL_F"))

sd <- filter_dat %>% group_by(Sport, Eyes, POS) %>% summarise(sd(Y))
names(sd)[4] <- "SD"
means$upper <- means$Agmean + 1.96*sd$SD/sqrt(3) #since xbar has sd sqrt(sigma^2/n)
means$lower <- means$Agmean - 1.96*sd$SD/sqrt(3) #same

#actually creates the plot (with ribbons instead of bands for uncertainty)
O <- ggplot(filter(means,Eyes == 'Open'), aes(POS,Agmean,group = Sport)) +
  geom_line(aes(color = Sport), size = 3) + ylab("Mean Area (mm)") +
  geom_ribbon(aes(ymin = lower, ymax = upper, fill = Sport), alpha = .2)+
  theme_bw() +theme(plot.title = element_text(hjust = 0.5), legend.position = "none") + ggtitle("Eyes Open")

#for eyes closed
C <- ggplot(filter(means, Eyes == 'Closed'), aes(POS, Agmean, group = Sport))+
  geom_line(aes(color = Sport), size = 3) + ylab("Mean Area (mm)") +
  geom_ribbon(aes(ymin = lower, ymax = upper, fill = Sport), alpha = .2)+
  theme_bw() +theme(plot.title = element_text(hjust = 0.5)) + ggtitle("Eyes Closed")

#create the plot with ggarrange
library(ggpubr)
ggarrange(O,C)
```



3) Replicate the models of the authors for the “Area”, also called “AOS”. You should be able to replicate the F-statistics at the end of page 4 and top of page 5. Report your code and an ANOVA table and note the results that match.

I replicated the models the authors fit with a standard 3-factor ANOVA model. The F-statistics for all of the effects match (within small rounding error), and are shown in Table 1.

```
## Replication of the ANOVA
#this fits the cell-means version
aov1 <- lm(Y ~ POS * Eyes * Sport, data = filter_dat)
#summary(aov1) #to get parameter estimates
set.caption("Analysis of Variance results matching paper.")
pander(anova(aov1))
```

Table 2: Analysis of Variance results matching paper.

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
POS	2	32807182	16403591	103.5	1.11e-29
Eyes	1	27655115	27655115	174.5	1.683e-27
Sport	2	1414131	707065	4.46	0.01301
POS:Eyes	2	13266767	6633383	41.85	2.234e-15
POS:Sport	4	682252	170563	1.076	0.3702
Eyes:Sport	2	1177330	588665	3.714	0.02649
POS:Eyes:Sport	4	519638	129909	0.8195	0.5145
Residuals	162	25680206	158520	NA	NA

4) Were the F-tests that they reported based on Type I, II, or III tests? Or is it impossible to tell here? Why?

Were the F-tests based on Type I, II, or III tests?

Based on the estimated F-statistics in the model, it is clear that they did not use Type III sums of squares (nor would I expect them to have, given the level of statistical sophistication evident throughout the paper). Because Type III SS consider effects entering into the model conditionally (i.e. based on the other features already included), it includes an intercept estimate as well, thus the difference.

It is impossible to distinguish whether they used Type II or Type I Sums of Squares because the design was balanced, with each of the treatment/respondent combinations having the same number of replications. For reference, I used the car package in R to calculate the other two types of SS - they are shown below.

```
library(car)
```

```
## Warning: package 'car' was built under R version 3.5.3
```

```
## Loading required package: carData
```

```
##
```

```
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
```

```
##
```

```
##      recode
```

```
Anova(aov1, type = "II")
```

```
## Anova Table (Type II tests)
```

```
##
```

```
## Response: Y
```

	Sum Sq	Df	F value	Pr(>F)
POS	32807182	2	103.4798	< 2.2e-16 ***
Eyes	27655115	1	174.4584	< 2.2e-16 ***
Sport	1414131	2	4.4604	0.01301 *
POS:Eyes	13266767	2	41.8458	2.234e-15 ***
POS:Sport	682252	4	1.0760	0.37022
Eyes:Sport	1177330	2	3.7135	0.02649 *
POS:Eyes:Sport	519638	4	0.8195	0.51445
Residuals	25680206	162		

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Anova(aov1, type = "III")
```

```
## Anova Table (Type III tests)
```

```
##
```

```
## Response: Y
```

	Sum Sq	Df	F value	Pr(>F)
(Intercept)	24889180	1	157.0099	< 2.2e-16 ***
POS	16099122	2	50.7795	< 2.2e-16 ***
Eyes	6012156	1	37.9268	5.584e-09 ***

```
## Sport          1893301    2    5.9718  0.003145 **
## POS:Eyes       4859355    2   15.3273 8.009e-07 ***
## POS:Sport      1164697    4    1.8368  0.124239
## Eyes:Sport      765996    2    2.4161  0.092481 .
## POS:Eyes:Sport  519638    4    0.8195  0.514453
## Residuals      25680206 162
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

5) Return to the model that they used that included the three-way interaction of sport, stance, and eyes open/closed. Note their model assumptions and use the study design and appropriately made diagnostic plots for their model to discuss/ assess those assumptions. You should find some clear issues with their model.

The 3-way interaction model that was used by the researchers assumes several important things:

- **Independence** of the individual response values. This assumption is likely violated. While we might reasonably assume that the participants were independent (and even this is, frankly, debateable to a degree), the researchers made no attempt to address the fact that they were measuring the same participant 24 times (with three measurements thrown out for a total of 21 used measurements).

We can use the plots shown in Figure X to assess the other two assumptions of ANOVA: + **Normality**: The residuals of the model (and therefore the responses) should be normally distributed. + **Constant Variance**: The variation of response values in each treatment group should be constant for all treatment combinations.

```
par(mfrow = c(2,2))
plot(aov1, pch = 20, col = "blue3")
```

The top left panel of Figure X shows the residuals vs. the fitted plot, which is helpful in diagnosing the homoskedasticity (constant variance) of the residuals. It is pretty obvious that the responses have a non-constant variance as for fitted values that are small, the spread of residuals is quite narrow whereas for large fitted values, the spread is substantially wider.

The assumption of normality is also likely unjustified, despite the relatively large sample size in the data. The QQ plot indicates that the distribution of the residuals is more heavy-tailed than would be expected under a normal model. While I am less concerned about this issue than the non-constant variance, it is still an important violation to note.

Moreover, I find it interesting how the authors addressed these assumptions in the paper. On page 4, they note that “The Shapiro-Wilks test was used for normality, and homogeneity of variance was investigated using Levene’s test”(2019). They do not show results for either test. I do not normally prefer these tests to model diagnostics like those shown above since the tests themselves make assumptions about normality/variance (i.e. Levene’s test is not great given the demonstrated lack of normality). However, I went ahead and carried both tests out below.

It is possible that they mis-interpreted the results (in both cases, a small p-value provides evidence of the undesirable outcome), or the omission is disingenuous as the results were inconvenient. With Levene’s test, I was able to generate a large p-value if I only included Sport as the group (perhaps this is what they did). The results indicate further evidence that the assumptions were violated.

```
# not sure about their Shapiro-Wilks test
shapiro.test(filter_dat$Y)
```

```
##
## Shapiro-Wilk normality test
##
## data:  filter_dat$Y
## W = 0.8019, p-value = 2.346e-14
```

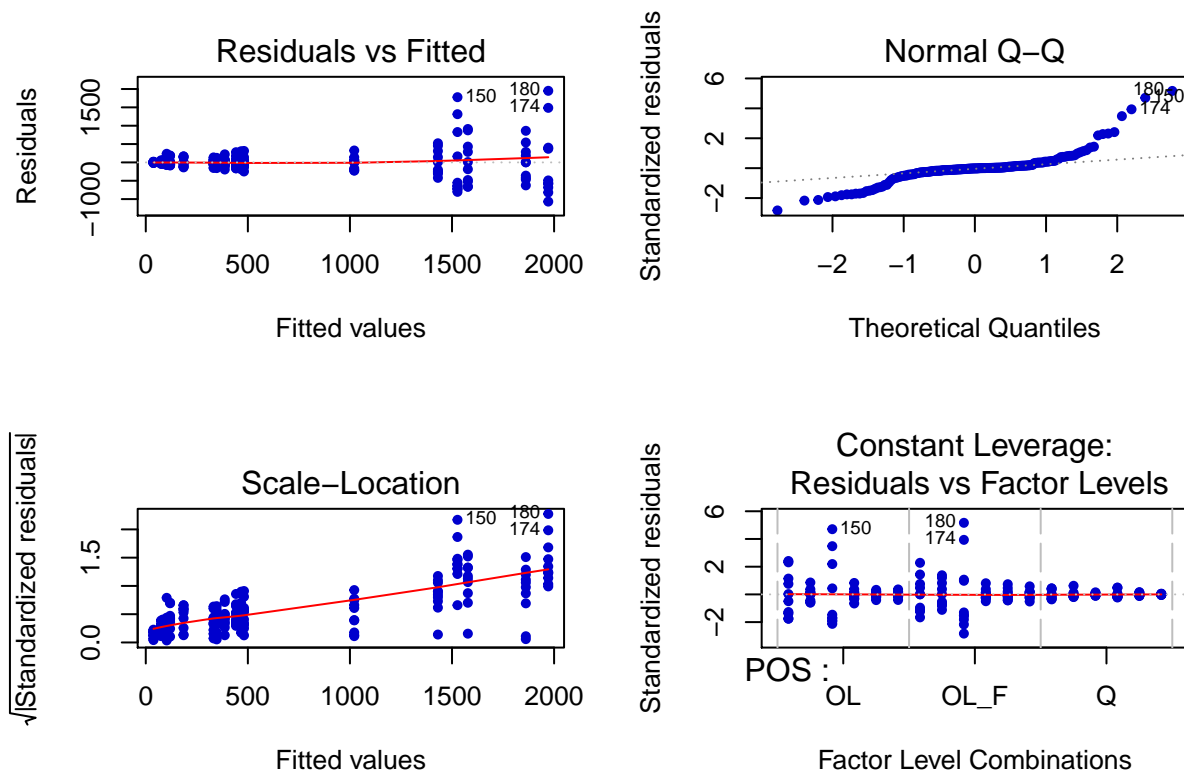


Figure 1: QQ Plots and Residual Plots to assess observations

```
leveneTest(filter_dat$Y, group = interaction(filter_dat$Sport,filter_dat$Eyes,filter_dat$POS))
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##      Df F value    Pr(>F)
## group 17   5.354 2.468e-09 ***
##      162
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

I suspect they grouped by Sport when they ran this because that's the only non-significant result

#non-significant p-value if I do it this way: Maybe this is what they did?

```
leveneTest(filter_dat$Y, group = filter_dat$Sport)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##      Df F value Pr(>F)
## group  2   1.455 0.2362
##      177
```

6) You should not be terribly impressed by the model they chose and how they worked with it. Fit a more appropriate model for this response variable that assesses the same 3-way interaction. Directly compare your results for the tests they report and explain any differences.

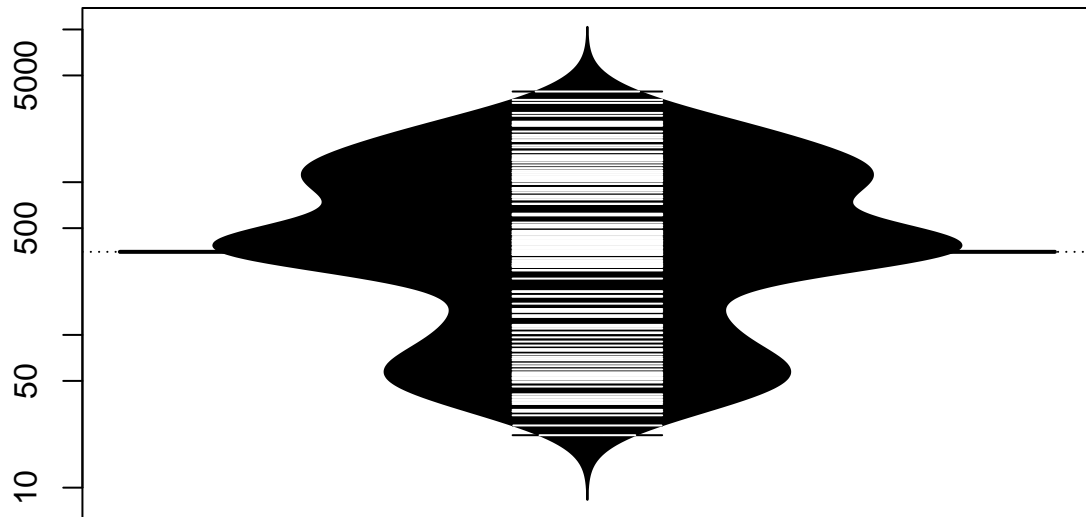
At minimum, they should have considered the non-constant variance plaguing their results. Additionally, they should have controlled for the correlated responses within each participant (i.e. repeated measures).

Prior to jumping into a mixed model, I considered transforming the response variable with a log-transformation. The log-transformation makes the most sense because the problem with constant variance and non-normality is exacerbated by large values that would be mitigated when log-transformed.

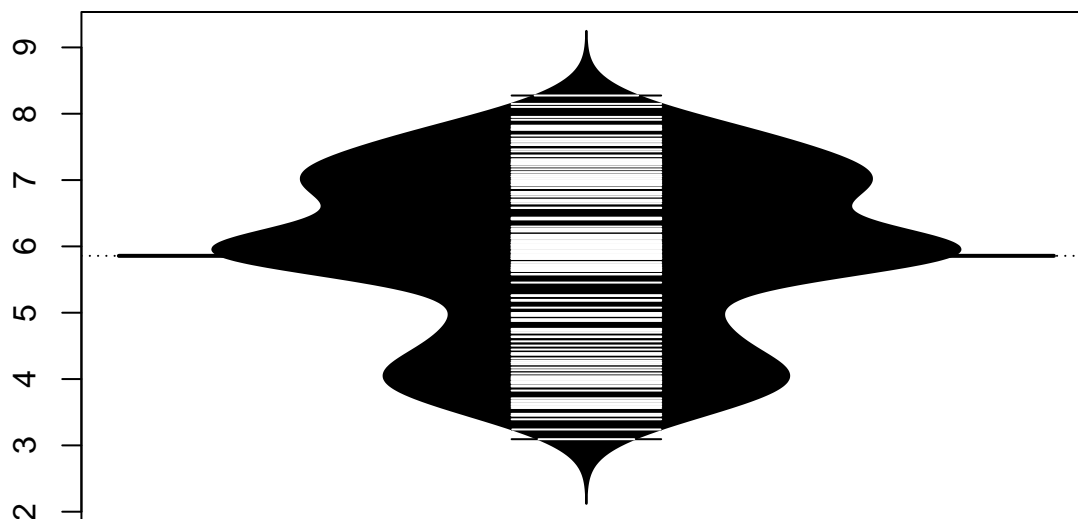
```
beanplot(filter_dat$Y, main = "Beanplot of Raw Response"); beanplot(log(filter_dat$Y), main = "Beanplot
```

```
## log="y" selected
```


Beanplot of Raw Response



Beanplot of Log-Transformed Responses



Refitting the model with a log-transformed response gives the following diagnostics. For interpretability reasons, I typically dislike transforming the data; however, in this case, the log-transform is a major improvement over the untransformed data. The residual plot is much improved with equal spreads for all the fitted values, and the heavy-tail problem is greatly mitigated in QQ plot in the upper right panel of Figure X. This is not the only change that needs to be made, but if it were, it would still improve the results of the model.

```
filter_dat$LogY <- log(filter_dat$Y)
aov2 <- lm(LogY ~ POS * Eyes * Sport, data = filter_dat)
par(mfrow = c(2,2))
plot(aov2, pch = 20, col = "blue3")
```

Log-transforming the response solves several of the problems with the model but does not fix the issue with repeated measures. I instead fit a random-intercept model that accounts for the repeated measures on each respondent. (Note that this model is very similar to a GLS model with a Compound Symmetry correlation structure).

```
mm1 <- lmer(LogY ~ POS * Eyes * Sport + (1|Participant), data = filter_dat)
summary(mm1)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: LogY ~ POS * Eyes * Sport + (1 | Participant)
## Data: filter_dat
##
## REML criterion at convergence: 223.7
##
```

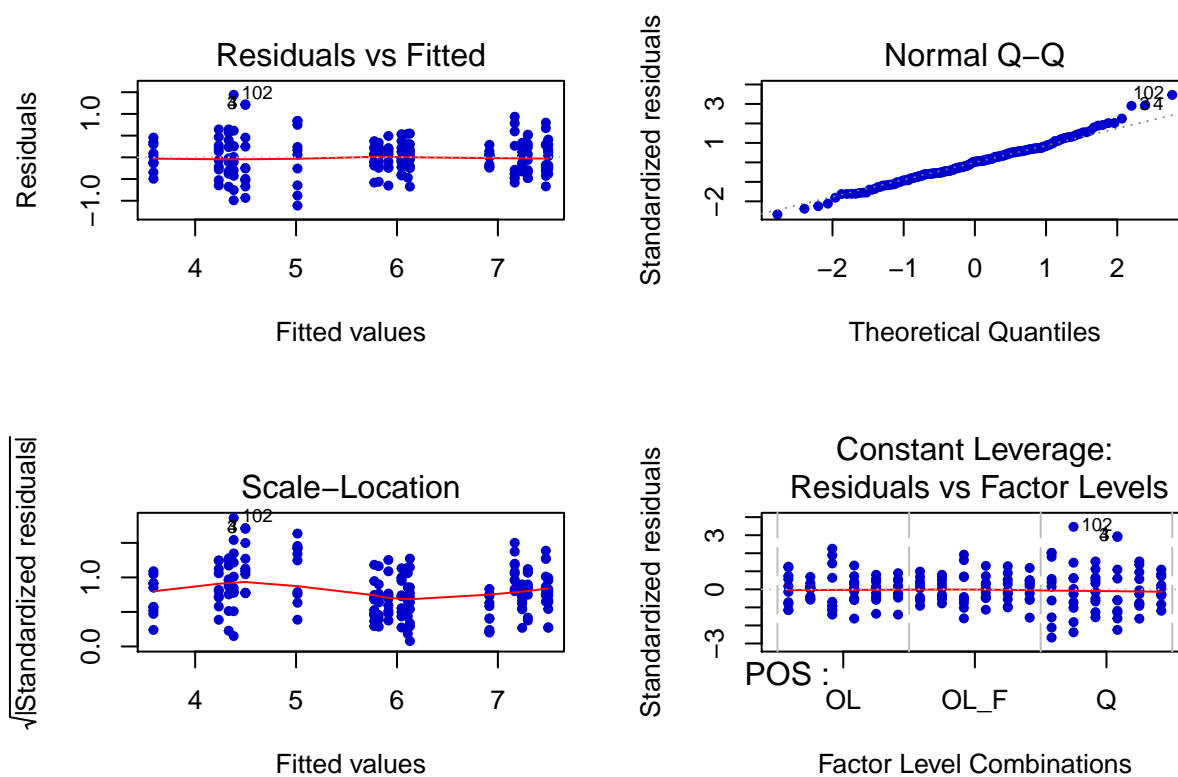


Figure 2: Diagnostics from the log-transformed model.

```
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.3715 -0.6672  0.0378  0.4957  3.4140
##
## Random effects:
##   Groups      Name      Variance Std.Dev.
## Participant (Intercept) 0.03601  0.1898
## Residual              0.15605  0.3950
## Number of obs: 180, groups: Participant, 30
##
## Fixed effects:
##              Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)      7.29702    0.13858 137.77930  52.654 < 2e-16
## POSOL_F           0.20245    0.17666 135.00000   1.146  0.2538
## POSQ             -2.28279    0.17666 135.00000 -12.922 < 2e-16
## EyesOpen        -1.16889    0.17666 135.00000  -6.617 7.91e-10
## Sport2          -0.38062    0.19599 137.77930  -1.942  0.0542
## Sport3          -0.12999    0.19599 137.77930  -0.663  0.5083
## POSOL_F:EyesOpen -0.22319    0.24984 135.00000  -0.893  0.3733
## POSQ:EyesOpen     0.65273    0.24984 135.00000   2.613  0.0100
## POSOL_F:Sport2     0.12229    0.24984 135.00000   0.489  0.6253
## POSQ:Sport2       -0.25158    0.24984 135.00000  -1.007  0.3158
## POSOL_F:Sport3     0.10599    0.24984 135.00000   0.424  0.6721
## POSQ:Sport3       -0.65018    0.24984 135.00000  -2.602  0.0103
## EyesOpen:Sport2    0.07161    0.24984 135.00000   0.287  0.7748
## EyesOpen:Sport3   -0.22290    0.24984 135.00000  -0.892  0.3739
## POSOL_F:EyesOpen:Sport2 0.12498    0.35332 135.00000   0.354  0.7241
## POSQ:EyesOpen:Sport2 0.39511    0.35332 135.00000   1.118  0.2654
## POSOL_F:EyesOpen:Sport3 0.05496    0.35332 135.00000   0.156  0.8766
## POSQ:EyesOpen:Sport3 0.09193    0.35332 135.00000   0.260  0.7951
##
## (Intercept)      ***
## POSOL_F
## POSQ             ***
## EyesOpen         ***
## Sport2           .
## Sport3
## POSOL_F:EyesOpen
## POSQ:EyesOpen     *
## POSOL_F:Sport2
## POSQ:Sport2
## POSOL_F:Sport3
## POSQ:Sport3       *
## EyesOpen:Sport2
## EyesOpen:Sport3
## POSOL_F:EyesOpen:Sport2
## POSQ:EyesOpen:Sport2
## POSOL_F:EyesOpen:Sport3
## POSQ:EyesOpen:Sport3
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##
## Correlation matrix not shown by default, as p = 18 > 12.
```

```
## Use print(x, correlation=TRUE) or
##      vcov(x)          if you need it
```

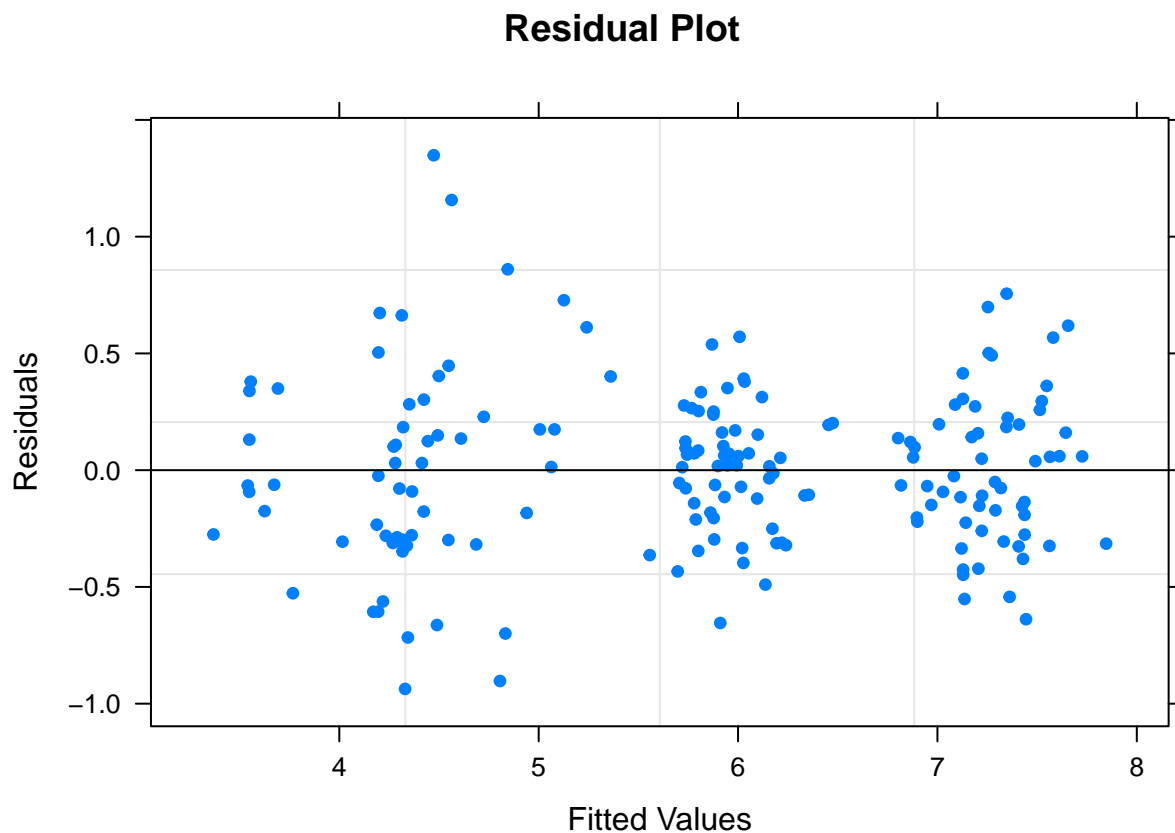
```
summary(mm1)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: LogY ~ POS * Eyes * Sport + (1 | Participant)
##      Data: filter_dat
##
## REML criterion at convergence: 223.7
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.3715 -0.6672  0.0378  0.4957  3.4140
##
## Random effects:
##      Groups      Name      Variance Std.Dev.
## Participant (Intercept) 0.03601  0.1898
## Residual              0.15605  0.3950
## Number of obs: 180, groups: Participant, 30
##
## Fixed effects:
##
##              Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)      7.29702    0.13858 137.77930  52.654 < 2e-16
## POSOL_F           0.20245    0.17666 135.00000   1.146  0.2538
## POSQ             -2.28279    0.17666 135.00000 -12.922 < 2e-16
## EyesOpen         -1.16889    0.17666 135.00000  -6.617 7.91e-10
## Sport2           -0.38062    0.19599 137.77930  -1.942  0.0542
## Sport3           -0.12999    0.19599 137.77930  -0.663  0.5083
## POSOL_F:EyesOpen -0.22319    0.24984 135.00000  -0.893  0.3733
## POSQ:EyesOpen     0.65273    0.24984 135.00000   2.613  0.0100
## POSOL_F:Sport2    0.12229    0.24984 135.00000   0.489  0.6253
## POSQ:Sport2       -0.25158    0.24984 135.00000  -1.007  0.3158
## POSOL_F:Sport3    0.10599    0.24984 135.00000   0.424  0.6721
## POSQ:Sport3       -0.65018    0.24984 135.00000  -2.602  0.0103
## EyesOpen:Sport2   0.07161    0.24984 135.00000   0.287  0.7748
## EyesOpen:Sport3  -0.22290    0.24984 135.00000  -0.892  0.3739
## POSOL_F:EyesOpen:Sport2 0.12498    0.35332 135.00000   0.354  0.7241
## POSQ:EyesOpen:Sport2 0.39511    0.35332 135.00000   1.118  0.2654
## POSOL_F:EyesOpen:Sport3 0.05496    0.35332 135.00000   0.156  0.8766
## POSQ:EyesOpen:Sport3 0.09193    0.35332 135.00000   0.260  0.7951
##
## (Intercept)      ***
## POSOL_F
## POSQ             ***
## EyesOpen         ***
## Sport2           .
## Sport3
## POSOL_F:EyesOpen
## POSQ:EyesOpen     *
## POSOL_F:Sport2
## POSQ:Sport2
## POSOL_F:Sport3
```

```
## POSQ:Sport3          *
## EyesOpen:Sport2
## EyesOpen:Sport3
## POSOL_F:EyesOpen:Sport2
## POSQ:EyesOpen:Sport2
## POSOL_F:EyesOpen:Sport3
## POSQ:EyesOpen:Sport3
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Correlation matrix not shown by default, as p = 18 > 12.
## Use print(x, correlation=TRUE) or
##      vcov(x)          if you need it
```

```
par(mfrow = c(1,2))
qqnorm(resid(mm1), main = "QQ Plot");qqline(resid(mm1))
plot(mm1, pch = 20, main = "Residual Plot", xlab = "Fitted Values", ylab = "Residuals")
```



Now, before we can directly compare the results between this model and the one fit by the researchers, we need to make a decision on which degrees of freedom method should be used for the mixed-effects model. Kuznetsova, Brockhoff, and Christensen (2017) offer a bit of insight that “Satterthwaite’s method can be considered as a good alternative [and]... is generally faster than Kenward-Roger’s” (pp 16), although in a relatively simple model such as this they likely perform the same.

The results from comparing Type II SS indicate that the test statistics for Position (POS), Eyes, and Sport’s main effects are all larger than the author’s model. Moreover, the mixed effects model finds evidence in favor of all three two-way interactions

whereas the author's ANOVA does not detect an interaction effect between Position and Sport. Neither of the models find evidence of a three-way interaction, and considering the number of additional parameters to be estimated, I would consider dropping it from the model.

```
anova(aov1)
```

```
## Analysis of Variance Table
##
## Response: Y
##              Df    Sum Sq Mean Sq F value    Pr(>F)
## POS           2 32807182 16403591 103.4798 < 2.2e-16 ***
## Eyes          1 27655115 27655115 174.4584 < 2.2e-16 ***
## Sport         2 1414131   707065   4.4604  0.01301 *
## POS:Eyes      2 13266767 6633383  41.8458 2.234e-15 ***
## POS:Sport     4  682252   170563   1.0760  0.37022
## Eyes:Sport    2 1177330   588665   3.7135  0.02649 *
## POS:Eyes:Sport 4  519638   129909   0.8195  0.51445
## Residuals    162 25680206  158520
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(mm1, type = "II")
```

```
## Type II Analysis of Variance Table with Satterthwaite's method
##              Sum Sq Mean Sq NumDF DenDF  F value    Pr(>F)
## POS          208.063  104.032     2    135 666.6743 < 2.2e-16 ***
## Eyes          45.183   45.183     1    135 289.5512 < 2.2e-16 ***
## Sport          2.171    1.086     2     27   6.9565 0.0036582 **
## POS:Eyes       8.240    4.120     2    135  26.4032 2.099e-10 ***
## POS:Sport      3.501    0.875     4    135   5.6087 0.0003306 ***
## Eyes:Sport     1.329    0.664     2    135   4.2577 0.0161026 *
## POS:Eyes:Sport 0.227    0.057     4    135   0.3638 0.8340486
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Note: Recall that the study design asked participants to stand still on a plate for 60 seconds. They were asked to do this 24 times, often standing only on one foot or balancing on their toes without being able to steady themselves (as this would be antithetical to the goal of the experiment). To me, this sounds quite tiring - I would expect that balance would deteriorate as the experiment went on. Although the researchers randomized the order of treatments, this effect could still have been useful in estimating some sort of time-varying correlation structure for measurements within a respondent (AR1, etc). However, this information is not made available in the data so order-driven correlation structures are unavailable to us at this juncture.

7) Suppose that another researcher had published a study that estimated the difference between eyes open and eyes closed for baseball players in the unipedal-foam level. Use your model to generate estimates for open eyes, baseball, unipedal-foam and closed eyes, baseball, unipedal-form combinations and confidence intervals for each estimate. Then generate an estimate and a 95% confidence interval for the difference in those two combinations of levels. You do not need to report this as part of the final report unless you think it or something similar provides utility in summarizing results.

Using the emmeans package, it is possible to generate predicted means in each of the combinations of factors. We can also get a conservative estimate of the difference between any pair of combinations using the Tukey method to control for multiple comparisons. My perspective is that if the goal is to compare the results of this analysis to another paper from a different researcher, it would be best to use a conservative method like Tukey to control for multiple comparisons.

Then, if we do detect a difference between the groups, we have hedged our methodology enough that our results ought to be detecting something relatively real.

With that methodology in mind, we estimate difference between Open and Closed eyes for baseball players standing in the unipedal position on foam:

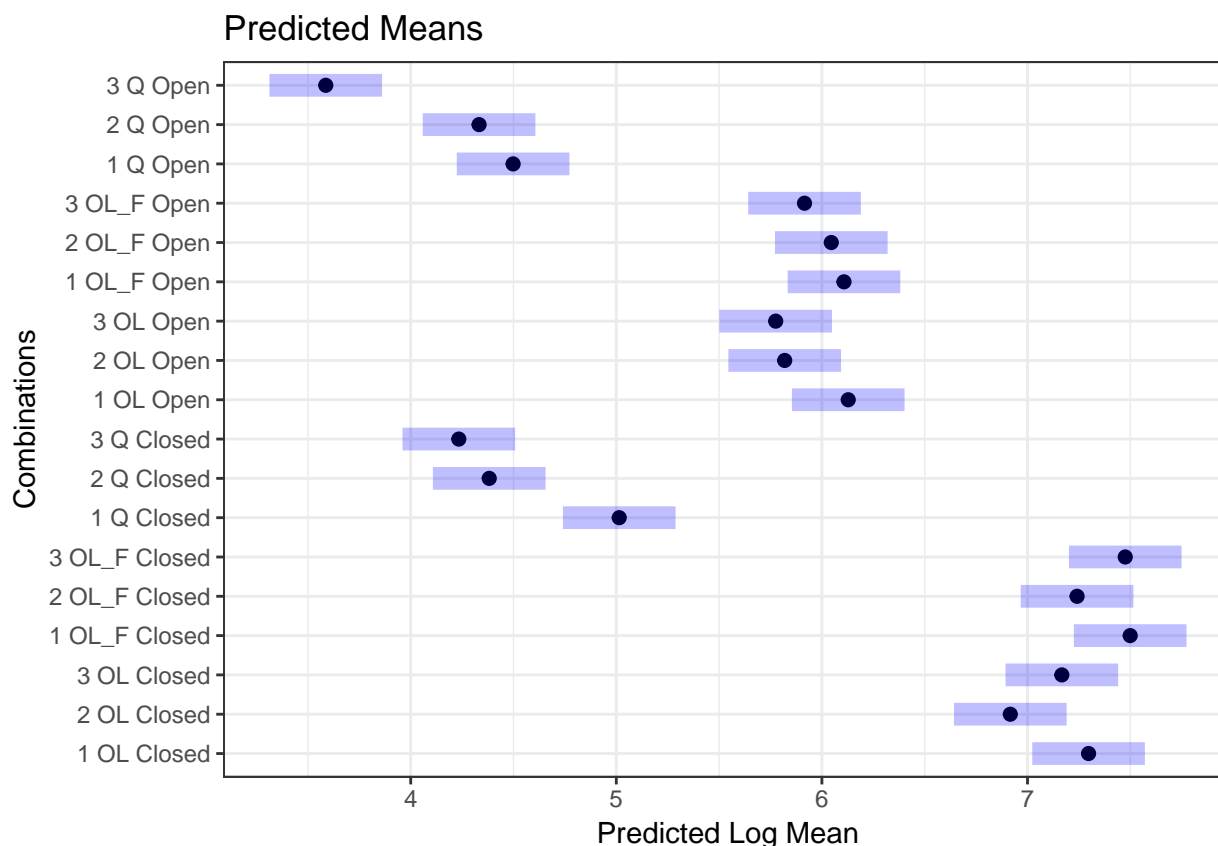
```
library(emmeans)
```

```
## Warning: package 'emmeans' was built under R version 3.5.3
```

```
em_sport <- emmeans(mm1, c("Sport", "POS", "Eyes"))
em_sport
```

```
##   Sport POS   Eyes   emmean    SE  df lower.CL upper.CL
##   1     OL Closed   7.30 0.139 138    7.02    7.57
##   2     OL Closed   6.92 0.139 138    6.64    7.19
##   3     OL Closed   7.17 0.139 138    6.89    7.44
##   1    OL_F Closed   7.50 0.139 138    7.23    7.77
##   2    OL_F Closed   7.24 0.139 138    6.97    7.52
##   3    OL_F Closed   7.48 0.139 138    7.20    7.75
##   1      Q Closed   5.01 0.139 138    4.74    5.29
##   2      Q Closed   4.38 0.139 138    4.11    4.66
##   3      Q Closed   4.23 0.139 138    3.96    4.51
##   1     OL Open    6.13 0.139 138    5.85    6.40
##   2     OL Open    5.82 0.139 138    5.55    6.09
##   3     OL Open    5.78 0.139 138    5.50    6.05
##   1    OL_F Open    6.11 0.139 138    5.83    6.38
##   2    OL_F Open    6.05 0.139 138    5.77    6.32
##   3    OL_F Open    5.92 0.139 138    5.64    6.19
##   1      Q Open    4.50 0.139 138    4.22    4.77
##   2      Q Open    4.33 0.139 138    4.06    4.61
##   3      Q Open    3.59 0.139 138    3.31    3.86
##
## Degrees-of-freedom method: kenward-roger
## Confidence level used: 0.95
```

```
plot(em_sport) + ggtitle("Predicted Means") + theme_bw() + xlab("Predicted Log Mean") + ylab("Combinati
```

```
pairs(em_sport)[57,]
```

```
## contrast          estimate    SE  df t.ratio p.value
## 1,OL_F,Closed - 1,OL_F,Open    1.39 0.177 135  7.880  <.0001
```

```
confint(pairs(em_sport)[57,])
```

```
## contrast          estimate    SE  df lower.CL upper.CL
## 1,OL_F,Closed - 1,OL_F,Open    1.39 0.177 135    1.04    1.74
##
## Confidence level used: 0.95
```

```
#1,OL_F,Closed - 1,OL_F,Open
```

8) Write a report that contains a more appropriate analysis of this response variable that starts with addressing the eyes open/closed, stance, and sport interaction. You can potentially pursue model refinement/modifications as you see fit, clearly reporting the model you used and how you arrived at it, and should attempt to interpret/explain results based on more than just p-values. The report should contain sections such as introduction, methods, results, and a scope of inference. This should be around 5 pages of double-spaced text with figures and tables following the text or around 10 pages with double-spaced text and tables and figures integrated within the document. Any analyses not performed in the prior questions should be included as an appendix to this analysis.

Report attached as a separate document.

Session Information:

Additional information about the R session I worked in is given here:

```
getRversion()
```

```
## [1] '3.5.2'
```

```
sessionInfo()
```

```
## R version 3.5.2 (2018-12-20)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 18342)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] stats      graphics  grDevices  utils      datasets  methods    base
##
## other attached packages:
## [1] emmeans_1.3.3  car_3.0-2      carData_3.0-2  ggpubr_0.1.9
## [5] bindrcpp_0.2.2 lmerTest_3.1-0 lme4_1.1-20    Matrix_1.2-15
## [9] beanplot_1.2   readxl_1.3.1   corrplot_0.84  tidyr_0.8.2
## [13] pander_0.6.3   tibble_2.1.1   magrittr_1.5    dplyr_0.7.8
## [17] ggplot2_3.1.0  readr_1.1.1
##
## loaded via a namespace (and not attached):
## [1] Rcpp_1.0.0      mvtnorm_1.0-8   lattice_0.20-38
## [4] zoo_1.8-4        assertthat_0.2.0 digest_0.6.18
## [7] utf8_1.1.4       R6_2.3.0        cellranger_1.1.0
## [10] plyr_1.8.4       evaluate_0.12   coda_0.19-2
## [13] pillar_1.3.1     rlang_0.3.1     multcomp_1.4-8
## [16] lazyeval_0.2.1   curl_3.2        minqa_1.2.4
## [19] data.table_1.11.8 nloptr_1.2.1    rmarkdown_1.11
## [22] labeling_0.3      splines_3.5.2   stringr_1.3.1
## [25] foreign_0.8-71    munsell_0.5.0   compiler_3.5.2
## [28] numDeriv_2016.8-1 xfun_0.4        pkgconfig_2.0.2
## [31] htmltools_0.3.6   tidyselect_0.2.5 codetools_0.2-15
## [34] rio_0.5.16        fansi_0.4.0     crayon_1.3.4
## [37] withr_2.1.2       MASS_7.3-51.1   grid_3.5.2
## [40] nlme_3.1-137      xtable_1.8-3    gtable_0.2.0
## [43] scales_1.0.0      zip_2.0.0       estimability_1.3
## [46] cli_1.0.1         stringi_1.3.1   sandwich_2.5-0
## [49] cowplot_0.9.3     openxlsx_4.1.0  TH.data_1.0-9
## [52] tools_3.5.2       forcats_0.4.0   glue_1.3.0
## [55] purrr_0.2.5       hms_0.4.2       parallel_3.5.2
```

```
## [58] pbkrtest_0.4-7    survival_2.43-3    abind_1.4-5
## [61] yaml_2.2.0        colorspace_1.3-2  knitr_1.22
## [64] bindr_0.1.1       haven_2.1.0
```

Additional Things to Think About: